PULSE: A Real Time System for Crowd Flow Prediction at Metropolitan Subway Stations*

Ermal Toto[1], Elke A. Rundensteiner[1], Yanhua Li[1], Richard Jordan[2], Mariya Ishutkina[2], Kajal Claypool[2], Jun Luo[3], and Fan Zhang[3]

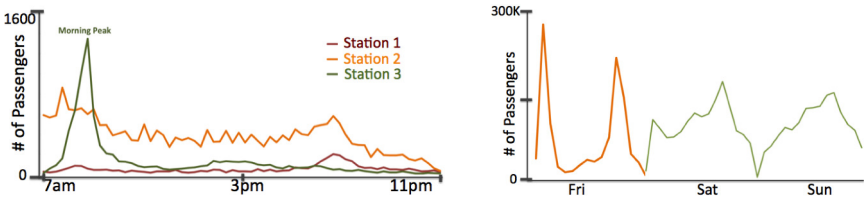
[1] Worcester Polytechnic Institute, USA
[2] MIT Lincoln Laboratory, USA
[3] Chinese Academy of Sciences, Shenzhen Institutes of Advanced Technology, China

Abstract. The fast pace of urbanization has given rise to complex transportation networks, such as subway systems, that deploy smart card readers generating detailed transactions of mobility. Predictions of human movement based on these transaction streams represents tremendous new opportunities from optimizing fleet allocation of on-demand transportation such as UBER and LYFT to dynamic pricing of services. However, transportation research thus far has primarily focused on tackling other challenges from traffic congestion to network capacity. To take on this new opportunity, we propose a real-time framework, called PULSE (**P**rediction Framework For **U**sage **L**oad on Subway Syst**E**ms), that offers accurate multi-granular arrival crowd flow prediction at subway stations. PULSE extracts and employs two types of features such as streaming features and station profile features. Streaming features are time-variant features including time, weather, and historical traffic at subway stations (as time-series of arrival/departure streams), where station profile features capture the time-invariant unique characteristics of stations, including each station's peak hour crowd flow, remoteness from the downtown area, and mean flow, etc. Then, given a future prediction interval, we design novel stream feature selection and model selection algorithms to select the most appropriate machine learning technique for each target station and tune that model by choosing an optimal subset of stream traffic features from other stations. We evaluate our PULSE framework using real transaction data of 11 million passengers from a subway system in Shenzhen, China. The results demonstrate that PULSE greatly improves the accuracy of predictions at all subway stations by up to 49% over baseline algorithms.

1 Introduction

Subway systems provide unobstructed transit throughout an urban area. Starting in the early 90s, in order to streamline fare collection, subway authorities have implemented smart card enabled entry and exit systems [20]. These widely

* This work is sponsored by the Department of Air Force under Air Force Contract FA 8722-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and not necessarily endorsed by the United States Government.



(a) (b)

Fig. 1: (1a) Time series of passenger arrivals at 3 stations during a Monday. (1b) System wide traffic during three consecutive days.

adopted systems generate a large amount of fine grain data about passengers' mobility throughout the transportation network. Offering new opportunities in gaining in-depth insights into the performance and effectiveness of the system as well as the passenger mobility patterns. However a recent survey of smart card transaction usage [20] found that current research is limited to simple post-hoc analysis of generalized mobility patterns, thus risks missing potentially valuable opportunities for new mobility-related services. Predictions of crowd flow arriving at subway stations based on fine grain smart card transaction streams is with foremost importance and opens tremendous new opportunities for novel services, including optimizing fleet allocation and introducing dynamic fares in on-demand systems. In addition, traditional transportation modes such as buses would also benefit from mobility prediction capabilities that would allow them to dynamically adjust stop frequency and routes [9, 11]. These new classes of services increase quality of service and reduce emissions. In the literature, traffic prediction on road networks has been studied extensively, and many prediction models have been applied and developed [24, 7, 23, 14, 26, 29, 12]. However, when applying these methods directly on solving the arrival crowd flow prediction at subway stations, they fail to achieve high prediction accuracy, because these (general) methods do not explicitly take into account the unique features and characteristics of subways systems, such as the pairwise crowd flow between stations, attrition rate of subway stations, etc. Such arrival crowd flow prediction problem is challenging in practice. Figure 1(a) shows that the arrival crowd flow at different stations exhibit completely different time-series patterns, while Figure 1(b) shows that for the same station, the arrival crowd flow are with different patterns over days.

Given these challenges, in this paper, we make the first attempt to study the crowd flow prediction problem at subway stations. We propose a novel real-time framework, called PULSE (**P**rediction Framework For **U**sage **L**oad on Subway Syst**E**ms), that offers accurate multi-granular arrival crowd flow prediction at subway stations. Below, we summarize our main contributions in this paper.

• PULSE extracts two types of features for the arrival crowd flow prediction, i.e., streaming features and station profile features. Streaming features are time-variant features including time, weather, and historical stream traffic at subway stations (as time-series of arrival/departure streams), where station profile features capture the time-invariant unique characteristics of stations, including each

station's peak hour crowd flow, remoteness from the downtown area, and mean flow, etc. (See Section 4.)
• PULSE employs a novel stream feature selection algorithm and a model selection algorithm to select the most appropriate machine learning technique for each target station and tune that model by choosing an optimal subset of stream traffic features from other stations. (See Section 5 and Section 6.)
• We evaluate our PULSE framework using real transaction data of 11 million passengers from a subway system in Shenzhen, China. The results demonstrate that PULSE greatly improves the accuracy of predictions at all subway stations by up to 49% over baseline algorithms. (See Section 7.)

