# Predicting Citywide Passenger Demand via Reinforcement Learning from Spatio-Temporal Dynamics

Xiaodong Ning
University of New South Wales
z5122770@student.unsw.edu.au

Lina Yao
University of New South Wales
lina.yao@unsw.edu.au

Xianzhi Wang
University of Technology Sydney
sandyawang@gmail.com

Boualem Benatallah
University of New South Wales
boualem.benatallah@unsw.edu.au

Flora Salim
RMIT University
flora.salim@rmit.edu.au

Pari Delir Haghighi
Monash University
Pari.Delir.Haghighi@monash.edu

## ABSTRACT

The global urbanization imposes unprecedented pressure on urban infrastructure and public resources. The population explosion has made it challenging to satisfy the daily needs of urban residents. 'Smart City' is a solution that utilizes different types of data collection sensors to help manage assets and resources intelligently and more efficiently. Under the Smart City umbrella, the primary research initiative in improving the efficiency of car-hailing services is to predict the citywide passenger demand to address the imbalance between the demand and supply. However, predicting the passenger demand requires analysis on various data such as historical passenger demand, crowd outflow, and weather information, and it remains challenging to discover the latent relationships among these data. To address this challenge, we propose to improve the passenger demand prediction via learning the salient spatial-temporal dynamics within a reinforcement learning framework. Our model employs an information selection mechanism to focus on the most distinctive data in historical observations. This mechanism can automatically adjust the information zone according to the prediction performance to find the optimal choice. It also ensures the prediction model to take full advantage of the available data by introducing the positive and excluding the negative correlations. We have conducted experiments on a large-scale real-world dataset that covers 1.5 million people in a major city in China. The results show our model outperforms state-of-the-art and a series of baselines by a large margin.

## 1 INTRODUCTION

The past decades have witnessed people increasingly migrate from rural into urban areas following the rapid pace of worldwide urbanization. Estimators project that approximately 70% of the world's population will live in urban before 2050 [2]. While the urbanization can bring some positive effects such as creating new employment opportunities, advancing technologies, infrastructures, and economic growths, it also leads to the problem of how to balance the limited public resources and rapidly increasing populations. The notion of 'Smart city' has been proposed to integrate information and communication technology (ICT), using various physical devices connected to the network (Internet of Things or IoT) to optimize the efficiency of city operations and services and connect to citizens. Smart city involves several aspects: traffic and transportation systems, power plants, water supply networks, waste management, law enforcement, information systems, schools, libraries, hospitals, and other community services. Among all these aspects, we focus on the traffic and transportation sector, which has been transformed by the evolution of the Internet. As an integral mode of transport in urban cities, taxis serve a large number of passengers on a daily basis. However, the traditional taxis hailing system may suffer low efficiency due to the information asymmetry between passengers and drivers, as well as adverse external conditions like weather.

Several online peer-to-peer ride-sharing services (e.g., Uber and Didi) have emerged in recent years aimed to address this problem. They bring the supplies and demands for ride and delivery services on a unified platform where customers can book their rides via mobile apps, and the drivers can pick up the orders based on the proximity. These services usually employ dynamic pricing models and have revolutionized the taxi markets in many countries significantly. Although these services can alleviate the information asymmetry problem, they cannot fully crack the code between passenger demand and ride supply. On the one hand, a passenger might experience a long waiting for a driver to pick his order in some high-demand area; on the other hand, a driver may find it difficult to receive an order in some areas where the number of passenger demand is low [15]. The imbalance between taxi demand and supply can waste the public transportation resources, influence the passenger experience, and further affect driver's income. Therefore, a more efficient car-hailing prediction model is crucial for reducing the delay time for the passenger order.

However, it is very challenging to accurately predict the passenger demand of a city in a real-time manner because the passenger demand can fluctuate in a relatively wide range and be affected by a

variety of factors. For example, the passenger demand can boost in some extreme weather (e.g., snowy days, stormy days) or during the holidays while it may decrease during the weekdays. Besides, the demand patterns of passengers in an area are highly related to the characteristics of the area. The area characteristics consist of many aspects such as the point-of-interests in that area, the population of the area, and the size of the area. These factors do not only affect the passenger demand independently but also mutually impact each other. For example, the meteorological factor can have a different impact on the passenger demand in different areas. The passenger demand of the central areas of a city can increase more than the remote areas because of the large rigid commuting requirements. Furthermore, due to the characteristic of traffic flow, the passenger demand pattern of a particular region can also be influenced by other areas. In this context, it is necessary to formulate a model which can capture the complicated spatial and temporal relationships of multiple sources of information across different domains for predicting the passenger demand. Some of the previous work attempt to tackle this problem. For example, some researchers [8, 14] employ the time-series models to capture the temporal relationship for taxi prediction, but they do not take other available data (e.g., spatial and weather information) into consideration. Some previous research [9, 18, 19, 22] has already confirmed that there exists a strong correlation between spatial information and taxi demand. Hence, it is essential to explore the complex relationships between different kinds of data sources and efficiently utilize them for accurate prediction. In recent years, deep learning based technique [15, 17, 20, 21] have been widely used to model the complex relationships between various data sources. However, these models either utilize data from all regions or only consider surrounding regions. None of them use the temporal and spatial correlations fully and directly.

In this work, we propose an information zone selection based deep learning framework to predict the passenger demand. Our model incorporates different data sources together: historical passenger demand, historical crowd outflow, PoI data, weather data, air quality data, and time meta. The previous deep learning based methods either utilize all the historical data (all the regions) or fixed partial data (surrounding regions). This way, they can sometimes introduce irrelevant or negative correlations into the model or miss some significant correlations. Our model differs from them in the ability to find the most distinctive temporal-spatial information zone from the historical observations automatically. The contributions of our model are as follows:

- We propose a deep learning based model that jointly incorporates temporal information, spatial information, and supplementary data for passenger demand prediction to improve the efficiency of car hailng services.
- Our model employs a reinforcement learning method to choose the most distinctive information zone from all the available historical observations instead of utilizing all the historical observation directly.
- We conduct extensive experiments on a real-world large-scale dataset collected from a major city in China covering 1.5 million people, and demonstrate that our method outperforms a series of benchmark methods and state-of-the-art methods.

