Study on Taxi Demand Prediction using Context and Spatio-Tamporal Data

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: January 29th, 2020

Sakura Yamaki
(F5118106-1)

Advisor: Wataru Kameyama
Research guidance: Research on Multimedia Information Distribution Systems

ABSTRACT

Taxi demand is closely linked to human travel habits. Accurately forecasting taxi demand is very important for passengers, drivers, ride platforms and city managers, but is very difficult in three aspects:

I. **Diversity of ST correlations.** Diversity of such spatio-temporal correlations, which vary from location to location and depend on the surrounding geographical information, *e.g.*, land use and road networks.
II. **Long-term periodic dependency.** In generally, traffic data show a strong daily and weekly periodicity and the dependency based on such periodicity can be useful for prediction. However, one challenge is that the traffic data are not strictly periodic.
III. **Sudden demand changes.** Spatio-temporal sudden demand changes, which is temporary increasing or decreasing demand due to weather condition, *e.g.*, rain and snow, traffic conditions, *e.g.*, accident delays, and large event, *e.g.*, concerts and sports games.

To tackle these challenges, we use a deep-meta-learning based model, named ST-MetaNet method [1]. To the best of our knowledge, this study is the first to study the inherent relationship between geographical information and ST correlations by applying meta learning on the related application, *i.e.*, urban traffic prediction. This research is the state-of-art traffic demand prediction method, to collectively predict traffic in all location at once. ST-MetaNet leverages the meta knowledge extracted from geo-graph attributes to generate the parameter weights of a graph attention network and a recurrent network within sequence-to-sequence architecture. Geo-graph attributes are consist of node attributes and edge attributes. As a result, it can capture the inherent relation between geo-graph attributes and diverse types of ST correlations, these solve first and second challenging aspects. On the other hand, this research is NOT considered spatio-temporal sudden demand changes which is third aspect. So, in order to solve this problem, we added contextualized dynamic spatio-temporal data by expanding ST-MetaNet method. Dynamic contextual factors of demand include such as weather condition, large events occurrence. Experiments were conducted based on two kind of real-world datasets from NYC TLC Trip Record to illustrate the high accuracy taxi demand prediction.

Keywords: Taxi demand prediction, Spatio-temporal data, Meta-knowledge learning,

TABLE OF CONTETNS

FIGURE INDEX

TABLE INDEX

CHAPTER 1: INTRODUCTION

1.1 Introduction and Background

The transport system is in the city, as the blood tissue is in the human body, which is the key to urban construction. In the future, it will play an integral role in building smart cities. With the development of Uber and Lyft, online car healing has become a travel habit for people. Therefore, extensive data on taxi demand have been collected for research. For example, as of February 1, the New York City Taxi and Limousine Commission (TLC) is requiring all High-Volume For-Hire Services (HVFHS) – including Uber, Lyft, Via and Juno – to pay drivers at least the minimum per-trip payment amount, as per the TLC's new Driver Pay Rules. Accurate taxi demand forecasting helps passengers avoid hot spots rationally and save waiting time. It also helps drivers to reasonably select hot spots and improve the balance between benefits and efficiency. With an online car healing platform, you can better plan, pre-allocate resources and maximize profits. For city managers, we can provide reference suggestions for infrastructure construction and transportation planning. Therefore, accurately predicting future taxi demand in different parts of the city based on historical data is a hot area of research.

1.2 Purpose and Motivation

Urban Traffic Prediction. The urban traffic prediction is a typical spatio-temporal forecasting problem, whose main factors include spatial and temporal correlations that are implicitly affected by the characteristics of nodes (locations) and edges (the mutual relation between two nodes) .Intuitively, the characteristic of a node is influenced by its attributes, like GPS locations and nearby POIs. Likewise, the characteristic of an edge depends on its attributes, e.g., road connectivity and the distance between locations. In a typical traffic forecasting setup, it is necessary to evenly forecast the traffic in the next time slot, taking into account past traffic data. Many researchers have studied traffic forecasts for decades. In the time-series community, autoregressive integrated moving average (ARIMA) and Kalman filtering are widely applied to traffic prediction problems. Recent studies have begun to consider spatial information (eg, adding regularization of model similarity for nearby places) and external contextual information (eg, adding venue information, weather conditions, local event capabilities) [2],[3]. In addition, spatial information has also been explicitly modeled in recent studies. For example, Deng et al. [4] used matrix factorization on road networks to learn the latent space between road connected regions for predicting traffic volume. However, these approaches are still based on traditional time series or machine learning models and do not capture complex nonlinear spatiotemporal dependencies.

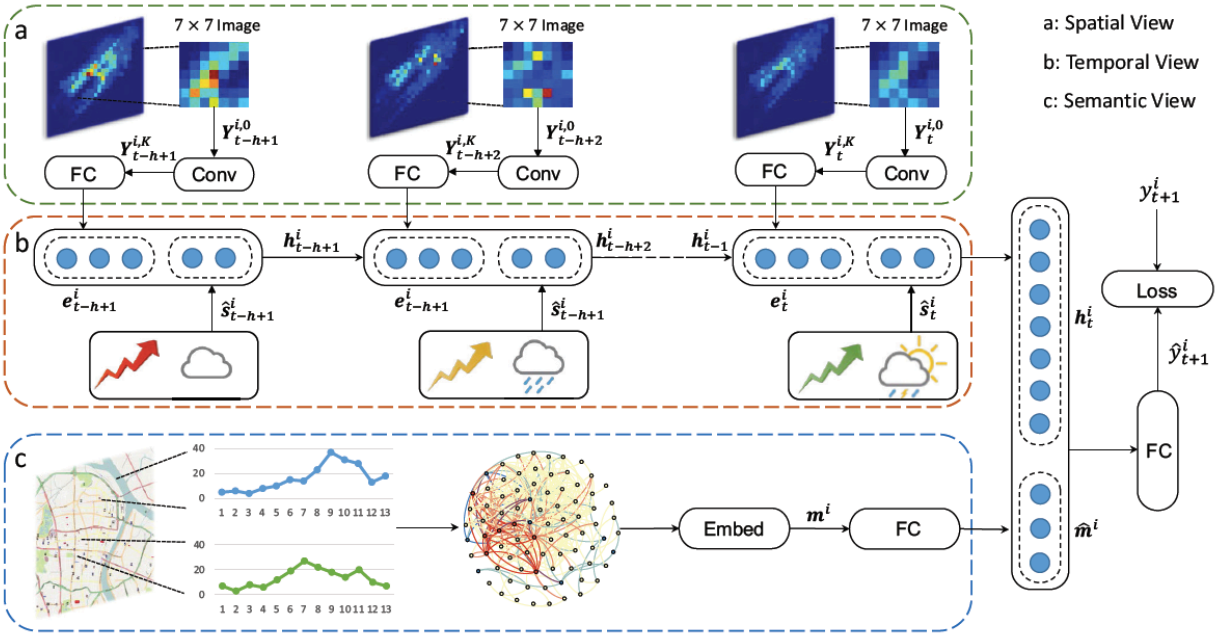


Figure 1: The Architecture of DMVST-Net proposed from Yao et al (2018) Cited from [7].

Deep Learning for Spatio-Temporal Prediction. Recently, deep learning has been a great success for many challenging learning tasks. This success has stimulated several studies applying deep learning techniques to traffic prediction problems. For example, several studies [5] modeled city-wide traffic as heatmap images and used convolutional neural networks (CNN) to model nonlinear spatial dependencies. To model non-linear time dependence, researchers have proposed using a recurrent neural network RNN-based framework from Yu et al (2017) [6]. Yao further proposes a way to model both spatial and temporal dependencies by integrating CNN with long-term short-term memory (LSTM) [7] as shown in Figure 1. Although deep learning of traffic prediction takes into account both spatial dependence and temporal dynamics, existing methods have three major limitations.

1. **Diversity of ST correlations.** First, the spatial dependence between locations depends only on the historical traffic similarity [5], and the model learns static spatial dependencies. However, dependencies between locations can change over time. For example, in the morning, the dependency between residential areas and business centers may be stronger. On the other hand, in the evening, the relationship between these two places may be very weak. However, while work on urban traffic forecasting and similar ST forecasting tasks has increased significantly, the aforementioned problems have not yet been fully resolved. First, some works [8], [9]. focus on modeling ST correlations with a single model of all locations. However, these methods

cannot explicitly model the inherent relationship between geographic graph attributes and various types of ST correlations as shown in Figure 2.

2. **Long-term periodic dependency.** A second limitation is that many existing studies ignore long-term cyclical dependency shifts. Traffic data shows strong daily and weekly periodicities, and dependencies based on such periodicities are useful for forecasting. One challenge, however, is that traffic data is not strictly periodic. For example, weekday peak times typically occur late in the afternoon, but can vary from 4:30 pm to 6:00 pm on some days. Previous studies [5] consider periodicity, but do not consider the sequential dependence of the periodicity and the time shift.
3. **Sudden demand changes.** The third limitation is that in many existing studies, ignore extreme in short-term demand due to weather conditions such as rain and snow, traffic conditions such as accident delays, and large events such as concerts, parades, and sports games. Most existing work focuses on predicting taxi demand in the near future by learning patterns from historical data. However, events with unusually high demands are non-repetitive and will fail to capture unusual events because they violate common assumptions, such as how smoothly the demand changes over time.

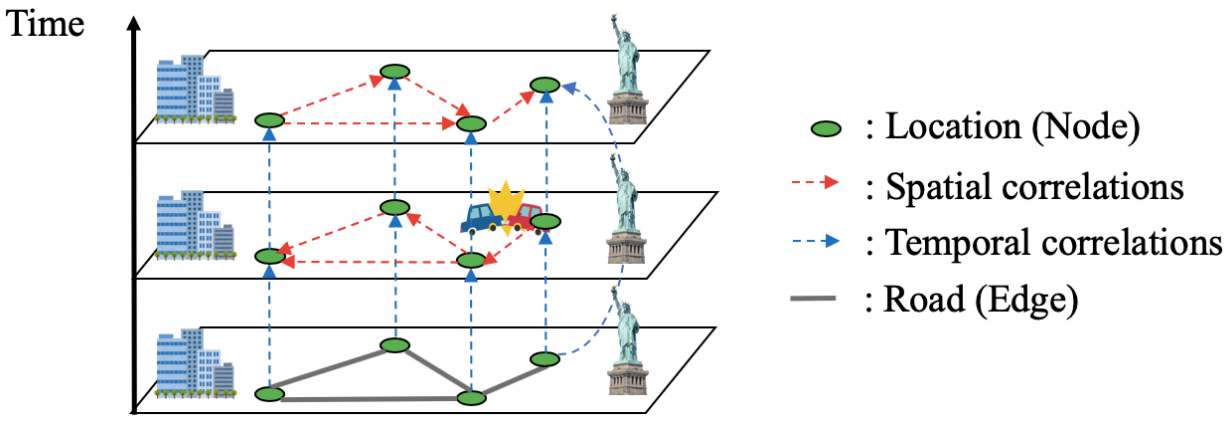


Figure 2: Composition of ST correlations.

To tackle the above challenges, we use ST-MetaNet [1] as a spatio-temporal prediction method based on deep learning using geographic feature and dynamic feature to predict taxi demand. Based on these insights, ST-MetaNet first extracts the meta knowledge (i.e. characteristics) of nodes and edges from their attributes, respectively. The meta knowledge is then used to model the spatio-temporal correlations, namely, generating weights of the prediction networks. And by adding dynamic features to ST-MetaNet, ST-MetaNet can consider both spatio-temporal correlation and dynamic changes. The main intuition is drawn in Figure 2. The contributions of our work are three folds:

- We use a novel deep meta learning based model, named ST-MetaNet, to predict urban yellow taxi and HVFHV in New York city. ST-MetaNet leverages the meta knowledge extracted from geo-graph attributes to generate the parameter weights of a graph attention network and a recurrent network within sequence -to-sequence architecture. As a result, it can capture the inherent relation between geo-graph attributes and diverse types of ST correlations.
- Then we add the dynamic features such as weather, event data of NYC and time information. And embedded them into RNN layer to capture extreme short-term and periodic long-term components for demand prediction.
- We evaluate ST-MetaNet on two datasets: yellow taxi and HVFHV demand prediction. The experiment results verify that ST- MetaNet can significantly improve taxi demand prediction, and learn better traffic-related knowledge from geo-graph attributes and temporal dynamic features.

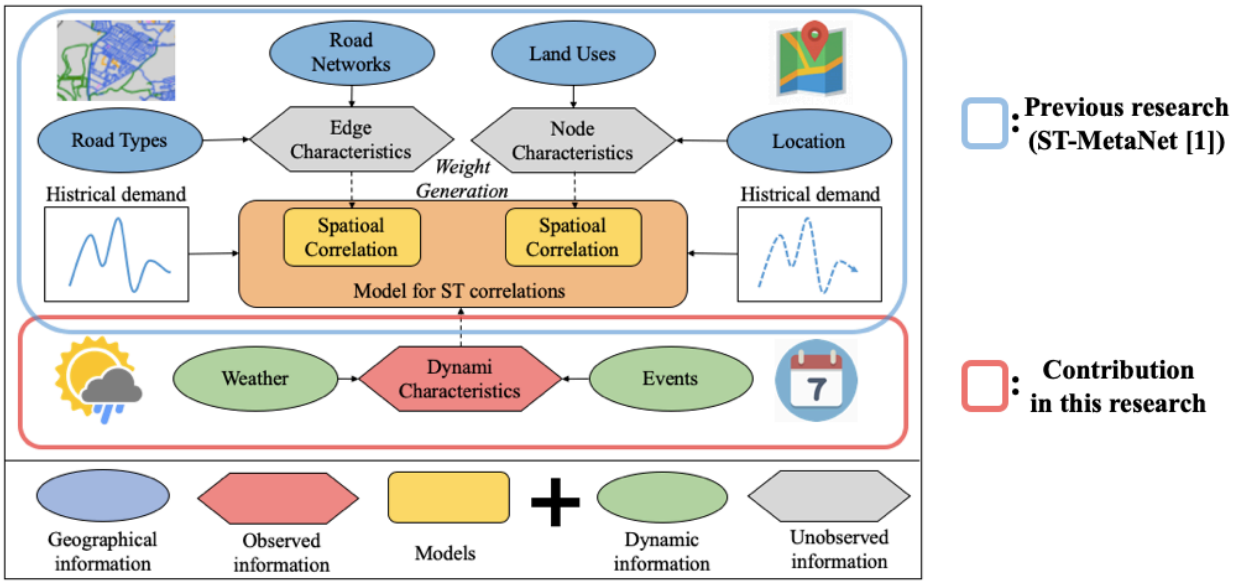


Figure 3: Insight of the framework.

1.3 Structure of Thesis

The rest of this paper is organized as follows: Chapter 2 describes the investigation of the spatio-temporal prediction method. Chapter 3 describes the details of the data, how to process the data, and how to draw the results of the data processing using maps. Chapter 4 provides definitions and problem descriptions and describes the method in detail. Chapter 5 describes the experimental conditions and the results of the prediction model and evaluates the prediction results for each baseline. Finally, the paper concludes with Chapter 6 which discusses future work. An appendix is attached at the end to support the reproducibility of the results of this paper.

CHAPTER 2: RELATED WORK

Machine learning is widely used in a variety of areas, including recommendation systems, service computing, prediction problems, and edge computing. In this 10 years, deep learning became also widely used for many areas to deal with multi-tasking learning to improve application performance. In recent years, deep learning has been widely used in many research fields, with great success in computer vision and natural language processing. The traffic data forecasting problem using deep learning is one of the hot topic problem. This is because deep learning-based prediction models can extract more complex spatiotemporal features than previous machine learning models. The traffic data forecasting problem (our problem), which consists of traffic volume, traffic speed, travel time, and taxi demand. These approaches can be divided into two groups: traditional approaches, deep learning approaches and Meta Learning for Neural Networks' Weight Generation.

2.1 Traditional Approach

Time series algorithms are first introduced into predicting traffic data in an ARIMA-like model. Hamed et al. [10] developed an ARIMA model to predict the traffic volume on urban arterials.From here on, to improve prediction performance, researchers applied many variants of ARIMA for traffic prediction. On the other hand, machine learning algorithms are also widely used in this filed. Wu et al. [11] applied support vector regression for travel-time prediction, K-nearest neighbor model to predict the duration of traffic accidents. k-NN models are also widely applied in predicting traffic speeds and volume due to its simple nature [12]. These methods focus on the temporal correlation of traffic data and ignore spatial correlation. However, the traffic conditions in the current area are affected not only by neighboring areas but also by distant areas. For example, a traffic accident at an intersection can cause roads to become impassable, resulting in a dramatic increase in traffic at remote transportation hubs.

2.2 Deep Learning Approach

In recent years, many researchers have widely used deep learning techniques to predict traffic data. CNN has proven to be effective in extracting features from images. Therefore, by treating the traffic situation of the entire city as an image, many researchers have adopted CNN for predicting traffic data. Deng et al. [4] divided the city into many tiny grids, converting city traffic speed into images and use CNN for predicting traffic speed. Zhang et al. [5] employed CNN modeling temporal dependent and spatial dependent for predicting traffic flow, rent/return of bikes and traffic flow. Later, Zhang et al. [5] used a residual neural network, a parametric-matrix-based fusion mechanism, and external information to improve the performance in predicting crowd flows. These studies focus more on the spatial correlation of traffic data. On the contrary, in modeling

time correlation, CNN is a simple fusion function extracted through a neural network and does not make full use of time correlation. On the other hand, the success of RNNs and their variants in continuous learning tasks [7], long-term short-term memory (LSTM) and gated repetitive units (GRU), has led many researchers to predict traffic data based on them. became. . However, it does not make full use of spatial correlation. To maximize the use of spatio-temporal correlation, many researchers have combined CNN and RNN structures for traffic prediction, such as graph convolution [13] and graph attention [14]. In these studies, similarities between regions are based on static distance or road structure. They also overlook long-term periodic effects and time shifts in time series forecasting. Very recent studies [8], [9] show that attention mechanisms enable RNNs to capture dynamic spatiotemporal correlations of geosensory data.

2.3 Meta Learning for Neural Networks' Weight Generation

[15] proposed a dynamic filter network that generates a convolution filter conditioned on input. [16] The learner was used to predict pupil network parameters from a single sample. [17] We used hyper networks to generate large network weights. This can be considered as weight sharing between layers. Recently [18] proposed meta-multitask learning of NLP tasks. [19] suggested that to amortize the cost of neural architecture search, embed a neural architecture and employ a hypernetwork to generate its weights. Unlike the above works, we aim to model various ST correlations. To our knowledge, [1] first studies the unique relationship between geographic information and ST correlation, and applies meta-learning to a related application: urban traffic forecasting.

CHAPTER 3: DATA DESCRIPTION AND PROCESSING

In this chapter, we descript 5 kinds of data used for out dataset. And then we introduce data processing of each data. All original data are public released and Taxi data, Load usage data, Road data and Event data can be download from NYC OpenData [22]. Weather data can be downloaded from Data Transmission Network and Dataline website [23]. Links are attached below.

3.1 Taxi Demand Data

NYC Taxi & Limousine Commission (TLC) [24] has released public datasets from January 2009 that contain data for taxi trips in NYC, including timestamps, pickup & drop-off from 263 taxi zones as shown in Figure 3, number of passengers and so on. We regard one taxi zone as one node. TLC provide the yellow taxi, green taxi, For-Hire Vehicle ("FHV") and High Volume FHV trip records. We use yellow taxi and High Volume FHV (HVFHV) trip records as datasets which are relatively large volume data. Yellow taxi and HVFHV data dictionary are shown in appendix section. Please refer to appendix. The data processing is conducted by counting the number of pick up and drop off taxi every 10 minutes in each 263 zones. Pickup value is used as target variable of each node. As a result, the amount of yellow taxi pickup is 36,721,325 and the amount of HVFHV pickup is 109,089,532 from 1st on Feb. to 31st June in 2019 all taxi zones. Figure 4 show the color map of taxi pickups of yellow taxi and HVFHV data in each taxi zone.

3.2 Land Use Data

New York City's Department of City Planning keep a geospatial database of tax lots and their associated records that goes by the name of PLUTO (Primary Land Use Tax Output) [25]. It includes polygons for the city's 870,000+ tax lots, and over 80 attribute columns including assessments, easements, number of units, number of floors, zoning status, etc. The Department of City Planning The Department of City Planning has created 11 land use categories and assigns each BUILDING CLASS to the most appropriate land use category (LUs) and provide the area of each land use category. We summarize the value description of LUs category in Table 1. Figure 5 shows map f 11 land use categories area. Data processing is conducted using LUs to extract each node features by calculating the area of the 11 land use categories included every 263 taxi zones and used them as node attributes. Comparing Figure 4 with Figure 5, it can be found that there is correlation between taxi demand and LUs.