2 Related Work

In this section, we briefly discuss two research areas that are closely related to this work, namely, urban computing and traffic prediction.
Urban computing studies the impact and application of technology in urban areas, including the collection and usage of smart card transactions. Analyzing smart card records is an effective way of understanding human mobility patterns in urban areas [17], [20]. Various studies [6], [5], [17], [15] show that city wide mobility follows a common pattern that is consistent across cities and modes of transportation. These studies describe mobility patterns, but fall short of developing a framework for fine grain predictions of human mobility. To our knowledge this study is the first to directly address prediction of arrival crowd flow in a subway network.
traffic prediction in road networks has been studied extensively [24] [7] [23] [14] [26] [29] [12]. In this study, we compare and contrast the most commonly used machine learning models as baseline methods. One of these baselines (Multiple Linear Regression–MLR) is described in [24], where it is used to capture short term traffic trends. In another study [7] non parametric models similar to K-Nearest Neighbours (KNN) are used for road traffic flow predictions. The concept of using ensembles of models is used in [23], where a state machine switches among different Auto-regressive Moving Average Models (ARIMA) [14]. In [12] Random Forest models are used for short term context aware predictions. All these traffic prediction methods are addressing vehicle traffic prediction problem and utilize a fixed (sometimes ensemble) model to conduct the traffic prediction. Thus, when applying to our crowd flow prediction problem at subway stations, these methods would fail to capture unique features and choose appropriate models for subway system.

In summary, PULSE is the first framework that enables fine-grained arrival crowd flow predictions at subway stations, using smart card transaction data, weather data, and calendar data.

3 Overview

In this section, we define the subway traffic prediction problem and outline the framework of our methodology.

3.1 Preliminary and Problem Definition

We worked on transaction data generated from subway system in Shenzhen, China. Similar to many other subway systems in different cities, such as Beijing Subway [4], and London Subway [5], a passenger needs to swipe his smart card at both the entering and leaving stations. Such paired transaction records capture the trip information of passengers. Below, we explicitly highlight key terms used in the paper, and define the subway station traffic prediction problem.

Definition 1 (Trip). *$tr = (p_{id}, s_d, t_d, s_a, t_a)$ represents a trip made by passenger with ID p_{id}, who departs from station s_d at time t_d and arrives the station s_a at time t_a. $\boldsymbol{TR}$ represent the set of all trips, i.e., $tr \in \boldsymbol{TR}$.*

Definition 2 (Subway Trajectory). *A subway trajectory is a sequence of subway stations that a passenger enters and leaves in the subway system as a function of time. Each record thus consists of a passenger ID p_{id}, subway station ID s, and a time stamp t.*

Definition 3 (Subway Network). *A subway network consists of a set of subway stations connected by subway lines. We represent a subway network as a undirected graph $G = (V, E)$, where V represent the subway stations and E contain the edges between neighboring subway stations via subway lines.*

Problem Definition. Given a set of historical trips **TR**, the subway network G, and the current time t, we want to predict the number of passengers arriving a subway station $s \in V$ (from other stations) during the consecutive time intervals $[t + T * (k - 1), t + T * k]$, with $1 \leq k \leq K$. T is a time aggregation interval, which is usually 15 minutes. K the number of future intervals to be predicted, and we use $K = 6$ in this paper.

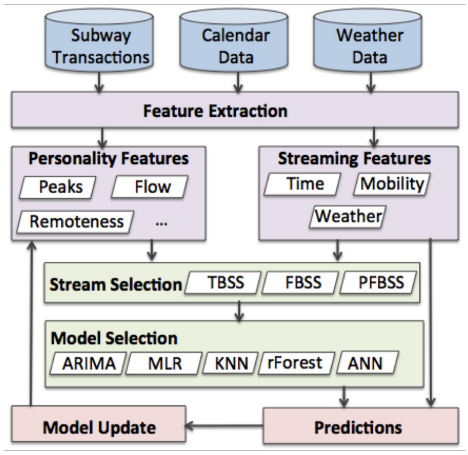
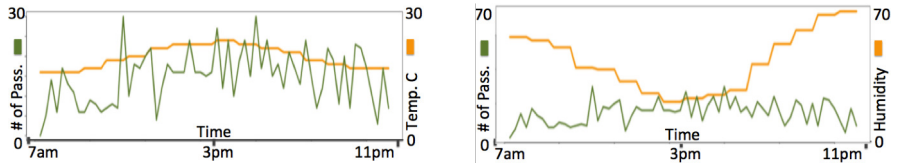


Fig. 2: The PULSE framework.

3.2 The PULSE Framework

To tackle the above subway station traffic prediction problem, we introduce PULSE framework (**P**rediction Framework For **U**sage **L**oad on **S**ubway Syst**E**ms) as shown in Figure 2. PULSE takes the historical trip data, calendar information, and weather data as input, to predict future traffic flows at each subway station at fine grained periodic intervals e.g., every 15 minutes. This

[4] `http://www.bmac.com.cn`
[5] `https://oyster.tfl.gov.uk/oyster/entry.do`



(a) (b)

Fig. 3: Temperature, Humidity and Arrivals at subway station during Saturday.

task is achieved in three core steps, namely, feature extraction, traffic prediction, and model update, as outlined next.