## 2 RELATED WORK

There exists some previous work on the taxi flow forecasting area. Moreira and Li [8, 14] try to capture the temporal relationship between passenger demand using the time series model auto-regressive integrated moving average (ARIMA) and its variant. Moreira et al. [14] aim to solve the best taxi stand choice problem after a passenger drop-off. Their work combines three time-series model (Time-Varying Poisson Model, Weighted Time-Varying Poisson Model, ARIMA model) to predict the number of services that will emerge at a given taxi stand. Li et.al [8] propose an improved ARIMI model to predict the spatial-temporal variation of passengers in hotspots. However, their work make predictions based on the GPS trajectory data, which may not reflect the actual taxi flow. Besides the temporal information, Zhang et.al [22] and Li et.al [9] also add the spatial and external data as the supplementary information. Zhang et al. [22] apply the exponentially weighted moving-average (EWMA) model to recommend the top $k$ hotspots for taxi drivers. However, their method can predict the hotpots instead of predicting an accurate demand number. Li et al. [9] try to predict the short-term traffic demand utilizing a wave SVM model which can take fully advantages of two model. However, they do not consider meteorological data which is crucial in passenger demand prediction. Zheng et al. [24] infer the real-time and fine-grained air quality information throughout a city based on the historical air quality report and a variety of data sources including meteorology, traffic flow, human mobility, the structure of road networks, and point of interests (POIs). Although this work explores the relationships between the traffic flow and other data, it aims to infer the air quality instead of predicting the traffic flow.

Besides traditional methods, deep learning techniques have also been widely used in the traffic forecasting area. Wang et al. [15] propose Deep Supply-Demand model (DeepSD) which utilizes a novel deep neural network structure to predict the gap between the car-hailing supply and demand in a specific area in the future. Though the work successfully utilizes multiple data sources, it requires hand-crafted features and cannot accurately capture the sequential relationship. Zhang et al. [21] propose a deep spatiotemporal residual network (ST-ResNet) to collectively forecast the inflow and outflow of crowds in every region of a city. They design three residual networks to model the temporal closeness, period, and trend properties of crowd traffic. For each property, they design a branch of residual convolutional units, each of which models the spatial properties of crowd traffic. However, the objective of this paper is to predict the taxi trajectories which differs from the real passenger demand. Another disadvantage of this model is that it takes the information about all the available regions into consideration which may introduce weak or negative correlations. Yao et al. [17] propose a Deep Multi-View Spatial-Temporal Network (DMVST-Net) to model both spatial and temporal relations. Their model consists of three views: temporal view (modeling the temporal relationship via LSTM), spatial view (modeling spatial correlation via CNN) and semantic view (modeling correlations among similar regions). This work is actually most similar to ours, but it only applies to the city data partitioned by the grid-based method. Also, the model does not leverage advanced partitioning methods such as road network based partition. Different from the previous studies, Yu et al. [20] focus on the extreme condition traffic forecasting. They apply deep LSTM to forecast

peak-hour traffic and manage to identify unique characteristics of the traffic data. They also improve the model for the post-accident forecast with Mixture Deep LSTM model. As their primary goal is to forecast traffic flow under extreme condition, we do not compare ours with their work.

## 3 BACKGROUND

In this section, we will introduce the background of our model. We introduce reinforcement learning, dueling deep Q network and data description in Section 3.1, 3.2 and 3.3 respectively.

### 3.1 Reinforcement Learning

Reinforcement learning is an important type of machine learning where an agent learns how to behave in an environment by performing actions and seeing the results.

A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent receives an observation $o_t$, which typically includes the reward $r_t$. It then chooses an action $a_t$ from the set of available actions, which is later sent to the environment. The environment moves to a new state $s_{t+1}$ and the reward $r_{t+1}$ associated with the transition $(s_t, a_t, s_{t+1})$ is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. The agent can (possibly randomly) choose any action as a function of the history.

The performance difference between an agent and the agent that acts optimally gives rise to the notion of regret. To act near optimally, the agent must reason about the long-term consequences of its actions (i.e., maximize future income), although the immediate reward associated with this may be low.

Thus, reinforcement learning is particularly well-suited to problems that include a long-term versus short-term reward trade-off. It has been applied successfully to various problems, including robot control, elevator scheduling, telecommunications, backgammon, checkers and go (AlphaGo).

### 3.2 Dueling Deep Q Network

Dueling Deep Q Network is proposed by Wang et.al[16] in 2015. Actually, Dueling Deep Q Network is a variant of original Deep Q Network(DQN). So we will introduce the Deep Q Network firstly. Deep Q Network is the the first deep reinforcement learning method proposed by DeepMind[12]. Q learning is a reinforcement learning method which produces a Q-table that an agent uses to find the best action to take given a state. However, it is not efficient to create and update a Q table for environment. In this context, Deep Q learning has been proposed, which utilizes neural network to approximate, given a state, the different Q-values for each action. The DQN can be regarded as a combination of CNN and Q learning. It contains four technique advantages to overcome stable learning problem: experience replay, target network, clipping rewards and skipping frames. Based on the DQN, a new variant Dueling Deep Q Network is proposed. It explicitly separates the representation of state values and (state-dependent) action advantages. The dueling architecture consists of two streams that represent the value and advantage functions, while sharing a common convolutional feature learning module.

### 3.3 Data Description

There are five types of data in our dataset: Passenger Demand Data, Crowd Out Flow Data, Meterological Data, Poi Data and Time Data. We will introduce them detailed in the following:

- Passenger Demand Data: This data contains taxi request data of Didi Chuxing. Each item contains the time and location (latitude and longitude) of a request. We pre-process this dataset to map requests to related regions and time intervals and set the time interval to 1 hour in our experiment.
- Meteorological Data: The meteorlogical data contains information about weather and air quality, including temperature, wind speed, visibility, weather state and air quality level. Temperature, wind speed, and visibility are continuous data and update every one hour. Weather state and air quality level are categorical data.
- Time Data: Time meta includes the hour of the day, the day of the week, and holiday information.
- PoI data: We collected PoI data of 12 categories, including offices, entertainment facilities, hotels, shopping malls, residences (i.e., apartments), schools, banks, restaurants, government facilities, bus stations, tourist attractions and hospitals. Each PoI item contains name and location (latitude and longitude). We pre-process this data by mapping PoI data to related regions.
- Crowd Outflow Data: it comes from the cellular networks which cover more than $1.5 \times 106$ mobile users. This data is also mapped to the related regions and time intervals, with the time interval set to 1 hours.

