IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 6, 2020, accepted June 21, 2020, date of publication July 6, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007538

MetroEye: A Weather-Aware System for Real-Time Metro Passenger Flow Prediction

JIANYUAN WANG[1], BIAO LENG[1,2,3], (Member, IEEE), JUNJIE WU[4], (Member, IEEE), HENG DU[5], AND ZHANG XIONG[1], (Member, IEEE)

[1]School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2]Shenzhen Institute, Beihang University, Shenzhen 518057, China
[3]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
[4]School of Economics and Management, Beihang University, Beijing 100191, China
[5]Beijing Urban Rail Transit Control Center, Beijing 100101, China

Corresponding author: Biao Leng (lengbiao@buaa.edu.cn)

This work was supported in part by the Science, Technology and Innovation Commission of Shenzhen Municipality Foundation under Grant JCYJ20180307123632627, in part by the Beijing Municipal Natural Science Foundation under Grant L182014, in part by the National Natural Science Foundation of China under Grant 61972014, and in part by the Project of the State Key Laboratory of Software Development Environment under Grant SKLSDE-2019ZX-19. The work of Junjie Wu was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2101804, and in part by the National Natural Science Foundation of China under Grant 71531001, Grant 71725002, and Grant U1636210.

ABSTRACT Real-time passenger flow prediction plays an important role in subway network design and management. Most of the existing prediction algorithms only consider the sequence of passenger flow volume, however, ignore the influence of other outer factors, for example, the weather conditions, air quality and temperature. In this paper, a systematic framework, MetroEye, is proposed for weather-aware prediction of real-time passenger flow. The framework contains an offline system and an online system. The offline system adopts a conditional random field (CRF) model to establish the relationship between passenger flow volume and weather factors. Experimental results show the superior prediction accuracy of the model, especially in large stations. The online system provides efficient methods to simulate the real-time passenger flow volume. Due to its high practicality, MetroEye has been adopted by Beijing Urban Rail Transit Control Center to monitor the passenger flow status of the Beijing subway system.

INDEX TERMS Passenger flow prediction, subway network, conditional random field, intelligent transportation.

I. INTRODUCTION

The subway transportation is an important way to solve the urban traffic problems. In the developed cities, subway traffic occupies a very large proportion in public transportation especially in rush hours. In Beijing, the subway system serves more than 10 million passengers everyday, while half of the passengers gather in the morning and evening peak (7 am to 9 am, and 5 pm to 7 pm). When enormous amount of passengers rush into one subway station, health and safety risks can be caused on crowded subway station platforms and escalators. Therefore, the real-time prediction of passenger flow volume is essential for passenger flow control in the subway daily operation and management.

In most subway networks, the automatic fare collection (AFC) system is adopted to collect the passenger information. The passengers should swipe the trip-card when they get-in or out of the system. However, the real-time passenger location cannot be collected and their destination stations are not clear before they get out. Especially, in some cases, the passengers don't need to swipe the trip-card again when get transfer. It is therefore difficult to directly infer the subway real-time passenger flow from AFC system.

Most existing real-time prediction methods only consider the sequence of passenger flow as the mainly factor in prediction models (autoregressive moving average, neural network, support vector machine, etc.). However, an important factor, weather, which has significant influence on people's travel mode, has been rarely considered in the prediction of passengers' traffic flow.

The associate editor coordinating the review of this manuscript and approving it for publication was J. D. Zhao.

Our analysis in Section III-A shows that weather such as temperature, air quality, and rainy/sunny conditions significantly affects the passenger flow volume at different subway stations, especially stations of suburban lines. For example, there are more passengers at stations of suburban lines on sunny days than on rainy days. Stations close to entertainment places are more crowded when air quality is better and temperature is more pleasant. Therefore, this paper targets on predicting the real-time passenger flow volume at each station of a subway network with the consideration of weather conditions.

In order to predict the real-time metro passenger flow volume, we propose an approach (called MetroEye) for simulating passengers' travel paths from their origins to destinations based on collected historic records. MetroEye contains an offline and online system. The offline system adopts a conditional random field to model the influence of weather factors on subway passenger flow volume. The online system predicts the destination station and path, then simulates the real-time passenger flow and predicts the flow volume. Experimental evaluation on a data set from Beijing subway system demonstrates the effectiveness and efficiency of MetroEye. Due to its high practicality, MetroEye has been applied by the government to monitoring the passenger flow status of the Beijing subway system. More application details will be found in Section VII.

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

- We propose a practical systematic framework, MetroEye, to predict the real-time passenger flow volume in subway system given only the entrance information of passengers. MetroEye has been applied in Beijing Traffic Control Centre (TCC), the government department to manage the daily operation of Beijing subway system.
- We consider the weather factor to establish conditional random field models for predicting the destination selections in the offline system of MetroEye. The weather-aware model makes more accurate prediction on passenger flow volume than other models without weather factors.
- We design the online system to simulate the real-time position of massive passengers. The simulation is driven by passenger thread and train thread according to AFC system and train schedule respectively. To speed up the simulating process, MetroEye adopts some optimization strategies such as Huffman tree for destination selections.

The remainder of this paper is organized as follows. Related works for passenger flow prediction is presented in Section II. The overview of MetroEye system is introduced in Section III. The weather factors and their influence are shown in Section III-A. In Section IV and VI, the destination prediction model (CRF) is illustrated in detail. Section V presents other specifics in real-time simulation. Section VII shows applications of MetroEye. Finally, conclusion and future works are discussed in Section VIII.

II. RELATED WORK

The prediction of traffic flow volume is strategically important in transportation management systems [1]. It is of great help for the traffic managers to detect the traffic condition and improve traffic service. In particular, real-time prediction are frequently used in monitoring traffic condition, which is very important in real-time traffic management [2], [3]. Methods for traffic volume prediction can be generally divided into three categories: linear algorithm, non-linear algorithm and hybrid algorithm, which will be introduced in the following three subsections. In the last subsection, prediction methods based on multiple factors will be presented.

A. LINEAR ALGORITHM

Linear prediction algorithm appears earlier and mainly includes historical average algorithm [4], [5], autoregressive moving average algorithm (ARMA) [6], [7], Kalman filtering algorithm [8], etc. Historical average algorithm simply uses an average of past traffic volumes to forecast future traffic volume. Autoregressive moving average algorithm is a statistical time-series model, which can capture the cyclic pattern of traffic demand over time and performs better than linear regression and historical average. Kalman filter automatically provides dynamic error-bounds on its estimates as well. Ghosh *et al.* [9] proposed a structural time-series model to predict the serious traffic congestion in Dublin, using a parsimonious and computationally simple multivariate real-term traffic condition predicting algorithm. Further more, Habtemichael *et al.* [10] proposed a non-parametric and data-driven methodology for real-term traffic predicting based on identifying similar traffic patterns using an enhanced K-nearest neighbor algorithm.

The advantages of linear algorithms are obvious: easy to implement, low-cost computation, and relatively accurate results. For a long history, linear algorithms are the most frequently used methods in real-term prediction of traffic flow volume. However, they often fail to model the fluctuations in the passenger flow sequence. Therefore, sophisticated non-linear models are proposed to improve the prediction accuracy.

B. NON-LINEAR ALGORITHM

Non-linear prediction algorithm has been widely developed in the recent ten years and mainly includes artificial neural network algorithm [11]–[13], the support vector machine (SVM) algorithm [14], [15], the gray algorithm [16] and so on [17]–[19]. Artificial neural network algorithm usually contains input, output and hidden layers to establish the relationship between future passenger flow and historical flow sequence. Support vector machine algorithm embodies the structural risk minimization principle, by involving both empirical and anticipant risk in the training cost function. The gray model treats all variables as a grey quantity within a certain range, then collects available data to obtain the internal regularity. In order to further improve the accuracy of the prediction algorithms, various data preprocessing methods and models were designed. Chen *et al.* [20] proposed

the pretreatment method of residual error sequence, removing the intraday trend of data, which is shown to be effective on improving the accuracy of the prediction algorithm. Duan *et al.* [21] developed a deep learning algorithm with consideration of both temporal and spatial factors for traffic data to improve the traditional neural network algorithm. Zhang *et al.* [22] developed a deep neural network to predict equipment running data according to time-series data. Li *et al.* [23] proposed a dynamic radial basis function (RBF) neural network to forecast outbound passenger volumes and improve passenger flow control.

Non-linear algorithms overcome some fatal flaws in linear algorithms and more appropriately model the influence of multiple factors on passenger flow. However, most of the non-linear models ignored global information of the system. In addition, complicated models [24] are easy to get over-fitted when training data are relatively in a small size.

C. HYBRID ALGORITHM

Hybrid algorithms combine more than one method, and takes the advantage of all combined methods [25]. For example, Sun *et al.* [26] proposed a hybrid algorithm by combining wavelet transform and SVM to predict passenger flows in Beijing subway system. They divided the prediction into three stages: Decomposition stage, to decompose data into high and low frequency information by wavelet; Prediction stage, to predict the high and low frequency information by SVM respectively; Reconstruction stage, to reconstruct predicted information series by wavelet. Jiang *et al.* [27] proposed a hybrid model by combining wavelet and neural network, which incorporates the self-similar, singular, and fractal properties discovered in the traffic flow. The wavelet frame is designed to provide adaptable translation parameters in traffic flow. Moreover, empirical mode decomposition (EMD) [28] is also used frequently in data pretreatment. In these methods the data will be pretreated into another form, and transform back after prediction. Some researchers also combined deep learning method to improve prediction. Li *et al.* [29] proposed a deep feature leaning approach combined with multiple steps to predict short-term traffic flow. Jia *et al.* [30] integrates a long short term memory neural network (LSTM) and stacked auto-encoders (SAEs)to predict short-term passenger flows. Gu *et al.* [31] proposed a Bayesian combination model with deep learning (IBCM-DL) for traffic flow prediction.

Hybrid algorithms often have good performance on predicting accuracy. However, they mainly focused on the time series or the traffic flow sequences, excluding other factors which may have impact on the traffic flow. The external factor like weather has not been considered in previous work.

D. MULTI-FACTOR ALGORITHM

Most of the traffic flow prediction algorithms only analyze the trend and fluctuation of passenger flow sequence by linear, non-linear, or hybrid methods. In fact, some people's travel patterns also depend on other multiple factors, such as weather, air quality, traffic congestion, and so on. For example, entertainment activities may be cancelled in rainy days; citizens will travel less on polluted days; while the commuting is almost unaffected by the weather. Usually for subway stations, the distribution of travel patterns is different. Thus, weather factors may have a potential impact on passenger flow of subway stations. Most of the existing algorithms predict traffic volume without considering these external factors, and thus have lower accuracy at the circumstances of extreme weather and special conditions.

Therefore, multi-factor algorithms have been attempted on traffic volume prediction. In literature [32], Zheng *et al.* analyzed the relationship among the weather, air quality index (AQI) and taxi data, and proposed a model for predicting the AQI of Beijing considering the factors of weather, traffic volume and historic AQI, based on Conditional random fields (CRF). CRF is an identify probability model, widely used in signal recognition, image segmentation, natural language processing and other fields. Ristovski *et al.* [33] proposed an effective CRF model for structured regression on large, fully connected graphs. Tseng *et al.* [34] present a Chinese word segmentation system by CRF models. He *et al.* [35] used CRF models for image labeling, to classify every pixel of a given image into one of several predefined classes. Sutton *et al.* [36] present dynamic CRFs to improve performance of the traditional CRF model in natural language processing task. Radosavljevic *et al.* [37] used CRF in remote sensing regression to predict output variables that have some internal structures. Djuric *et al.* [38] proposed a series of CRF model to improve traffic speed prediction accuracy.