Feature extraction module aggregates the time-varying data sources, such as the transaction data, weather data, calendar data, at certain time granularity, e.g., $15min$. Then, we extract and model both *streaming* and *profile* features. Streaming features are direct aggregates of the time-varying datasets, including aggregated traffic volumes entering and leaving a subway station and weather statistics. Profile features describe relatively stable characteristics of each station, including remoteness of a station, peak-hour traffic, average inflow at a station. See more details in Section 4.2.

Traffic prediction. When predicting the entering and leaving traffic at a subway station s_i, the traffic prediction module employs an automatic feature and model selection algorithm that achieves high prediction accuracy. A prediction model is chosen and a subset of subway stations are selected to include their streaming features as training data. The model and features selected are used to perform predictions on the future entering and leaving traffic at each subway station. Section 5 describes this process in more details.

Model update module keeps track of the performance of the PULSE system overtime. It automatically re-selects features and rebuilds the models.

4 Feature Extraction for PULSE

The feature extraction module explores two sets of key features,namely streaming features and station profile features. The former capture the dynamics of departing/arriving traffic at different stations and the meteorological features over time; while the latter characterize the time-invariant profiles of different subway stations, including remoteness from the city center, the mean flow, peak-hour traffic, etc.

4.1 Streaming Features

4.1.1 Time Features F^t. As discussed earlier, the departing and arriving transaction data are aggregated at a certain time granularity, e.g., $T = 15$ minutes. We observe that the daily operation time of a subway system, denoted as T_0, is usually less than 24 hours. For example, in Shenzhen, the subway system operates between 7am and 11pm every day, that is, a total of $T_0 = 16$ hours operation time. Hence, given the time aggregation interval T, the daily operation time T_0 is divided into a fixed number of time slots with equal length of

T minutes. For example, a total of 64 such intervals are obtained given $T = 15$ minutes and $T_0 = 16$ hours. We then use the interval id $F_{int} \in [1, 64]$ to represent the **time of day** as a feature. As observed in [17], [15], and [5] this feature is significant in urban human mobility predictions. Similarly, we introduce the feature **day of the week**, that distinguishes between weekdays from Monday to Sunday, which can be represented using the weekday id, namely, $F_{day} \in [1, 7]$. As shown in Figure 1b, The traffic patterns vary significantly during different days in a week as also observed in [17] and [15].

4.1.2 Traffic Stream Features F^s. Given an aggregation interval T, we can obtain the arrival and departure traffic at each subway station during each time interval T. For one station s_i, we denote the vector $F_i^{arr} = [a_1, a_2, \ldots, a_N]$ as the **arrival stream feature** of a station s_i. Given a starting time t_0, each a_ℓ represents the number of passengers who arrived the station s_i, during the ℓ-th time interval, namely, $T_\ell = [t_0 + T * (\ell - 1), t_0 + T * \ell]$. Hence, each a_ℓ can be obtained from the trip data as follows.

$$a_\ell = \sum_{tr \in \mathbf{TR}} I(tr.s_a = s_i, tr.t_a \in T_\ell), \tag{1}$$

where $I(\cdot)$ is the indicator function, which is 1 if the condition holds, and 0 otherwise. Similarly, we define the **departure stream feature** of a station s_i as a vector $F_i^{dep} = [d_1, d_2, \ldots, d_N]$. Each d_ℓ can be represented as $d_\ell = \sum_{tr \in \mathbf{TR}} I(tr.s_d = s_i, tr.t_d \in T_\ell)$. When considering pair-wise flows between station pairs, $F_{i,j}^{pair} = [p_1, p_2, \ldots, p_N]$ is the **pairwise flow feature**. p_ℓ representing the number of trips from station s_i to station s_j during the time interval T_ℓ, namely, $p_\ell = \sum_{tr \in \mathbf{TR}} I(tr.s_d = s_i, tr.s_a = s_j, tr.t_d \in T_\ell, tr.t_a \in T_\ell)$. We also take into account of $F_{i,j}^{dur} = [\pi_1, \pi_2, \ldots, \pi_N]$ as the vector **average trip duration feature** from station s_i to s_j during the time interval T_ℓ. Each $\pi_\ell = \frac{1}{p_\ell} \sum_{tr \in \mathbf{TR}} (tr.s_a - tr.s_d) I(tr.s_d = s_i, tr.s_a = s_j, tr.t_d \in T_\ell, tr.t_a \in T_\ell)$.

4.1.3 Weather Features F^w. The traffic at subway stations are affected by meteorology. Hence, we identify two features that are correlated with the subway stations traffic, namely temperature and humidity. Figure 3a (resp. Figure 3b) shows the correlation between the subway station traffic and the temperature (resp. humidity) feature, using the data we collected during 03/20/2014-03/31/2014 in Shenzhen. We can see that the temperature (resp. humidity) is positively (resp. negatively) correlated with subway station traffic. Apparently, a higher temperature leads to a high traffic volume, and a high humidity usually causes low traffic volume.

4.2 Station Profile Features

In this section, we present the profile features extracted from each subway station, which are time-invariant, and capture the unique profile of each subway station from different aspects, such as peak-hour traffic, mean flow, and remoteness from the city center.

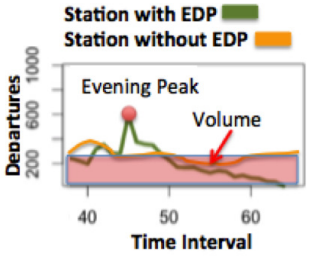


Fig. 4: Equivalent traffic volumes, but different peak patterns.