Table 2 shows the details of meteorological and time data. Among all the five kinds of data, meteorological, time and historical passenger demand data has already shown their capability of passenger demand order prediction in the previous work [10, 17, 24]. However, the previous work seldom uses PoI data and crowd outflow data for passenger demand prediction. Generally, different regions in a city can have different patterns of passenger demand. The pattern of a specific region is highly dependent on its characteristics such as the PoIs in that region. For example, the region with a lot of shopping malls and restaurants will have a much higher amount of passenger demand on weekends while the demand will decrease significantly on weekdays. Also, if a region has many tourist attractions, it will have a burst in the passenger demand on holidays. Therefore, PoI data can be a useful index to characterize regions and help find the regions with similar passenger demand patterns. Analyzing the historical passenger demand of regions with similar patterns to a target region can provide much useful information and help the final prediction. In this context, we take the PoI data as auxiliary information in our work. When collecting the crowd outflow data, we consider passengers as a subset of the crowd outflow. Thus passenger demand should be closely related with crowd outflow data. Modeling the correlation between passenger demand and crowd outflow could help predict passenger demand.

## 4 PROBLEM FORMULATION

In this section, we will first introduce the terms used in our paper and formalize the passenger demand prediction problem.

**Table 1: Details of meteorological and time data.**

|  | Subtypes | Data Condition |
|---|---|---|
| Meteorological Data | Weather State | 4 Types |
|  | Temperature($°C$) | [-21.24,6.11] |
|  | Wind Speed (m/s) | [0,7.15] |
|  | Visibility (km) | [0.16,16.09] |
|  | Air Quality | 4 Types |
| Time Data | Hour of Day | [1,24] |
|  | Day of Week | [1,7] |
|  | Holiday | 2 Types |

**Region** We utilize the same city partition method as Deng et.al[3] did. It is a more flexible partition which divides the entire city into many blocks. We divide the city into M regions (M is 35 in our work). The $i_{th}$ region is denoted as $R_i$ where i ∈ [1, M].

**Time Data** In our paper, we utilize $T$ to represent a time point and $t$ to represent a time interval $[T, T + \Delta T]$. Accordingly $t_{i+1}$ represents the next time interval $[T + \Delta; T + 2\Delta]$ after $t_i$. As we also take the time data as a kind of feature for passenger demand prediction, then we denote $A(t)$ to represent the time feature in time interval t.

**Point of Interest** We use $P_i$ to denote the all the POI information of the region i. The $P_i$ is a K dimensional vector (K is 12 in our work) with each element of it representing the correct PoIs category (such as schools, restaurant, shopping malls and so on) in region i.

**Passenger Demand** For a region $R_i$, the passenger demand of region $R_i$ in a given time interval $t$ is defined as the number of taxi request originating in this region during this time period, which can be represented as $D_t(R_i)$. Additionally, we denote $D_t = [D_t(R_1), D_t(R_1), \cdots, D_t(R_M)]$ (a M dimensional vector) to represent the passenger demand of all the M regions in time interval t.

**Meteorological Data** Meteorological Data consists of the weather and air quality information in each region. We denote the $M_t$ to represent the meteorological data in time interval t.

**Crow Outflow** Similar to the definition of passenger demand, we denote $C_t(R_i)$ as the crowd outflow of region $R_i$ and $C_t = [C_t(R_1), C_t(R_1), \cdots, C_t(R_M)]$ (a M dimensional vector) to represent the crowd outflow of all the M regions in a given time interval $t$.

Given all the historical information including the passenger demand and crow outflow in all the M regions, the meteorological data of region $i$ in time interval $t$, our goal is to predict the passenger demand $D_t(R_i)$ of region $i$ in time interval t. We formulate the problem as the following equation:

$$D_t(R_i) = F(D_{t-h}, \cdots, D_{t-2}, D_{t-1}, C_{t-h}, \cdots, \\ C_{t-2}, C_{t-1}, M_t, A(t)) \qquad (1)$$

where $D_t(R_i)$ is the passenger demand of region i in time interval t, $h$ is the historical window of time that is used for prediction, $D_{t-1}$ is the passenger demand of all regions in time interval t, $C_{t-1}$ is the crowd outflow of all regions at time interval t, $M_t$ is the meteorological data at time interval t, and $A(t)$ is the time feature at time interval t.

# 5 PROPOSED METHOD

As discussed in Section 3, we aim to predict the passenger demand of a region based on the historical observations supplemented with meteorological and time features. Previous studies [17, 21] have revealed that a region's passenger demand is not only dependent on its own historical data but also related to other regions' demand. Therefore, exploiting the spatial relationship between regions could help improve the prediction performance. Based on this reason, we decide to consider the historical observations from both the target region and other related regions[23]. This way, we can exploit the temporal and spatial relationships of passenger demand in different regions. The most critical question now turns into the following: how to efficiently exploit the latent temporal-spatial dependency between the historical observations of all the regions with the current passenger demand of the target region. To deal with this problem, we propose a spatial-temporal attention reinforcement learning model (STAR) to predict the passenger demand by automatically selecting the most informative historical observations. We explain the motivation of our proposed model in Section 5.1. The structure of our model is shown in Figure 1. The proposed approach contains three components: 1) Data Organization and Preparation model; 2) Information Zone Selection model; 3) Convolutional Neural Network based prediction model. We will introduce them in 5.2, 5.3, 5.4, respectively.

## 5.1 Motivation

The proposed model focuses on utilizing the historical observations of all the regions efficiently. This problem involves two sub-problems: (1) selecting the optimal time window of historical observations (considering the temporal relationship) (2) selecting the best combinations of relevant regions for the target region (considering the spatial relationship). The previous research efforts solve the two problems by making the above selections manually in a data-driven manner. However, manual selection cannot guarantee the optimal time window and the best combinations of relevant regions. Manual selection can sometimes introduce weak or negative correlations into prediction model which can influence the final performance. Besides, it is unfeasible for humans to select the combinations manually when the number of regions is enormous. Based on the above reasons, we design a historical and spatial zone selection mechanism to select the time window and relevant regions combination automatically. In our work, we transform the selection of time window and relevant regions into a problem that choosing the most distinctive subset from the overall historical observations of all regions. It is difficult to find the distinctive subset(denoted as information zone in the rest of our paper) manually by trial and error. In this context, we decide to design a selection mechanism which can find the optimal information zone automatically instead of utilizing exhaustive attack method manually.To optimize the information zone automatically, we employ deep reinforcement learning as the optimization framework for its excellent performance in policy optimization[13].