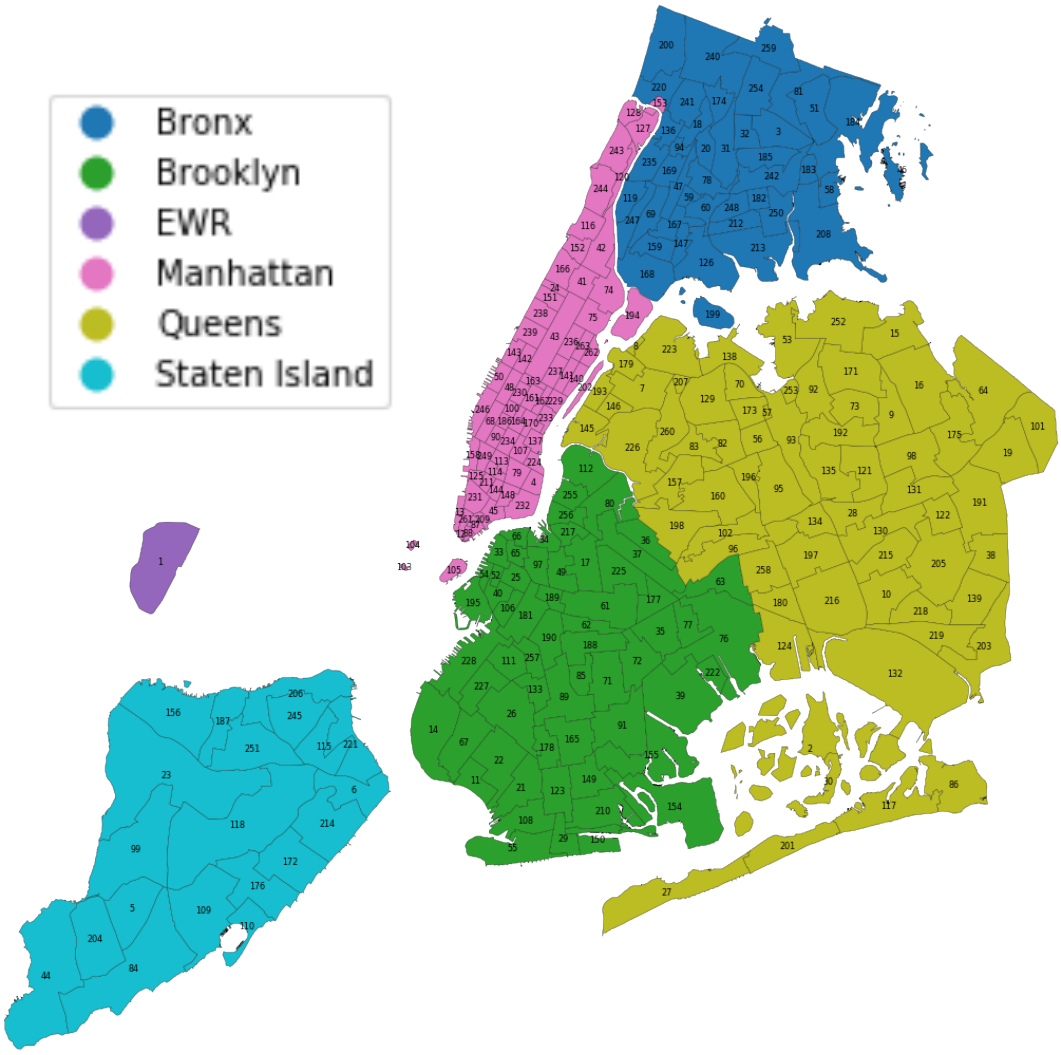


Figure 4: Taxi zone map (drawn by author using [24]).

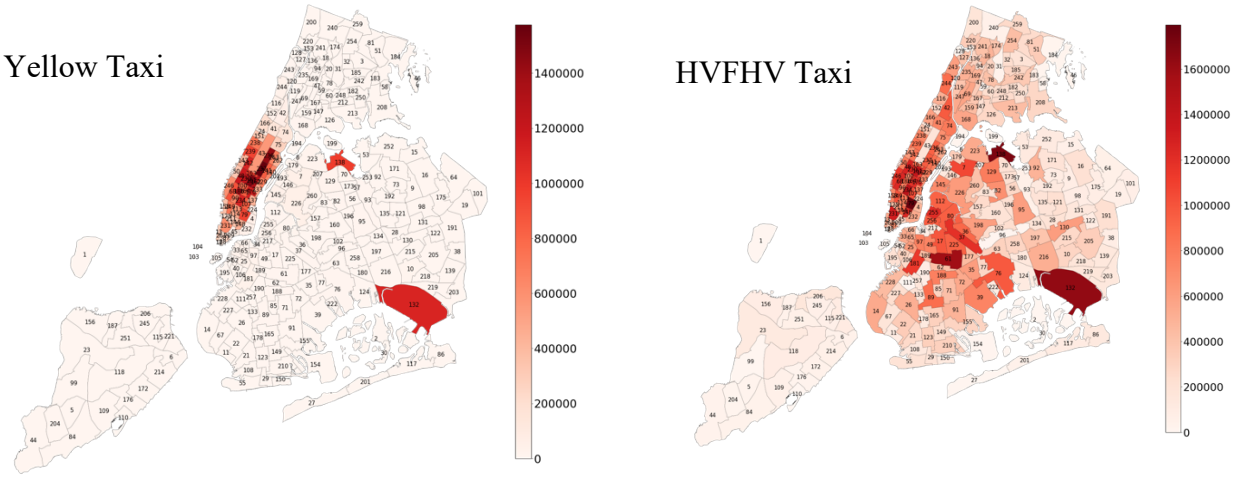


Figure 5: Mapped amount of taxi pickups of yellow taxi and HVFHV data (drawn by author using [24]).

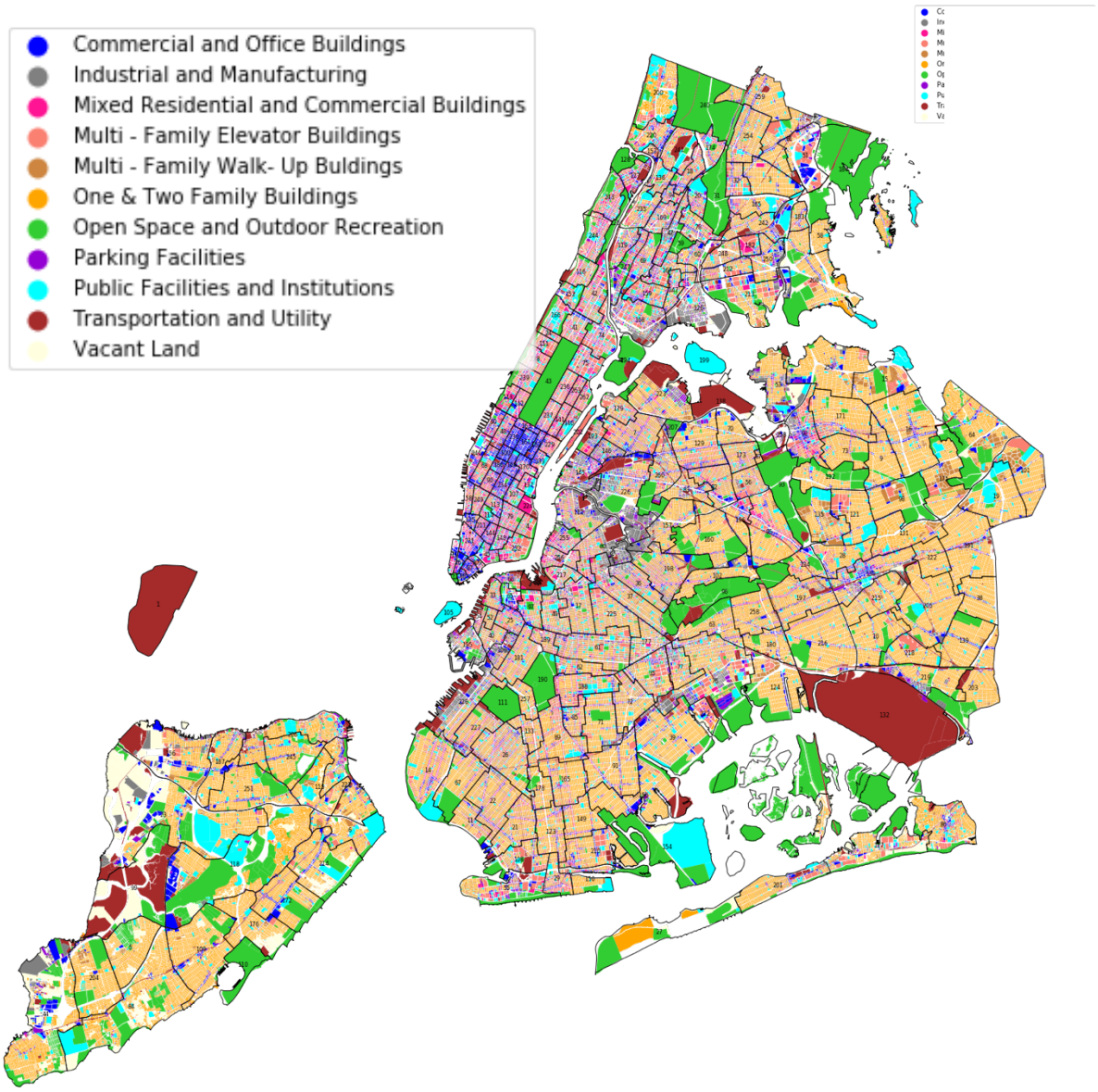


Figure 6: Map of 11 land use categories area (drawn by author using [25]).

Table 1: The value description of LUs category in PLUTO data.

Value	Description
01	One & Two Family Buildings
02	Multi-Family Walk-Up Buildings
03	Multi-Family Elevator Buildings
04	Mixed Residential & Commercial Buildings
05	Commercial & Office Buildings
06	Industrial & Manufacturing
07	Transportation & Utility
08	Public Facilities & Institutions
09	Open Space & Outdoor Recreation
10	Parking Facilities
11	Vacant Land

3.3 Road Data

The NYC Street Centerline (CSCL) is a road-bed representation of New York City streets containing address ranges and other information such as traffic directions, road types, segment types, it is called LION file. We extract node attributes from roadway type as shown in Table 2 by counting the number of roads contained in every 263 taxi zones per each road type as shown in Figure 6. And we also count the number of connected road between each taxi zone per road type, which is used for edge attributes. There are 14 roadway types but 10 of 14 type contain in road network data as shown in Table 2. In this study, we conduct prediction using edge-meta-knowledge learning. Almost ST deep-learning method can consider the connectivity of nodes, but it cannot consider edge-meta-knowledge such as roadway type. On the other hand, ST-MetaNet can consider not only the neighborhoods of node importance, but also the edge type between neighborhoods. It should be advantage, for example, as shown in Figure 6, in the case of highway, highway network connects major area for efficient transportation. Thus, it can be assumed that the place connected by highway would have high correlation even if the places connected by highway is not close. In this way, ST-Meta-Net is empowered to grasp higher ST-correlation than any others.

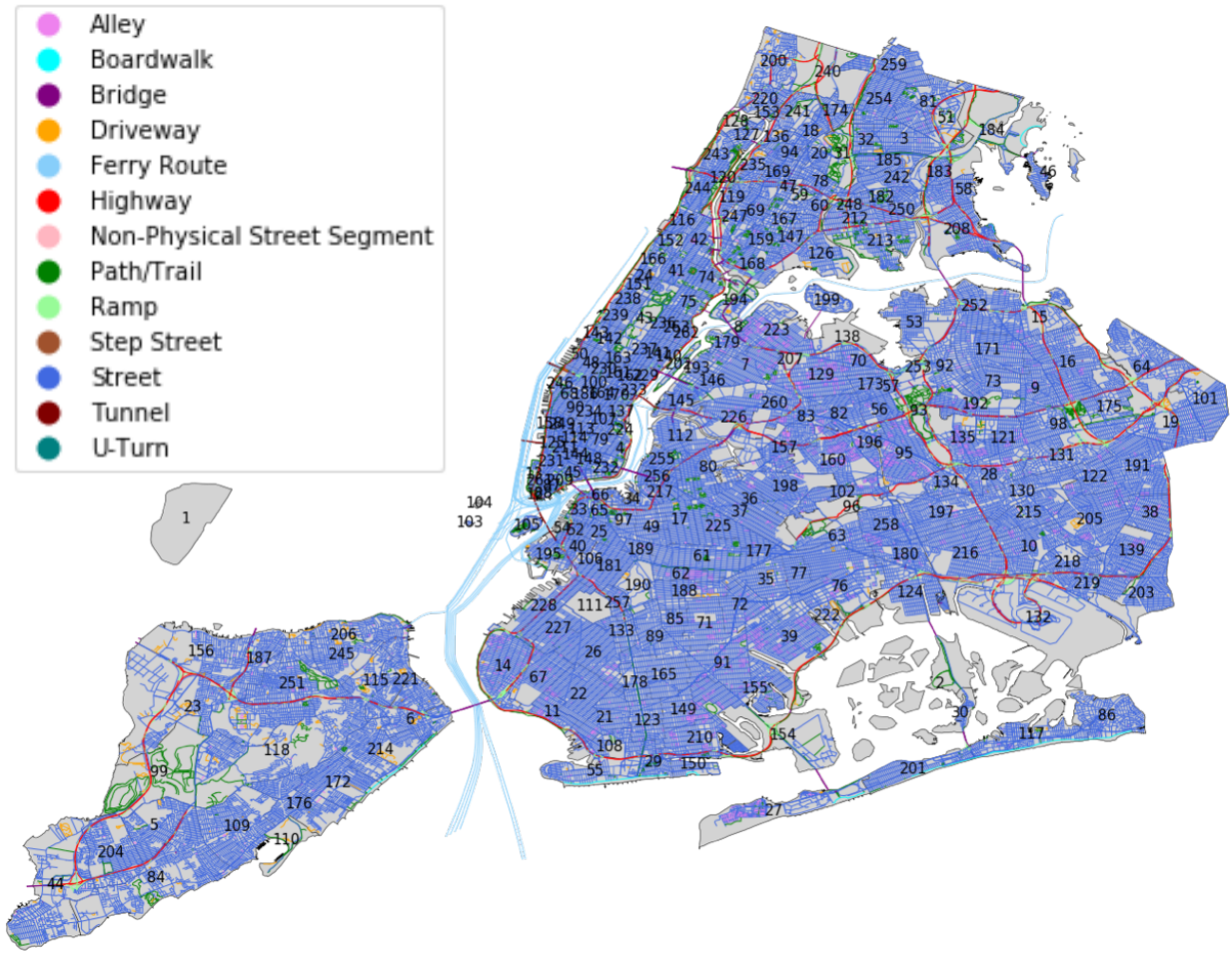


Figure 7: NYC road network (drawn by author using [22]).

Table 2: The value description of roadway type in LION files.

Value	Description
1	Street
2	Highway
3	Bridge
4	Tunnel
5	Boardwalk
6	Path/Trail
7	Step Street
8	Driveway
9	Ramp
10	Alley
13	U-Turn

3.4 Event Data

Office of Citywide Event Coordination and Management (CECM) provides past and future High-impact events information held in NYC from 2008. This list contains information on approved event applications that will occur within the next month. Data processing is conducted by adding the coordinate of event place using geo-scrapping from address because of lack of coordinate and by weighting the effect of event each taxi zone using simple way. Firstly, we extract the events whose open time within 6 hours. Then in order to take into account of the effect of events, adding the number to timestamp sequence filled with zero as follow: +1 before 20 minutes, +2 before 10 minutes, +3 on the end time, +4 after 10 minutes, +3 after 20 minute, +2 after 30 minutes and +1 after 40 minutes from end time of event. That is because people are most likely to use a taxi well after the start of the event. To understand easily, please show Figure 7 as example how we weight the number to timestamps for event effect.

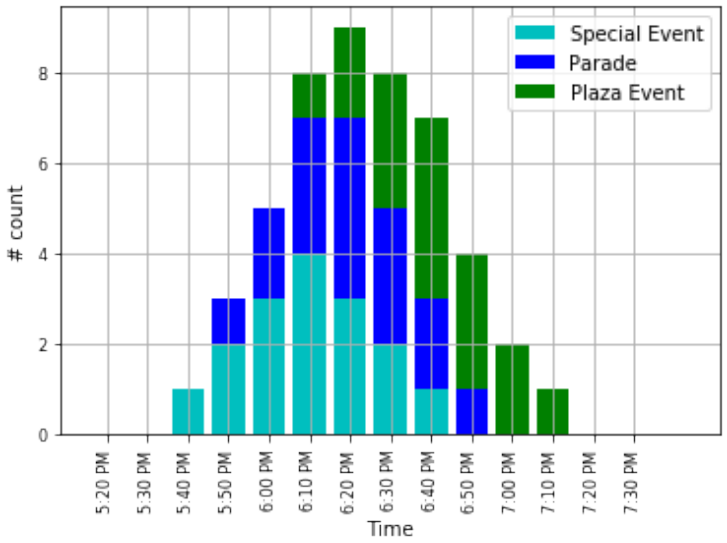


Figure 8: The example how to weight the effect of event.

Table 3 show the summary of event type and the number of events held from 1st Feb to 30th Jun in 2019. In order to understand the frequency of event per taxi zone, Figure 8 shows color map of event occurrence count average per month from 1st Feb to 30th June in taxi zones. As shown in Figure 8, it can be seen that there are places where the event is held frequently and places where it is not. While it is difficult to predict sudden demand in areas where events are infrequent, it is easier to predict areas where frequent events occur, such as the dark areas such as # 43 at Central Park in Figure 8. You can see that the event is held about 10 times a day on average. Figure 9 show the weighted number of Special Event occurrence from 1st Feb to 30th June at Central Park (#43). Figure 9 indicates that the periodicity of the event varies depending

on middle term such as a month duration. Therefore, it is likely that seasonal trends need to be factored in to predict these events.

Table 3: The summary of event type and the number of events.

Index	Event type	# events held
1	Special Event	7374
2	Parade	387
3	Street Event	163
4	Block Party	110
5	Construction	79
6	Production Event	61
7	Plaza Event	42
8	Religious Event	34
9	Plaza Partner Event	30
10	Athletic Race / Tour	30
11	Health Fair	18
12	Weekend Walk	18
13	Miscellaneous	17
14	Clean-Up	15
15	Single Block Festival	10
16	Press Conference	5
17	Rally	4
18	Street Festival	4
19	Stationary Demonstration	2
20	Athletic	1

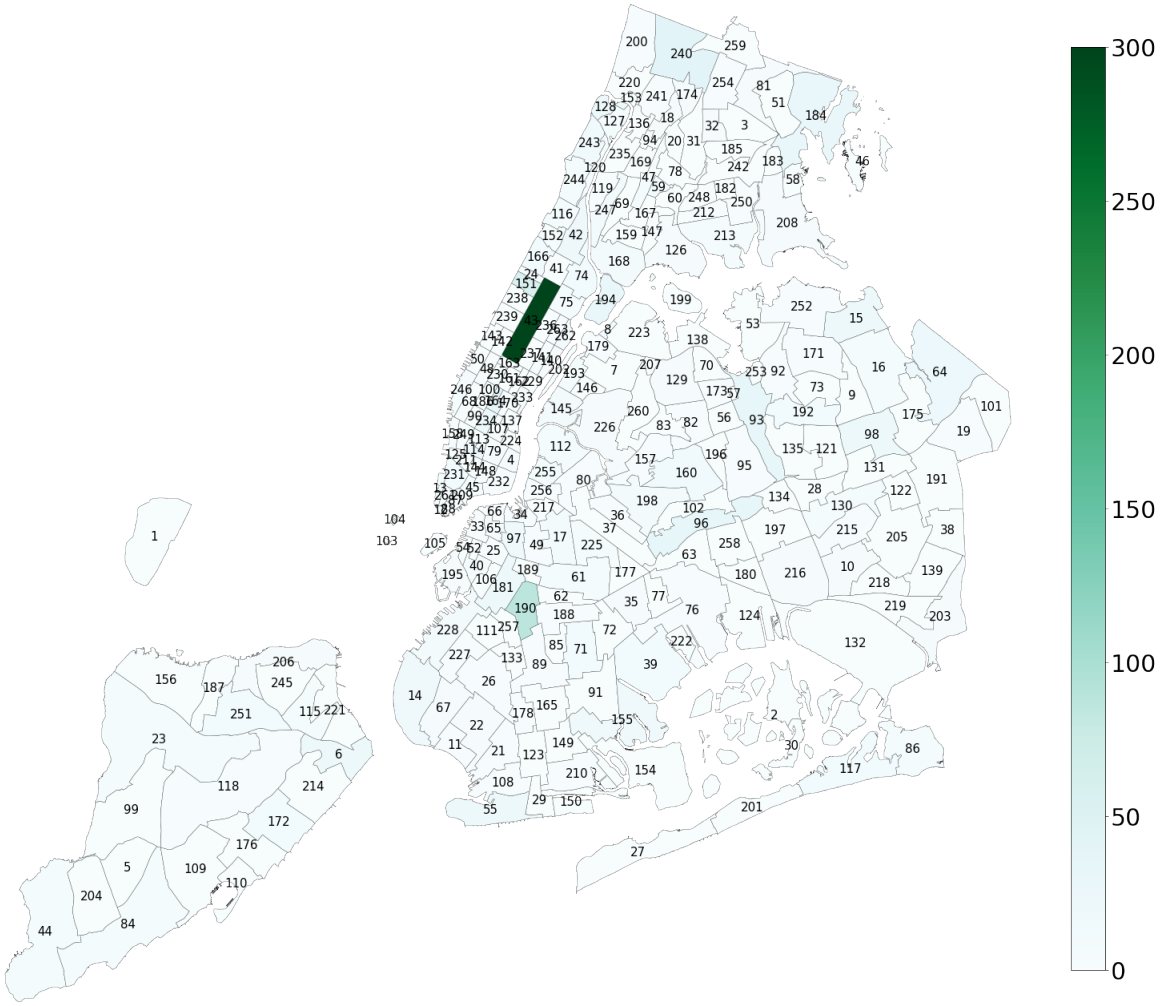


Figure 9: The color map of all event occurrence count average per month (drawn by author using [22]).