4.2.1 Peak Traffic F^P Crowd movement during commute hours shows unique and characteristic peak patterns. These patterns vary between stations, but are relatively stable over time. In our study, we choose the peak hour as 7-11am and 5-11pm. The peak-hour behavior as a feature precisely capture the "signature" of a station. A naive way of characterizing the peak-hour behavior is to use total traffic volume occurred in the peak hour. However, it may miss important information of the underlying traffic dynamics in the peak-hour. For example, as shown in Figure 4, two stations have exactly the same peak-hour traffic volume, namely, the total area between the traffic curve and the x-axis. However, we observe that the station 1 shows a flat traffic pattern during the peak-hour, while station 2 has one significant spike. To capture such spike, we employ the Tukey[25] outlier detection method to identify the outliers in the peak-hour, and count the number of outliers as the **peak-hour traffic feature**.

In Figure 5, we use the morning arrival peak-hour traffic as an example. To generate these scores we first extract arrivals during the morning interval 7-11am. Then, we construct the frequency distribution of the arrival values.

We find the first quantile Q_1 and third quantile Q_3 of the distribution, and compute the inter-quantile range as $IQR = Q_3 - Q_1$. A data point Y is labeled as an outlier when: $Y > Q_3 + 1.5IQR$ and the morning arrival peak score is the average number of outliers in all the morning intervals. Similarly, we can obtain the peak-hour traffic for evening arrival, evening departure, and morning departure.

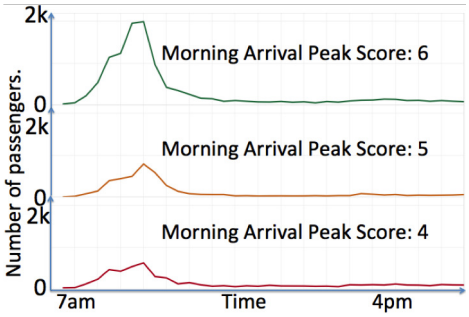
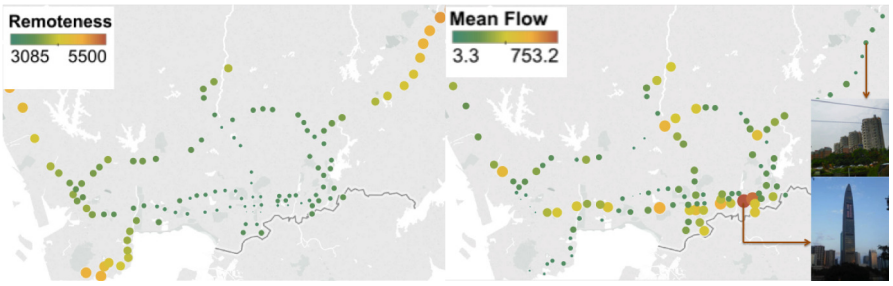


Fig. 5: Arrival streams with different morning peak scores

4.2.2 Flow Related Features F^F. We introduce two types of flow related features, including attrition rate and mean flow of a station.

Attrition Rate. For a station s_i we define the attrition rate of a station s_i as the relative difference between departures and arrivals as the attrition feature Att_i. As is observed in [17], most departure trips from a station s_i have a matching arrival trip, since the majority of passengers return to their home station. However, attrition rates in Shenzhen subway data vary considerably as illustrated in Figure 6. $Att_i = (|F_i^{dep}| - |F_i^{arr}|)/|F_i^{arr}|$.



(a) (b)

Fig. 7: Geographic distribution of remoteness and mean station flow.

Mean Flow of a station s_i is defined as F_i^{flow} and is the average number of arrivals per interval, which can be calculated as $F_i^{flow} = |F_i^{arr}|/N$. Figure 7b illustrates the flow at each subway station. As expected, downtown areas and commercial centers show high concentrations of passenger arrivals.

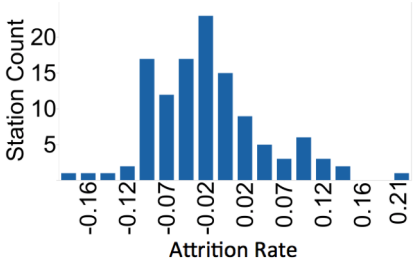
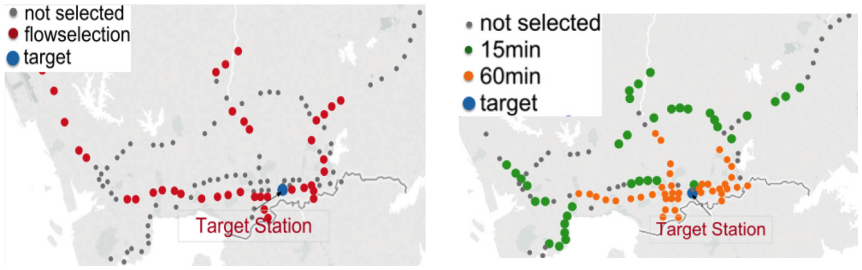


Fig. 6: Distribution of Attrition Rate.

4.2.3 Remoteness F^R. From the subway transaction data, we observe that in general stations located farther away from the downtown area tend to have similar traffic patterns and overall fewer traffic. This motivates us to extract the remoteness of station s_i as a feature, i.e., F_i^R. F_i^R is the average duration of the historical trips arriving to s_i, namely, $F_i^R = \sum_{tr \in \mathbf{TR}} (tr.t_a - tr.t_d) I(tr.s_a = s_i)$. Figure 7a illustrates the geographic distribution of remoteness.