## 5.2 Data Preparation and Organization

**Meteorological and Time Data** As the passenger demand of region $i$ in a time interval of $t$, $D_t R_i$, is only related to the meteorological and time data of this region in the same time interval, we do not
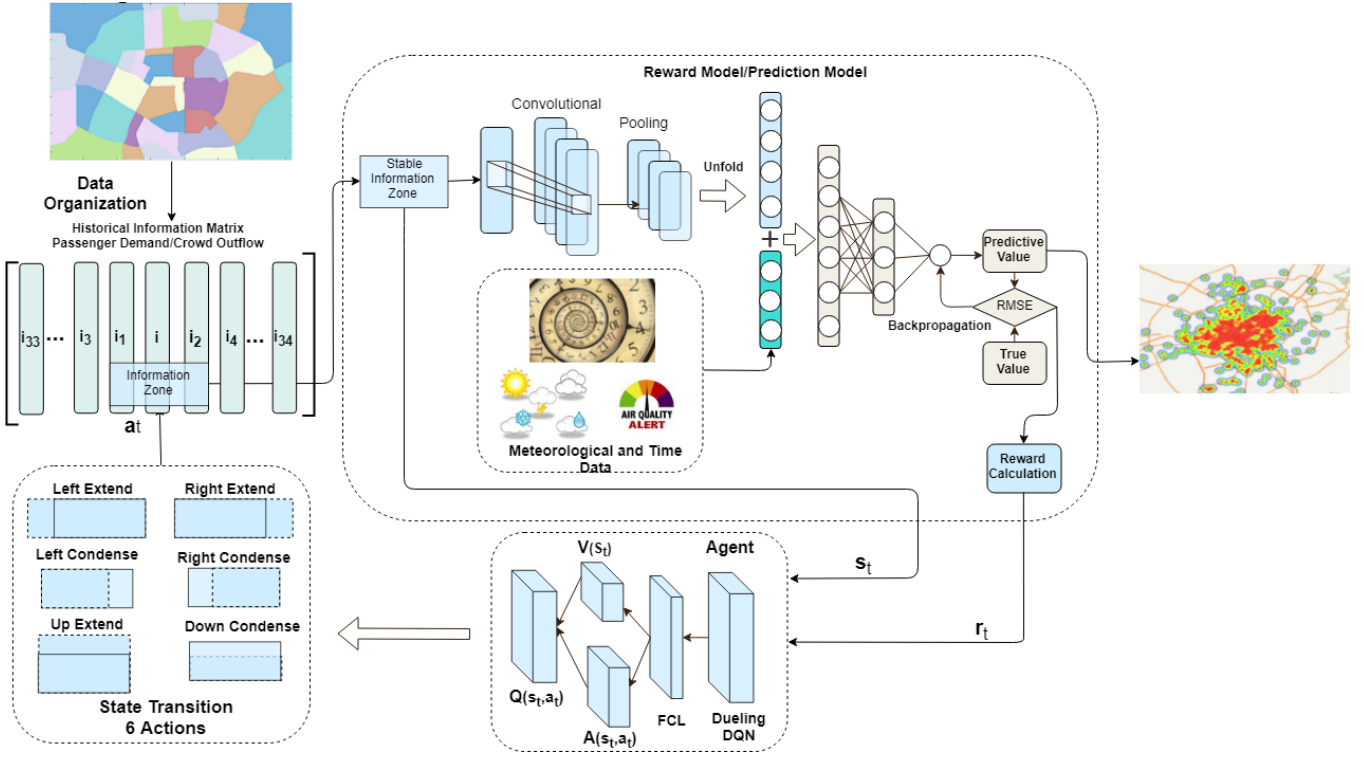
**Figure 1: Flowchart of the proposed model. The information zone is a selected fragment from the historical information matrix to feed to the state transition and the reward model. In each step $t$, the state transition selects one action to update $s_t$ based on the agent's feedback. The reward model evaluates the quality of the information zone. The dueling DQN finds the optimal information zone and pass it as the input of the prediction model. The prediction model serves as a reward model as well. The state transition contains six actions. Each action adjusts the information zone by one unit. The solid and dashed rounded rectangles denote the positions of the information zone before and after an action, respectively.**

need to consider any historical information of this kind of data or the meteorological data in other regions. Therefore, we can utilize the meteorological and time data directly without any additional processing and organization.

**Historical Passenger Demand and Crowd Outflow** As discussed above, selecting the optimal historical time window and the combinations of relevant regions remains a significant challenge, as most of the previous work choose both the time window and the relevant regions manually in a data-driven manner. That can be both time-consuming and resource-consuming when a large amount of data is concerned. There are generally two ways to select the relevant regions for a target region:(1) Considering all the regions as relevant with each other, treating the whole city as an image (a two-dimensional matrix), and then applying CNN directly to the image to capture relationships among regions. This method introduces not only the positive correlations but also the weak and negative correlations between regions, which may adversely impact the final prediction results. Additionally, the method may consume significant resources and time if there is a large number of regions in a city. (2) Instead of taking all the regions into account, the second method focuses on discovering local relationships, given that the adjacent regions are more likely to be relevant than the distant regions. This

method only uses the historical information from the target region and its surrounding regions but neglects the remote regions which have similar characteristics to the target region. Such remote regions usually share similar passenger demand patterns as the target region, and exploiting the relationships between these remote regions can also provide rich and discriminate information. Based on the above discussion, we design a new methodology to calculate the relevance between regions by considering both the distance and characteristics. The overview of our historical data organization is shown in figure 2. Since the existence of PoIs can be a key index of the characteristics of a region and consequently impact the passenger demand pattern of the region, we define the relevance score between two regions (region $i$ and region $j$) as follows:

$$Rel_{ij} = \frac{1}{||PoI_i - PoI_j||_1 * ||C_i - C_j||_2} \quad (2)$$

Where $PoI_i$ and $PoI_j$ are both $N$-dimensional vectors ($N$ is 12 in this work) representing the PoI information of the corresponding regions, $C_i$ and $C_j$ are both two-dimensional vectors representing the horizontal and vertical coordinates of the geometric center in region $i$ and region $j$, $Rel_{ij}$ denotes the relevance score between the region $i$ and region $j$. The above equation considers both the distance and characteristics of regions to calculate the relevance
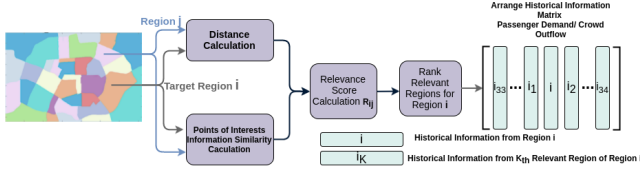
**Figure 2: Organization of historical data. PoI stands for point-of-interest. The light cyan rectangle labeled with $i$ denotes the historical information(passenger demand or crowd outflow) in region i while rectangel labeled with $i_K$ denotes the historical information from $K_{th}$ relevant region of region i**

scores of regions. The relevance score is inversely proportional to the distance between the geometric centers of the two regions and the difference of PoIs in the two regions. After calculating the relevance scores, we propose a method to reorganize the historical observations of different regions. The historical passenger demand and crowd outflow data are organized separately but in the same manner. For a given region $i$, we rank the remaining M-1 regions by the relevance scores between them and region $i$. Then, we manually set the upper limit of historical time window as $H$ to extract the $H$ hours historical information (passenger demand or crowd outflow) and can get a $H$-dimensional vector $V_k$ ($k \in [1, M]$) for each of the $M$ regions. To select the most informative zone from the overall historical information for region $i$, we first form a matrix using the $M$ vectors. Instead of combining them randomly, we put the M-1 vectors $V_k$ (k $\in [1, M]k \neq i$) at the two sides of the vector $V_i$ successively according to region relevance. Finally, we obtain a $M \times H$ historical information matrix where the target region and its highly relevant regions are placed in the central part while the lowly relevant regions are near the edge.