In this paper, a CRF model is designed to predict the destination choice of passengers. The relationship of the passenger flow, weather factors and time period series will be established by CRF models in the offline system.

III. MetroEye: MOTIVATION AND OVERVIEW

This paper targets at predicting the real-time passenger flow volume of each station and section in a subway network, with the passenger information gathered by the so-called automatic fare collection (AFC) systems [41]. The system records the origin station and the entrance time when passengers swipe cards to pass through gate channels. With such limited information, it is difficult to directly infer the passenger flow volume at each station, as the AFC system will not receive any information during traveling until passengers arrive at their destinations and swipe cards to get out.

Be provided with a three-month real-world data set from Beijing subway, we are able to have an after-event analysis of the influence of weather on the traffic flow volume of subway, which motivated our study of a weather-aware approach. We will firstly present the analysis results and then give an overview of our proposed approach MetroEye in this section.

A. WEATHER EFFECT

We mainly consider three weather factors: temperature, air quality index (AQI) and rainy/sunny condition. These three factors are easy to be perceived and accessed from weather

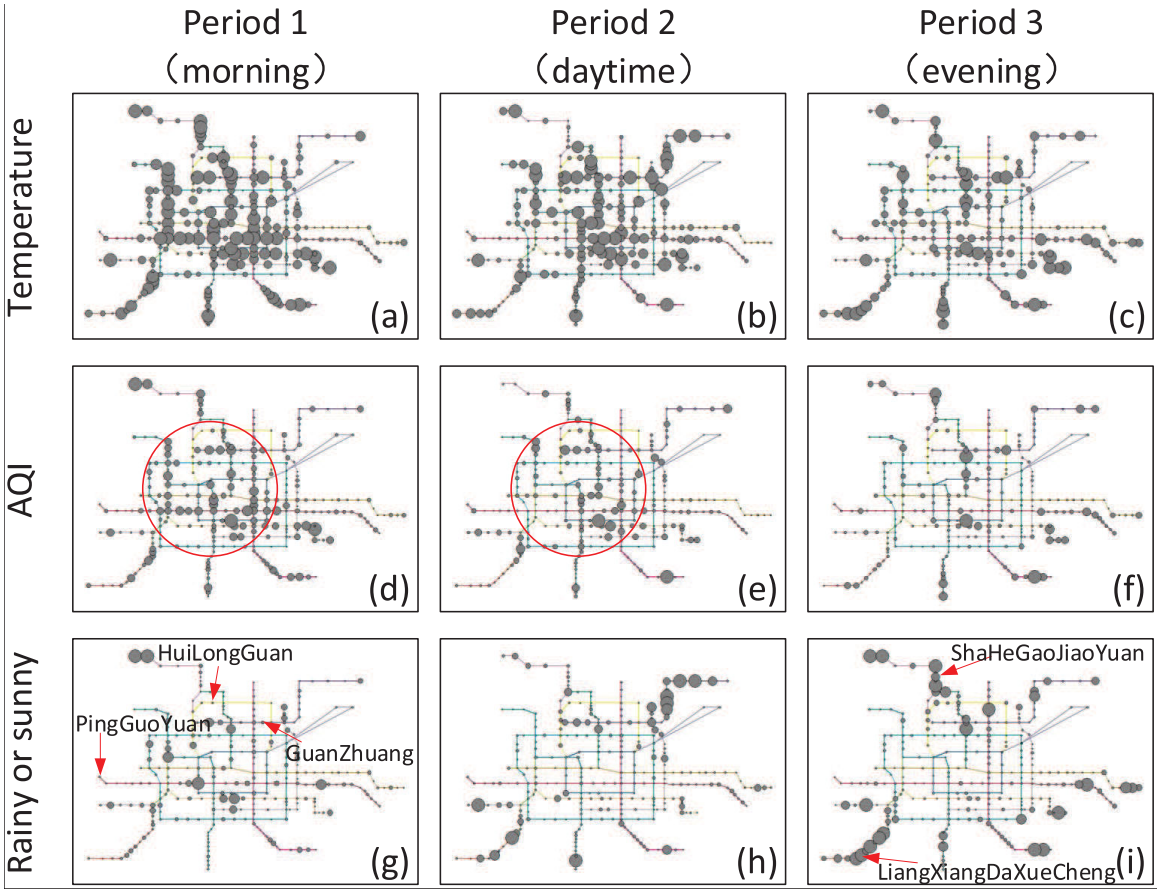


FIGURE 1. Variation of passenger flow distribution entropy in main subway stations under different weather conditions averaged on March to May. The larger radius means larger difference, which indicates stronger influence of the weather factor.

forecasting reports. More importantly, they have obvious influence on people's travel plans. For example, people are more likely to go out in mild weather and good air quality. So the weather has an overall effect on the passenger flow, especially the elastic passenger flow.

We study the weather influence by considering the distribution of passenger flow at different subway stations. For a subway station O, we define the destination distribution from station O to other stations as a vector $\boldsymbol{r}_{O,t} = [r_{1,t}, r_{2,t}, \ldots, r_{n,t}]$, where $r_{i,t}$ is the percentage of passengers who travel from station O to the i-th station at time period t among all passengers who started their travel from station O, and n is the total number of stations. Notice that passengers' destinations are known and only used in the after-event analysis, as they were recorded after passengers finish their travels. In real-time prediction, the destinations are unknown and need to be inferred. In this paper, we consider t as three different time periods in a day: *morning* (from 5:00 to 9:00); *daytime* (from 9:00 to 17:00); and *evening* (from 17:00 to 23:00). The entropy e of station O at t can be calculates as:

$$e_{O,t} = -\sum_{i=1}^{n} r_{i,t} \ln r_{i,t} \tag{1}$$

Entropy is a measure of uncertainty, and used here as an indicator of the randomness of the passenger flow distribution at a subway station at t. A station at t with larger entropy means the distribution of destinations is more chaos, passengers from which perform more flexible, unlikely to concentrate on some destinations.

By checking the weather at t, we can define different sets of station entropies such as

$$\boldsymbol{e}_{O,t}^{T_{high}} = \{e_{O,t} | \text{temperature at } t > \mu_{temperature}^{t} + 2^{\circ}C\},$$
$$\boldsymbol{e}_{O,t}^{T_{low}} = \{e_{O,t} | \text{temperature at } t < \mu_{temperature}^{t} - 2^{\circ}C\},$$

which are collections of station entropies when people travel in *warm* days and *cool* days ($\mu_{temperature}^{t}$ is the historic average temperature at t in the three months), respectively. Notice that t is a specific time period (e.g., *morning*, *daytime*, *evening*) in different days (March to May in our studied data sets).

Similarly, we can define the sets

$$\boldsymbol{e}_{O,t}^{A_{clean}} = \{e_{O,t} | \text{AQI at } t < 100\},$$
$$\boldsymbol{e}_{O,t}^{A_{polluted}} = \{e_{O,t} | \text{AQI at } t > 150\},$$

which are collections of station entropies when people travel in *clean* days and *polluted* days, respectively. The division

is according to *Technical Regulation on Ambient Air Quality Index (on trial)(HJ 633-2012)* in China, in which AQI < 100 means Good or Moderate, and AQI > 150 means Unhealthy.

$$\boldsymbol{e}_{O,t}^{W_{rainy}} = \{e_{O,t} | \text{it's rainy at } t\},$$
$$\boldsymbol{e}_{O,t}^{W_{sunny}} = \{e_{O,t} | \text{it's sunny at } t\},$$

which are collections of station entropies when people travel in *rainy* days and *sunny* days, respectively. In addition, other weather type such as cloudy or windy are also integrated to sunny. Passengers have to carry an umbrella during travelling outside, and some outdoor activities will be greatly affected by rainy days. As a result, rainy may cause obvious trouble to passengers and we put it as a special weather.

For each set, we can calculate its average $\bar{e}_{O,t}^{Weather}$, and measure the influence of different weather factors on the entropy of destination distribution from each station:

$$d_{O,t}^{T} = |\bar{e}_{O,t}^{T_{high}} - \bar{e}_{O,t}^{T_{low}}| \quad (2)$$
$$d_{O,t}^{A} = |\bar{e}_{O,t}^{A_{clean}} - \bar{e}_{O,t}^{A_{polluted}}| \quad (3)$$
$$d_{O,t}^{W} = |\bar{e}_{O,t}^{W_{sunny}} - \bar{e}_{O,t}^{W_{rainy}}| \quad (4)$$

Fig. 1 demonstrates the influence in different time periods and under different weather conditions in the whole Beijing subway network. The circle radius at each station indicates the magnitude of the difference. There are several interesting observations from Fig. 1. First, the temperature has the strongest influence on passengers' travel patterns, including the stations in central area and suburban lines, comparing to the other two factors. It seems that the temperature has an overall influences, however, there is also a hidden reason. Temperature varies obviously among different months. In March, it is usually a little cold outsides, while in May, it is rather hot. This temperature factor is mainly affected by the month. Therefore, the travel patterns maybe change monthly, and would repeat annually.

Second, air quality (AQI) has more significant influence in morning and daytime, especially in central area marked in red circle in Fig. 1(d)(e). There is a conjecture that can explain this phenomenon. People are more sensitive to air quality during the daytime. Moreover, the air pollution is more serious in the city center.

Third, rainy or sunny conditions affect passengers in suburban lines more than those in central area, especially in *evening* period. For example, the LiangXiangDaXueCheng station and the ShaHeGaoJiaoYuan station, marked in Fig. 1(i), are located at suburban line. These stations are near to several college campuses where people study and also live there. Passengers taking subway at these stations are mostly students who are not for commuting, but for other flexible demands. They usually take subway to other shopping and entertainment places, but can easily change their travel plans if weather is not good. Therefore, the passenger flow volumes of these stations are stronger correlated with weather factors, than other stations close to large residential areas such as HuiLongGuan, PingGuoYuan and GuanZhuang, marked in Fig. 1(g), where passengers need to commute on every workday regardless weather changes.

The above analysis results motivated us to build a model for learning the influence of weather factors on passenger flow volume. This model will be used for making real-time prediction of the passenger flow at t given the weather information at t.

B. OVERVIEW OF MetroEye

The overview of our proposed framework named "MetroEye" is shown in Fig. 2. Given the weather conditions, MetroEye is designed to monitor the real-time passenger flow volumes on the subway according to metro card swiping records only. In other words, real-time passenger flow prediction will be based only on the entrance information of passengers, without knowing their destinations and their preferences to travel paths in advance. MetroEye divides the real-time prediction task into three steps:

1) when knowing the current weather conditions, predict the destination of a passenger's trip, given her entrance record from the origin station;

2) given a pair of origin station (O) and predicted destination station (D) (or *OD-pair* for short) of a passenger, determine her travel path from O to D;

3) infer the exact position of a passenger at any time given her estimated travel path and the trains' running schedule. By positioning all passengers on the subway, MetroEye can finally "view" the real-time passenger flow continuously and report the passenger flow volume at each station.