5 Station Stream Selection

Our focus in this work is *arrival traffic prediction* at subway stations. Given a target station s_i, its historical traffic data as a time-series can be used to predict its future arrival traffic, e.g., [14]. In general, subway stations are interconnected, and the arrival traffic at one particular subway station s_i is affected and generated by the traffic from all other stations (in V/s_i). However, given s_i, it is computationally efficient in practice to include a subset of stations (instead of all stations) which contribute significantly to the arrival traffic at s_i, i.e., they are geo-graphically close by, or there are a large amount of traffic flow to the target station. In this section, we present our stream selection algorithm, that can identify the subset of stations, whose departure traffic (as key features) have the most contribution to the traffic at the target station. Our selection algorithm combines three criteria, including Time Based Stream Selection (TBSS), Flow Based Stream Selection (FBSS), and Profile Based Stream Selection (PBSS).



(a) (b)

Fig. 8: Selecting streams based on pairwise flow (8a) and temporal distance (8b).

Below, we elaborate each selection criterion and the overall stream selection algorithm.

Time Based Stream Selection (TBSS). Given the current time t, a time interval $T = 15$ minutes, a target station s_i, we aim to predict the arrival traffic at s_i during the future time interval $\phi = [t + T * (k-1), t + T * k]$ with a positive integer $k > 0$. For example, when $k = 1$, the prediction yields the arrival traffic for the immediate time interval T from the current time t. Hence, we choose those stations that have average arrival time during the prediction interval ϕ. We use the following criterion (in Equation 2) to select θ_L such stations. Recall that the average trip time feature $F^{dur}_{j,i} = [\pi_1, \cdots, \pi_N]$ include the pairwise trip time from a station s_i to s_j over time.

$$L_{i,\phi}(\theta_L) = \underset{B^{\theta_L} \subset V/s_i}{\operatorname{argmin}} \sum_{s_j \in B^{\theta_L}} \left(\sum_{\pi \in F^{dur}_{j,i}} \left| T\left(k - \frac{1}{2}\right) - \pi \right| \right) \tag{2}$$

where $L_{i,\phi}(\theta_L)$ is the set of θ_L selected stations. The value of θ_L is selected by the model selection module (See Section 6) to achieve high prediction accuracy. Figure 8b illustrates the set of stations selected by TBSS for with $\theta_L = 20$, $T = 15$ minutes, and two values of k (orange, $k = 1$ and green, $k = 4$).

Flow Based Stream Selection (FBSS). FBSS is based on the intuition that future traffic at station s_i will come from (departures of) stations with most historical trips to s_i. Recall that the pairwise flow feature $F^{pair}_{j,i} = [p_1, \cdots, p_N]$ include the numbers of pairwise trips from a station s_i to s_j over time. $M_{i,\phi}(\theta_M)$ is the set containing θ_M stations with the highest number of trips to s_i, as illustrated in Equation 3. Again, θ_M is chosen by the model selection module. An example of stations selected by FBSS is given in Figure 8a.

$$M_{i,\phi}(\theta_M) = \underset{B^{\theta_M} \subset V/s_i}{\operatorname{argmax}} \sum_{s_j \in B^{\theta_M}} |F^{pair}_{j,i}| \tag{3}$$

where $|F^{pair}_{j,i}|$ indicates the total number of trips from station s_j to s_i.

Profile Based Stream Selection (PBSS). Profile features characterize the overall traffic patterns of subway stations. The stations with similar profile features tend to have similar traffic pattern over time. Given a target station s_i,

its profile feature vector is $PF_i = [F_i^P, F_i^F, F_i^R]$, where F^P, F^F and F^R represent the peak traffic features, flow related features, and remoteness features, respectively. PF_i is compared to PF_j for each $s_j \in V$ and a set $K_{i,\phi}(\theta_K)$ of the θ_K nearest (in terms of profile features) stations is selected as illustrated in Equation 4. The optimal value for θ_K is determined during model selection.

$$K_{i,\phi}(\theta_K) = \underset{B^{\theta_K} \subset V/s_i}{\operatorname{argmin}} \sum_{s_j \in B^{\theta_K}} \left(\sqrt{\sum_{n=1}^{|PF|} \left(PF_i^n - PF_j^n\right)^2} \right) \tag{4}$$

Stream selection. The final set of stations are simply the union set of the results from three criteria, i.e., $L_{i,\phi}(\theta_L) \cup M_{i,\phi}(\theta_M) \cup K_{i,\phi}(\theta_K)$.

The pseudocode for the stream selection is given in Algorithm 13. In Lines 2–6, the procedure iterates through all stations $s_j \in V/s_i$ and calculates the time distances, pairwise flows, and profile feature Euclidean distances between stations s_i and s_j. In lines 7–12, these distances are sorted, and the first θ_L,θ_M, and θ_K, streams are selected. Line 13 merges and returns the three stream sets.