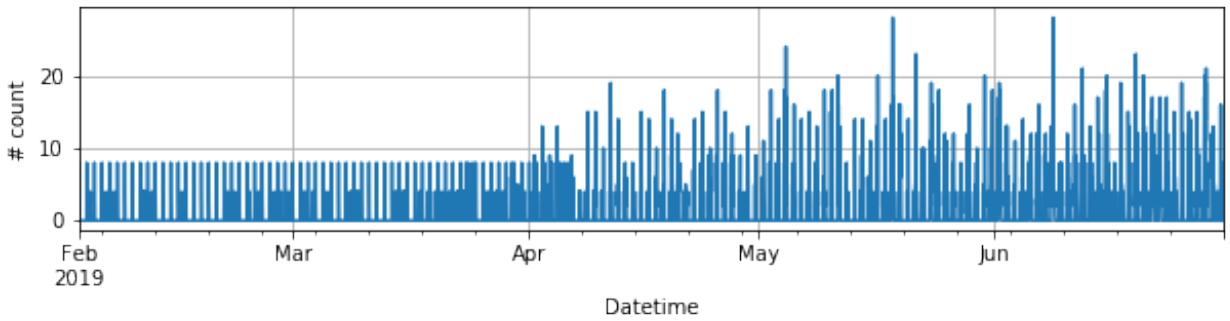


Figure 10: The weighted number of Special Event occurrence.

3.5 Weather Data

We collect historical hourly weather data at EWR airport located near to central NYC. Because time interval of taxi demand is 10 minutes in spite of hourly weather data observation, we interpolate it linearly. We use precipitation amount, temperature and wind speed as dynamic spatio-temporal features for all of taxi zone. The feature name and of weather data is summarized in Table 4. Figure 10 show the plotted value of all features in weather data from st Feb to 30th June at EWR airport.

Table 4: The summary of weather feature

Name	Description
FeelsLikeC	The "feels like" temperature is a measurement of how hot or cold it really feels like outside. The “Feels Like” temperature relies on environmental data including the ambient air temperature, relative humidity, and wind speed to determine how weather conditions feel to bare skin. (unit: ℃)
windspeedKmph	Wind speed (unit: Km/ph)
precipMM	Precipation (unit: mm)

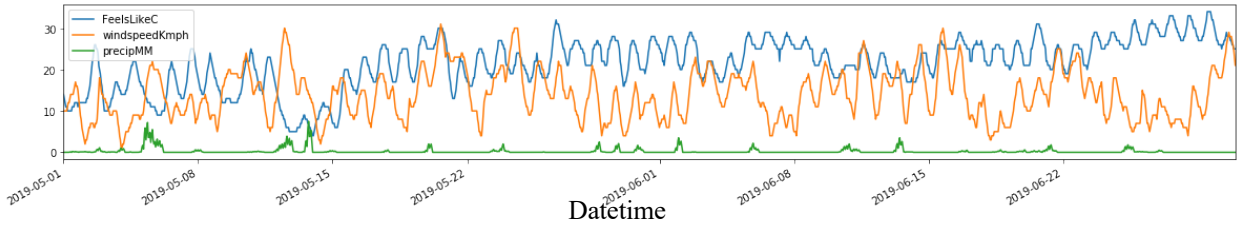


Figure 11: Plot of features of weather data at EWR airport in NYC.

3.6 Others

We add geometric features such as latitude, longitude, shape length, area. And finally add the number of stations belong to each taxi zone.

Table 5: The summary of Others features

Index	Name	Description
1	Latitude	Latitude of the barycentric coordinates of the node.
2	Longitude	Longitude of the barycentric coordinates of the node.
3	Shape length	Shape length of the barycentric coordinates of the node.
4	Area	Area of the barycentric coordinates of the node.
5	Station count	Number of stations belong to each taxi zone

CHAPTER 4: METHODOLOGIES

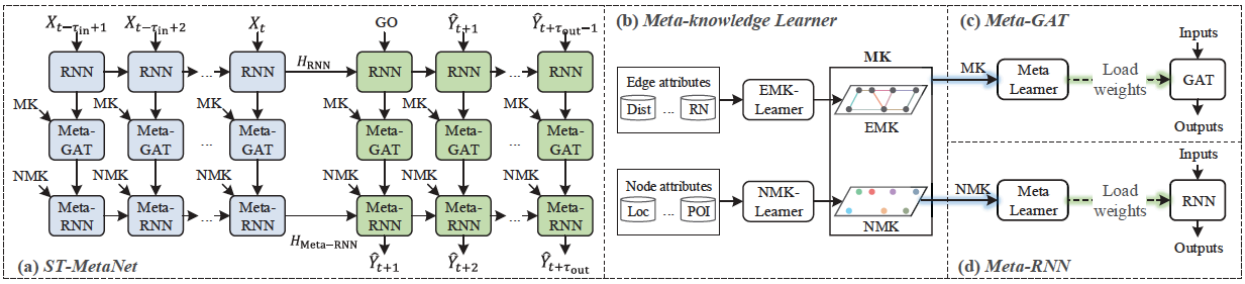


Figure 12: Overview of ST-MetaNet cited from [1].

In this chapter, we describe the architecture of ST-MetaNet for traffic prediction, as shown in Figure 11 (a). Following the sequence-to-sequence (Seq2Seq) architecture [20], ST-MetaNet is composed of two separate modules: the encoder and the decoder .The former one is used to encode the sequence of input, *i.e.*, historical information of taxi demand $\{X_{t-\tau_{in}+1}, \ldots, X_t\}$, producing the hidden states $\{H_{RNN}, H_{Meta-RNN}\}$, which are used as the initial states of the decoder that further predicts the output sequence $\{\hat{Y}_{t+1}, \ldots, \hat{Y}_{t+\tau_{out}}\}$.

More specifically, the encoder and the decoder have the same network structures, consisting of four components:

1) ***RNN (recurrent neural network).*** We employ RNN to embed the sequence of not only historical taxi demand but also weather condition and event occurrence to capture dynamic spatio-temporal feature, capable of learning long range temporal dependencies.
2) ***Meta-knowledge learner.*** A shown in Figure 11 (b), we use two fully connected networks (FCNs), named node-meta-knowledge learner (NMK-Learner) and edge-meta-knowledge learner (EMK- Learner) , to respectively learn the meta-knowledge of nodes (NMK) and edges (EMK) from node attributes (*e.g.*, Land use and Road density) and edge attributes (*e.g.*, road connectivity and road type). Then the learned meta knowledge is further used to learn the weights of another two networks, *i.e.*, graph attention network (GAT) [21] and RNN. Taking a certain node as an example, the attributes of the node are fed into the NMK-Learner, and it outputs a vector, representing the meta knowledge of that node.
3) ***Meta-GAT (meta graph attention network)*** is comprised of a meta learner and a GAT, as shown in Figure 11 (c). Author propose employing a FCN as the meta learner, whose inputs are meta knowledge of all nodes and edges, and outputs are the weights of GAT.Meta-GAT can capture diverse spatial correlations by individually broadcasting locations' hidden states along edges.
4) ***Meta-RNN (meta recurrent neural network)*** is comprised of a meta learner and a RNN. The meta learner here is a typical FCN whose inputs are meta knowledge of all nodes and outputs

are the weights of RNN for each location. Meta-RNN can capture diverse temporal correlations associated with the geographical information of locations.

4.1 Preliminaries

In this chapter, we introduce the definitions and the problem statement a. For brevity, we present a table of notations in Table 6nd detail of metholorogies.

Suppose there are N_l, locations, which report D_t, types of traffic information on N_t, timestamps respectively.

Table 6: Notations.

Notations	Description
$N_l,\ N_t$	Number of locations/timestamps.
$\tau_{in},\ \tau_{out}$	The number of timestamps for historical/future traffic.
$\chi = \{X_t\}$	The traffic readings at all timestamps.
$\mathcal{V} = \{v^{(ij)}\}$	The node attributes of location i.
$\mathcal{E} = \{e^{(ij)}\}$	The edge attributes between location i.
$\mathcal{N}_i$	Neighborhoods of location i.
$\mathrm{NMK}(\cdot)$	The function to learn meta knowledge of node.
$\mathrm{EMK}(\cdot)$	The function to learn meta knowledge of edge
$g_\theta(\cdot)$	The function to generate weights of parameter θ.

DEFFINITION 1. ***Urban traffic.*** *The urban traffic is denoted as a tensor* $\chi = \left(X_1, \dots, X_{N_t}\right) \in \mathbb{R}^{N_t \times N_l \times D_t}$, where $X_t = \left(x_t^{(1)}, \dots, x_t^{(N_l)}\right)$ *denotes all locations' traffic information at timestamp t.*

DEFFINITION 2. *Geo-graph attribute. Geo-graph attributes rep- resent locations' surrounding environment and their mutual relations, which corresponds to node attributes and edge attributes respectively. Formally, let* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ *represent a directed graph, where* $\mathcal{V} = \left\{v^{(1)}, \dots, v^{(N_t)}\right\}$ *and* $\mathcal{E} = \{e^{(ij)} | 1 \le i, j \le N_l\}$ *are lists of vectors denoting the geographical features of locations and relations between locations, respectively. Moreover, we use* $\mathcal{N}_i = \{j | e^{(ij)} \in \mathcal{E}\}$ *to denotes the neighbors of node i.*

PROBLEM 1. Given previous τ_{in} trafficinformation $\{X_{t-\tau_{in}+1}, \dots, X_t\}$, and the geograph attributes $\mathcal{G}$, predict the traffic information for all locations in the next τ_{out} timestamps, denoted as$\{\hat{Y}_{t+1}, \dots, \hat{Y}_{t+\tau_{out}}\}$.

4.2 Recurrent Neural Network

To encode the temporal dynamics of urban traffic, we employ a RNN layer as the first layer of Seq2Seq architecture. There are various types of RNN implementation for time series analysis. Among them, as gated recurrent unit (GRU) [9] is a simple but effective structure, we introduce GRU as a concrete example to illustrate ST-MetaNet. Formally, a GRU is defined as:

$$h_t = \mathrm{GRU}(Z_t, h_{t-1} | W_\Omega, U_\Omega, b_\Omega),$$

where $z_t \in \mathbb{R}^D$ and $h_t \in \mathbb{R}^{D'}$ are the input vector and the encoding state at timestamp *t*, respectively. $W_\Omega \in \mathbb{R}^{D' \times D}$ and $U_\Omega \in \mathbb{R}^{D' \times D'}$ are weight matrices. $b_\Omega \in \mathbb{R}$ are biases ($\Omega \in \{u, r, h\}$). GRU derives the vector representations of a hidden state, which is expressed as:

$$u = \sigma(W_\mathrm{u} z_t + U_u h_{t-1} + b_u)$$

$$r = \sigma(W_\mathrm{r} z_t + U_r h_{t-1} + b_r)$$

$$h_t = u \circ h_{t-1} + (1 - u) \circ \phi(W_t z_t + U_t(r \circ h_{t-1}) + b_h),$$

where $\circ$ is the element-wise multiplication, $\sigma(\cdot)$ is sigmoid function, and $\phi(\cdot)$ is tanh function. In urban traffic prediction, we collectively encode the traffic of all nodes (locations). To make nodes' encoding states in the same latent embedding space such that we can quantify the relations between them, RNN networks for all nodes share the same parameters. Suppose the input is $Z_t = \left(z_t^{(1)}, \ldots, z_t^{(N_t)}\right)$ where $z_t^{(i)}$ is the input of node *i* at timestamp *t* , we obtain hidden states for all nodes, denoted as, by the following formula:

$$H_t = \left(h_t^{(1)}, \ldots, h_t^{(N_l)}\right)$$

$$h_t^{(i)} = \mathrm{GRU}\left(z_t^{(i)}, h_{t-1}^{(i)} \middle| W_\Omega, U_\Omega, b_\Omega\right), \forall i \in \{1, \ldots, N_l\},$$

More specifically, in ST-MetaNet the inputs of RNNs in the encoder and decoder are $\{X_{t-\tau_{in}+1}, \ldots, X_t\}$,and $\{\hat{Y}_{t+1}, \ldots, \hat{Y}_{t+\tau_{out}}\}$, respectively, as shown in Figure 11 (a).

4.3 Meta-Knowledge Learner

In urban area, characteristics of locations and their mutual relation- ship are diverse, depending on geographical information, e.g., LUs and RNs. Such diverse characteristics bring about various types of ST correlations within urban traffic. Hence, we propose two meta-knowledge learners to learn traffic-related embeddings (meta knowledge) from geographical information, i.e., NMK-Learner and EMK-Learner. As shown in Figure 11 (b), two meta-knowledge learn- ers respectively employ different FCNs, in which input is the at- tribute of a node or an edge, and the corresponding

output is the embedding (vector representation) of that node or edge. Since such embeddings are used for generating weights of GAT and RNN to capture ST correlations of urban traffic, the learned embeddings can reflect traffic-related characteristics of nodes and edges. For simplicity, we use $\mathrm{NMK}(v^{(i)})$ and $\mathrm{EMK}(e^{(ij)})$ to denote the learned meta knowledge (embedding) of a node and an edge, respectively.

4.4 Meta Graph Attention

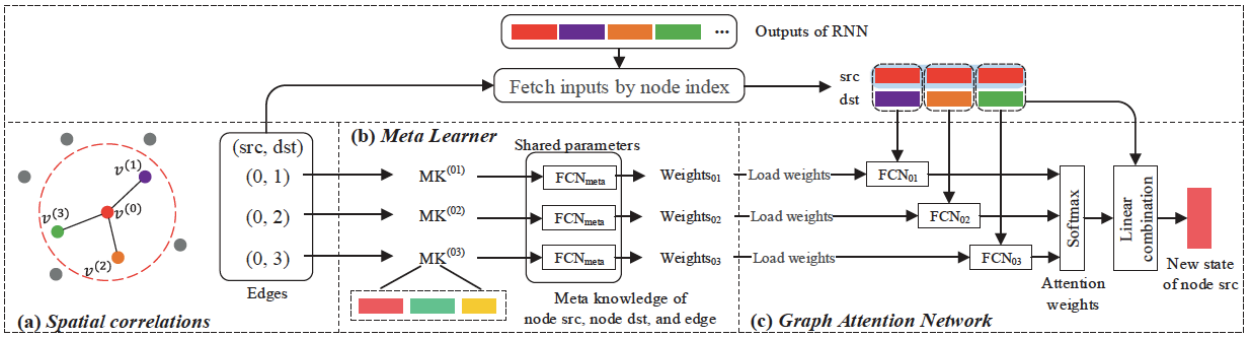


Figure 13: Structure of meta graph attention network cited from [1].

Urban traffic has spatial correlations that some locations are mutu- ally affected and such impact is highly dynamic. In addition, spatial correlations between two nodes are related to their geographical information, and such correlations are diverse from node to node and edge to edge. Inspired by graph attention network (GAT) [21], we propose employing attention mechanism into the framework to capture dynamic spatial correlations between nodes. However, it is inappropriate to directly apply GAT because all nodes and edges would use the same attention mechanism, ignoring the relationship between geographical information and spatial correlations.

To capture diverse spatial correlations, we propose a meta graph attention network (Meta-GAT) as shown in Figure 12, which employs attention network whose weights are generated from the meta knowledge (embeddings) by the meta learner. Consequently, the networks for spatial correlation modeling are different from node to node and edge to edge. Formally, suppose the inputs of meta graph attention network are $H = \left(h^{(1)}, \ldots, h^{(N_l)}\right) \in \mathbb{R}^{N_l \times D_h}$ (*i.e.*,outputs of RNN a teach timestamp) and geo-graph attributes $G = \{\mathcal{V}, \mathcal{E}\}$, while the output is $\bar{H} = \left(\bar{h}^{(1)}, \ldots, \bar{h}^{(N_l)}\right) \in \mathbb{R}^{N_l \times D_h}$,where D_h is the dimension of nodes' hidden states. The meta graph attention mechanism for each node mainly contains 2 steps: 1) attention score calculating for each edge; and 2) hidden state aggregation. As shown in Figure 12, we give an example to show Meta-GAT, that calculates

the impact to the red node from its neighborhoods (the purple, orange, and green node) along edges. The details of Meta-GAT are as follows:

- Attention score calculation. The score of edge *ij* is related to the hidden states of node *i* and node *j*, as well as the meta knowledge of nodes and edge learned from geographical information. As shown in Figure 12, for edge ij, the first step is fetching the hidden states of nodes by index, *i.e.*, $h^{(i)}$ and $h^{(j)}$ and meta knowledge $\mathrm{MK}^{(ij)}$, which is a composition of meta knowledge of nodes and edge:

$$\mathrm{MK}^{(ij)} = \mathrm{MK}(v^{(i)}) \parallel \mathrm{MK}(v^{(j)}) \parallel \mathrm{EMK}(e^{(ij)})$$

  where $\parallel$ is the concatenation operator. Then we apply a function to calculate the attention score, denoted as:

$$w^{(ij)} = a\left(h^{(i)}, h^{(j)}, \mathrm{MK}^{(ij)}\right) \in \mathbb{R}^{D_h}$$

  where $w^{(ij)}$ D_h dimension vector, denoting the importance of how $h^{(i)}$ impacts $h^{(i)}$ at each channel. Like GAT [21] shown in Figure 12 (c), we employ a single fully connected layer to calculate function $\mathrm{a}(\cdot)$. However, different pairs of nodes have different node and edge attributes, resulting in different patterns of edge attention given the nodes' states. To model such diversity, we employ an edge- wise fully connected layer, followed by an activation of LeakyRelu:

$$a\left(h^{(i)}, h^{(j)}, \mathrm{MK}^{(ij)}\right) = \mathrm{LeakyRelu}(\mathrm{W}^{(ij)}\left[h^{(i)} \parallel h^{(j)}\right] + b^{(ij)})$$

  where$\mathrm{W}^{(ij)} \in \mathbb{R}^{D_h \times 2D_h}$, $b^{(ij)} \in \mathbb{R}$ are edge-wise parameters of the fully connected layer. In particular, these parameters are generated by the meta learner from the meta knowledge as shown in Figure 12 (b). Formally, let g_w and g_b be FCNs within the meta learner to generate $\mathrm{W}^{(ij)}$ and $b^{(ij)}$ respectively, then for any edge (*i,j*):

$$\mathrm{W}^{(ij)} = g_w\left(\mathrm{MK}^{(ij)}\right) \in \mathbb{R}^{D_h \times 2D_h}$$

$$b^{(ij)} = g_b\left(\mathrm{MK}^{(ij)}\right) \in \mathbb{R}$$

  Note that the output of a FCN is a vector, so we need to reshape the output to the corresponding parameter shape.

- Hidden state aggregation. Like GAT, we firstly normalize the attention score for a node across all its neighborhoods by softmax:

$$a^{(ij)} = \frac{\exp\left(w^{(ij)}\right)}{\sum_{j \in N_i} \exp\left(w^{(ij)}\right)}$$

Then for each node, we calculate the overall impact of neighbohoods by linear combinations of the hidden states corresponding to the normalized weights and apply a nonlinearity function σ (e.g., ReLU), which is expressed as $\sigma\left(\sum_{j \in N_i} a^{(ij)} h^{(j)}\right)$. After that, let $\lambda^{(ij)} \in (0, 1)$ be a trainable parameter denoting the weights of neighborhoods' impact to node *i*. And finally, the hidden state for node *i* with consideration of spatial correlations is expressed as:

$$\bar{h}^{(i)} = \left(1 - \lambda^{(i)}\right) h^{(j)} + \lambda^{(i)} \sigma\left(\sum_{j \in N_i} a^{(ij)} h^{(j)}\right)$$

Since we extract meta knowledge from features of locations and edges, and use such information to generate the weights of graph attention network, Meta-GAT can model the inherent relationship between geo-graph attributes and diverse spatial correlations.

4.5 Meta Recurrent Neural Network

Since temporal correlations of urban traffic vary from location to location, a simple shared RNN (like Eq. 3) is not sufficient to capture diverse temporal correlations. To model such diversity, we adopt the similar idea of Meta-GAT to generate the weights of RNN from node embeddings, which is learned by NMK-Learner from node attributes (e.g., LUs and RNs).

Here we introduce Meta-GRU as a concrete example of Meta- RNN. It adopts the node-wise parameters within GRU. Formally, we define Meta-GRU as:

$H_t =$ Meta-GRU$(Z_t, H_{t-1}, \mathcal{V})$,

Where $Z_t = (z^{(1)}, \ldots, z^{(N_l)})$ and $H_t = (h^{(1)}, \ldots, h^{(N_l)})$ are the inputs and the hidden states at timestamp *t*, respectively, and $\mathcal{V} = (v^{(1)}, \ldots, v^{(N_l)})$ is the node attributes. Then for any node *i*, the current hidden states can be calculated by:

$$W_\Omega^{(i)} = \mathcal{g}_{W_\Omega}\left(\mathrm{NMK}(v^{(1)})\right)$$

$$U_\Omega^{(i)} = \mathcal{g}_{U_\Omega}\left(\mathrm{NMK}(v^{(1)})\right)$$

$$b_\Omega^{(i)} = \mathcal{g}_{b_\Omega}\left(\mathrm{NMK}(v^{(1)})\right)$$

$$h_t^{(i)} = \mathrm{GRU}\left(z_t^{(i)}, h_{t-1}^{(i)} \middle| W_\Omega, U_\Omega, b_\Omega\right),$$

where$W_\Omega^{(i)}$, $U_\Omega^{(i)}$ and $b_\Omega^{(i)}$arenode-wise parameters generated from $v^{(1)}$ by a meta learner which consists of three types of FCNs g_{W_Ω} , $g_{U_\Omega}, g_{b_\Omega}$ $(\Omega \in \{u, r, h\})$. As a result, all nodes have their individual RNNs respectively, and the models represent the temporal correlations related to the node attributes. In ST-MetaNet, we take the outputs of Meta-GAT as the inputs of Meta-RNN (i.e., Z_t), accordingly, both diverse spatial and temporal correlations are captured.