The framework of MetroEye consists of two major parts, the offline system and the online system, as shown in Fig. 2. The offline system is mainly for the storage of historical data, the modeling of passenger flow volumes with time and weather, and the computation of travel path selection ratios. The online system is mainly for the selection of passengers' destinations and travel paths, and also the simulation of real-time passenger flows. In what follows, we describe the two systems in detail.

C. THE OFFLINE SYSTEM

The offline system contains various historical datasets about weather conditions, traffic volumes in each OD-pair (or *OD-flow* for short) collected after passengers completed their trips, train schedules, and basic subway structures. This part is mainly concerned with modeling the passenger flow volumes according to travel time and weather conditions, as well as the travel path selections (or *path ratio* for short) in each OD-pair.

As we know, an OD-flow changes in different time periods of a day, and can be influenced by various external factors such as the weather conditions. Thus the offline system builds a model for learning the influence of time (e.g., morning, daytime, or evening) and weather (including temperature, air quality and rainy/sunny condition) on the passenger flow volumes, based on historical data. The learned model will be used for the real-time prediction in the online system given

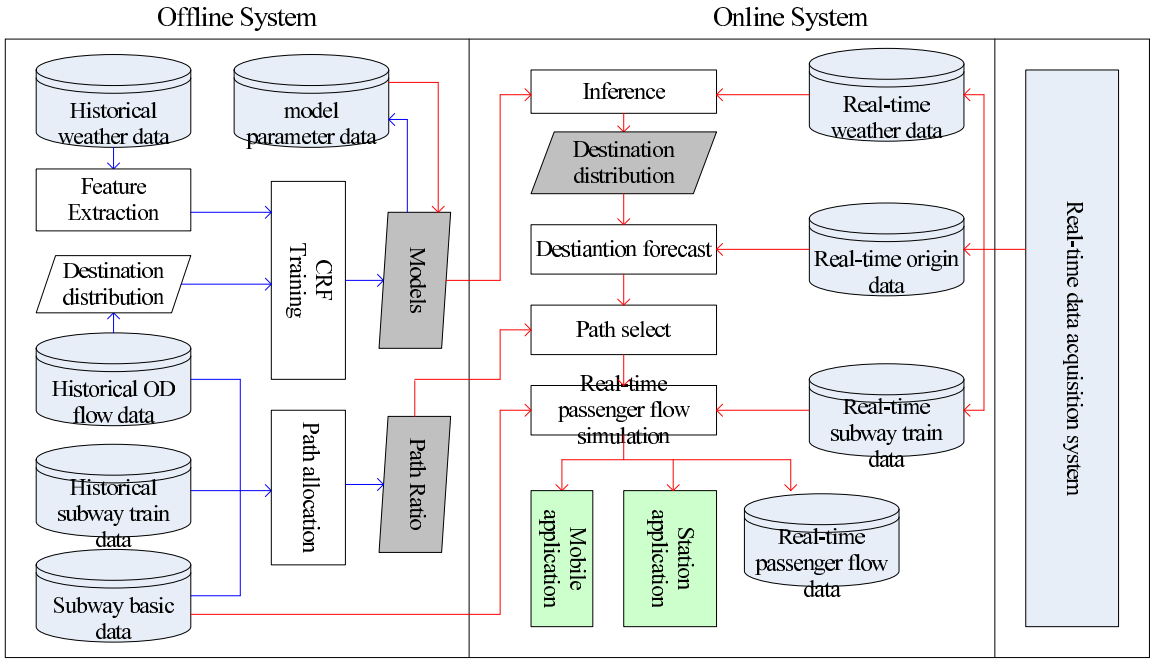


FIGURE 2. Framework of MetroEye.

the current time period and weather. Section IV will introduce the detailed model based on *conditional random fields* (CRF).

Considering that the weather factors, including temperature, air quality and rain/shine, may have an impact on the travel patterns, the model *conditional random fields* (CRF) [37] is adopted here to establish the relationship between passenger flow volume and the weather factors. The adoption of CRF also enables the modeling of interaction effect of traffic volumes between two consecutive time periods. Note that for each OD-pair in a specific time period, a CRF model will be built and saved to the offline system. We leave the details of the model to Section IV.

After predicting the destination, the travel path selected by the passenger should be determined. In Beijing subway network, passengers don't need to swipe cards when get transfer, and therefore the travel path is not collected directly in the data. So we have to infer the path from limited information. In this part, given an OD-pair, the offline system firstly detects the feasible paths from O to D, and then calculates the theoretical travel time for each path according to historical subway train schedules and network structures. Secondly, the system computes each passenger's historical real travel time from O to D, which is then compared to all paths' theoretical travel time to find which path is actually selected by that passenger. Then the system can aggregate the counts of passengers' path selection in this OD-pair, and finally calculate its path ratio by normalizing the counts among different paths. In addition, the path ratios are relatively stable when subway lines and train schedules are fixed. So they are only needed to calculate once in the offline system, which are then saved for the further use in the online system. The path ratios will be updated aperiodically when the subway system has a substantial change in metro lines or running schedules.

D. THE ONLINE SYSTEM

The main task of the online system is to simulate real-time passenger flow in the subway network. The implementation consists of four steps, i.e., destination ratio inference, destination selection, path selection, and real-time passenger flow simulation.

Firstly, given each origin station O, the CRF models learnt in the offline system are employed with real-time weather forecasting data to infer the passenger flow volumes to all possible destinations, which are then normalized to obtain the destination distribution (or *destination ratio* for short) of O. In this way, the online system can learn the real-time destination ratios for every origin station.

Secondly, passengers' real-time entrance data (including the origin station and the time) are acquired from the AFC system. For each passenger, randomly select her destination according to the destination ratio of the origin in the entering period. To further accelerate the destination selection for huge volumes of incoming passengers, a Huffman tree [39], [40] is adopted with technical details given in Section V.

Thirdly, given each simulated OD-pair, the online system randomly selects the travel path from the effective paths based on the path ratio computed in the offline system. This process is also accelerated by the Huffman tree to meet the computing challenge.

Finally, the simulation of all passengers' real-time positions on the subway is conducted according to the estimated travel paths, estimated transfer time, and the train

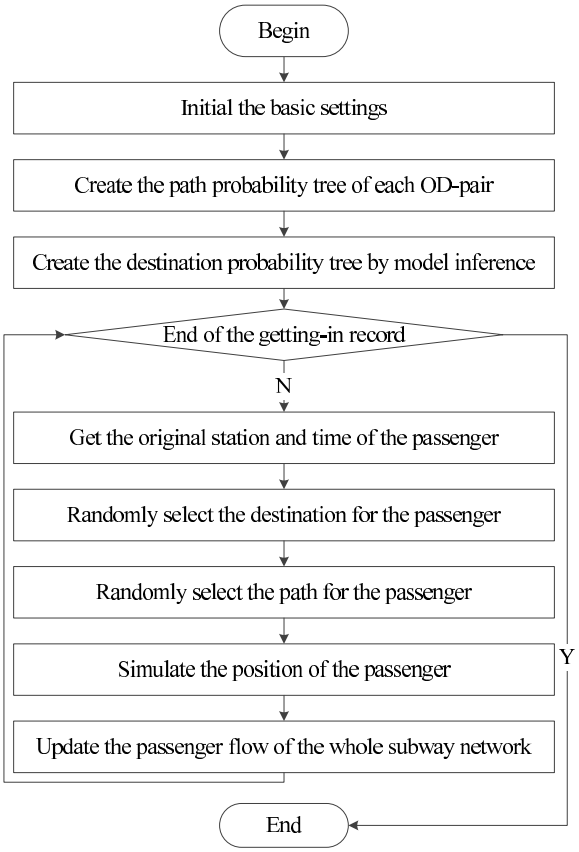


FIGURE 3. Flow chart of the online system.

schedules. The simulation is conducted periodically to reflect the dynamics of passenger flow volumes on the subway. Fig. 3 shows the running procedure of the online system.

E. DISCUSSIONS

The most attractive characteristic of MetroEye lies in that it provides a simple yet feasible way to monitor the real-time passenger flows on the subway given only the entrance information of passengers. The separate design of offline and online systems makes MetroEye effective on modeling the passenger flow w.r.t. time and weather and efficient on real-time positioning the massive passengers. More application details will be found in Section VII.

The practicality of MetroEye is under the support of the well-designed offline and online systems, which serve the goals of efficiently simulating passengers' travel paths from only their entrance information. In the offline system, the introduction of the CRF model enables the modeling of the weather influences to passengers' destination selections, which to our best knowledge is rare in most existing algorithms for real-time passenger flow prediction. Moreover, to speed up the simulation process for real-time positioning of massive passengers, MetroEye adopts some optimization strategies such as introducing the Huffman tree for both destination and path selections.

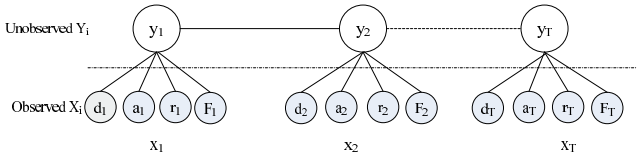


FIGURE 4. Presentation of the CRF model. The temperature *m*, air quality *a*, rainy or sunny *w* and historical passenger flow volume *p* are observed as *x*, while the real passenger flow volume *y* is output.

TABLE 1. The symbols in CRF model.

Symbol	Description
m	Temperature
a	Air quality index (aqi)
w	Weather (rainy or sunny)
p	Historical passenger flow volume
X	Observed factors
Y	Real passenger flow volume
s	Length of the sequence
v	Weather state (v_1=rain, v_2=shine)
k	Number of weather state (k=2)
l	Number of observed factors except rain state (l=3)
α, β	Weight parameters in potential function $g_t(y_t, x_t; \alpha, \beta)$
λ	Weight parameters in potential function $f_t(y_t, y_{t+1}, x_t; \lambda)$

IV. CRF MODEL IN THE OFFLINE SYSTEM

The CRF model in offline system is adopted to predict destination ratio of passenger flow given an origin station, and this part is divided into 3 steps. Firstly, the offline system uses CRF model to learn the relationship between historical passenger flow volume and other factors. Secondly, the online system put the weather forcast results into CRF model to infer the actual passenger flow volume. Finally, the passenger flow volume is normalized to obtain the destination ratio and then provided to online simulation. This CRF model considered the effect of weather condition to improve the accuracy of prediction. Next in this section, the representation, learning, and inference of the CRF model will be introduced.

A. MODEL REPRESENTATION

In this model, a linear-chain CRF is used to detect the relationship of passenger flow volume with the weather factors consisting of temperature, air quality index and rainy or sunny. For one OD's model in period t, the observed factor $\boldsymbol{x}_t = [m_t, a_t, w_t, p_t]$ is consisted of temperature m_t, air quality index a_t, weather w_t and historical passenger flow volume p_t. The output y_t is the real passenger flow volume. All the variables are normalized to [0, 1].

Fig. 4 shows the graphical structure G of the CRF model and Table 1 shows the description of the main symbols. In the graphical structure G, the gray nodes $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_s\}$ represent observe variables, while the white nodes $Y = \{y_1, y_2, \ldots, y_s\}$ represent the real passenger flow volume, s stands for the length of the sequence of a day.