## 5.3 Information Zone Selection

After the data preparation, we get two matrices of historical observations (historical passenger demand and crowd outflow) for all the regions. The historical observations consist of two independent parts with the same dimension (historical passenger demand and historical crowd outflow), and we select the same information zone for the two parts in one unified selection model. In the following section, we take one of the two historical observation matrices as an example to explain the selection model. It is notable that the two historical information matrices actually share the same information zone. Hence, we only show on matrix and information zone in figure 2. We aim to find the most distinctive information zone (optimal time window and best combinations of relevant regions) from the whole matrix and feed it into the subsequent prediction model.

According to previous experience [17, 21], the passenger demand of target region is more related to the historical information of itself and other regions that have high relevance score with it. Based on this reason, we first initialize the information zone by choosing an $h$-hour historical observation of the target region and get a $1 \times h$ matrix. Then, our selection mechanism can include or exclude other regions into the information zone or adjust the length of time window step by step. After a certain number of operations, our selection mechanism can find the most distinctive zone by setting the optimal time window

and region combinations, and then feed the stable information zone from the matrices into the following prediction model.

The informative zone selection is shown in Figure 1, which includes two key components: the environment (including state transition and reward model) and the agent. The two components interact by exchanging three elements: the state $s$, the action $a$, and the reward $r$. All of the three elements are customized based on the specific situation. Next, we introduce the design of the crucial components of our deep reinforcement learning structure:

- The **State** $S = \{s_t, t = 0, 1, ..., T\} \in \mathbb{R}^3$ describes the size of information zone, where $t$ denotes the time stamp. Since the information zone is a variable-size fragment in the historical observation matrix, we design three parameters to denote the state at time interval $t$: $s_t = \{Start_{id}^t, End_{id}^t, TW_{id}^t\}$, where $Start_{id}^t$ and $End_{id}^t$ denote the start index and end index of information zone at time interval t, and $TW^t$ denotes the length of time window at time stamp t. For example, for a historical information matrix (number of regions is 3 and time window is 3 hours) [3,1,3,4;2,6,7,11;1,3,8,7], the state $\{Start_{id}^t = 1, End^{id_t} = 2, TW_{id}^t = 1\}$ is sufficient denoted the information zone as [1,3;6,7;3,8]. We set the upper limit of the historical time window to $H$ hours in data organization section. The size of historical information matrix is $35 \times H$(where 35 is the total number of regions). During the training, s0 is initialized from the historical information of target region and denoted by $[M - 1/2, M + 1/2, h]$, where $h$ is the initial time window set by us manually.

- The **Action** $A = \{a_t, t = 0, 1, 2, ..., T\} \in \mathbb{R}^6$, describes which the agent could choose to act on the environment. In our case, we define six categories of actions for the information zone (as described in the State Transition part in Figure 1): combining the left region, combining the right region, excluding the left region, excluding the right region, extending the time window, shortening time window. Each kind of action only adjusts the information zone by one unit on the historical observation matrix. At time stamp t, the state transition only chooses one action to implement following the agent's policy: $s_{t+1} = \pi(s_t, a_t)$

- The **reward** $R = \{r_t, t = 0, 1, 2, ..., T\} \in \mathbb{R}^1$ is calculated by the reward model, which also serves as the prediction model at the final stage. The reward is model is a convolutional neural network (to be detailed later). The principle of the reward model is to receive the current state and returns an evaluation as the reward.

- We employ the Dueling DQN (Deep Q Networks [16]) as the optimization **policy** $\pi(s_t, a_t)$ to learn the state-value function efficiently. Dueling DQN learns the Q value $V(s_t)$ and the advantage function $A(s_t, a_t)$ and combines them: $Q(s_t, a_t) \leftarrow V(s_t), A(s_t, a_t)$. The primary reason that we employ a dueling DQN to optimize the information zone is that it updates all the six Q values (since we have six actions in $a_t$, the $Q(s_t, a_t)$ contains six $Q$ values. This arrangement is similar to the one-hot label) at each step while other policies only update one value at each step.

**Reward Model**. The purpose of the reward model is to evaluate how the current state (the information zone) impact our final target which

refers to the passenger demand prediction in our case. Intuitively, the state which can lead to better prediction performance should gain a higher reward from the reward model: $r_t = F(s_t)$. The standard reinforcement learning framework mostly focuses on the classification problem, so the original reward model regards the accuracy as the reward. In contrast, our framework aims to predict the passenger demand, which is essentially a regression problem. Therefore, we can regard the root mean square error as the index to evaluate the state. As the amount of the training samples is relatively small in our case, we reuse the final prediction model as the reward model, which is a convolutional neural network. The specific structure of the reward model is as follows: The first and third layers are convolutional layers, the second and fourth are max-pooling layers, and the fifth and sixth are two fully-connected layers.