4.6 Algorithm of Optimization

Suppose the loss function for framework training is $\mathcal{L}_{train}$, which measures the difference between the prediction values and the ground truth. We can train ST-MetaNet end-to-end by backpropagation. In specific, there are two types of trainable parameters. Let ω_1 denote the trainable parameters in common neural networks (Chaper 5.1), ω_2 denote all trainable parameters in the meta-knowledge learners (Chaper 5.2), and meta learners (Chaper 5.3 and Chaper 5.4). For parameter ω_1 in common neural networks, the gradient of it is $\nabla\omega_1\mathcal{L}_{train}$ As for parameter ω_2 in meta-knowledge learner and meta learner which generates parameter θ , the gradient of ω_2 can be calculated by chain rule because both meta-knowledge learner and meta learner are differentiable neural networks:

$$\nabla\omega_1\mathcal{L}_{train} = \nabla\omega_1\mathcal{L}_{train}\nabla\omega_2\theta.$$

Algorithm 1 outlines the training process of ST-MetaNet. We first construct training data (Lines 1-5). Then we iteratively optimizer ST-MetaNet by gradient descent (Lines 7-12) until the stopping criteria is met. Algorithm 1 outlines the training process of ST-MetaNet. We first construct training data (Lines 1-5). Then we iteratively optimizer ST-MetaNet by gradient descent (Lines 7-12) until the stopping criteria is met.

Algorithm 1: Training algorithm of ST-MetaNet

input : Urban traffic $\mathcal{X} = (X_1, \ldots, X_{N_t})$.
Node attributes $\mathcal{V} = \{v^{(1)}, \ldots, v^{(N_l)}\}$.
Edge attributes $\mathcal{E} = \{e^{(ij)}\}$.

$\mathcal{D}_{\text{train}} \leftarrow \emptyset$
for *available* $t \in \{1, \ldots, N_t\}$ **do**
    $X_{\text{input}} \leftarrow (X_{t-\tau_{\text{in}}+1}, \ldots, X_t)$
    $Y_{\text{label}} \leftarrow (X_{t+1}, \ldots, X_{t+\tau_{\text{out}}})$
    put $\{X_{\text{input}}, \mathcal{V}, \mathcal{E}, Y_{\text{label}}\}$ into $\mathcal{D}_{\text{train}}$
initialize all trainable parameters
do
    randomly select a batch $\mathcal{D}_{\text{batch}}$ from $\mathcal{D}_{\text{train}}$
    forward-backward on $\mathcal{L}_{\text{train}}$ by $\mathcal{D}_{\text{batch}}$
    $\omega_1 = \omega_1 - \alpha \nabla_{\omega_1} \mathcal{L}_{\text{train}}$ // α is learning rate
    $\omega_2 = \omega_2 - \alpha \nabla_{\omega_2} \mathcal{L}_{\text{train}}$
until *stopping criteria is met*
output learned ST-MetaNet model

CHAPTER 5: EVALUATION

5.1 Experimental Settings

Task Description. We illustrate the details of experimental setting is shown in Table 7. In this task, we use previous 2-hour dynamic data to predict up to the next 30-minute taxi demand. We partition the data on the time axis into nonoverlapping training, evaluation, and test data, by the ratio of 8: 1:1. The details of the datasets in taxi demand prediction is shown in Table 8.

Table 7: Details of dataset. (Inside of () show # features included in each dataset.)

Task	Taxi demand prediction
Kinds of datasets	Yellow taxi, HVFHV
Prediction target	The number of pick up taxi
Timespan	5/1/2019 – 6/30/2019
Time interval	10 minutes
# timestamps	8784
Dataset split ratio (Train:Eval:Test)	8:1:1
Input length (ML model / DL model)	1 / 12 (Previous 2 hours)
Output length	1 / 3 (up to the next 30-minute)
Metrics	MAE, RMSE, R^2
# nodes	263 (# Taxi zones)
# edges	665 (# Connected roads)
# node attributes	26 (Road(10) + LUs(11) + Others(5))
# edge attributes	10 (Road(10))
# dynamic features	67 (Taxi(2) + Time(42) + Event(20) +Weather(3))

ML: Machine Learning, DL: Deep Learning

Scaling. All categorical numeric are scaled by dummy conversion and all quantitative numeric are scaled by z-score.

Metrics. We use mean absolute error (MAE), rooted mean square error (RMSE) and coefficient of determination (R^2) to evaluate prediction model.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Table 8: Details of the datasets in taxi demand prediction.

Datasets Name	Property	Value
Taxi	Attribute type	Dynamic
	# features	2
	Features Name	Pickup, Dropoff
Time	Attribute type	Dynamic
	# features	42
	Features Name	The categorical data of month (5 to 6), number of weekday (0 to 6), holiday or not (0 or 1), hour (0 to 23) and minute index (0, 10, 20, 30, 40, 50). And repeated number of the day period (0 to 143).
Land Use	Attribute type	Node
	# features	11
	Features Name	Shown in Table1
Road	Attribute type	Node and Edge
	# features	10
	Features Name	Shown in Table 2
Event	Attribute type	Dynamic
	# features	20
	Features Name	Shown in Table 3
Weather	Attribute type	Dynamic
	# features	3
	Features Name	Shown in Table 4
Others	Attribute type	Node
	# features	5
	Features Name	Shown in Table 5

Baselines.

- **HA (**Historical average**)**. We model the urban traffic as the seasonal process, where the period is one day. In our research, the period is 144 (6 periods per 24 hours) per one day. We use the average values of the train dataset.
- **Lasso / XGBoost (XGB) / Multiple Perceptron Layer (MLP).** These are basic regression models. For each future step (next 30 minutes), we train a single model, and predict the taxi demand, where the input consists of timeseries sequence of previous 2-hour taxi pickup and drop off demand, node attributes, dynamic features every 263 taxi zones. The code is implemented by scikit-learn which is a python library for machine learning. The parameter of Lasso, α is set as 1.0. The parameter of XGB is set as follow: {n_estimators 500, min_child_weight: 2, max_depth: 25, subsample: 0.3, learning_rate: 0.1, gamma: 0.15, num_boost_round: 1000, early_stopping_rounds: 5}. The parameter of MLP is set as follow: {activation: relu, solver: adam, early_stopping: True, validation_fraction: 877/(7015+877), alpha: 0.0001, hidden_layer_sizes: (256,128,64)}.
- Seq2Seq [20]. We implement a two-layer sequence-to-sequence model for urban traffic prediction, where GRU is chosen as the RNN implementation. It contains 2 layers of GRU, each of which has 128 hidden units. The features of nodes, i.e., node attributes and dynamic features are firstly embedded by a two-layer FCN with hidden units [32,32]. Then the outputs of FCN fuse with the outputs of the decoder in Seq2Seq. Finally, the fused vectors are linearly projected into two-dimensional vectors, as the predictions of pickup and drop off value.
- GAT-Seq2Seq [21]. We employ graph attention network and Seq2Seq to model spatial and temporal correlations, respectively. It applies the similar structure as ST-MetaNet, which consists of a GRU layer, a GAT layer, and a GRU layer. All these layers have 128 hidden units, respectively. Like Seq2Seq, we also embed the node attributes and dynamic features by a two-layer FCN with hidden units [32,32], and fuse the embeddings with the outputs of decoder. Finally, the fused vectors are linearly projected into two-dimensional vectors, as the predictions of pickup and drop off value.

Settings of ST-MetaNet. The structure of ST-MetaNet for taxi flow prediction is as follow:

- RNN. We adopt GRU as the implementation of RNN, in which the dimension of hidden state is 64.
- NMK-Learner. It is a two-layer FCN with hidden units [32, 32].
- EMK-Learner. It is a two-layer FCN with hidden units [32, 32].
- Meta-GAT. The meta learner to generate the weights of GAT is a FCN with hidden units [16, 2, n], where n is the dimension of the target parameters. The Meta-GAT outputs a 64 dimensional hidden states for each node.

- Meta-RNN. We adopt Meta-GRU as Meta-RNN implementation, in which the dimension of hidden state is 64. The meta learner to generate the weights of GRU is a FCN with hidden units [16, 2,n], where n is the dimension of the target parameters.

The Framework is trained by Adam optimizer with learning rate decay. The initial learning rate is 1e-2, and it is divided by 10 every 20 epochs. We also apply gradient clipping before updating the parameters, where the maximum norm of the gradient is set as 5. To tackle the discrepancy between training and inference in Seq2Seqarchitecture, we employ inverse sigmoid decay for schedule sampling in the training phase:

$$\epsilon_i = \frac{k}{k + \exp(i/k)}$$

where we set k = 2000. The batch size for taxi flow prediction is set as 16.

Environment. Our implementation is based on MXNet 1.5.1 2, and DGL 0.1.3 3, tested on Ubuntu 16.04 with a Gefrase GTX 1080. Machine spec is Intel (R) Core (TM) i7-7700K CPU @ 4.20GHz

5.2 Performance Results

The performance of the competing baselines and ST-MetaNet in the case of yellow taxi and HVFHV dataset are shown in Table 9 and Table 10. For deep models, we train and test each of them five times, and present results as the format: "mean ± standard deviation".

As a result, it is shown that the most accurate model for prediction up to 30 minutes ahead is ST-MetaNet using all types of data. On the other hand, ST-MetaNet using dynamic data of only taxi data has the highest accuracy in the case of 10 minutes ahead prediction. As shown in Table 9 and 10, ST- MetaNet outperforms all the baselines on both metrics. Specifically, ST-MetaNet shows about 9.0% and 9.4% improvements in the case of yellow taxi dataset and over 8.4% and 8.1% improvements in the case of HVFHV dataset on overall MAE and RMSE beyond other baselines respectively for prediction up to future 30 minutes. Compared with GAT-Seq2Seq, ST-MetaNet also achieves significant improvement, because it explicitly models the inherent relationship between geo-graphical information and ST correlations. Seq2Seq does not con-sider spatial correlations and diversity of temporal correlations, so it gets much larger error compared with other deep models. Conventive methods, *i.e.*, Lasso, XGB and MLP are also good model of urban traffic. However, due to the limitation of model expressiveness and the incapability to fully leverage geographical information, the error become large compare with ST-MetaNet.

Table 9: Performance results on yellow taxi demand prediction.

Models		HA	Lasso (All)	XGB (All)	MLP (All)	Seq2Seq (All)	GAT-Seq2Se2 (All)	ST-MetaNet (All)	ST-MetaNet (Taxi)
All	R^2	0.8515	0.9397	0.9482	0.9562±0.0003	0.9455±0.0000	0.9473±0.0000	**0.9616±0.0000**	0.9614±0.0000
	RMSE	6.441	4.105	3.804	3.499±0.013	3.603±0.000	3.558±0.000	3.186±0.000	**3.181±0.000**
	MSE	1.960	1.693	1.334	1.236±0.015	1.213±0.000	1.200±0.000	1.087±0.000	**1.081±0.000**
Next 10 min	R^2	0.8518	0.9512	0.9558	0.9624±0.0004	0.9590±0.0000	0.9604±0.000	0.9697±0.0000	**0.9704±0.0000**
	RMSE	6.438	3.695	3.516	3.243±0.019	3.168±0.000	3.129±0.000	2.845±0.000	**2.806±0.000**
	MSE	1.959	1.553	1.261	1.170±0.011	1.085±0.000	1.076±0.000	0.983±0.000	**0.963±0.000**
Next 20 min	R^2	0.8515	0.9399	0.9498	0.9572±0.0009	0.9456±0.0000	0.9475±0.0000	**0.9611±0.0000**	0.9608±0.0000
	RMSE	6.441	4.096	3.746	3.456 ±0.035	3.595±0.000	3.551±0.000	**3.203±0.000**	3.204±0.000
	MSE	1.960	1.693	1.316	1.236±0.005	1.219±0.000	1.205±0.000	**1.100±0.000**	1.101±0.000
Next 30 min	R^2	0.8512	0.9279	0.9445	0.9529±0.013	0.9312±0.0000	0.9332±0.0000	**0.9540±0.0000**	0.9528±0.0000
	RMSE	6.444	4.486	3.936	3.626±0.048	3.997±0.000	3.946±0.000	**3.478±0.000**	3.500±0.000
	MSE	1.961	1.832	1.366	1.287±0.022	1.336±0.000	1.319±0.000	**1.179±0.000**	1.182±0.000
# features		1	378	378	378	93	93	93	2
# params		-	-	-	138,619	351,682	419,780	261,248	236,672

Inside of () show kind of dynamic feature used in dataset. The best for each metric is indicated by bold face.

Table 10: Performance results on HVFHV taxi demand prediction.

Models		HA	Lasso (All)	XGB (All)	MLP (All)	Seq2Seq (All)	GAT-Seq2Se2 (All)	ST-MetaNet (All)	ST-MetaNet (Taxi)
All	R^2	0.7942	0.9156	0.9278	0.9340±0.0014	0.9206±0.0000	0.9186±0.0000	**0.9375±0.0000**	0.9354±0.0000
	RMSE	10.029	6.422	5.941	5.682±0.060	5.616±0.000	5.653±0.000	**5.144±0.000**	5.178±0.000
	MSE	5.432	4.151	5.731	3.577±0.040	3.546±0.000	3.558±0.000	**3.258±0.000**	3.281±0.000
Next 10 min	R^2	0.7944	0.9314	0.9353	0.9412±0.0017	0.9353±0.0000	0.9338±0.000	**0.9484±0.0000**	0.9479±0.0000
	RMSE	10.027	5.791	5.623	5.363±0.076	5.112±0.000	5.145±0.000	4.679±0.000	**4.674±0.000**
	MSE	5.432	3.848	3.624	3.460±0.044	3.272±0.000	3.287±0.000	2.990±0.000	**2.981±0.000**
Next 20 min	R^2	0.7943	0.9156	0.9270	0.9312±0.0013	0.9218±0.0000	0.9205±0.0000	**0.9373±0.0000**	0.9351±0.0000
	RMSE	10.029	6.423	5.972	5.800 ±0.054	5.576±0.000	5.609±0.000	**5.147±0.000**	5.186±0.000
	MSE	5.433	4.156	3.741	3.606±0.033	3.538±0.000	3.548±0.000	**3.278±0.000**	3.306±0.000
Next 30 min	R^2	0.7940	0.8999	0.9210	0.9198±0.045	0.9042±0.0000	0.9007±0.0000	**0.9265±0.0000**	0.9229±0.0000
	RMSE	10.032	6.995	6.213	3.257±0.175	6.114±0.000	6.158±0.000	**5.569±0.000**	5.630±0.000
	MSE	5.433	4.451	3.826	3.779±0.077	3.828±0.000	3.838±0.000	**3.506±0.000**	3.554±0.000
# features		1	378	378	378	93	93	93	38
# params		-	-	-	138,619	351,682	419,780	261,248	2366,72

Inside of () show kind of dynamic feature used in dataset. The best for each metric is indicated by bold face

5.3 Consideration

This section discusses the results of the 5.2 evaluation. First, in order to extract feature values with high importance from many feature values, the top 50 feature values were plotted using gain, which is an evaluation index of the degree of improvement of feature values in a tree-based model. Figure from 13 to 18 show plot of top 50 feature importance by gain in XGB model for 10, 20 and 30 minutes ahead prediction using Yellow Taxi data and HVFHV. From these 6 figures, you can see the following three things mainly. The first is that the value of pickup, which is the closest to the prediction, is significantly most important. Second, following the past value of the pick-up value, the importance of the feature value of the event is high. In other words, this figure shows that more accurate forecasts can be calculated by adding the event information of future events to the features in advance. And finally, the third shows that geographic features such as nodes and edge features do not exist at the top. As described earlier, this is probably because the machine learning model represented by XGB cannot take geographic correlation into account.

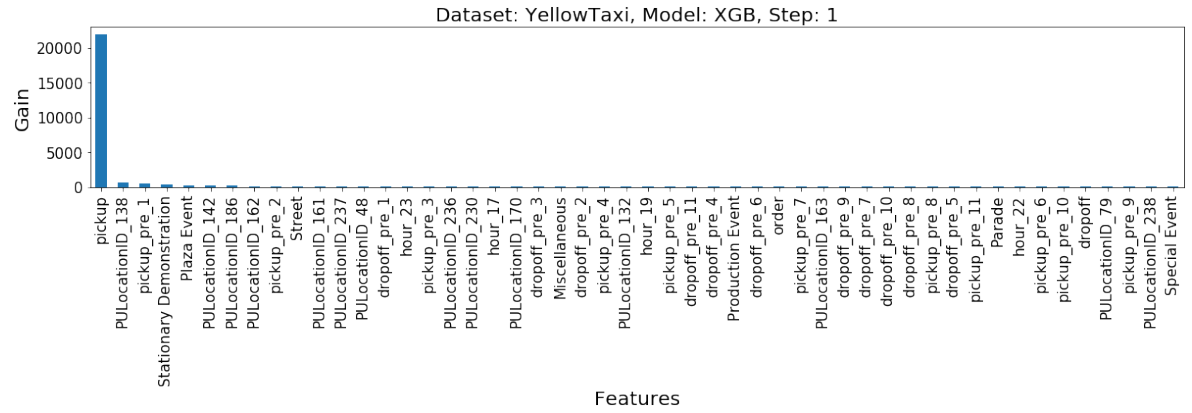


Figure 14: Figure 12: Plot of top 50 feature importance by gain in XGB model for 10 minutes ahead prediction using Yellow Taxi data.

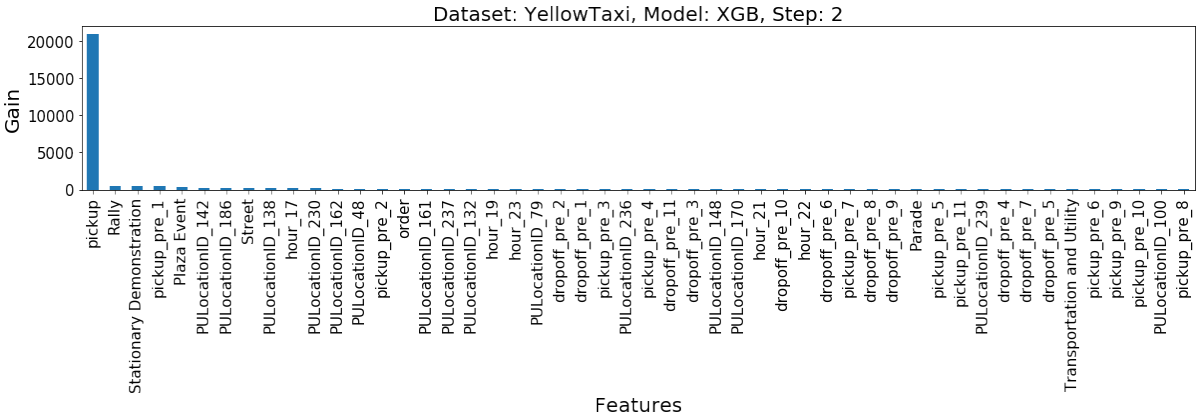


Figure 15: Plot of top 50 feature importance by gain in XGB model for 20 minutes ahead prediction using Yellow Taxi data.

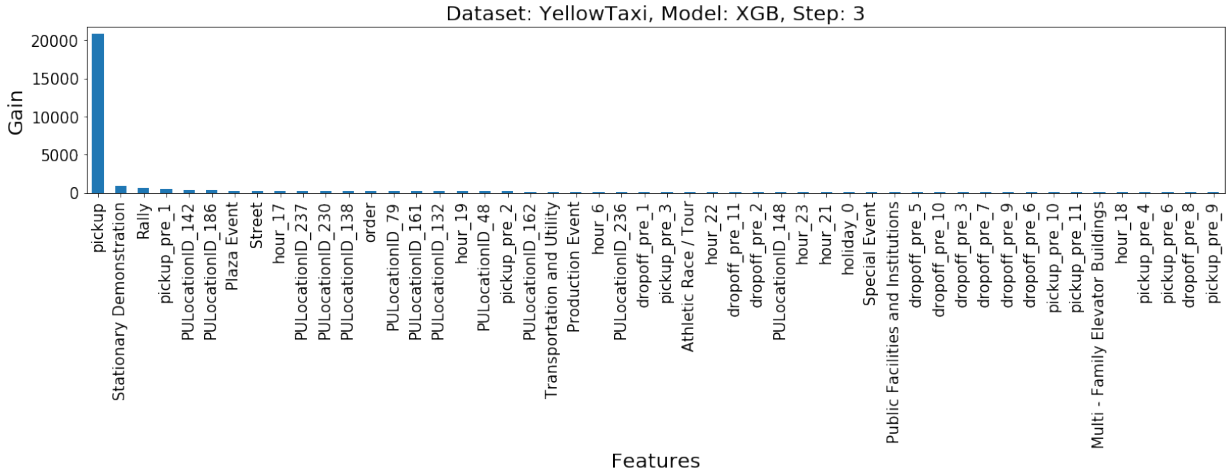


Figure 16:Figure 14: Plot of top 50 feature importance by gain in XGB model for 30 minutes ahead prediction using Yellow Taxi data.