Inspired by the practical application of continuous conditional random fields model [37], a real valued potential function $f_t(y_t, y_{t+1}, \boldsymbol{x}_t; \boldsymbol{\lambda})$ is used to model interactions among output Y. This function establishes the relationship

between two connected periods, the bigger value indicating stronger relationships. Simultaneously, the interaction between observed factors X and output Y is denoted by the function $g_t(y_t, \boldsymbol{x}_t; \boldsymbol{\alpha}, \boldsymbol{\beta})$.

Actually, the values of elements in vector Y and X are continuous except the value of rainy or sunny, so the discrete variable w_t is separated from other continuous variables in $\boldsymbol{x}_t$, that is $\tilde{\boldsymbol{x}}_t = [m_t, a_t, p_t]$. And an indicator function is introduced to show the effect of w_t as follows:

$$I(w_t = v_j) = \begin{cases} 1; & if\ w_t = v_j \\ 0; & if\ w_t \neq v_j \end{cases} \tag{5}$$

As a result, the feature functions $f_t(y_t, y_{t+1}, \boldsymbol{x}_t; \boldsymbol{\lambda})$ and $g_t(y_t, \boldsymbol{x}_t; \boldsymbol{\alpha}, \boldsymbol{\beta})$ can be expressed as equation (6) and (7). The output y_t is set to be quadratic to simplify the calculation. $\boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ are weight parameters in the functions.

$$\begin{aligned} &f_t(y_t, y_{t+1}, \boldsymbol{x}_t; \boldsymbol{\lambda}) \\ &= \exp(-\sum_{j=1}^{k} I(w_t = v_j)(\tilde{\boldsymbol{\lambda}}_t^j \tilde{\boldsymbol{x}}_t y_t y_{t+1})) \\ &= \exp(-\sum_{j=1}^{k} I(w_t = v_j)(\sum_{i=1}^{l} \lambda_{t,i}^j x_{t,i} y_t y_{t+1})) \end{aligned} \tag{6}$$

$$\begin{aligned} &g_t(y_t, \boldsymbol{x}_t; \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \exp(-\sum_{j=1}^{k} I(w_t = v_j)(\frac{1}{2}\tilde{\boldsymbol{\alpha}}_t^j \tilde{\boldsymbol{x}}_t y_t^2 - \tilde{\boldsymbol{\beta}}_t^j \tilde{\boldsymbol{x}}_t y_t)) \\ &= \exp(-\sum_{j=1}^{k} I(w_t = v_j)(\frac{1}{2}\sum_{i=1}^{l} \alpha_{t,i}^j x_{t,i} y_t^2 - \sum_{i=1}^{l} \beta_{t,i}^j x_{t,i} y_t)) \end{aligned} \tag{7}$$

Finally, the conditional probability expression of CRF model can be represented as:

$$\begin{aligned} &p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \\ &= \frac{1}{Z(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})} \prod_{t=1}^{s} g_t(y_t, \boldsymbol{x}_t; \boldsymbol{\alpha}, \boldsymbol{\beta}) \prod_{t=1}^{s-1} f_t(y_t, y_{t+1}, \boldsymbol{x}_t; \boldsymbol{\lambda}), \end{aligned} \tag{8}$$

where $Z(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})$ is a normalized factor.

Up till now, the relationship of weather factors and passenger flow sequence is represented by a quadratic function with parameters, then the parameters can be learned from historical data by the basic method of machine learning.

B. LEARNING METHOD

The learning task is to find the parameter values that maximize the conditional log-likelihood function LCL. The parameter set is $\Pi = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}\}$. Besides, regularization parameters $\{u_\alpha, u_\beta, u_\lambda\}$ are introduced to avoid the overfitting. In practical applications, these parameters can be set as $u_\alpha = u_\beta = u_\lambda$ for simplicity. So the LCL can be expressed as follows:

$$\begin{aligned} LCL = &\log p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) - \frac{u_\alpha}{2}\sum_{t=1}^{s}\sum_{j=1}^{k}\sum_{i=1}^{l}(\alpha_{t,i}^j)^2 \\ &- \frac{u_\beta}{2}\sum_{t=1}^{s}\sum_{j=1}^{k}\sum_{i=1}^{l}(\beta_{t,i}^j)^2 - \frac{u_\lambda}{2}\sum_{t=1}^{s-1}\sum_{j=1}^{k}\sum_{i=1}^{l}(\lambda_{t,i}^j)^2 \end{aligned} \tag{9}$$

Different from the classic discrete CRF models, in this model, the value of elements in vector Y and X are continuous, which brings a difficulty in calculating the value of $p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})$. To solve this problem, $p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})$ is assumed as multivariate Gaussian, and transformed to a multivariate Gaussian distribution by representing $Z(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})$ as follows:

$$\begin{aligned} &Z(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \\ &= \int \prod_{t=1}^{s} g_t(y_t, \boldsymbol{x}_t; \boldsymbol{\alpha}, \boldsymbol{\beta}) \prod_{t=1}^{s-1} f_t(y_t, y_{t+1}, \boldsymbol{x}_t; \boldsymbol{\lambda}) dy \\ &= \int \exp(-\sum_{t=1}^{s}\sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)(\frac{1}{2}\alpha_{t,i}^j x_{t,i} y_t^2 - \beta_{t,i}^j x_{t,i} y_t) \\ &\quad - \sum_{t=1}^{s-1}\sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)\lambda_{t,i}^j x_{t,i} y_t y_{t+1}) dy \\ &= \int \exp(-\frac{1}{2}\boldsymbol{y}^T A \boldsymbol{y} + b^T \boldsymbol{y}) dy \\ &= (2\pi)^{s/2} |A|^{-1/2} \exp(\frac{1}{2} b^T A^{-1} b), \end{aligned} \tag{10}$$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_s]$; A is an $s \times s$ matrix where $A_{t,t} = \sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)\alpha_{t,i}^j x_{t,i} (1 \leq t \leq s)$ and $A_{t,t+1} = A_{t+1,t} = \sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)\lambda_{t,i}^j x_{t,i}$, $(1 \leq t \leq s-1)$. The other entries of A are zero. b is a column vector and $b_t = \sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)\beta_{t,i}^j x_{t,i} (1 \leq t \leq s)$.

Note that maximization of LCL is a constrained optimization problem and it should be guaranteed that $A_{t,t} = \sum_{j=1}^{k}\sum_{i=1}^{l} I(w_t = v_j)\alpha_{t,i}^j x_{t,i} > 0$. Since $x_{t,i} > 0$, the learning only need to maintain $\alpha_{t,i}^j > 0$. To address this problem, $\log\alpha$ is adopted instead of α when maximize LCL. As a result, the new optimization issue becomes unconstrained and the gradients of LCL to $\log\alpha_{t,i}^j$, $\beta_{t,i}^j$ and $\lambda_{t,i}^j$ are given as:

$$\begin{aligned} &\frac{\partial LCL}{\partial \log \alpha_{t,i}^j} \\ &= -\alpha_{t,i}^j (\frac{\partial \log Z(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial \alpha_{t,i}^j} + I(w_t = v_j)\frac{1}{2} x_{t,i} y_t^2 + u_\alpha \alpha_{t,i}^j) \end{aligned} \tag{11}$$

$$\frac{\partial LCL}{\partial \beta_{t,i}^{j}} = -\left(\frac{\partial \log Z(\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial \beta_{t,i}^{j}} - I(w_t = v_j)x_{t,i}y_t + u_\beta \beta_{t,i}^{j}\right) \quad (12)$$

$$\frac{\partial LCL}{\partial \lambda_{t,i}^{j}} = -\left(\frac{\partial \log Z(\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial \lambda_{t,i}^{j}} + I(w_t = v_j)x_{t,i}y_t y_{t+1} + u_\lambda \lambda_{t,i}^{j}\right) \quad (13)$$

For each step, supposing the learning rate is η, the parameters $\log\alpha$, β and λ are updated as:

$$\log\alpha_{new} = \log\alpha_{old} + \eta \times \frac{\partial LCL}{\partial \log\alpha_{old}} \quad (14)$$

$$\beta_{new} = \beta_{old} + \eta \times \frac{\partial LCL}{\partial \beta_{old}} \quad (15)$$

$$\lambda_{new} = \lambda_{old} + \eta \times \frac{\partial LCL}{\partial \lambda_{old}} \quad (16)$$

Then let the parameter set $\Pi = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}\}$ reach the optimal through iteration of gradient ascent, the relationship of weather factors and passenger flow is confirmed.

C. MODEL INFERENCE

In online inference, the prediction is equal to find the optimal $\boldsymbol{y}$ that makes the model expression $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda})$ reach its maximum value with input weather factors.

$$\begin{aligned}\boldsymbol{y} &= \arg\max p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}) \\ &= argmax(-\frac{1}{2}\boldsymbol{y}^T A\boldsymbol{y} + b^T\boldsymbol{y}) \\ &= A^{-1}b\end{aligned} \quad (17)$$

In summarize, the whole prediction steps are as follows:

Step 1: (in offline system) normalize the training data: historical passenger flow volume and weather (temperature, aqi, rainy or sunny);

Step 2: random initialize $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}$;

Step 3: while (the max number of iterations is not reached)
maximize the log-likelihood *LCL*
update the gradients parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}$
if (log-likelihood has converged)
break
end while;

Step 3: output $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}$;

Step 4: (in online system) input the real weather of the day to predict real passenger flow volume $\boldsymbol{y}$ by equation (17);

Step 5: compute the destination ratio by normalize the passenger flow volume from the same origin station;

V. PATH SELECTION AND REAL-TIME SIMULATION IN ONLINE SYSTEM

As introduced in section III-D, MetroEye online system mainly involves 4 steps: destination ratio inference, destination selection, path selection and real-time passenger flow simulation. The first step can be easily conducted by applying the learned CRF model in the offline system with real-time weather conditions. Then for each origin station, we have a destination ratio vector indicating the probability of reaching other stations as destinations. Next in this section, we introduce the details of the other three steps to follow.

A. DESTINATION SELECTION

Given a passenger's real-time entrance data such as the origin station and entrance time, the online system will infer his/her destination according to the inferred destination ratio vector. As the selection occurs frequently in online system and the number of destination options is very large, we adopt a Huffman tree to make the sampling process efficient.

The steps to create Huffman tree are as follows:

Step 1: create the leaf nodes $Leaf_i$, the weight w_t^i of each node i is the probability to the station i, where $\sum_{i=1}^{N} w_t^i = 1$;

Step 2: sort the leaf nodes by the weight and put them into a queue;

Step 3: pick up the two leaf nodes $(Leaf_m, Leaf_n)$ of the least two weight $(w_t^m \le w_t^n)$, create an *InnerNode*, with a weight as the sum of the two leaf nodes, the left subtree is $Leaf_m$ and the right subtree is $Leaf_n$;

Step 4: put the *InnerNode* back into the queue according to the weight. Return to step 3 until the queue contains only one node.

The steps to sampling in the Huffman tree are as follows:

Step 1: for each passenger record of getting in, randomly generate a probability $p \in [0, 1]$ and initial the current state at the root of the Huffman tree;

Step 2: if $(p \le Node_{current}.P_{leftchild})$
set the current state at the left child node
else
set the current state at the right child node and
$p = p - Node_{current}.P_{leftchild}$
end if;

Step 3: return to step 2 until the current state is a leaf node.

Step 4: the station corresponding to the current state is the destination of the record.