The first and third convolutional layers in our model take a set of independent filters and slide them over the whole information zone. Along the way, the dot product is made between the filters and chunks of the input information zone. Filters are used to generate the feature maps in each filter zone. In this way, the original feature map is projected into a stack of feature maps. Additionally, we apply the rectified linear unit (ReLU) activation function on the dot product from filters. After the convolutional layers, the hidden relationships between features can be combined by the dot product from each filter. The multiple filters in convolutional layers can be updated automatically via the evolution of the network. Following each convolutional layer, we add one max-pooling layer, which down-samples the feature representation, reduces its size and further allows for assumptions to be made about features contained in the sub-regions binned. The critical features are selected after the max-pooling layer. The information zone $I_t(i)$ selected at time interval $t$ of target region $i$ is fed into the convolutional and max-pooling layers firstly, and then unfolded to a one-dimensional vector $I'_t$. Then, we combine the one-dimensional vector and meteorological data $M_{t+1}$ at time interval $t + 1$, followed by feeding them into the following fully connected layers. The neurons in each fully-connected layer have full connections to all the activations in the previous layer, as in regular Neural Networks. Their activations can, therefore, be computed using matrix multiplication followed by a bias offset. As our case is a regression problem, we replace the softmax function used in a traditional CNN with a linear function. The output $D'_{t+1}(i)$ from the last year is the predicted passenger demand of region $i$. The loss function is defined as follows

$$\mathcal{L} = RMSE(D'_{t+1}(i) - D_{t+1}(i)) \tag{3}$$

where RMSE is the root mean square error, L denotes the loss. The loss function can gradually converge via the training of the reward model, and it stops when the number of training epochs reaches the pre-defined value or when the updated value meets the criteria. After the convergence of the reward model, we define the reward value at time interval $t$ as a function of the loss:

$$r_t = \frac{1}{\mathcal{L}} \tag{4}$$

After obtaining the reward value $r_t$ at time interval $t$, we can feed it back to the agent for choosing the next action. By doing so, we can link the reward with the performance of the selected information zone. The reward becomes larger when the loss of reward model decreases.

## 5.4 Prediction Model

As we have mentioned in the above section, our prediction model also serves as the reward model. After several training epochs, the selection model will gradually converge, and the information zone will be optimized. Note that the reward model is trained at each time stamp of the selection model. Once the information zone is set, it is sent to the prediction model as the input, and the prediction model will be trained for the last time.

## 6 EXPERIMENT

In this section, we will describe the experiment settings(dataset and comparison methods) in Section 6.1, and detail the hyper-parameters used in our model in Section 6.2. Finally, the comparison results are shown in Section 6.3

## 6.1 Experiment Settings

We evaluate our method with the real-world dataset described in Section 4.1. This dataset is collected in a major city in China provided by Chu et al. [1]. Before feeding data into the model, we first transfer categorical features such as the weather state, air quality level, hour of the day, the day of week and holiday by one-hot encoding. Continuous features including temperature, wind speed, and visibility are normalized. We set the length of the time interval to one hour, the upper limit of time window length $H$ to 12 hours, and the initial time window $h$ to four hours. As for the comparative models which require one-dimensional input (MLP and DBN), we unfold the two historical information matrices into one-dimensional vectors and combine them with the meteorological and time data as the final input. As for the other three deep learning methods(LSTM, CNN and CNN-SVR), we feed the historical passenger demand and crowd outflow as initial input. Similar to our model, we combine the meteorological and time data with the initial input before they are fed into the first fully connected layer. As for the LSTM+CNN model, we utilize the convolutional layer and long short term memory layer to process the historical information respectively, and then combine them with meteorological and time data as the above methods do. We compare our model with a series of methods including benchmark baselines and state-of-art method. For each method, we tune its parameters to report the best performance.

- **HA**: The historical average model predicts future passenger demand by calculating the average value of previous passenger demand in the same related time interval at the same region.
- **ARIMA**: The Auto-Regressive Integrated Moving Average model is a widely used time series prediction model which is a generalization of the Auto-Regressive Moving Average (ARMA) model.
- **SARIMA**: The Seasonal Auto-Regressive Integrated Moving Average model [11] is a variance of ARIMA, which can capture the seasonality in a time series data.
- **OLSR**: The Ordinary Least Square Regression [7] model is a kind of linear regression model. It can estimate the relationship between multiple variables.
- **MLP**: Multiple Layers Perception is a typical class of feedforward neural network. In our case, we combine all the

data into a one-dimensional vector and feed it into the MLP network.

- **Deep Belief Network Regression (DBN)**: A deep belief network [4] where deep belief network sites at the bottom for unsupervised feature learning with a linear regression layer set at the top of supervised prediction.
- **Convolutional Neural Network (CNN)**: is a variation of the various traditional perceptions. It utilizes the filters in the convolutional layers to capture the relationships between various features.
- **Convolutional Neural Network with Support Vector Regressor (CNN-SVR)**: is the regression version of Huang's work [6] which utilizes the support vector regressor to replace the output layer of original CNN. The parameters are set to the same configurations as the CNN with support vector regressor.
- **Long Short Term Memory Neural Network (LSTM)**: LSTM is a variation of recurrent neural networks[5], which is prominent in sequence data processing. LSTM employs all the data to predict future passenger demand.
- **Long Short Term Memory and Convolutional Neural Network (LSTM+CNN)**: We combine the LSTM and CNN model, and employ them to capture the temporal and spatial relationship respectively.
- **DMVST-Net**:DMVST-Net [21] is a state-of-art method proposed by Yao et.al. DMVST-Net consists of three main components capturing the temporal view, spatial view and semantic view respectively. We reproduce this model on our dataset and tune the parameters for optimal performance.

## 6.2 Hyper-parameters Tuning

The hyper-parameters of the prediction model (CNN) are configured as follows: the number of layers in CNN is set to 6 with two convolutional layers, two max-pooling layers, and two fully connected layers. As the size of the information zone selected can differ in each step, we extend the information zone to keep the same input size ($35xH$) of CNN by filling zeroes. We set the number of filters in each convolutional layer to 10, and the convolutional kernel size set to $2x3$, the number of nodes in first and second fully connected layer to as 100 and 1, the loss function as RMSE.

## 6.3 Comparison Results

We show the overall root mean square error (RMSE) for all regions and mean absolute error (MAE) for each region in Table 2. We will explain the RMSE and MAE respectively in the following equation:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N}|R_i - P_i|^2}{T}} \qquad (5)$$