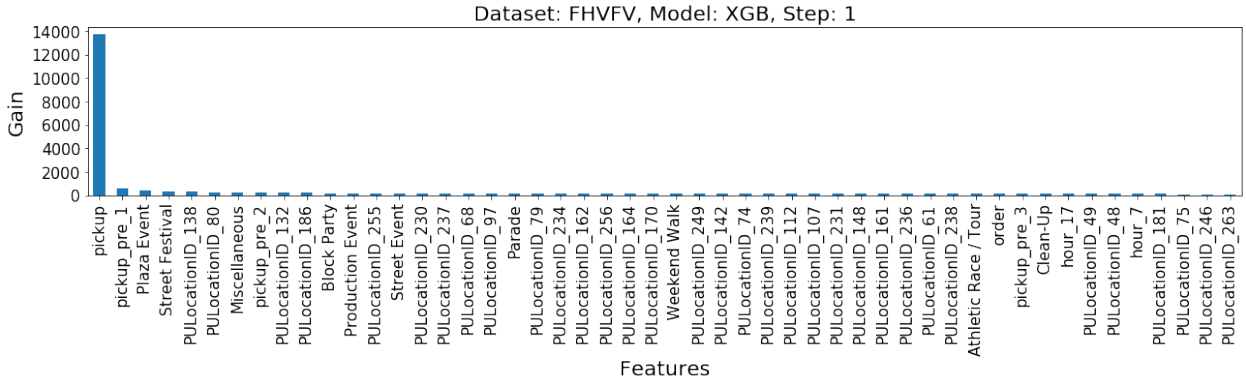


Figure 17: Plot of top 50 feature importance by gain in XGB model for 10 minutes ahead prediction using FHVFV data.

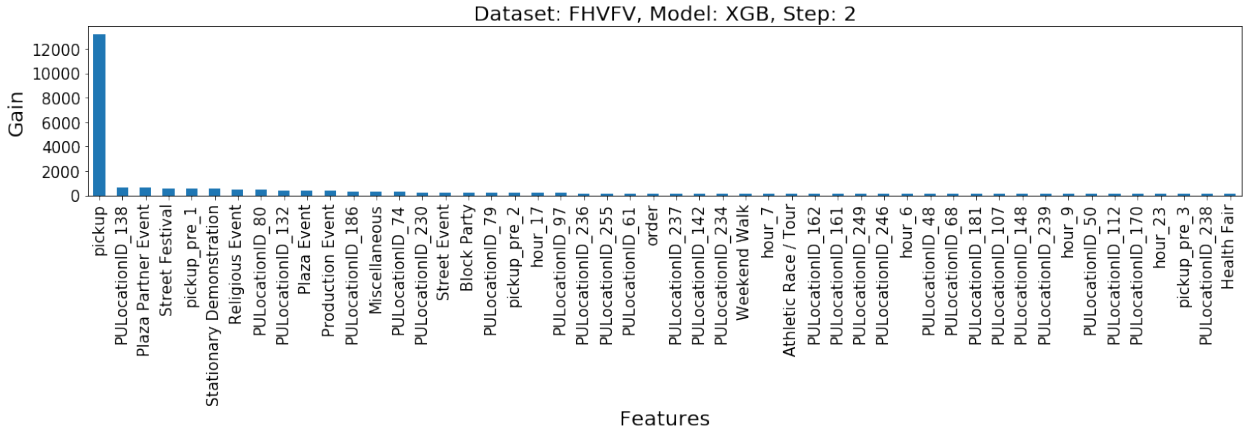


Figure 18: Plot of top 50 feature importance by gain in XGB model for 20 minutes ahead prediction using FHVFV data.

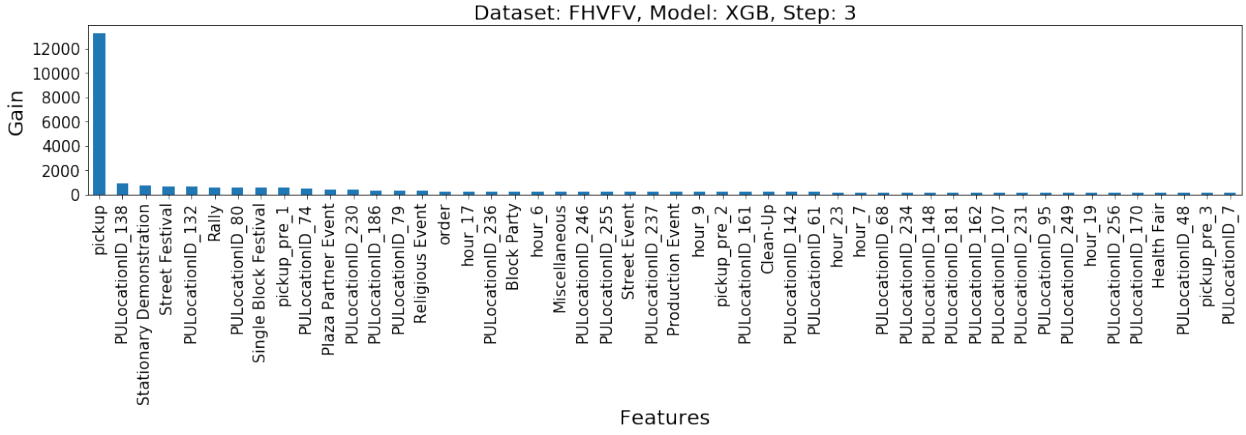


Figure 19: Plot of top 50 feature importance by gain in XGB model for 30 minutes ahead prediction using FHVFV data.

Figure 19, 20 and 21 show observation value, predicted value by MLP and ST-MetaNet (All) at the node number of 79, 138 and 132 for 10 minutes ahead prediction. The place whose node number 79 is center of manhattan near Wall street. The place whose node number 138 is LGA airport. The place whose node number 138 is JFK airport. These places are highest taxi demand places in all taxi zones. From these figures, it can be seen that the predicted value by ST-MetaNet fits the observed value better than the predicted value of MLP, which is the most accurate model among the baselines. This is probably because ST-MetaNet captures the correlation between various complex spatial features and temporal features.

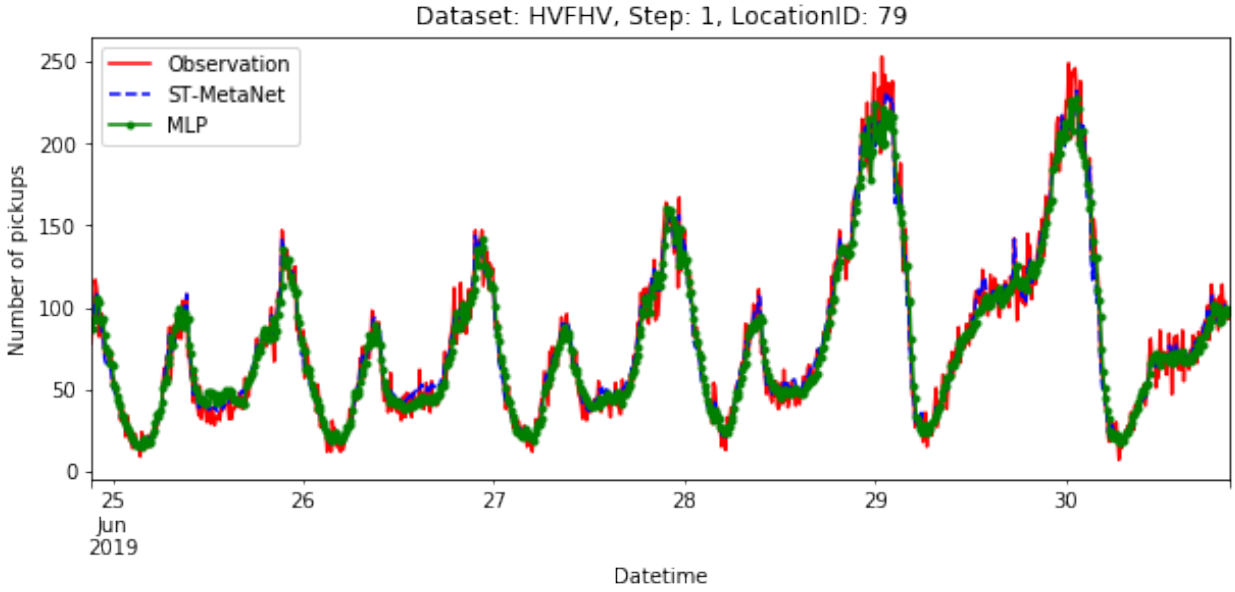


Figure 20: Observation value and predicted value by MLP and ST-MetaNet (All) at place where node number is 79 (near to Wall St.) for 10 minutes ahead prediction.

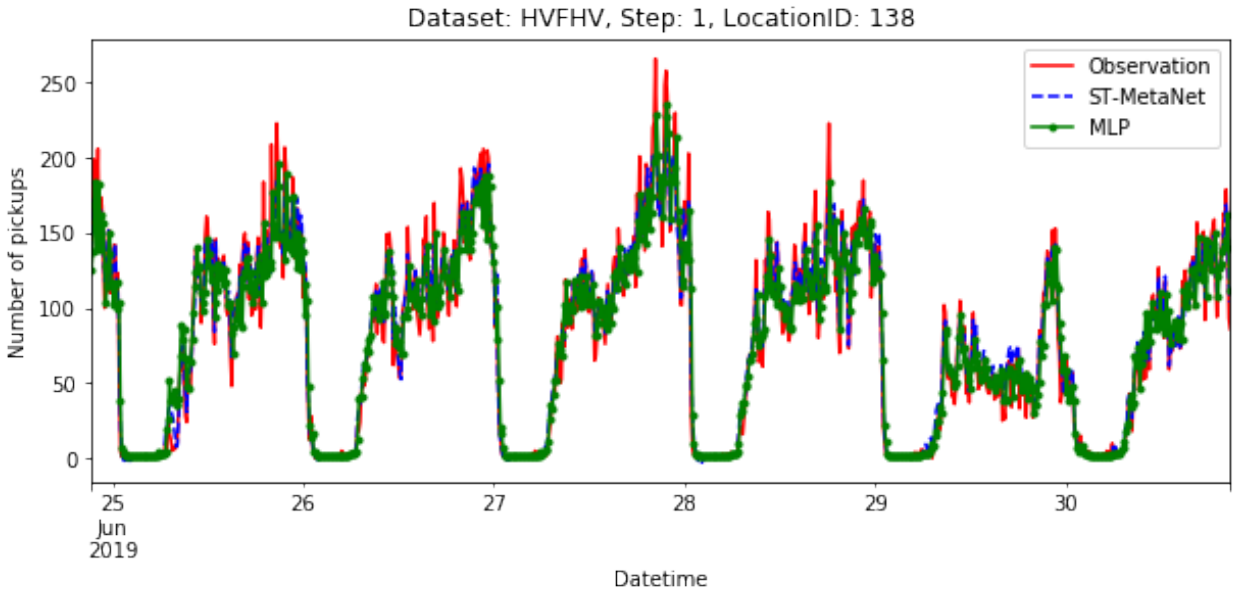


Figure 21: Observation value and predicted value by MLP and ST-MetaNet (All) at place where node number is 138 (LGA airport) for 10 minutes ahead prediction.

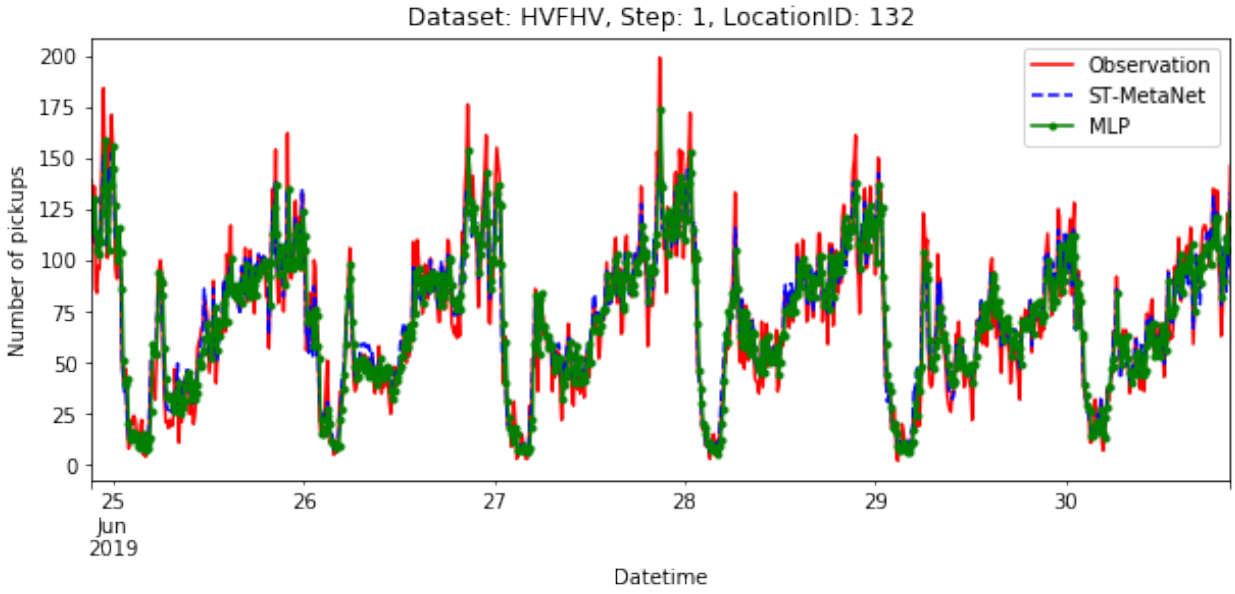


Figure 22: Observation value and predicted value by MLP and ST-MetaNet (All) at place where node number is 138 (JFK airport) for 10 minutes ahead prediction.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

In this study, spatio-temporal prediction of taxi demand is conducted using geographical attributes and context data, capable of learning traffic-related embeddings of nodes and edges from geo-graph attributes modeling both spatial and temporal correlations. And we also add contextual data such as time time series information, weather and event data to capture long-term periodicity and short-term rapid change. We evaluated our ST-MetaNet on two real-world tasks, achieving performance which significantly outper-forms 6 baselines. And it is found that the prediction models capture event features well and less weather features for precise prediction. In the future, we will extend our framework to a much broader set of urban ST prediction tasks.

The main remain work is parameter turning and of ST-MetaNet. Because ST-Meta has many components, RNN, EMK-Learner, NMK-Learner, Meta-GAT and Meta-RNN, so it is necessary to tune the proper hidden layer size and unit size of each layer. As shown in MAE loss curve attached in appendix section, it can be seen that the learning curve is not smooth. The cause of inefficient learning would be algorithm of optimization written in Section4 6. Loss function in backpropagation may have fallen into a local solution because of non-fit parameters such like learning rate.

REFERENCES

[1] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 1, pp. 1720–1730, 2019, doi: 10.1145/3292500.3330884.

[2] F. Wu, H. Wang, and Z. Li, "Interpreting traffic dynamics using ubiquitous urban data," *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, 2016, doi: 10.1145/2996913.2996962.

[3] Y. Tong *et al.*, "The Simpler The Better," pp. 1653–1662, 2017, doi: 10.1145/3097983.3098018.

[4] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, "Latent space model for road networks to predict time-varying traffic," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 1525–1534, 2016, doi: 10.1145/2939672.2939860.

[5] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, pp. 1–4, 2016, doi: 10.1145/2996913.2997016.

[6] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," *Proc. 17th SIAM Int. Conf. Data Mining, SDM 2017*, pp. 777–785, 2017, doi: 10.1137/1.9781611974973.87.

[7] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 2588–2595, 2018.

[8] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 2151–2158, 2018.

[9] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 3428–3434, 2018, doi: 10.24963/ijcai.2018/476.

[10] H. Mei, A. Ma, S. Poslad, and T. O. Oshin, "Short-term traffic volume prediction for sustainable transportation in an urban area," *J. Comput. Civ. Eng.*, vol. 29, no. 2, 2015, doi: 10.1061/(ASCE)CP.1943-5487.0000316.

[11] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, 2004, doi: 10.1109/TITS.2004.837813.

[12] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, Dec. 2015, doi: 10.1016/j.neucom.2015.12.013.

[13] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–16, 2018.

[14] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," *34th Conf. Uncertain. Artif. Intell. 2018, UAI 2018*, vol. 1, pp. 339–349, 2018.

[15] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 667–675, 2016.

[16] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 523–531, 2016.

[17] J. Johnson, "Hypernetworks," *Hypernetworks Sci. Complex Syst.*, pp. 151–176, 2014, doi: 10.1142/9781860949739_0006.

[18] J. Chen, X. Qiu, P. Liu, and X. Huang, "Meta multi-task learning for sequence modeling," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 5070–5077, 2018.

[19] C. Zhang, M. Ren, and R. Urtasun, "Graph hypernetworks for neural architecture search," *7th Int. Conf. Learn. Represent. ICLR 2019*, no. 2018, pp. 1–17, 2019.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, 2014.

[21] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–12, 2018.

[22] NYC OpenData: https://opendata.cityofnewyork.us/ (accessed 28 January 2020)

[23] NYC weather data: http://www.frontierweather.com/historicaldatasubscribers_hourly.html (accessed 28 January 2020)

[24] TLC Trip Record Data: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page (Accessed 28th January 2020.)

[25] PLUTO data: https://www1.nyc.gov/site/planning/data-maps/open-data/dwn-pluto-mappluto.page

ACKNOWLEDGEMENTS

I would like to express my gratitude to my Adviser Prof. Kameyama for their guidance in conducting scientific research and for their time and the patience they showed during our brainstorming sessions. My gratitude goes to Prof. Kameyama, for his precious help and support during Master’s course, not only in matters of studies but also with administrative issues. A special thank you goes to my family for their constant encouragement and support and without the help of whom I could not have pursued this Master's degree.

29 Jan. 2020 Sakura Yamaki

LIST OF PUBLICATIONS

[1] 八巻櫻，菅沼睦，亀山渉，“交通系 IC カード大規模データにおける非負値行列因子分解を用いた利用者行動分析の一検討”，2018 電子情報通信学会総合大会，D-4，2018 年 3 月.

[2] Sakura Yamaki, Shou-de Lin, and Wataru Kameyama, “Detection of Anomaly State Caused by Unexpected Accident using Data of Smart Card for Public Transportation”, 2019 IEEE International Conference on Big Data, BigD254, Dec 9-12, Los Angeles, CA, USA, Dec. 2019.

APPENDIX

A. Data dictionary provided by TLC

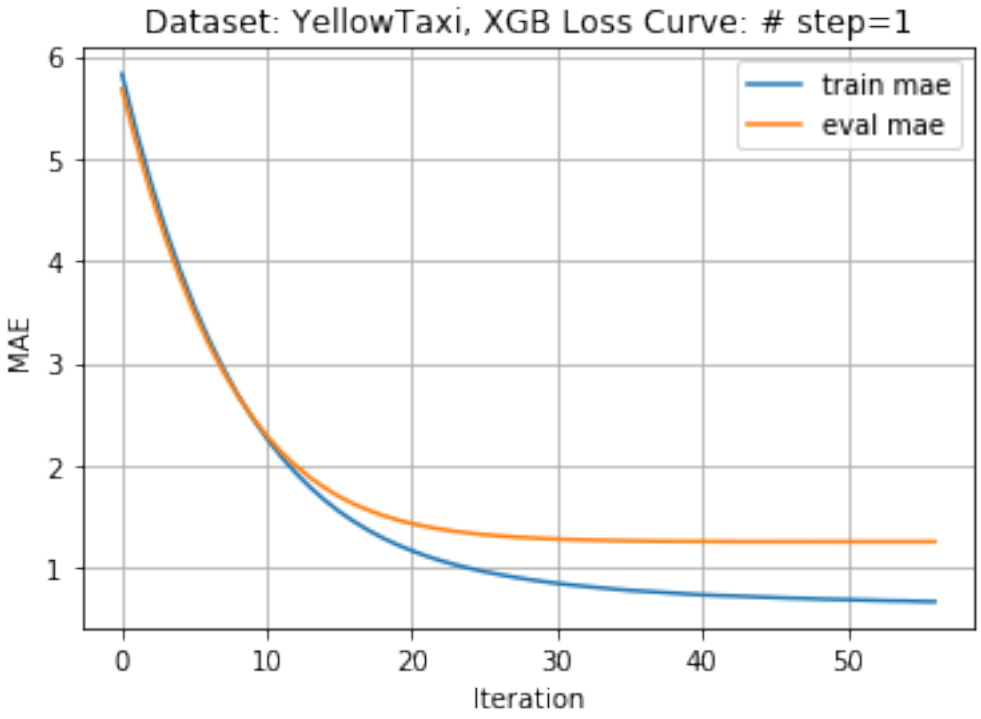
A.1: Yellow taxi data dictionary provided by TLC [24].

Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. **1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.**
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. **Y= store and forward trip** **N= not a store and forward trip**
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
MTA_tax	$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.