B. PATH SELECTION

The selected destination and the given origin station form an OD-pair. Then path selection is to select the real path of the passenger between this OD-pair. However in subway network, the feasible paths are usually more than one. Actually, only k-shortest paths, which are called *candidate paths*, with fewer transfer will be adopted by a real passenger. So that path selection in online system is to select one real path from the candidate paths.

Now the task is to find all candidate paths and the probability to choose each path. In most cases, passengers usually adopt fixed paths. So the candidate path and the probability of each OD-pair almost stay the same, under the circumstances that the structure of subway network does't change a lot, and this will be done by *path allocation* in offline system. Path

allocation includes three basic steps as follows: calculate theoretical travel time; select candidate path; and path matching.

1) THEORETICAL TRAVEL TIME

When a passenger travel along a path, the theoretical travel time includes the time to platform, the time to wait for train, the time of train travels, and the time to get out. If the path includes transfer station, the theoretical travel time needs to add the time to get transfer platform and the time to wait again. In addition, if the passenger need to transfer for more times, the theoretical travel time is also need to add the transfer time for more times. The details of the each pieces of time can be get from the train schedule and the basic information of subway networks.

2) SELECT CANDIDATE PATH

Generally, not all the paths are feasible to be adopted by passengers. People don't like to transfer too many times, and a penalty factor will be added to the theoretical travel time when transfer occurs. So there is no need to find all the paths between each OD-pair. Instead, the offline system only selects k-shortest paths as candidate paths. In practice, $k = 10$ is enough.

3) PATH MATCHING

For each historical passenger travel record, just compare the real travel time and the theoretical travel time, and find the candidate path closest to the real travel time. This candidate path is selected to match the record. When all the records were matched to the candidate paths, the path ratio of each OD-pair is easy to calculated by statistic analysis.

C. REAL-TIME SIMULATION

The real-time simulation includes all the trains and passengers traveling in the subway system, so the simulation is composed by two threads. One thread is for passenger processing, dealing with the information of passenger getting in or out at each station according to AFC system. The other thread is for train processing, dealing with all the trains arriving at and leaving each station according to the schedule. Besides, each station has two queue buffers (two directions) for the passengers waiting for the trains, and each train has a maximal capacity of passengers.

1) PASSENGER THREAD

When a passenger getting into an origin station according to AFC system, select the destination and path, then put him/her into the buffer of origin station and wait for the train. When a passenger arriving at the destination according to AFC system, delete the passenger immediately.

2) TRAIN THREAD

When a train arriving at a station, drop the passenger to get transfer or reach the end, and deliver them to passenger thread; When a train leaving a station, take as many as passengers from the waiting buffer of the station. If a passenger has arrived at the destination while he/she doesn't get out according to AFC system, select a new destination again.

VI. EXPERIMENTAL EVALUATION OF CRF MODEL

The experiments in this section are for validating the destination distribution prediction made by the CRF model, which is introduced in the previous section IV. Classical linear and non-linear algorithms are compared as baseline methods.

A. EXPERIMENTAL SETUP

1) DATA

The experimental data was the passenger flow data of Beijing subway in the working days from Mar. 1st to May 20th, 2016. The dataset included more than 500 million trips, and the count of passengers every 5 minutes for each OD-pair stations. The evaluation was made on the prediction tasks: predicting the passenger flow volumes on the working days from May 9th to May 20th based on the past 8 weeks data. For example, to predict passenger flow volumes on May 11th (rainy, AQI = 187) based on the past 8 weeks (March 16th to May 10th). The prediction error (the difference between the predicted volume and the after-event collected data) will be reported. The total flow volume and weather factors is shown in Table 2 left part. The temperature of these days were getting warmer, especially little hot in May.

2) BASELINES

The proposed model is compared with classical linear baseline autoregressive moving average algorithm (ARMA) [6], nonlinear baseline support vector machine algorithm (SVM) [14], artificial neural network algorithm (ANN) [12], and Long Short-Term Memory algorithm (LSTM) [22]. In order to evaluate of importance of considering the weather effect, a default conditional random field algorithm (CRF0) is conducted, in which the weather factors are set as default value: zeros (isolate weather factors).

3) METRICS

From the after-events collected data, we can have the ground truth of passenger flow volume in period t starting from a station O to each of the potential destination station, denoted by $\boldsymbol{y}_O^t = \{y_{O,1}^t, y_{O,2}^t, \ldots, y_{O,n}^t\}$. Denoting the predicted passenger flow volume in period t from a station O to other stations $\hat{\boldsymbol{y}}_O^t = \{\hat{y}_{O,1}^t, \hat{y}_{O,2}^t, \ldots, \hat{y}_{O,n}^t\}$. The prediction performance is evaluated by four metrics:

Mean Relative Error (MRE): The average relative error is a measure of the average of relative errors between the predicted passenger flow values $\hat{y}_{O,i}^t$ and the real passenger flow values $y_{O,i}^t$. For one station O (e.g., the k-th station), it is calculated as:

$$MRE_k^t = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_{k,i}^t - y_{k,i}^t|}{y_{k,i}^t} \tag{18}$$

Root Mean Square Error (RMSE): Root mean square error is the square root of the mean of the quadratic sum of the

TABLE 2. Comparison of the performance of evaluated algorithms.

Date	Flow Volume	Temperature	AQI	Weather	ARMA	SVM	ANN	LSTM	CRF0	CRF
May.9	5775587	$12 \sim 21°C$	58	sunny	84.19%	86.39%	81.53%	**87.12%**	86.50%	*87.01%
May.10	5916330	$16 \sim 17°C$	60	sunny	84.31%	87.27%	81.03%	*87.36%	86.00%	**87.61%**
May.11	5848701	$18 \sim 29°C$	187	rainy	83.23%	86.03%	80.54%	86.21%	*86.40%	**86.74%**
May.12	5752687	$9 \sim 23°C$	68	sunny	84.37%	87.12%	81.21%	86.47%	*87.21%	**87.92%**
May.13	6174111	$14 \sim 25°C$	55	rainy	83.09%	86.56%	81.68%	*87.07%	86.15%	**87.19%**
May.16	5928394	$16 \sim 29°C$	52	sunny	84.32%	**87.07%**	80.46%	86.76%	86.67%	*87.05%
May.17	5923300	$17 \sim 31°C$	90	sunny	83.18%	86.51%	80.49%	86.39%	*86.88%	**87.21%**
May.18	5945194	$18 \sim 28°C$	91	sunny	83.97%	86.07%	81.75%	**86.91%**	86.75%	*86.79%
May.19	5892702	$15 \sim 29°C$	124	sunny	83.97%	86.64%	79.98%	86.31%	*86.80%	**86.94%**
May.20	6281818	$15 \sim 30°C$	124	sunny	83.04%	86.68%	81.70%	86.30%	*87.11%	**87.48%**
AVERAGE	-	-	-	-	83.76%	86.63%	81.04%	*86.69%	86.65%	**87.19%**

difference between the predicted passenger flow values $\hat{y}^t_{O,i}$ and the real passenger flow values $y^t_{O,i}$. For one station O (e.g., the k-th station), it is defined as:

$$RMSE^t_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^t_{k,i} - y^t_{k,i})^2} \tag{19}$$

Absolute Error Sum (AES): Absolute error sum measures the absolute difference between the predicted passenger flow values $\hat{y}^t_{O,i}$ and the real passenger flow values $y^t_{O,i}$. For one station O (e.g., the k-th station), it is calculated as:

$$AES^t_k = \sum_{i=1}^{n}|\hat{y}^t_{k,i} - y^t_{k,i}| \tag{20}$$

Accuracy: Accuracy measures the accuracy of the prediction $\hat{y}^t_{O,i}$ to the real passenger flow values $y^t_{O,i}$. For one station O (e.g., the k-th station), it is calculated as:

$$Accuracy^t_k = 1 - \frac{AES^t_k}{\sum_{i=1}^{n} y^t_{k,i}} \tag{21}$$

By taking the average of MRE, RMSE and the sum of AES on all stations, we have the final performance metrics defined as:

$$MRE^t_{mean} = \frac{1}{n}\sum_{k=1}^{n} MRE^t_k \tag{22}$$

$$RMSE^t_{mean} = \frac{1}{n}\sum_{k=1}^{n} RMSE^t_k \tag{23}$$

$$AES^t_{sum} = \sum_{k=1}^{n} AES^t_k \tag{24}$$

$$Accuracy^t = 1 - \frac{AES^t_{sum}}{\sum_{k=1}^{n}\sum_{i=1}^{n} y^t_{k,i}} \tag{25}$$

Notice that the sum of AES indicates the total error of the subway network. Small values of MRE^t_{mean}, $RMSE^t_{mean}$, and AES^t_{sum} indicate good performance of the evaluated approaches.

Results of proposed method and different baseline methods will be reported in next subsection. All the The experiments were conducted in a standard desktop computer, with double core i7-4790 3.60GHz CPU, 8GB RAM.

B. EVALUATION RESULTS AT ALL STATIONS

Table 2 shows the results of the proposed CRF algorithm and the baseline algorithms when predicting the passenger flow volumes in working days from May 9th to 20th, 2016. The best results are highlighted in bold, and the second best is marked with stars. CRF model have the best performance, especially in bad weathers. LSTM is the second best model, which is a little better than CRF0. CRF0 and CRF employ a similar model, while CRF considers weather factor and CRF0 does not. Comparing the performance of CRF0 and CRF, we can see that the consideration of weather factor effectively increase the prediction accuracy. The following analyses focus on May 11th and 19th, the two days with bad weathers.

Table 3 shows the results at different time periods of May 11th and 19th, 2016. The accuracy of CRF in the three periods of the two days are:88.16%, 84.27%, 87.86%; 88.70%, 84.33%, 87.88%, always performs the best. For more details, in May 11th, the rainy weather and the bad air quality disturbed the prediction of the passenger flow. All the algorithm performs a little worse than that in May 19th. Actually, the weather situation in May 11th does not frequently appear, so the travel mode is relatively harder to predict. The CRF algorithm performed better than other algorithms in such day.

Fig. 5 shows the distribution of accuracy for three periods in May 11th 2016 with different methods. For most stations, the accuracy is distributed from 70% to 95%. CRF (red), CRF0 (blue), LSTM (black) and SVM (green) performs much better than other methods, and CRF is the best for most cases.

C. EVALUATION RESULTS AT SELECTED STATIONS

The influence of weather on passenger flow at different stations varies according to the location of these stations. The passenger flows of the stations in residential or working region usually hold steady regardless the weather condition. However, the entertainment districts are influenced greatly. We select three stations for evaluating and comparing the prediction performance of different algorithms. The selected subway stations are GuanZhuang station (in residential

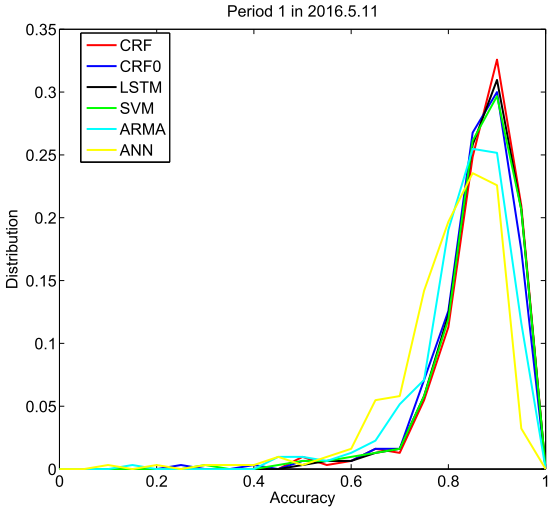
TABLE 3. Comparison of the performance in bad weathers.