Algorithm 1: Stream selection for station s_i

1 **function** StreamSelection $(s_i, \phi, F_{i,j}^{dur}, F_{i,j}^{pair}, PF, \theta_L, \theta_M, \theta_K)$;
  Input : Station s_i. Prediction interval ϕ. Sets $F_{i,j}^{dur}$,$F_{i,j}^{pair}$, and PF. Number of streams to be selected defined by θ_L, θ_M, and θ_K.
  Output: $L_{i,\phi}^{\theta} \cup M_{i,\phi}^{\theta} \cup K_{i,\phi}^{\theta}$
2 **for** $s_j \in V/s_i$ **do**
3 $timedistance[j] = |average(F_{i,j}^{dur}) - T * (k - 1/2)|$;
4 $flow[j] = |F_{i,j}^{pair}|$;
5 $pfdistances[j] = euclidiandistance(PF_i, PF_j)$;
6 **end**
7 $timedistances = sort(timedistances)$;
8 $flow = sort(flow)$;
9 $pfdistances = sort(pfdistances)$;
10 $L_{i,\phi}^{\theta} = getKeys(timedistances[1..\theta_L])$;
11 $M_{i,\phi}^{\theta} = getKeys(flow[1..\theta_M])$;
12 $K_{i,\phi}^{\theta} = getKeys(pfdistances[1..\theta_K])$;
13 return $L_{i,\phi}^{\theta} \cup M_{i,\phi}^{\theta} \cup K_{i,\phi}^{\theta}$;

6 Model Selection

To accurately predict the arrival traffic for a prediction interval ϕ at a target station s_i, we need to choose the right prediction model and the right set stream features from other stations, namely, θ_L, θ_M, θ_K. We consider five candidate prediction models used in literature for time-series data prediction, including Autoregressive integrated moving average (ARIMA) [14, 23], Artificial Neural Networks (ANN) [18, 28, 26, 29], K-Nearest Neighbours (KNN) [8, 10, 7], Random Forest (RF) [13, 16, 12], and Multiple Linear Regression (MLR) [24]. It also has to choose the optimal number of streams to include using the methods described in Section 5. In our study, the Shenzhen subway system have five subway lines

with 118 subway stations. Thus, θ_L, θ_M, and θ_K each has 118 possibilities, leading to a searching space of 118^3. Each model configuration setup requires training and testing using historical data.

To find the optimal configuration of model and stream set for a station s_i and prediction interval ϕ, it requires examining all configurations with different model and stream combinations. **A naive method** is to brute force all such configurations, and choose the one with the highest prediction accuracy. However, this is too costly to be implemented in practice. To be precise, we have five prediction models and 118^3 possibilities of stream set sizes. Let's consider 6 future prediction intervals and different temporal partitions, which in this set of experiments is two (weekdays and weekends). In total there are about 79 million different models. We ran our experiments in a server with 30 Intel(R) Xeon(R) CPU E5-4627 v2 @ 3.30GHz Cores. Each model training and testing would take about 1 to 15 seconds, thus leads to a total of 14 years to compare all configurations using our 30 core system.

Thus, we are motivated to employ the profile features to conduct **Gradient-based optimization of hyper-parameters** [3] [4] to optimize this process. Initially this method uses a pure gradient search approach to discover parameters. As more station profiles are matched to models, PULSE can initiate subsequent searches with model parameters from stations with similar profiles as described by Equation 5. Henceforth we refer to this method as Model Select (MSELECT). After a large number of stations have been assigned with prediction models, the process only takes a few seconds. Therefore this method is suitable as an online process for model updates based on changes in the profile features. Our gradient based model search takes approximately 2 hours to find the optimal prediction configuration.

$$Model_i = \underset{Model_j \in Models}{\operatorname{argmin}} \left[\sqrt{\sum_{n=1}^{|PF|} \left(PF_i^n - PF_j^n \right)^2} \right] \tag{5}$$

Model update. PULSE system monitors the prediction performance overtime. It automatically re-selects features and rebuilds the models when the average prediction accuracy goes below certain threshold value.

7 Evaluation of PULSE Model

To evaluate the performance of our PULSE framework on arrival traffic prediction, we conduct comprehensive experiments using real subway transaction dataset collected from Shenzhen subway system for 21 days in March 2014. By comparing with baseline algorithms, the experimental results demonstrate that PULSE can achieve a 26%-94% relative prediction accuracy, which is on average 20% higher than baseline algorithm. Below, we present the datasets, baseline algorithms, experiment settings, and results.

7.1 Dataset Description

For this work, we used 60 million smart card transactions from the subway system in the city of Shenzhen, China between March 10^{th} and March 31^{st}, 2014. The dataset contains 11 million unique passengers (identified by their smart card ids). Each transaction contains a timestamp, location coordinates, and whether the transaction is an departure from or an arrival at a station. During data preprocessing we matched entry and exit transactions for each passenger in order to generate trip record $tr = (p_{id}, s_d, t_d, s_a, t_a)$ containing a passenger identifier p_{id}, a starting station s_d, a destination s_a and respective departure and arrival times t_d, t_a.

7.2 Evaluation Settings

PULSE predicts the number of arrivals at a station s_i at future time intervals in $[t + T * (k-1), t + T * k]$ with $1 \leq k \leq K$. In our evaluation of PULSE, we used a variable $k \in [1 \cdots 6]$.

Prediction models for both PULSE and the baseline methods are trained using a sliding window containing a week of historical data, to predict the arrival traffic of a future interval specified by k. The accuracy of the predictions is defined as $accuracy = 1 - \frac{\sum |\hat{y}_i - y_i|}{\sum y_i}$. Again, we consider five prediction models used in literature for time-series data prediction, including Autoregressive integrated moving average (ARIMA) [14, 23], Artificial Neural Networks (ANN) [18, 28, 26, 29], K-Nearest Neighbours (KNN) [8, 10, 7], Random Forest (RF) [13, 16, 12], and Multiple Linear Regression (MLR) [24]. All these methods can be setup as both single stream (only using the features of the target station) or multi-stream models (using features from both the target station and other selected stations)[6]. In our experiments, we evaluate PULSE framework in two stages. In the *first stage*, we run all prediction models in a single-stream fashion using the arrival stream feature F_i^{arr} of the target station s_i, with vs without other streaming features, such as time feature F^T and weather features F^W. In the *second stage*, we evaluate the stream feature selection and model selection algorithms introduced in Section 5 and 6 in a multi-stream scenario. We compare our PULSE framework with each individual model under single-stream mode. The evaluation results are summarized in the next subsection.