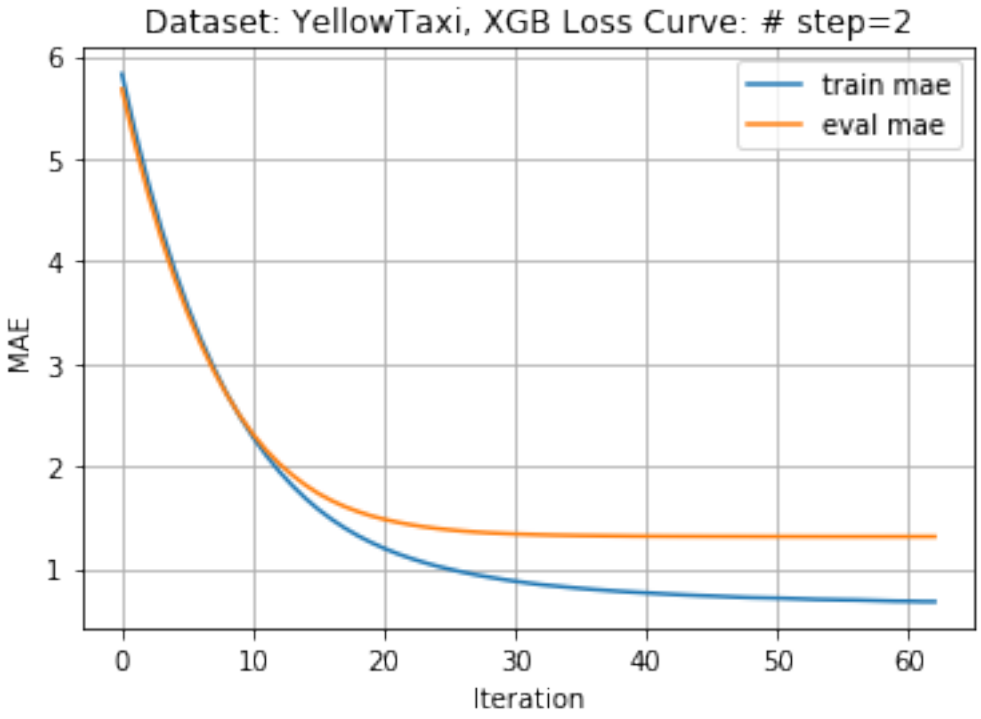
A. 2: HVFHV data dictionary provided by TLC

Field Name	Description
Hvfhs_license_num	The TLC license number of the HVFHS base or business As of September 2019, the HVFHS licensees are the following: • HV0002: Juno • HV0003: Uber • HV0004: Via • HV0005: Lyft
Dispatching_base_num	The TLC Base License Number of the base that dispatched the trip
Pickup_datetime	The date and time of the trip pick-up
DropOff_datetime	The date and time of the trip drop-off
PULocationID	TLC Taxi Zone in which the trip began
DOLocationID	TLC Taxi Zone in which the trip ended
SR_Flag	Indicates if the trip was a part of a shared ride chain offered by a High Volume FHV company (e.g. Uber Pool, Lyft Line). For shared trips, the value is **1**. For non-shared rides, this field is **null**. **NOTE:** For most High Volume FHV companies, only shared rides that were requested AND matched to another shared-ride request over the course of the journey are flagged. However, Lyft (hvfhs_license_num='HV0005') also flags rides for which a shared ride was requested but another passenger was not successfully matched to share the trip—therefore, trips records with SR_Flag=1 from those two bases could indicate EITHER a trip in a shared trip chain OR a trip for which a shared ride was requested but never matched. Users should anticipate an overcount of successfully shared trips completed by Lyft. Note also that Juno does not offer shared trips at this time.

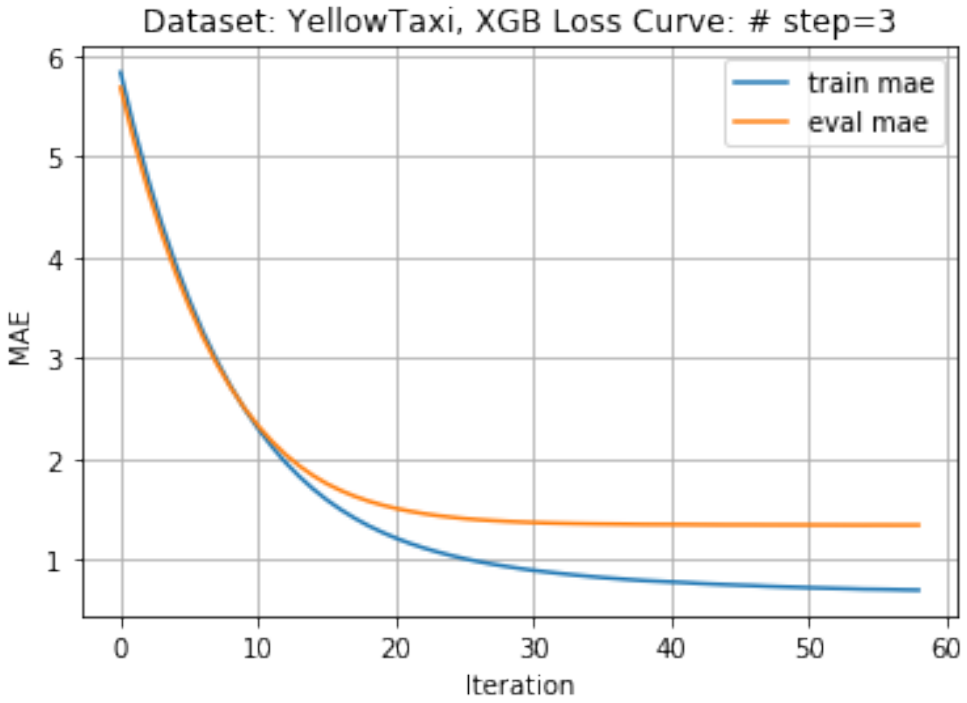
B. MAE loss curve by several prediction models



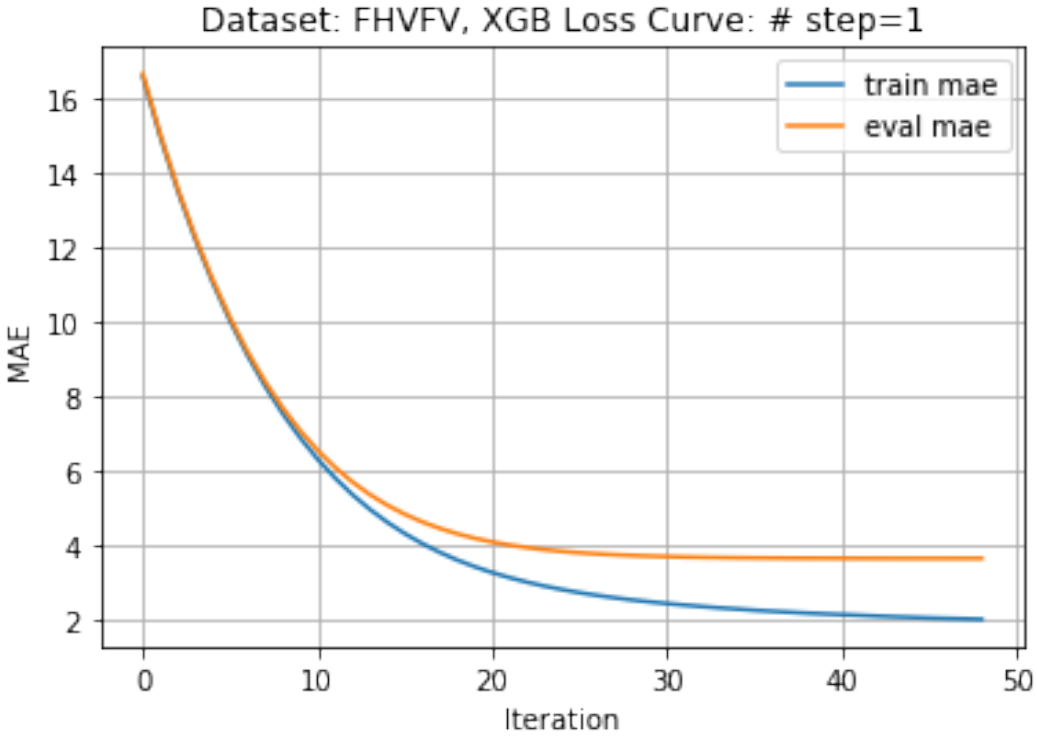
B.1: MAE loss curve by XGB model using Yellow Taxi dataset for future 10 minutes prediction.



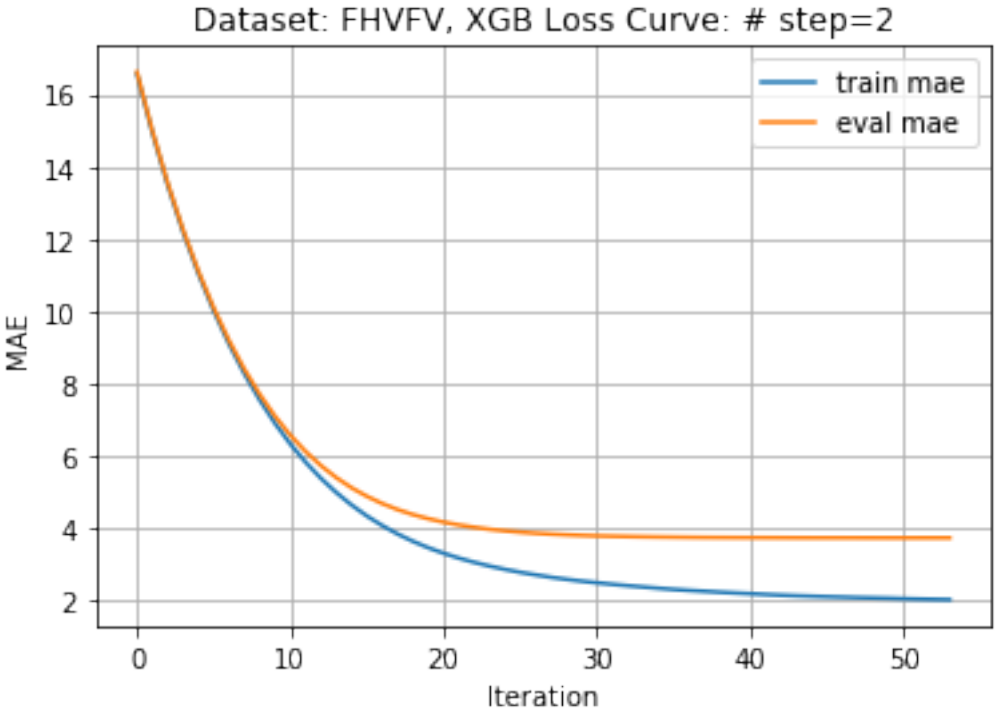
B.2: MAE loss curve by XGB model using Yellow Taxi dataset for future 20 minutes prediction.



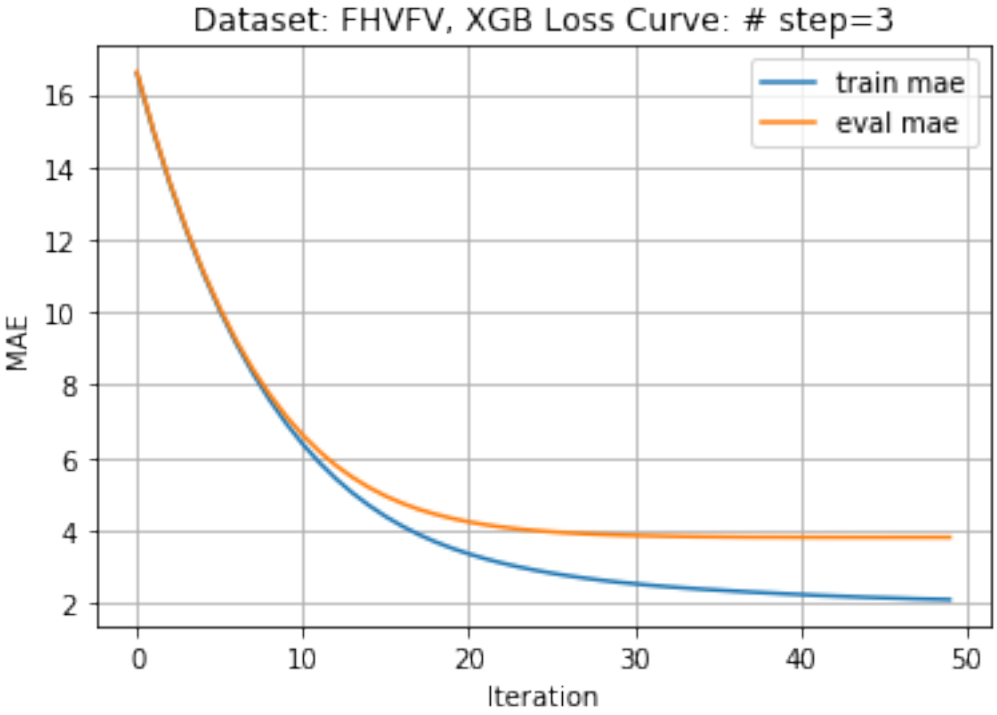
B.3: MAE loss curve by XGB model using Yellow Taxi dataset for future 30 minutes prediction.



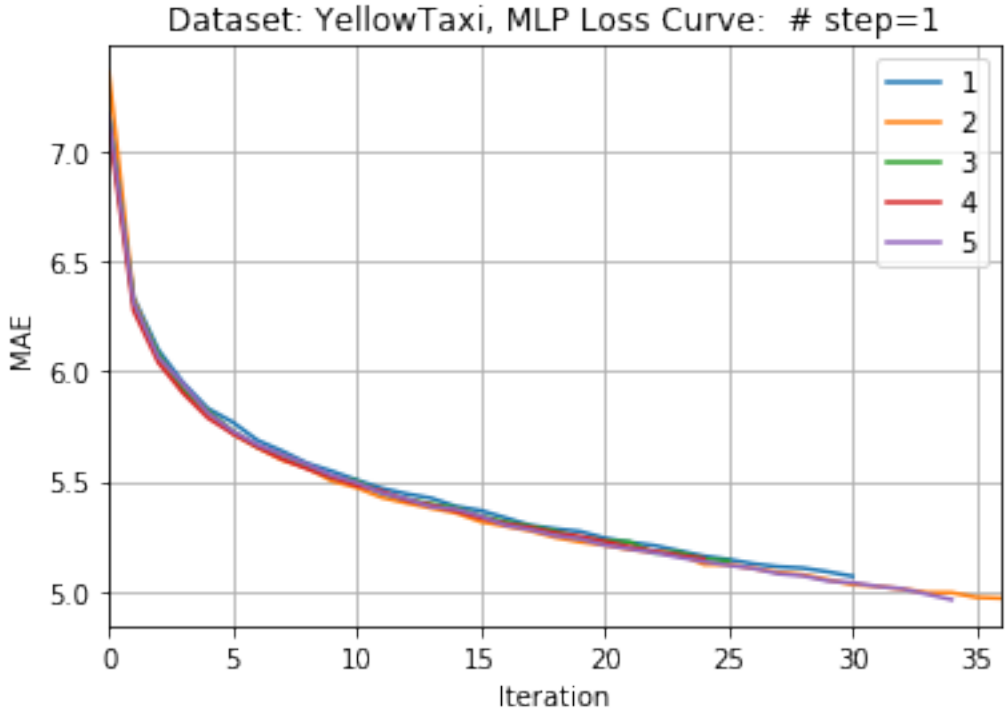
B.4: MAE loss curve by XGB model using FHVFV dataset for future 10 minutes prediction.



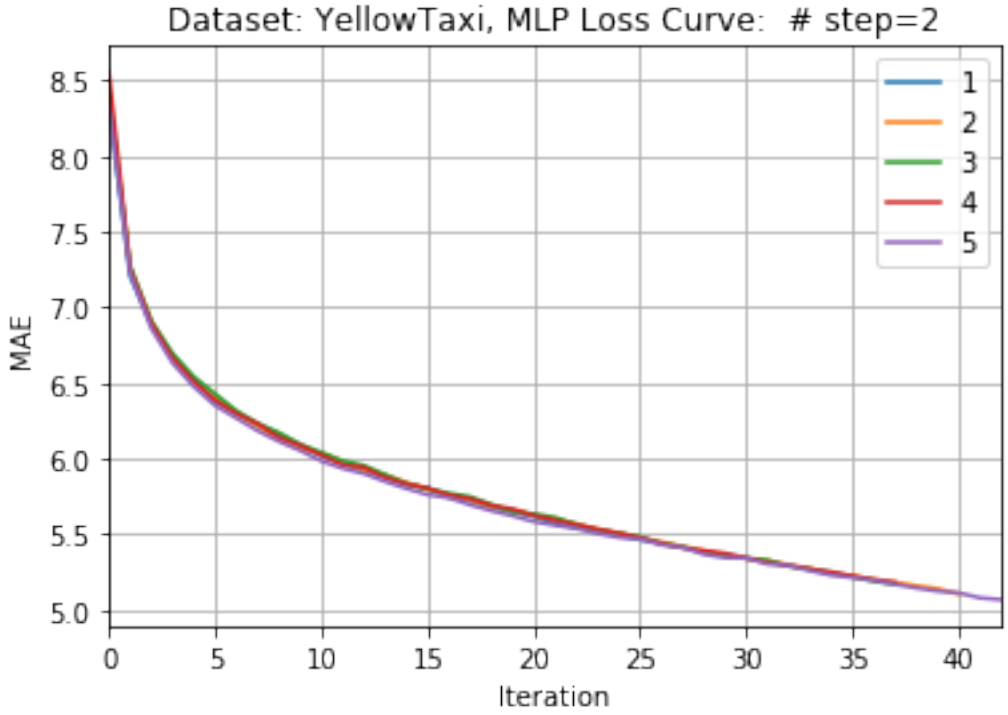
B.5: MAE loss curve by XGB model using FHVFV dataset for future 20 minutes prediction.



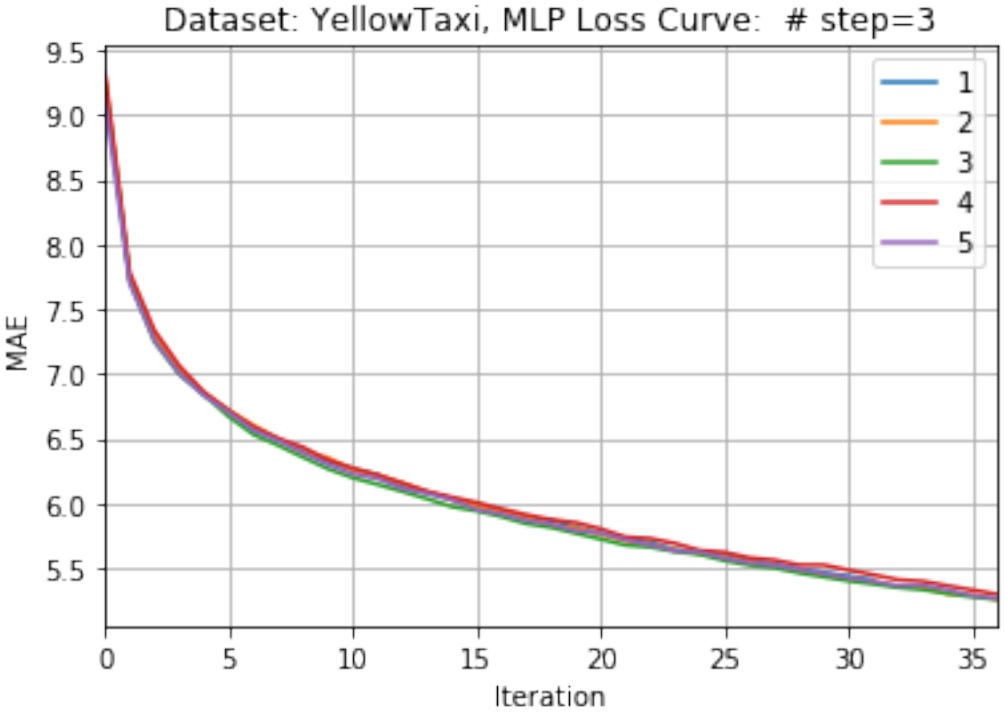
B.6: MAE loss curve by XGB model using FHVFV dataset for future 30 minutes prediction.



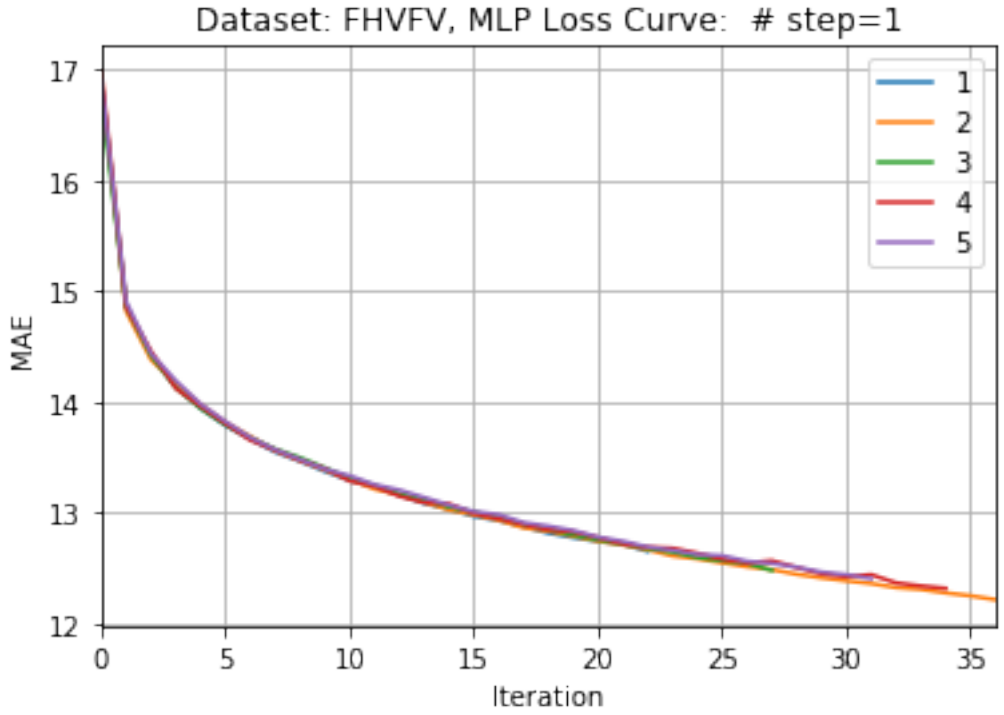
B.7: MAE train loss curve by MLP model using Yellow Taxi dataset for future 10 minutes prediction for 5 times.