Period Flow Volume	Algorithm	MRE	RMSE	AES	Accuracy
5.11-1 1827612	ARMA	0.3405	6.1993	260831	85.73%
	SVM	0.2836	5.4402	222263	87.84%
	ANN	0.3973	6.9380	300810	83.54%
	LSTM	*0.2819	*5.3588	*218347	*88.05%
	CRF0	0.2826	5.8606	229414	87.45%
	CRF	**0.2736**	**5.3140**	**216345**	**88.16%**
5.11-2 1982932	ARMA	0.4232	8.3703	398581	79.90%
	SVM	0.3435	7.5173	331099	83.30%
	ANN	0.5054	9.6265	471999	76.20%
	LSTM	0.3412	6.7681	320034	83.86%
	CRF0	*0.3408	**6.3853**	*313563	*84.19%
	CRF	**0.3353**	*6.5754	**311973**	**84.27%**
5.11-3 2038157	ARMA	0.4250	7.5670	321229	84.24%
	SVM	0.3008	5.5808	263724	87.06%
	ANN	0.4331	7.3878	365329	82.08%
	LSTM	0.3159	5.7320	268360	86.83%
	CRF0	*0.2966	*5.1060	*252611	*87.61%
	CRF	**0.2895**	**5.0393**	**247474**	**87.86%**
5.19-1 1828745	ARMA	0.3256	5.1762	239255	86.92%
	SVM	0.2812	4.4530	213167	88.34%
	ANN	0.4034	6.1274	295424	83.85%
	LSTM	0.2734	4.4324	210993	88.46%
	CRF0	*0.2709	*4.4101	*208733	*88.59%
	CRF	**0.2689**	**4.3815**	**206709**	**88.70%**
5.19-2 1977959	ARMA	0.4312	8.1435	381733	80.70%
	SVM	0.3414	**6.3415**	314113	84.12%
	ANN	0.5158	9.6725	480498	75.71%
	LSTM	0.3559	6.7732	332859	83.17%
	CRF0	*0.3351	6.5179	*313696	*84.14%
	CRF	**0.3344**	*6.4332	**309955**	**84.33%**
5.19-3 2085998	ARMA	0.3674	6.9914	323875	84.47%
	SVM	0.2989	5.1976	260168	87.53%
	ANN	0.4539	8.5190	403626	80.65%
	LSTM	0.3182	5.3087	262877	87.40%
	CRF0	*0.2908	*5.1529	*255346	*87.76%
	CRF	**0.2896**	**5.1132**	**252729**	**87.88%**

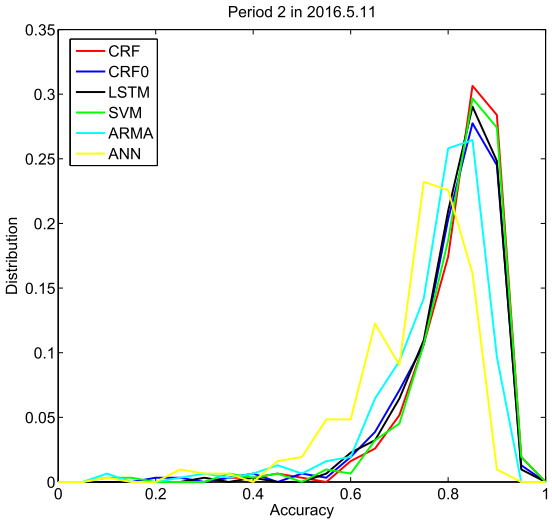
district), Agricultural Exhibition Center station (in entertainment district) and Biomedical Base station (in technology park).

The prediction result of GuanZhuang station is shown in Table 4. Since it is in a residential district, the inflow in the *morning* (period 1) is the largest of the all day. All algorithms thus perform quite good, comparing to the period 2 and 3 (*daytime* and *evening*). Especially in the evening when the inflow sharply decreases, the performance of compared algorithms decreases to different extents. However, CRF model significantly outperforms other baseline methods. This is because the passengers in GuanZhuang station in the evening have flexible travel demands and easier get affected by weather conditions. CRF model takes into consideration of the weather influence, and thus performs better on predicting the passenger flow volume.

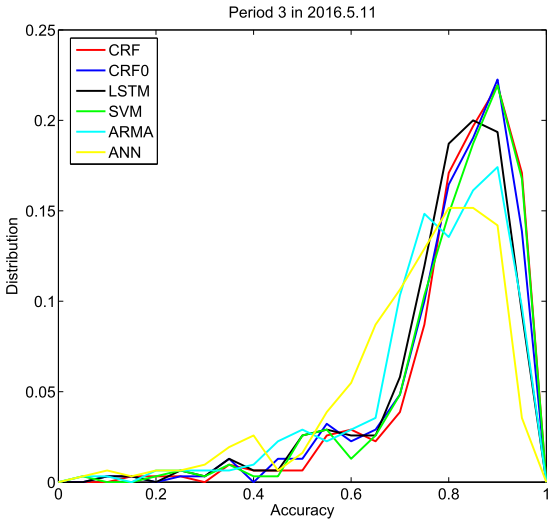
Table 5 shows the result of Agricultural Exhibition Center station, which is in an entertainment district, and usually attracts more passengers in the afternoon, and has a peak in the evening. Passengers gathered at this station has flexible and various travel purposes, which are easy to be affected by different factors and thus difficult to be predicted. CRF still demonstrates better performance than other baseline methods. More interestingly, the 18th Beijing international toys



(a) Distribution in period 1



(b) Distribution in period 2



(c) Distribution in period 3

FIGURE 5. Distribution of accuracy for each method in May 11th 2016. The CRF model (red) preforms the best.

TABLE 4. The result of GuanZhuang station in bad weathers.

Period Flow Volume	Algorithm	MRE	RMSE	AES	Accuracy
5.11-1 10093	ARMA	0.2439	5.7908	1004	90.05%
	SVM	0.2249	6.0622	973	90.36%
	ANN	0.2889	6.1233	1104	89.07%
	LSTM	*0.1983	*5.6803	*923	*90.86%
	CRF0	0.1994	5.9309	956	90.53%
	CRF	**0.1962**	**5.1734**	**897**	**91.11%**
5.11-2 5900	ARMA	0.4140	6.5060	1115	81.10%
	SVM	0.3725	5.5758	938	84.10%
	ANN	0.4880	7.1455	1304	77.89%
	LSTM	0.3836	5.8218	945	83.98%
	CRF0	*0.3453	*5.3919	*914	*84.50%
	CRF	**0.3381**	**5.2165**	**892**	**84.89%**
5.11-3 2627	ARMA	0.4605	3.6630	643	75.53%
	SVM	0.4040	3.5387	599	77.21%
	ANN	0.5217	4.7591	781	70.25%
	LSTM	0.3829	3.3290	584	77.77%
	CRF0	**0.3637**	*3.2907	*561	*78.65%
	CRF	*0.3666	**3.1943**	**533**	**79.73%**
5.19-1 10217	ARMA	0.2640	7.2894	1116	89.07%
	SVM	0.2660	*5.6133	944	90.76%
	ANN	0.3279	9.0073	1371	86.58%
	LSTM	0.2723	5.7820	956	90.64%
	CRF0	*0.2542	5.8001	*937	*90.83%
	CRF	**0.2496**	**5.5082**	**923**	**90.96%**
5.19-2 6168	ARMA	0.3776	6.9955	1170	81.03%
	SVM	0.3249	5.8793	984	84.04%
	ANN	0.5010	8.9222	1548	74.90%
	LSTM	0.3429	5.9923	996	83.85%
	CRF0	*0.3141	*5.8084	*982	*84.08%
	CRF	**0.3052**	**5.6378**	**954**	**84.53%**
5.19-3 2666	ARMA	0.4350	3.7342	654	75.47%
	SVM	0.4262	*3.3667	598	77.56%
	ANN	0.5253	4.9978	870	67.37%
	LSTM	0.3882	3.5083	574	78.47%
	CRF0	**0.3733**	3.3706	*566	*78.79%
	CRF	*0.3766	**3.2654**	**561**	**78.94%**

TABLE 5. The result of agricultural exhibition center station in bad weathers.

Period Flow Volume	Algorithm	MRE	RMSE	AES	Accuracy
5.11-1 1308	ARMA	0.4379	2.2403	391	70.10%
	SVM	0.3372	2.3041	403	69.22%
	ANN	0.4648	2.6632	464	64.51%
	LSTM	0.3523	2.4339	380	70.95%
	CRF0	*0.3210	*2.1984	*375	*71.35%
	CRF	**0.3193**	**2.0370**	**354**	**72.91%**
5.11-2 5224	ARMA	0.6143	11.9331	1891	63.79%
	SVM	0.6041	11.7998	1699	67.47%
	ANN	0.7537	13.9767	2258	56.78%
	LSTM	*0.5083	*11.4828	*1635	*68.70%
	CRF0	0.5124	11.6308	1697	67.52%
	CRF	**0.4905**	**11.3322**	**1610**	**69.18%**
5.11-3 3583	ARMA	0.4262	5.9709	985	72.52%
	SVM	0.3634	**4.9362**	826	76.95%
	ANN	0.5551	6.2859	1089	69.61%
	LSTM	0.3723	5.1289	833	76.75%
	CRF0	*0.3582	4.9787	*811	*77.34%
	CRF	**0.3497**	*4.9461	**808**	**77.44%**
5.19-1 1319	ARMA	0.3633	2.3557	384	70.89%
	SVM	0.3328	2.4554	409	69.01%
	ANN	0.3978	2.7504	462	64.97%
	LSTM	0.3387	2.4648	410	68.92%
	CRF0	*0.3010	*2.1572	*340	*74.22%
	CRF	**0.2986**	**2.1481**	**336**	**74.55%**
5.19-2 2571	ARMA	0.5933	8.5897	1059	58.82%
	SVM	0.5392	*4.6804	824	67.97%
	ANN	0.6210	7.0823	1126	56.19%
	LSTM	0.5282	5.1283	813	68.38%
	CRF0	**0.4686**	5.0105	*802	*68.81%
	CRF	*0.4809	**4.6624**	**799**	**68.92%**
5.19-3 3632	ARMA	0.4598	5.6506	983	72.94%
	SVM	0.3692	*3.9143	763	78.98%
	ANN	0.5451	7.3263	1195	67.10%
	LSTM	0.3522	4.0123	758	79.13%
	CRF0	*0.3498	3.9679	*738	*79.69%
	CRF	**0.3378**	**3.9065**	**730**	**79.91%**

& preschool tools exhibition was held during May 11th to 13th just near the station. Hence, the total inflow in period 2 on May 11th was much bigger than that on May 19th. The prediction of CRF model is still better than others in this situation.

Table 6 shows the prediction performance on Biomedical Base station, which is near to the end of a suburban line (DaXing line). The passenger flow volumes at stations of suburban lines are small in general. However, the variance of the distribution is relative large. Biomedical Base station is one of the stations in suburban line with the mixed feature of residential and working district. The destination distribution from this station often has large variance and can be greatly influenced by weather factors. The result shows that CRF algorithm performs better in the suburban stations than all other methods.