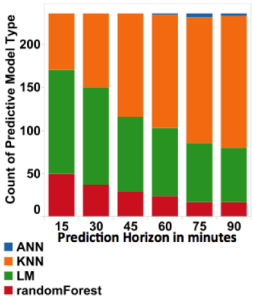


Fig. 9: Distribution of the best performing models over the prediction horizon.

7.3 Evaluation Results

Stage 1: Single-stream models. In table 1, the column *BaseL No SF* list the baseline results of single stream models, that only use the arrival stream feature

[6] Note that ARIMA can only be setup as a single stream model by its design in nature.

Table 1: Overall performance evaluation at 118 stations.

		BaseL No SF					BaseL SF				
	H.	**KNN**	**MLR**	**RF**	**ANN**	**ARIMA**	**KNN**	**MLR**	**RF**	**ANN**	**MSEL**
W	*15*	0.738	0.735	0.735	0.750	0.746	0.872	0.848	0.860	0.836	0.884
D	*30*	0.658	0.647	0.657	0.672	0.745	0.872	0.846	0.855	0.840	0.883
a	*45*	0.575	0.560	0.574	0.595	0.745	0.870	0.837	0.850	0.840	0.882
y	*60*	0.526	0.509	0.525	0.548	0.745	0.868	0.831	0.848	0.834	0.881
	75	0.498	0.477	0.498	0.524	0.745	0.865	0.824	0.845	0.832	0.880
	90	0.488	0.462	0.489	0.516	0.744	0.862	0.818	0.842	0.825	0.879
W	*15*	0.752	0.784	0.749	0.780	0.772	0.770	0.726	0.801	0.724	0.845
E	*30*	0.712	0.760	0.707	0.755	0.772	0.768	0.667	0.791	0.718	0.841
n	*45*	0.639	0.702	0.631	0.698	0.771	0.761	0.603	0.763	0.705	0.833
d	*60*	0.585	0.662	0.578	0.649	0.771	0.760	0.573	0.745	0.693	0.827
	75	0.540	0.623	0.535	0.610	0.769	0.762	0.572	0.731	0.687	0.820
	90	0.518	0.601	0.516	0.590	0.771	0.770	0.590	0.728	0.699	0.813
Av.		0.602	0.627	0.600	0.641	**0.758**	**0.817**	0.728	0.805	0.769	**0.856**

Table 2: Stations with top improvement in prediction accuracy.

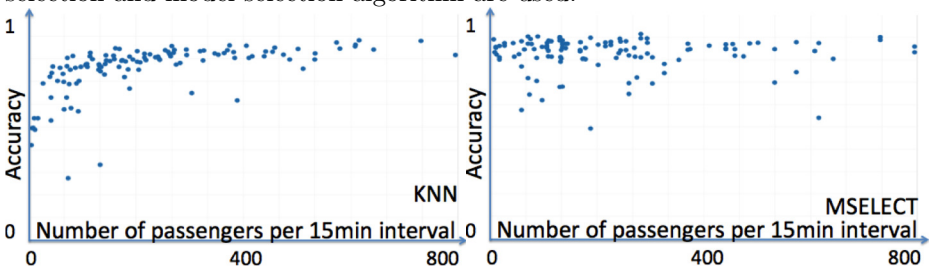
	Rank	Station ID	ML	H	TBSS	FBSS	PBSS	KNN	M.Select	Diff
Week	1	260011	LM	90	0	0	0	0.709	0.769	0.060
days	2	260024	RF	30	30	10	20	0.465	0.523	0.058
	3	260024	RF	45	30	40	0	0.465	0.521	0.056
	4	268028	RF	15	40	40	20	0.469	0.522	0.053
	5	268023	KNN	90	40	40	40	0.871	0.921	0.050
Week	1	261006	RF	45	0	0	10	0.264	0.755	0.491
ends	2	268023	KNN	60	30	0	40	0.334	0.814	0.481
	3	268012	KNN	60	20	20	30	0.618	0.854	0.236
	4	261006	KNN	90	0	20	10	0.481	0.716	0.234
	5	263013	KNN	15	30	0	10	0.512	0.739	0.228

of the target station. The column *BaseL SF* list the results of single stream models, that include both the arrival stream feature of the target station, and also other streaming features introduced in Section 4.1, such as the weather and time features. The results show that by introducing time and weather features, the prediction accuracy for the single-stream models is improved on average 13.4% and up to 21.7%, namely, from 60%–75.8% to 76.9%–81.7%. When we look at the different prediction horizons from 15 minutes to 60 minutes ahead of time, the accuracy of all models (except ARIMA) decreases as the prediction horizon increases. This is reasonable since it is in general harder to predict the arrival traffic in a long term future interval than an immediate future interval.

Stage 2: Multi-stream models. In table 1, the last column *MSEL* list the results of multi-stream models, when stream feature selection and model selection algorithms are applied to include departure stream features from other stations than the target station. We observed that the average prediction accuracy is

further improved to 85.6% over single-stream models, with an average of 7.6% improvement over *BaseL SF*, and 21% improvement over *BaseL No SF*.