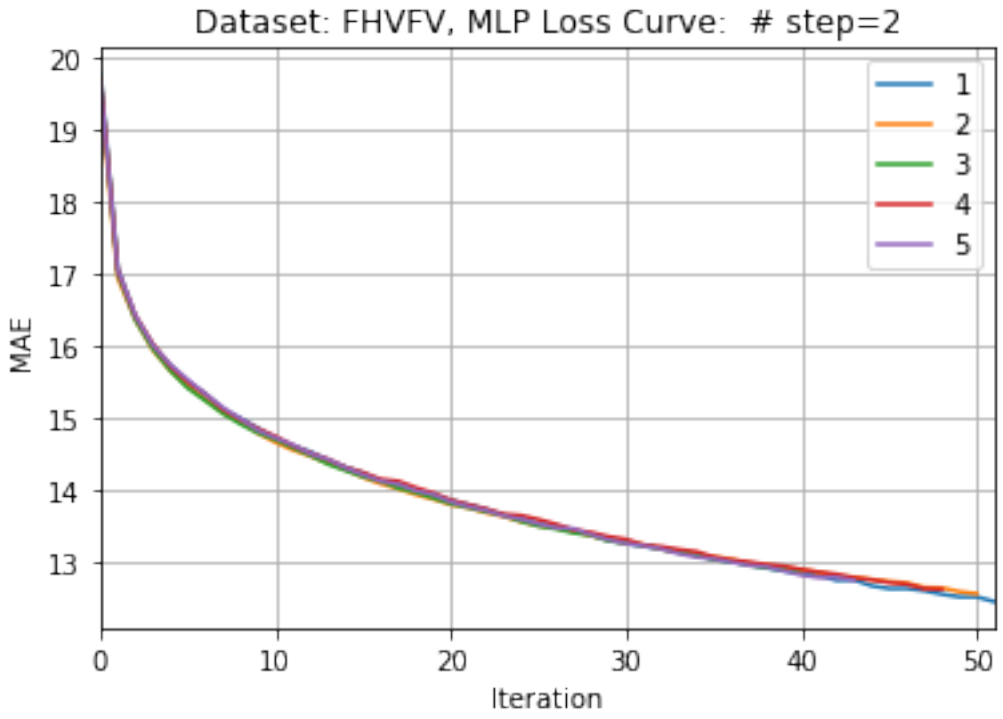
B.8: MAE train loss curve by MLP model using Yellow Taxi dataset for future 20 minutes prediction for 5 times.



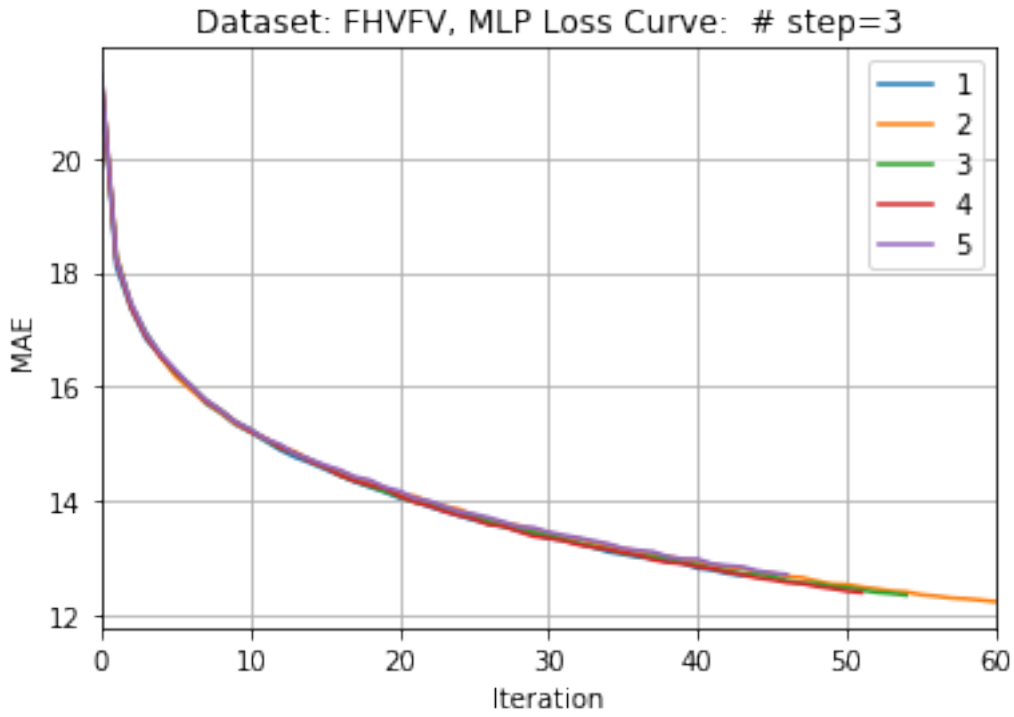
B.9: MAE train loss curve by MLP model using Yellow Taxi dataset for future 30 minutes prediction for 5 times.



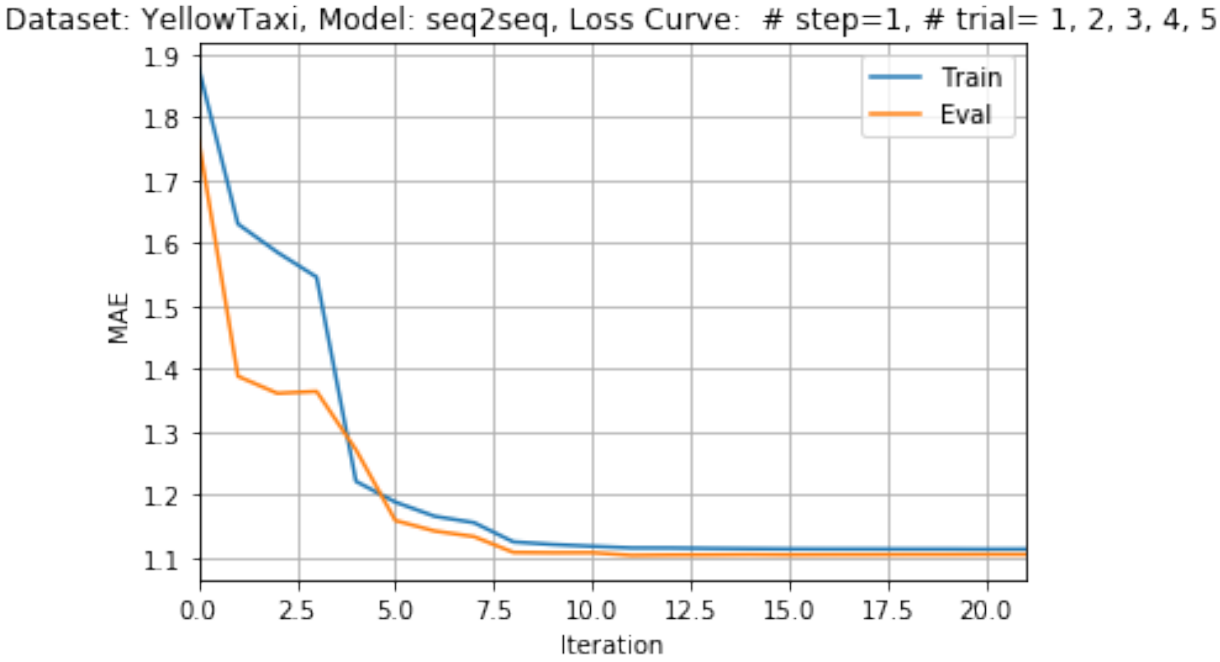
B.10: MAE train loss curve by MLP model using FHVFV dataset for future 10 minutes prediction for 5 times.



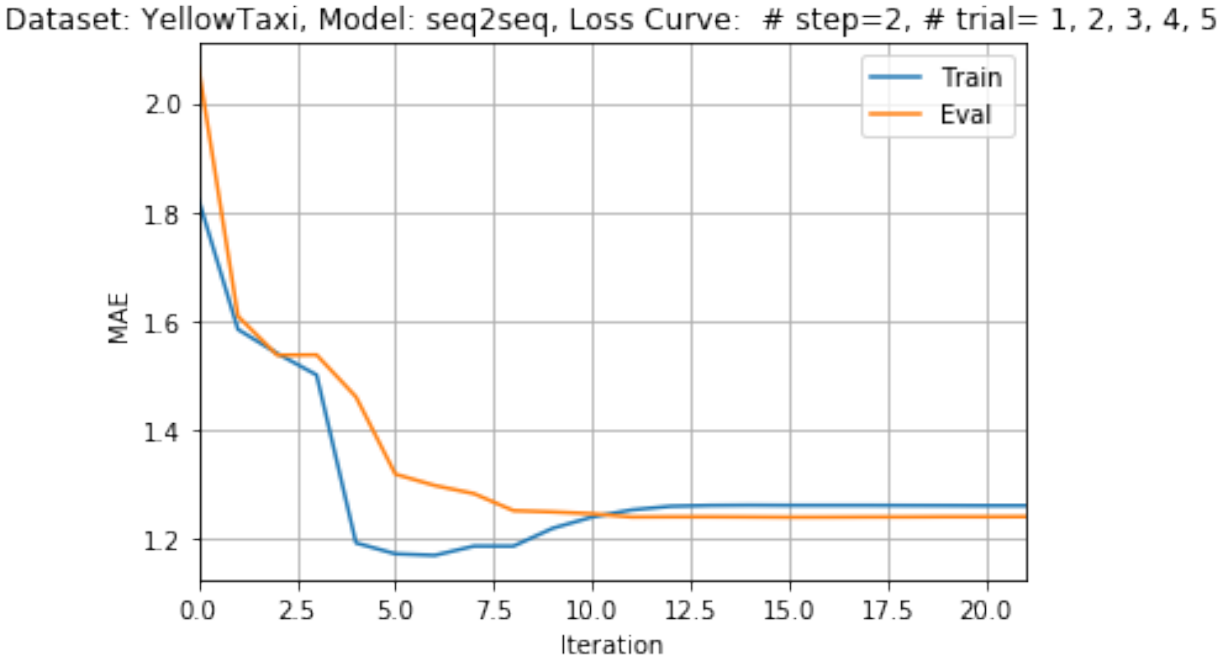
B.11: MAE train loss curve by MLP model using FHVFV dataset for future 20 minutes prediction for 5 times.



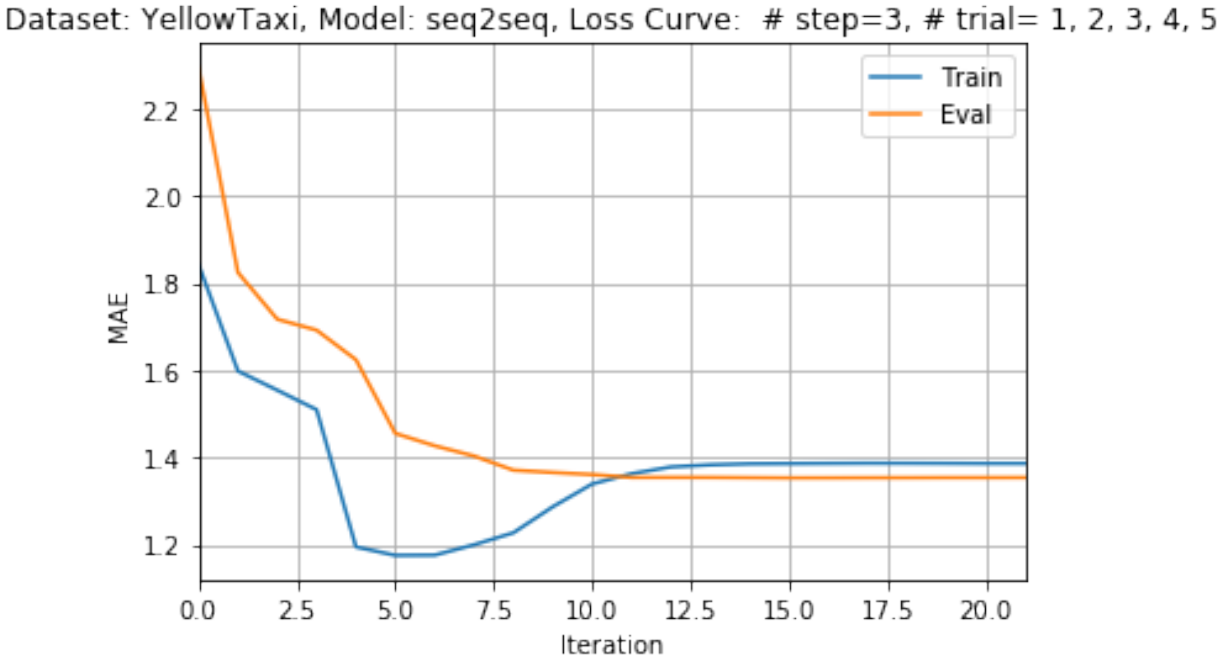
B.12: MAE train loss curve by MLP model using FHVFV dataset for future 30 minutes prediction for 5 times.



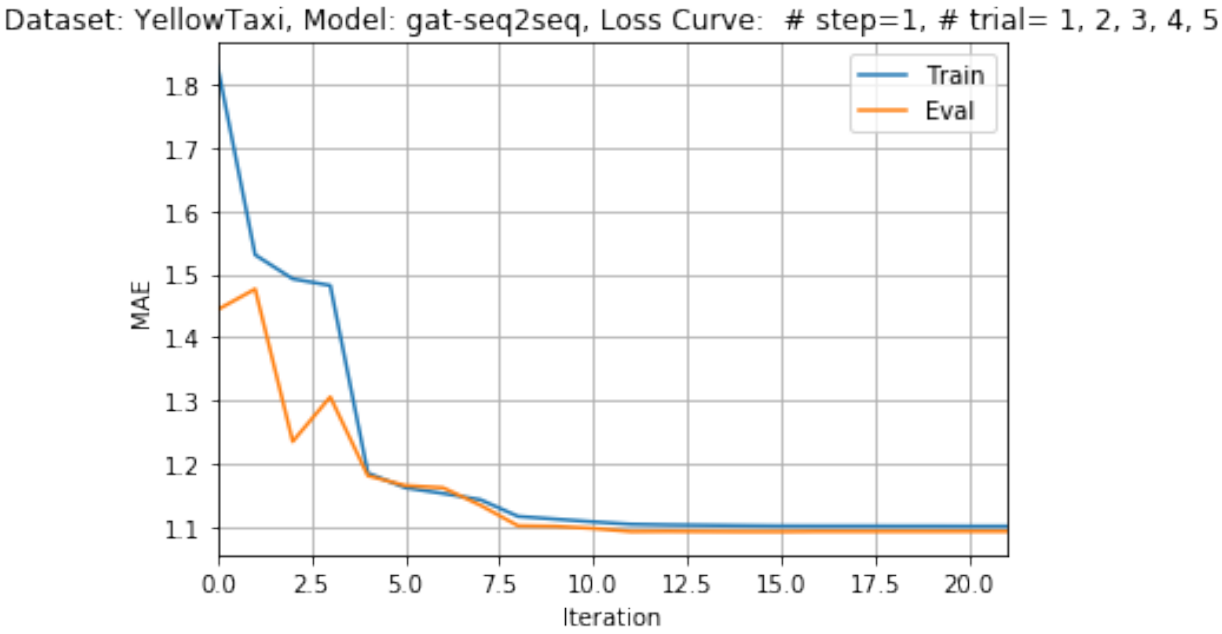
B.413: MAE loss curve by Seq2Seq model using Yellow Taxi dataset for future 10 minutes prediction for 5 times.



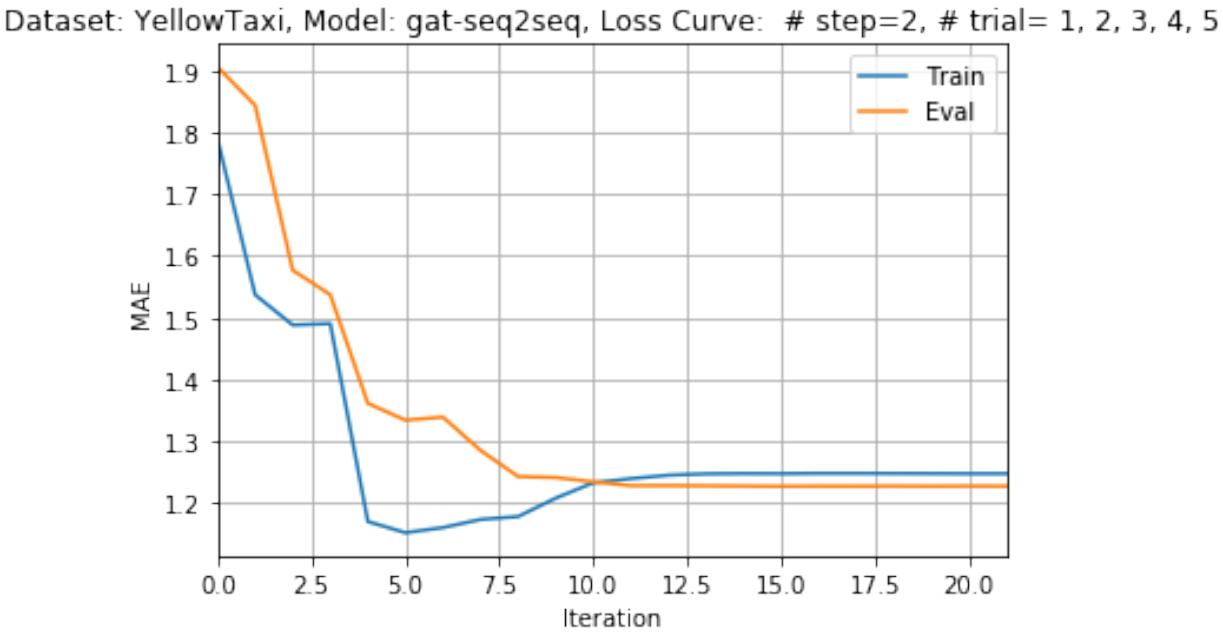
B.14: MAE loss curve by Seq2Seq model using Yellow Taxi dataset for future 20 minutes prediction for 5 times.



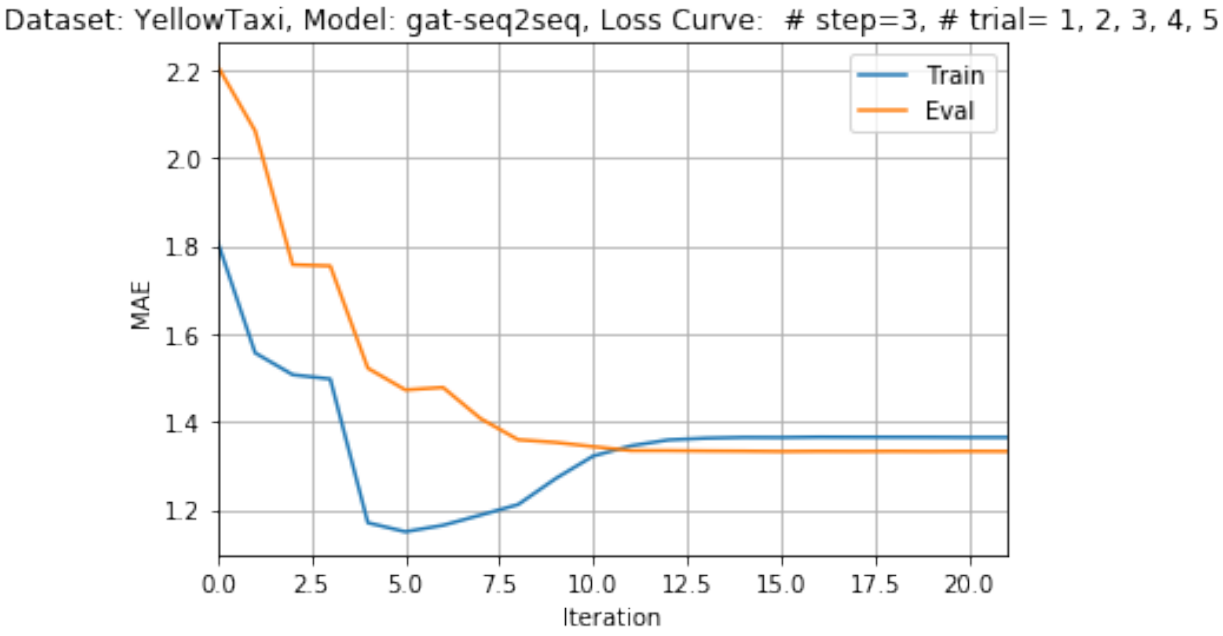
B.15: MAE loss curve by Seq2Seq model using Yellow Taxi dataset for future 30 minutes prediction for 5 times.