In Table 4, 5 and 6, we can see that CRF often has the best performance, and better than LSTM and SVM. The evaluation days May 11th and 19th are polluted days with different AQI values. The more accurate prediction of CRF on these days verifies that CRF effectively learns the influence of weather factors and guarantees that MetroEye will make accurate prediction of real-time passenger flow volume.

D. PREDICTION PERFORMANCE W.R.T. PASSENGER FLOW VOLUMES

It is interesting to study how the prediction performance varies w.r.t. the real passenger flow volumes at different stations. In other words, we evaluate whether stations with a large amount of passengers are easier or more difficult to predict than those with a small number of passengers. The x-axis in Fig. 6 are the index of stations ordered by their real passenger flow volumes, which is shown as the blue curve (with scales on the left y-axis). The AES in three periods on May 11th 2016 is also shown in the figure as the green curve (with scales on the left y-axis) to be compared with passenger flow volumes directly. The prediction accuracy of CRF model in the same period is shown as the red curve (with scales on the right y-axis).

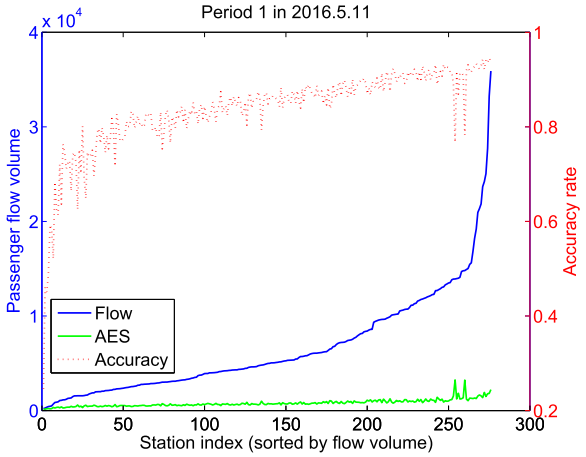
From Fig. 6, we can see that the accuracy rate increases when stations have a larger number of passengers. Especially, for the large stations with more than ten thousand passenger flow volume in the period, the CRF model has a high accuracy of prediction. It has great significance for subway systems. The large stations usually have a large scale of the passenger flow, which would lead to serious congestion in the downstream station especially in rush hours. Therefore, the accurate prediction for the large stations is essential for

TABLE 6. The result of biomedical base station in bad weathers.

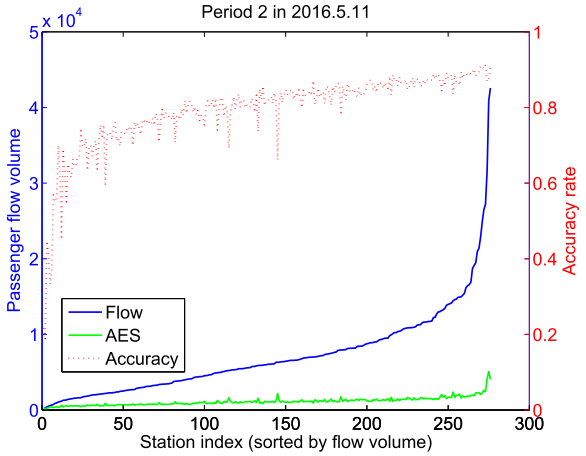
Period Flow Volume	Algorithm	MRE	RMSE	AES	Accuracy
5.11-1 4879	ARMA	0.3226	4.4236	783	83.94%
	SVM	0.2531	3.6627	685	85.96%
	ANN	0.3426	4.7944	866	82.26%
	LSTM	*0.2487	*3.5008	*641	*86.86%
	CRF0	0.2556	3.5264	661	86.46%
	CRF	**0.2406**	**3.4661**	**633**	**87.02%**
5.11-2 3223	ARMA	0.5078	5.2628	923	71.36%
	SVM	*0.4093	*3.9652	*759	*76.44%
	ANN	0.5767	5.4551	1064	66.97%
	LSTM	0.4317	4.0123	766	76.23%
	CRF0	0.4346	4.0200	769	76.14%
	CRF	**0.4043**	**3.8784**	**738**	**77.09%**
5.11-3 2049	ARMA	0.3852	2.8686	550	73.18%
	SVM	0.3558	*2.8518	*520	*74.62%
	ANN	0.4826	3.8657	697	65.99%
	LSTM	0.3662	3.1283	533	73.99%
	CRF0	*0.3401	2.9945	524	74.40%
	CRF	**0.3164**	**2.6024**	**467**	**77.22%**
5.19-1 4830	ARMA	0.2986	4.4396	696	85.59%
	SVM	**0.2358**	3.5799	635	86.85%
	ANN	0.3826	5.3966	902	81.33%
	LSTM	0.2502	3.5538	633	86.89%
	CRF0	0.2472	**3.4945**	*627	*87.02%
	CRF	*0.2464	*3.4948	**616**	**87.24%**
5.19-2 3290	ARMA	0.6308	6.6986	1043	68.29%
	SVM	**0.4193**	4.6367	*736	*77.63%
	ANN	0.6493	7.5663	1193	63.72%
	LSTM	0.4278	4.7232	741	77.48%
	CRF0	*0.4260	*4.6114	740	77.52%
	CRF	0.4331	**4.5106**	**725**	**77.96%**
5.19-3 2294	ARMA	0.4596	**3.4296**	*630	*72.53%
	SVM	0.4165	4.0203	692	69.84%
	ANN	0.5154	4.3397	795	65.33%
	LSTM	*0.3772	3.7183	635	72.32%
	CRF0	0.3810	3.7715	640	72.12%
	CRF	**0.3794**	*3.5704	**615**	**73.17%**

the whole subway system. Nevertheless, CRF model still has some weakness in prediction for the small stations with less than one thousand passenger flow volume in the period. The accuracy rate is usually less than 50% in these small stations. Considering that there are 276 stations in Beijing subway network, the passenger flows from the small stations are less than 4 on average for each OD-pair. Thus, the error of prediction is too small to cause a serious problem on the whole network.

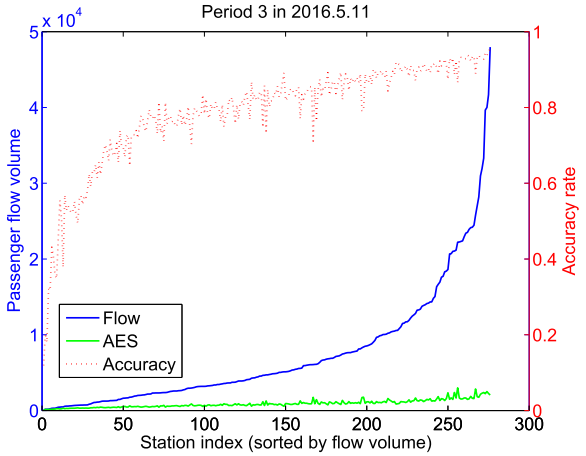
There are two exceptional large stations on which the prediction accuracy is low in period 1 (after Index 250 at x-axis shown in Fig. 6(a)). They are *Beijing West Railway Station* and *Beijing Railway Station*, where passengers arrived from other cities by over-night trains and might have high uncertainty on their destinations in Beijing. Thus, the prediction of their destination distribution is not as accurate as other large stations. Predictions in period 2 and 3 for these two stations are not obviously bad, which because the passengers have less choices for destination. Especially in the evening, the only choice may be to go home.



(a) Real flow volume and prediction accuracy in period 1



(b) Real flow volume and prediction accuracy in period 2



(c) Real flow volume and prediction accuracy in period 3

FIGURE 6. The variation of prediction performance at stations with different passenger flow volumes (from small to large on x-axis).

VII. APPLICATIONS OF MetroEye

There are some details in online simulation which will not be illustrated in this paper, such as destination selection, path selection, and real-time simulation. This section only introduce the application of MetroEye system to show the practicability of it.

The MetroEye framework has been adopted by the Beijing Urban Rail Transit Control Center (TCC) to monitor the

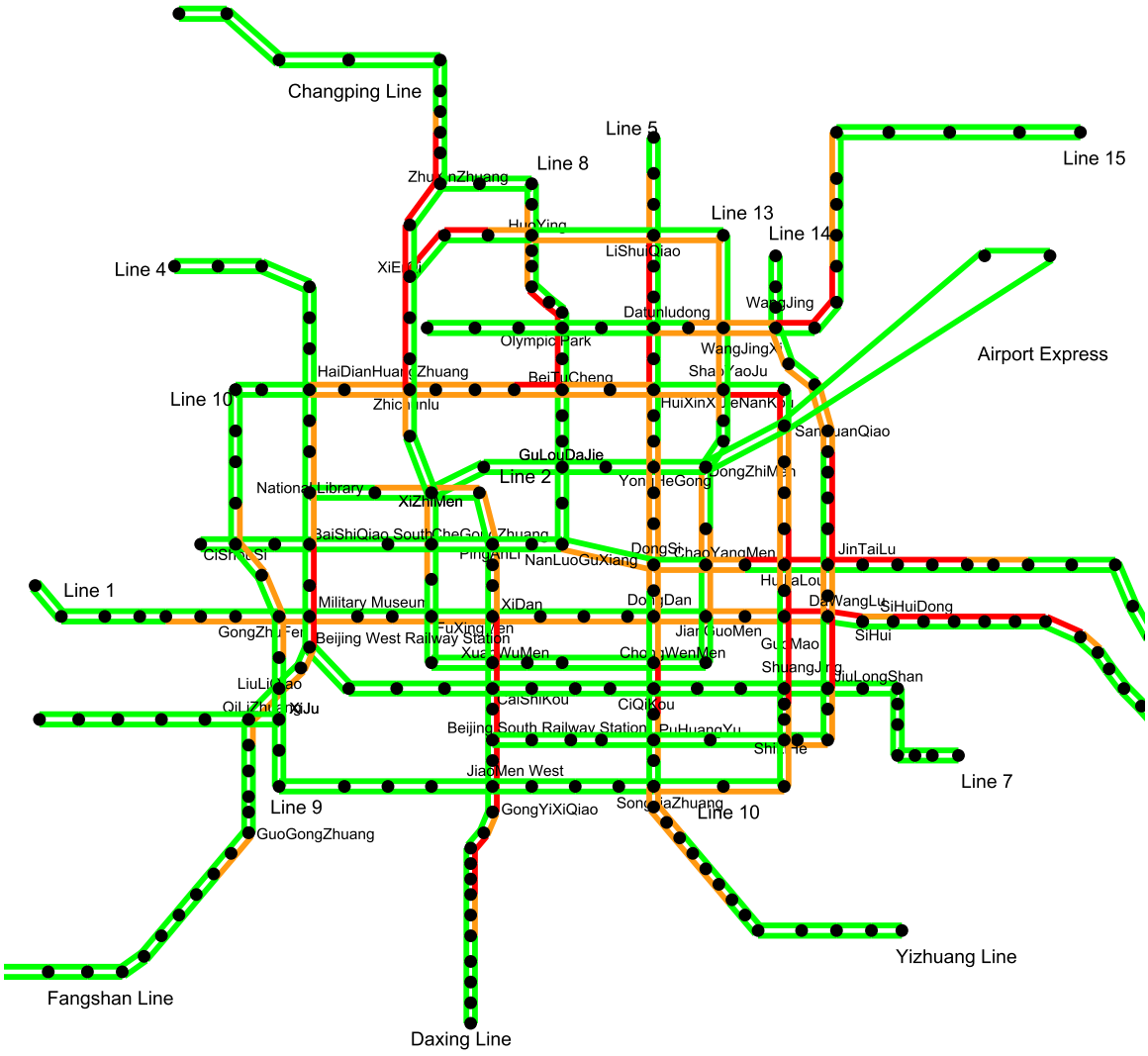


FIGURE 7. An example of real-time passenger flow prediction of Beijing subway network at 8:30 on May 11th 2016. The passenger flow load level is marked on the right side of each direction by green (free), yellow (crowded) and red (overload) according to the prediction and simulation. Only line names and transfer station names are marked. The sections from the residential regions to the working regions are crowded or even overload.

real-time network status. The information of real-time passenger flow volume will be published to the station software application and mobile software application.