where T is the total number of time intervals in testing data, N is the number of testing samples, $R_i$ denotes the real value of $i_{th}$ sample and $P_i$ denotes the predicted value of $i_{th}$ sample.
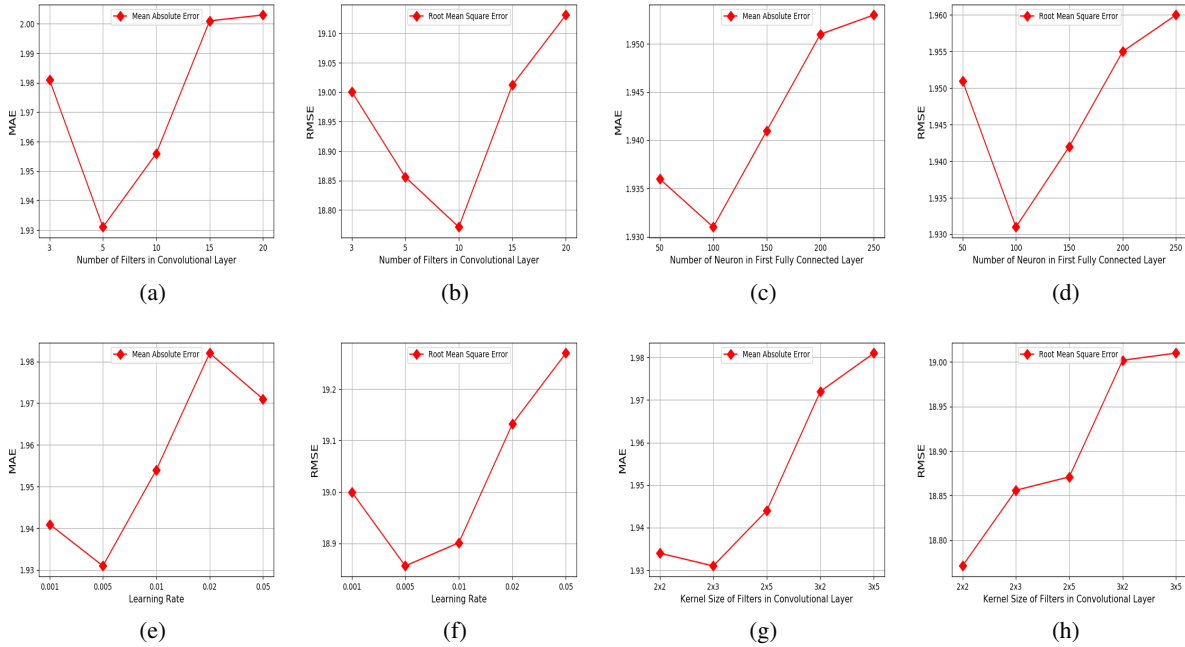
$$MAE = \frac{\Sigma_{i=1}^{N}|R_i - P_i|}{N} \qquad (6)$$

where N is the total number of testing samples, $R_i$ denotes the real value of $i_{th}$ sample and $P_i$ denotes the predicted value of $i_{th}$ sample.

From the table, we can observe that our model outperforms the baseline and state-of-art methods in both two evaluation metrics (18.856 in RMSE and 1.931 in MAE respectively). The DMVST-Net [21] performs best among all the comparison methods. Compared to the DMVST-Net, our model achieves 6.19% and 7.07% in RMSE and MAE respectively. Among all the baseline methods, deep learning based methods all outperform the other traditional methods, which confirms the effectiveness of neural networks in capturing the latent relationship between various data sources. The LSTM+CNN achieves the best performance among all the baseline methods due to its ability to capture the spatial and temporal correlations simultaneously. By introducing the information zone selection mechanism to our model, we improve the prediction performance by a large margin (9.2% and 10.8% in RMSE and MAE respectively) compared to the original CNN. As our model adds the information zone selection mechanism into the original CNN, the comparison between CNN and ours can be regarded as an ablation study. The significant improvement in the real dataset confirms the effectiveness of the selection component in our model. Furthermore, we also conduct extensive experiments on exploring how different parameters settings of prediction model can influence final performance and show the results in Figure 3. We can observe that the performance of our model is sensitive to four kinds of parameters settings: the number of filters in the convolutional layer, the kernel size of filters in the convolutional layer, the number of neurons in the first fully connected layer, and the learning rate of the prediction model. However, these parameters do not influence the final prediction performance in a wide range. Among all the four parameters, the learning rate is the most insensitive parameter to our model. Additionally, we also evaluate the effectiveness of various data in our work. We exclude the corresponding data from our model respectively and present the final results in Figure 5. We denote the four kinds of data as passenger demand (P), crowd outflow (C), meteorological data (M), and time data (T). We denote STAR-P, STAR-C, STAR-M, STAR-T and STAR-TM to represent our model excluding the passenger demand data, crowd outflow data, meteorological data, time data and meteorological plus time data. From Figure 5, we can find that excluding each kind of data can reduce the prediction performance. There is no doubt that passenger demand is the most crucial data in our model as excluding it from our model increases 9.6 in RMSE and 5.1 in MAE. This increase suggests that there exists a strong correlation between the historical and current passenger demand. Following the passenger demand, crowd outflow is the second most important data which can result in 11.1% and 8.09% improvement in prediction performance.

In order to provide a direct insight of our model performance, we choose four regions (region 3, region 7, region 14 and region 15 denoted in the dataset) as example to compare the true passenger demand and predicted passenger demand by our model. We utilize the latest 132 hours as testing time period. The comparison results are shown in figure 4. From the figure, we can observe that our model successfully capture the rise and fall of the true passenger demand. The peak time interval in our predicted trend can basically match the true passenger demand, which confirms the value of practical application in real life. Although the difference between predicted demand and real demand fluctuates over time, it does not show an abnormally large value which confirms the stability of our model.

(a)                                (b)                                (c)                                (d)



(e)                                (f)                                (g)                                (h)

**Figure 3: Mean Absolute Error (MAE) (a) and Root Mean Square Error (RMSE) (b) comparison between different number of filters in first convolutional layer; Mean Absolute Error (MAE) (c) and Root Mean Square Error (RMSE) (d) comparison between different number of neurons in first fully connected layer;Mean Absolute Error (MAE) (e) and Root Mean Square Error(RMSE) (f) comparison between different learning rate; Mean Absolute Error (MAE) (g) and Root Mean Square Error (RMSE) (h) comparison between different kernel size of filters in convolutional layer**

**Table 2: Comparison of different methods. RMSE represents root mean square error. MAE represents mean absolute error.**

| Method | RMSE | MAE |
|---|---|---|
| HA | 23.215 | 2.33 |
| ARIMA | 22.147 | 2.28 |
| SARIMA | 21.943 | 2.24 |
| OLSR | 21.570 | 2.16 |
| MLP | 21.628 | 2.18 |
| DBN | 21.613 | 2.191 |
| LSTM | 21.531 | 2.171 |
| CNN | 20.767 | 2.166 |
| CNN-SVR [6] | 20.911 | 2.134 |
| LSTM+CNN | 20.353 | 2.113 |
| DMVST-Net [21] | 20.102 | 2.078 |
| **Ours** | **18.856** | **1.931** |

## 7   CONCLUSIONS

In this paper, we propose a spatial-temporal dynamics based reinforced learning model for citywide passenger demand prediction which can solve the imbalance problem between supply and demand. Our model jointly incorporates various data into one framework and select the most distinctive information zone from historical observation. The selected information zone can introduce strong spatial and temporal correlations while avoiding to be influenced by weak or negative correlations. Extensive experiments on a real dataset show that our model outperforms a series of state-of-the-art method by reducing RMSE and MAE by 9.2% and 10.8%, respectively. Furthermore, our model can be easily extended to other prediction problems with complex relationships between various features. As part of our future work, we will explore word of mouth information such as tweets posted at real-time online as the extra hint for the prediction. Additionally, we hope to design an advanced data organization and information zone selection mechanism which can traverse each possible combination of region and time window and select the global optimal one efficiently. Furthermore, it is also worth to designing a user-friendly mobile interface for providing a convenient service for both passengers and drivers.