Table 2 lists the evaluation results of the stations with the top five improvement on the prediction accuracy for weekdays and weekends, respectively. During weekends, the first ranked station (in terms of model improvement) has a prediction accuracy as low as 26.4% at 45 minutes prediction horizon when using KNN (the best performing single-stream baseline) with all streaming features. By applying stream feature selection and model selection algorithms, PULSE increases the prediction accuracy of this model to 75.5% with a total of 49.1% improvement. This was achieved by using a Random Forest model with 10 streams that were selected using profile based stream selection (PBSS). Overall, the stream feature selection and model selection algorithms improve the prediction accuracy more during the weekends (up to 49.1% improvement) than the weekdays (up to 6%). This happens primarily because the arrival traffic in weekends is less stable than that during weekdays, and single-stream models have low prediction accuracy, providing more room to improve the performance when stream feature selection and model selection algorithm are used.



(a) (b)

Fig. 10: KNN vs MSELECT weekend prediction accuracy at 60min horizon, for stations with different mean passenger flow.

Summary and Observations. The above results with single-stream models demonstrate that by introducing time and weather features, the prediction accuracy is improved on average 13.4%. For multi-stream models, our PULSE framework further improves the prediction accuracy 7.6% on average. To better understand the evaluation results, Figure 10(a)(b) present the prediction accuracy distribution over stations in terms of their mean arrival flow for single stream model (KNN) in Figure 10(a) vs multi-stream models Figure 10(b). We observed that stations with lower mean arrival traffic had the most improvement. When we looked at the best models being selected by our model selection algorithm over different prediction horizons, we noticed that there is a clear shift in the machine learning models with increasing prediction horizons. For example, linear model (LM) and Random forest (RF) are used more for smaller prediction horizons (i.e., predicting the near future), while k-nearest neighbors (KNN) in general performs better for larger prediction horizons (i.e., predicting the long term future intervals). These observations shed light on the performances of different models in subway station traffic predictions.

8 Conclusion

In this study we present PULSE, a real-time system to predict arrival crowd flow at metropolitan subway stations. The system extracts streaming features and station profile features from heterogeneous urban data, including subway transaction data, weather data, and calendar data. PULSE employs novel stream feature selection and model selection algorithms to improve the prediction accuracy and running time. Experimental results on real subway transaction data from 11 million passengers in Shenzhen, China demonstrated that PULSE can increase the prediction accuracy by up to 49% over baseline algorithms.

References

1. Statistic Brief. World Metro Figures. *1st ed. Brussels, Belgium: UITP*, 2014.
2. Patricia Clarke Annez and Robert M Buckley. Urbanization and growth: setting the context. *Urbanization and growth*, 1:1–45, 2009.
3. Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.
4. James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.
5. Artem Chakirov and Alexander Erath. Use of public transport smart card fare payment data for travel behaviour analysis in singapore. 2011.
6. Yew-Yih Cheng, Roy Ka-Wei Lee, Ee-Peng Lim, and Feida Zhu. Measuring centralities for transportation networks beyond structures. In *Applications of Social Media and Social Network Analysis*, pages 23–39. Springer, 2015.
7. Stephen Clark. Traffic prediction using multivariate nonparametric regression. *Journal of transportation engineering*, 129(2):161–168, 2003.
8. Thomas M Cover and Peter E Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
9. Liping Fu, Qing Liu, and Paul Calamai. Real-time optimization model for dynamic scheduling of transit operations. *Transportation Research Record: Journal of the Transportation Research Board*, (1857):48–55, 2003.
10. Keinosuke Fukunaga and Patrenahalli M Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE*, 100(7):750–753, 1975.
11. Peter Furth and Adam Rahbee. Optimal bus stop spacing through dynamic programming and geographic modeling. *Transportation Research Record: Journal of the Transportation Research Board*, (1731):15–22, 2000.
12. Benjamin Hamner. Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow. In *ICDMW, 2010 IEEE International Conference on*, pages 1357–1359. IEEE, 2010.
13. Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.
14. GM Jenkins and GC Reinsel. Time series analysis: forecasting and control, 1976.
15. Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. Traffic prediction in a bike-sharing system. 2015.
16. Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

17. Liang Liu, Anyang Hou, Assaf Biderman, Carlo Ratti, and Jun Chen. Understanding individual and collective mobility patterns from smart card records: A case study in shenzhen. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–6. IEEE, 2009.
18. Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
19. United Nations. *World Urbanization Prospects 2014: Highlights*. United Nations Publications, 2014.
20. Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.
21. Lawrence R Rabiner and Bernard Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1, 1975.
22. Vsevolod Salnikov, Renaud Lambiotte, Anastasios Noulas, and Cecilia Mascolo. Openstreetcab: Exploiting taxi mobility patterns in new york city to reduce commuter costs. *arXiv preprint arXiv:1503.03021*, 2015.
23. Anthony Stathopoulos and Matthew G Karlaftis. A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, 11(2):121–135, 2003.
24. Hongyu Sun, Henry X Liu, Heng Xiao, Rachel R He, and Bin Ran. Short term traffic forecasting using the local linear regression model. In *82nd Annual Meeting of the Transportation Research Board, Washington, DC*, 2003.
25. John W Tukey. Exploratory data analysis. 1977.
26. Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies*, 13(3):211–234, 2005.
27. Eric W Weisstein. Fast fourier transform. 2015.
28. B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
29. Weizhong Zheng, Der-Horng Lee, and Qixin Shi. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, 132(2):114–121, 2006.