An example of real-time passenger flow prediction at 8:30am on May 11th 2016 is conducted and shown in Fig. 7. For visualizing the network traffic status in an easy-to-read manner, we publish the load level categorized as 'free' (less than 0.5), 'crowded' (0.5-0.8) and 'overload' (over 0.8), rather than the real number of passengers at stations and sections between stations.

The load level is calculated as the average number of passengers on the trains passing the section divided by the maximal capacity of the train during a piece of time. In the network shown in Fig. 7, the load level of each station and each section between two stations are displayed on the right side of each direction of each line, in color green (free), yellow (crowded) or red (overload).

As we can see, the sections from the residential regions (e.g., HuoYing) to the working regions (e.g., HaiDian-HuangZhuang) are crowded or even overload at 8:30am, which matches the impression of people. However, not all stations in the subway network are crowded and overload. Massive passengers gathered only at some large stations (e.g., GuoMao) and the downstream stations (e.g., the stations after GuoMao in Line 1 and Line 10.) for commuting. Thus, it is of great importance to accurately predict the passenger flow in the large stations, which is just the advantage of the MetroEye model.

Notice that the dataset used for evaluation in this paper are from weekdays. In real application of MetroEye, two different offline models are built for weekday and weekend traffic flow. This is because the feature of passenger flow in weekday is totally different from that in weekends.

VIII. CONCLUSION

Real-time passenger flow prediction is playing an important role in subway management. This paper proposed a systematic framework, MetroEye, to predict real-time passenger flow in subway system with weather awareness. The framework consists of an offline system and an online system. The offline system is mainly for OD-flow modeling, while the online system is for real-time simulation. Weather conditions also have influence on people's travel plans. Especially bad air quality and bad weathers have drawn much attention of the urban residents and make people re-plan their non-urgent and unnecessary travels.

A conditional random field model based on weather factors is proposed in the offline system, aiming at establishing the relationship between passenger flow volume and weather factors. Experimental results proved that the conditional random field model has higher accuracy among the compared algorithms. Especially for the large stations with over 10 thousands passengers, the accuracy is quite high, which is significant in subway passenger flow prediction.

The online system proposes a practical way to select the destination, path and conduct the real-time simulation. An example of real-time simulation on Beijing subway network is provided for showing the practicability of MetroEye. The offline-online system structure is also scalable for other prediction method. Each module's algorithm can be updated independently. The MetroEye framework has been adopted by the Beijing Urban Rail Transit Control Center.

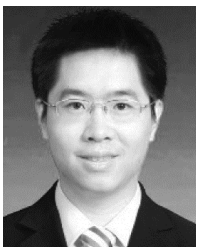
As a result of the limitation of the used data set, this paper concentrates on the prediction of working days. However, the weather effect on weekends/holidays can be totally different. It will be very interesting to analyze the difference of working days and weekends/holidays in the future.

REFERENCES

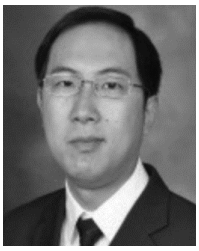
[1] A. Nuzzolo and A. Comi, "Advanced public transport and intelligent transport systems: New modelling challenges," *Transportmetrica A, Transp. Sci.*, vol. 12, no. 8, pp. 674–699, Sep. 2016.
[2] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, "A real-time passenger flow estimation and prediction method for urban bus transit systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3168–3178, Nov. 2017.
[3] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.
[4] H. Ye, Y. Hao, and H. Zhu, "'N-day' average volume based analysis and forecasting for daily passenger flow of Shanghai URT," in *Proc. Int. Conf. Transp. Eng.*, Jul. 2009, pp. 4080–4085.
[5] H.-Y. Zhu, "N-day average volume based time-series analysis for passenger flow of metro," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur. (MINES)*, 2010, pp. 384–387.
[6] S. Ng and P. Perron, "Unit root tests in ARMA models with data-dependent methods for the selection of the truncation lag," *J. Amer. Stat. Assoc.*, vol. 90, no. 429, pp. 268–281, Mar. 1995.
[7] W. Y. Szeto, B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate traffic forecasting technique using cell transmission model and SARIMA model," *J. Transp. Eng.*, vol. 135, no. 9, pp. 658–667, Sep. 2009.
[8] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, Feb. 1984.
[9] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 246–254, Jun. 2009.
[10] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transp. Res. C, Emerg. Technol.*, vol. 66, pp. 61–78, May 2016.
[11] F. Hai-Liang, C. Di, L. Qing-Jia, and C. Chun-Xiao, "Multi-scale network traffic prediction using a two-stage neural network combined model," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput. (WiCOM)*, Sep. 2006, pp. 1–5.
[12] C. Senfa and T. Changbao, "Neural network structure optimization and its application for passenger flow predicting of comprehensive transportation between cities," in *Proc. IEEE Int. Conf. Grey Syst. Intell. Services*, Nov. 2007, pp. 1087–1091.
[13] 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," *Transp. Res. C, Emerg. Technol.*, vol. 77, pp. 306–328, Apr. 2017.
[14] X. Yan, Y. Liu, Z. Mao, Z.-H. Li, and H.-Z. Tan, "SVM-based elevator traffic flow prediction," in *Proc. 6th World Congr. Intell. Control Autom. (WCICA)*, vol. 2, 2006, pp. 8814–8818.
[15] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, May 2009.
[16] C.-T. Lin and S.-Y. Yang, "Forecast of the output value of Taiwan's opto-electronics industry using the grey forecasting model," *Technol. Forecasting Social Change*, vol. 70, no. 2, pp. 177–186, Feb. 2003.
[17] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.
[18] E. I. Vlahogianni, "Enhancing predictions in signalized arterials with information on short-term traffic flow dynamics," *J. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 73–84, May 2009.
[19] E. Vlahogianni and M. Karlaftis, "Temporal aggregation in traffic data: Implications for statistical characteristics and model choice," *Transp. Lett.*, vol. 3, no. 1, pp. 37–49, Jan. 2011.
[20] C. Chen, Y. Wang, L. Li, J. Hu, and Z. Zhang, "The retrieval of intra-day trend and its influence on traffic prediction," *Transp. Res. C, Emerg. Technol.*, vol. 22, pp. 103–118, Jun. 2012.
[21] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.
[22] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, and S. Yang, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, 2018.
[23] H. Li, Y. Wang, X. Xu, L. Qin, and H. Zhang, "Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105620.
[24] W. Zhang, Y. Zhang, L. Xu, J. Zhou, Y. Liu, M. Gu, X. Liu, and S. Yang, "Modeling IoT equipment with graph neural networks," *IEEE Access*, vol. 7, pp. 32754–32764, 2019.
[25] M. Shenify, A. S. Danesh, M. Gocić, R. S. Taher, A. W. A. Wahab, A. Gani, S. Shamshirband, and D. Petković, "Precipitation estimation using support vector machine with discrete wavelet transform," *Water Resour. Manage.*, vol. 30, no. 2, pp. 641–652, Jan. 2016.
[26] Y. Sun, B. Leng, and W. Guan, "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system," *Neurocomputing*, vol. 166, pp. 109–121, Oct. 2015.
[27] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *J. Transp. Eng.*, vol. 131, no. 10, pp. 771–779, 2005.
[28] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A, Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998.
[29] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowl.-Based Syst.*, vol. 172, pp. 1–14, May 2019.
[30] F. Jia, H. Li, X. Jiang, and X. Xu, "Deep learning-based hybrid model for short-term subway passenger flow prediction using automatic fare collection data," *IET Intell. Transp. Syst.*, vol. 13, no. 11, pp. 1708–1716, Nov. 2019.
[31] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved Bayesian combination model for short-term traffic prediction with deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1332–1342, Mar. 2020.

[32] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1436–1444.
[33] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous conditional random fields for efficient regression in large fully connected graphs," in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 840–846.
[34] H. Tseng, P.-C. Chang, G. Andrew, D. Jurafsky, and C. D. Manning, "A conditional random field word segmenter for Sighan bakeoff 2005," in *Proc. 4th SIGHAN Workshop Chin. Lang. Process.*, Jeju Island, South Korea, 2005, pp. 168–171.
[35] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun./Jul. 2004, pp. II-695–II-702.
[36] C. Sutton, A. McCallum, and K. Rohanimanesh, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," *J. Mach. Learn. Res.*, vol. 8, pp. 693–723, Mar. 2007.
[37] V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous conditional random fields for regression in remote sensing," in *Proc. ECAI*, 2010, pp. 809–814.
[38] N. Djuric, V. Radosavljevic, and V. Coric, "Travel speed forecasting by means of continuous conditional random fields," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2263, no. 1, pp. 131–139, 2011.
[39] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 25–34.
[40] B. Leng, J. Zeng, and Z. Xiong, "Probability tree based passenger flow prediction and its application to the Beijing subway system," *Frontiers Comput. Sci.*, vol. 7, no. 2, pp. 195–203, 2013.
[41] J. J. Barry, R. Freimer, and H. Slavin, "Use of entry-only automatic fare collection data to estimate linked transit trips in New York City," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2112, no. 1, pp. 53–61, Jan. 2009.

JIANYUAN WANG received the B.S. degree from Beihang University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree.



BIAO LENG (Member, IEEE) received the B.S. degree from the National University of Defense Technology, in 2004, and the Ph.D. degree from Tsinghua University, in 2009. He is currently a Research Fellow with Beihang University.



JUNJIE WU (Member, IEEE) received the B.E. degree in civil engineering and the Ph.D. degree in management science and engineering from Tsinghua University, Beijing, China, in 2002 and 2008, respectively. He is currently a Full Professor with the School of Economics and Management, Beihang University, Beijing. He is also the Director of the Research Center for Data Intelligence and the Vice Director with the Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations. His current research interests include data mining, with a special interest in social computing, urban computing, and financial computing. He was a recipient of the various national academic awards in China, including the NSFC Distinguished Young Scientist, the MOE Changjiang Young Scholars, and the National Excellent Doctoral Dissertation.

HENG DU received the B.S. and M.S. degrees from Beihang University, Beijing, China, in 2013 and 2016, respectively. He is currently with the Beijing Urban Rail Transit Control Center.

ZHANG XIONG (Member, IEEE) received the B.S. degree from Harbin Engineering University, in 1982, and the M.S. degree from Beihang University, Beijing, China, in 1985. He is currently a Professor with the School of Computer Science and Engineering, Beihang University.