## REFERENCES

[1] Jing Chu, Kun Qian, Xu Wang, Lina Yao, Fu Xiao, Jianbo Li, Xin Miao, and Zheng Yang. 2018. Passenger Demand Prediction with Cellular Footprints. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE.

[2] Ashim Kumar Debnath, Hoong Chor Chin, Md Mazharul Haque, and Belinda Yuen. 2014. A methodological framework for benchmarking smart transport cities. *Cities* 37 (2014), 47–56.

[3] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1525–1534.

[4] Geoffrey E Hinton. 2009. Deep belief networks. *Scholarpedia* 4, 5 (2009), 5947.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

(a)

(b)

(c)

(d)
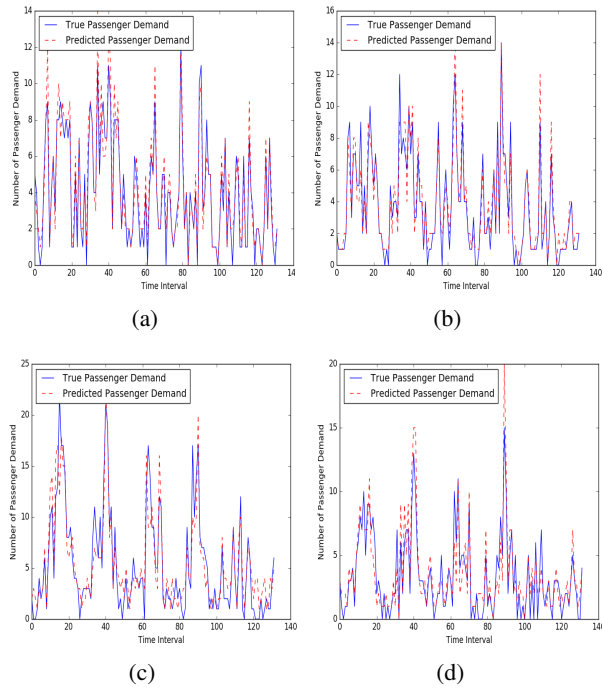
**Figure 4: The comparison between the true passenger demand trend and the predicted passenger demand trend during the last 132 hours in four regions: region 3 (a), region 7 (b), region 14 (c), region 15 (d). The blue solid line indicates the true passenger demand trend, the red dashed line indicates the predicted passenger demand by our model**



(a)

(b)
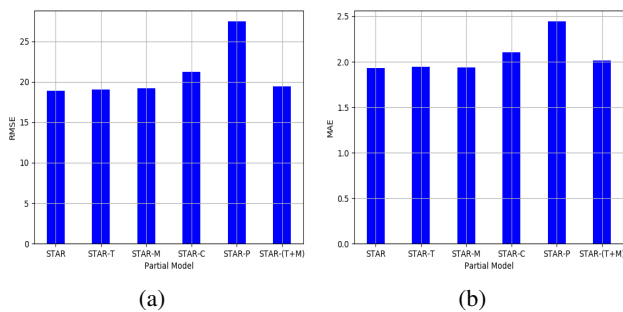
**Figure 5: Comparison of different partial models with respect to (a) RMSE and (b) MAE. STAR represents our original reinforced learning spatial-temporal model. STAR-T, STAR-M, STAR-C, STAR-P represent the variants of our model that excludes the time data, meteorological data, crowd outflow data, and passenger demand data, respectively. STAR-TM represents the version of our model that excludes both the time and meteorological data.**

[6] FJ Huang and Y LeCun. 2006. Large-scale learning with svm and convolutional netw for generic object recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[7] Ling Leng, Tianyi Zhang, Lawrence Kleinman, and Wei Zhu. 2007. Ordinary least square regression, orthogonal regression, geometric mean regression and their applications in aerosol science. In *Journal of Physics: Conference Series*, Vol. 78. IOP Publishing, 012084.

[8] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6, 1 (2012), 111–121.

[9] Yunxuan Li, Jian Lu, Lin Zhang, and Yi Zhao. 2017. Taxi booking mobile app order demand prediction based on short-term traffic forecasting. *Transportation Research Record: Journal of the Transportation Research Board* 2634 (2017), 57–68.

[10] Ye Liu, Yu Zheng, Yuxuan Liang, Shuming Liu, and David S Rosenblum. 2016. Urban water quality prediction based on multi-task multi-view learning. (2016).

[11] Edson Zangiacomi Martinez, Elisângela Aparecida Soares da Silva, and Amaury Lelis Dal Fabbro. 2011. A SARIMA forecasting model to predict the number of cases of dengue in Campinas, State of São Paulo, Brazil. *Revista da Sociedade Brasileira de Medicina Tropical* 44, 4 (2011), 436–440.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[14] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.

[15] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. 2017. DeepSD: supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 243–254.

[16] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. 2015. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581* (2015).

[17] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. *arXiv preprint arXiv:1802.08714* (2018).

[18] Lina Yao, Quan Z Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware point-of-interest recommendation using tensor factorization with social regularization. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, 1007–1010.

[19] Lina Yao, Quan Z Sheng, Xianzhi Wang, Wei Emma Zhang, and Yongrui Qin. 2018. Collaborative Location Recommendation by Integrating Multi-dimensional Contextual Information. *ACM Transactions on Internet Technology (TOIT)* 18, 3 (2018), 32.

[20] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 777–785.

[21] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.. In *AAAI*. 1655–1661.

[22] Kai Zhang, Zhiyong Feng, Shizhan Chen, Keman Huang, and Guiling Wang. 2016. A framework for passengers demand prediction and recommendation. In *Services Computing (SCC), 2016 IEEE International Conference on*. IEEE, 340–347.

[23] Xiang Zhang, Lina Yao, Chaoran Huang, Sen Wang, Mingkui Tan, Guodong Long, and Can Wang. 2018. Multi-modality Sensor Data Classification with Selective Attention. *arXiv preprint arXiv:1804.05493* (2018).

[24] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1436–1444.