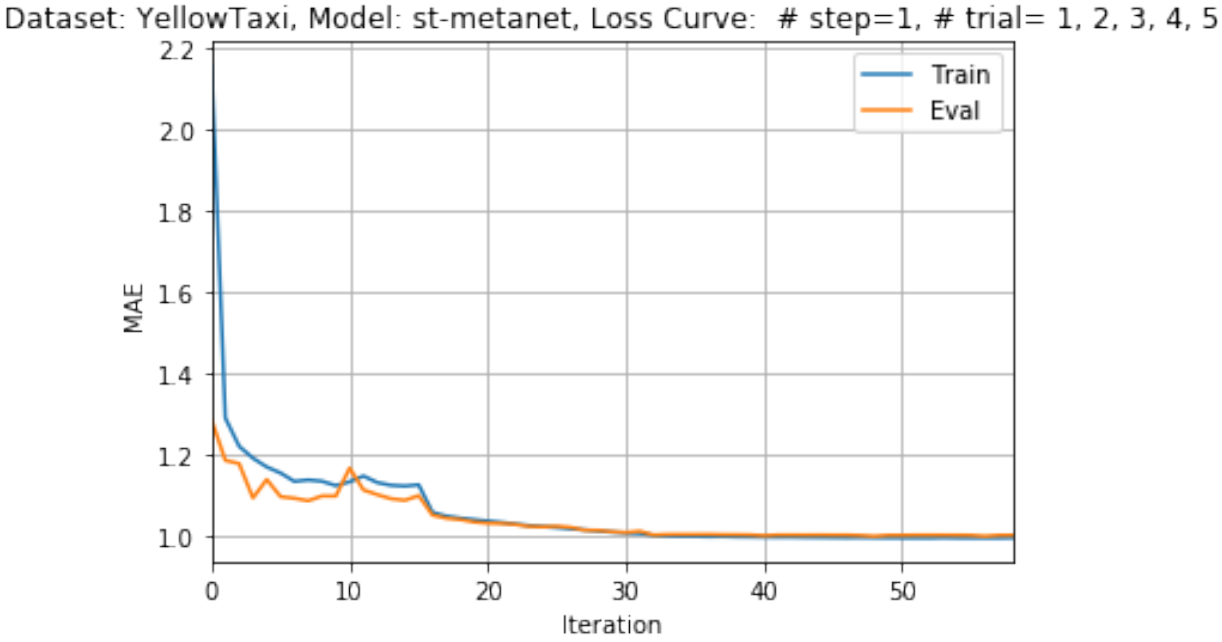
B.16: MAE loss curve by GAT-Seq2Seq model using Yellow Taxi dataset for future 10 minutes prediction for 5 times.



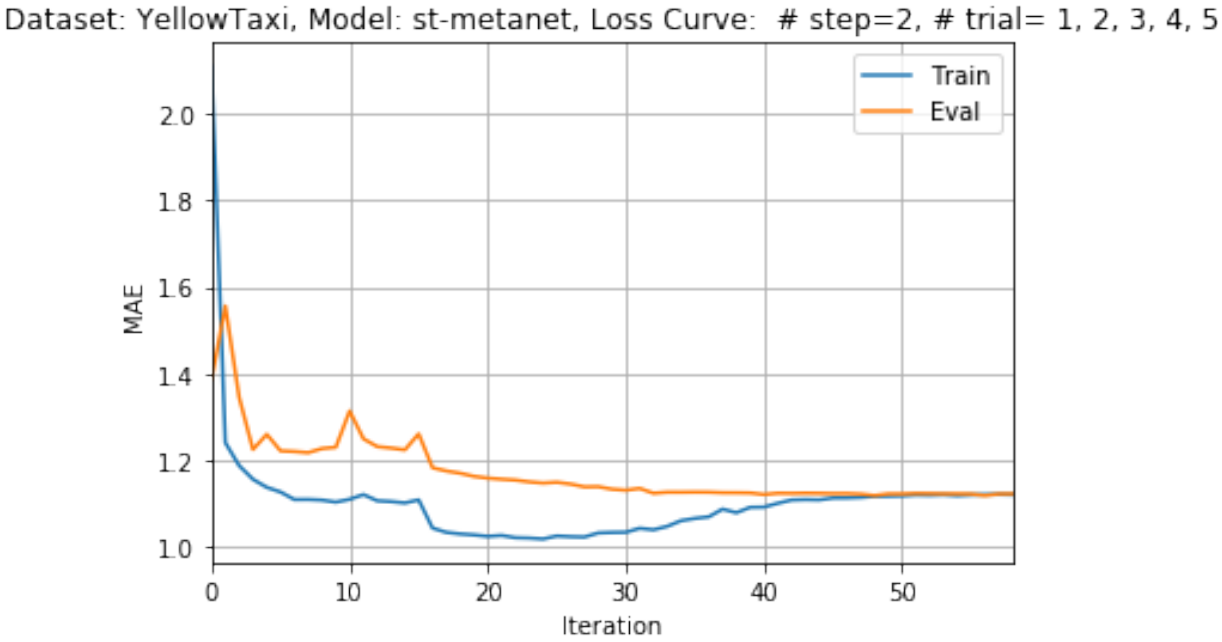
B.17: MAE loss curve by GAT-Seq2Seq model using Yellow Taxi dataset for future 20 minutes prediction for 5 times.



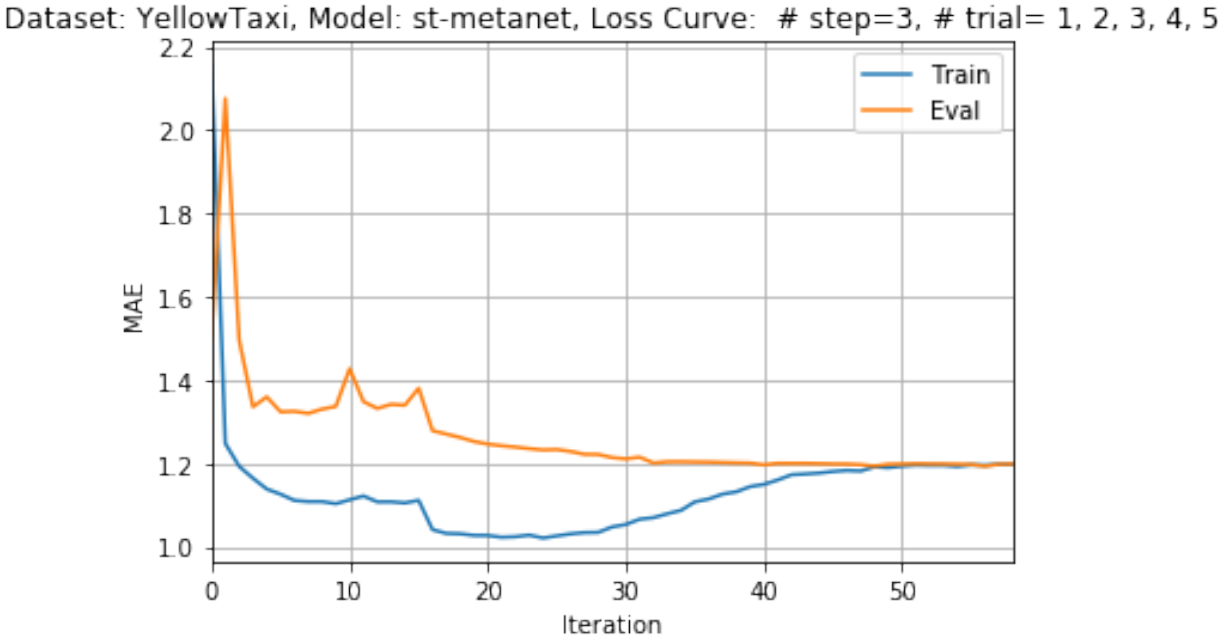
B.18 MAE loss curve by GAT-Seq2Seq model using Yellow Taxi dataset for future 30 minutes prediction for 5 times.



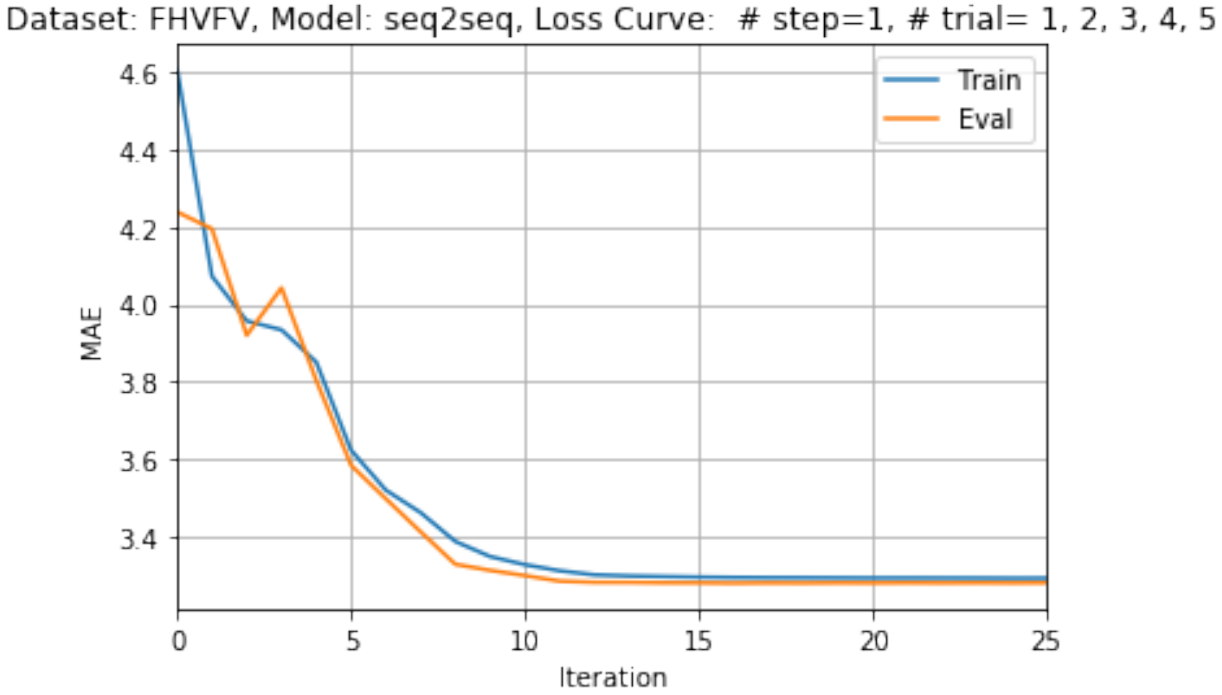
B.19: MAE loss curve by ST-MetaNet model using Yellow Taxi dataset for future 10 minutes prediction for 5 times.



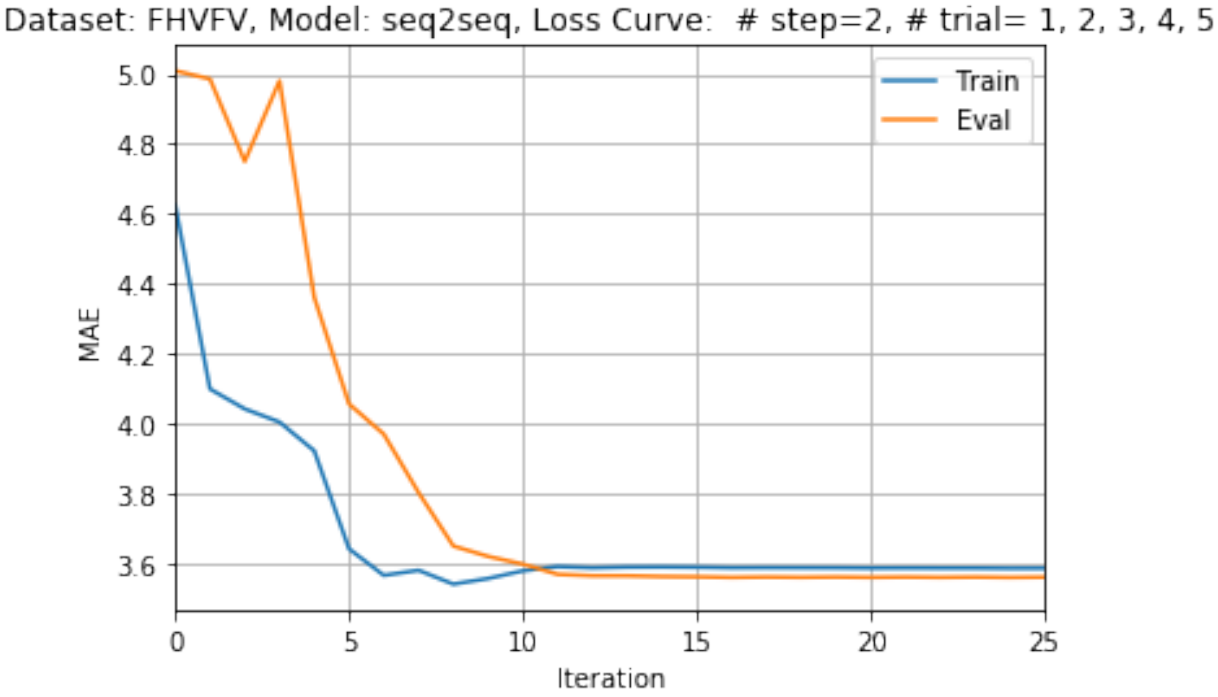
B.20: MAE loss curve by ST-MetaNet model using Yellow Taxi dataset for future 20 minutes prediction for 5 times.



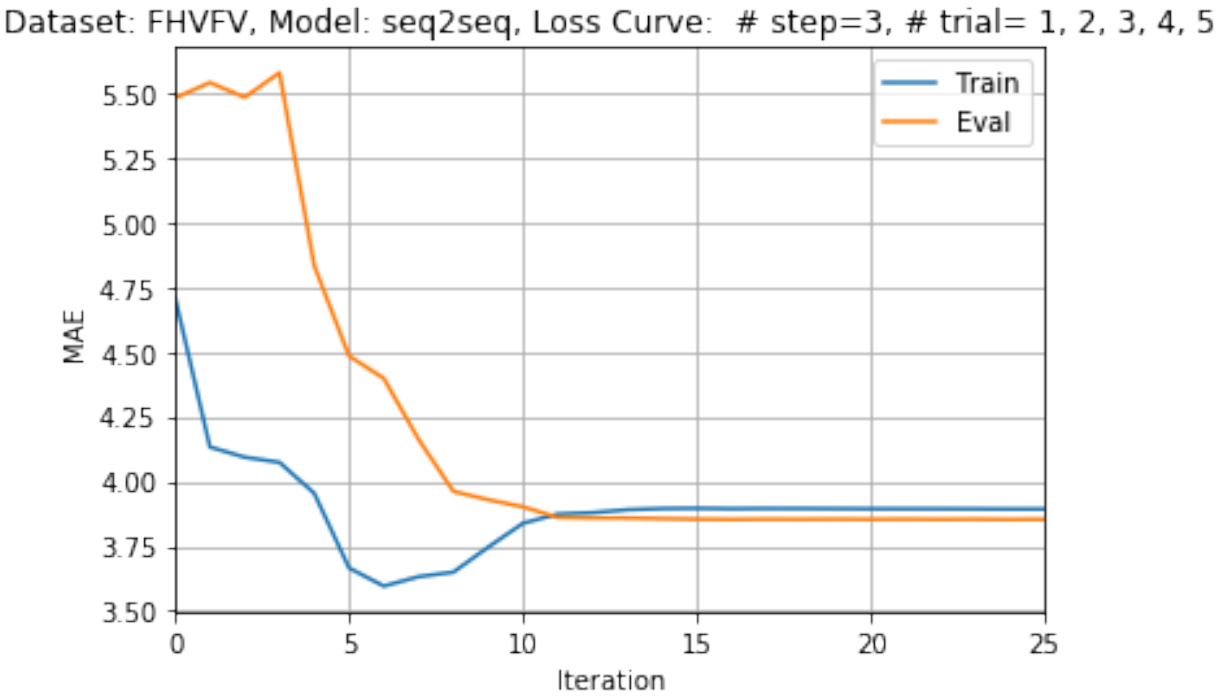
B.21: MAE loss curve by ST-MetaNet model using Yellow Taxi dataset for future 30 minutes prediction for 5 times.



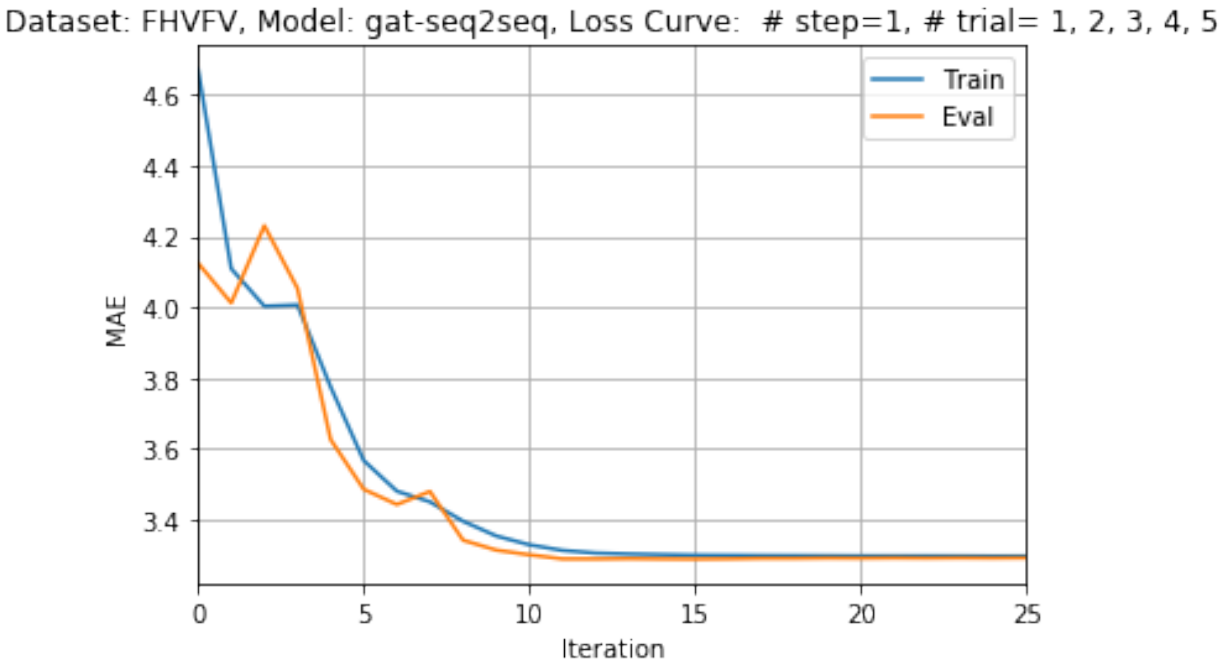
B.22 MAE loss curve by Seq2Seq model using FHVFV dataset for future 10 minutes prediction for 5 times.



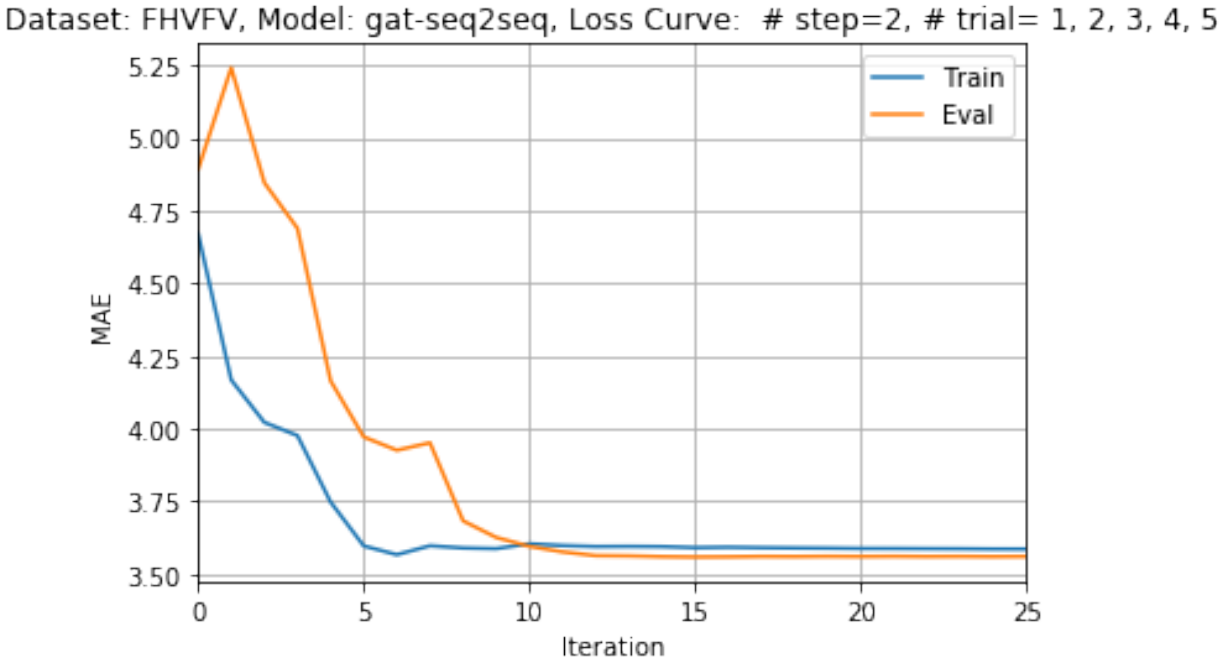
B.23: MAE loss curve by Seq2Seq model using FHVFV dataset for future 20 minutes prediction for 5 times.



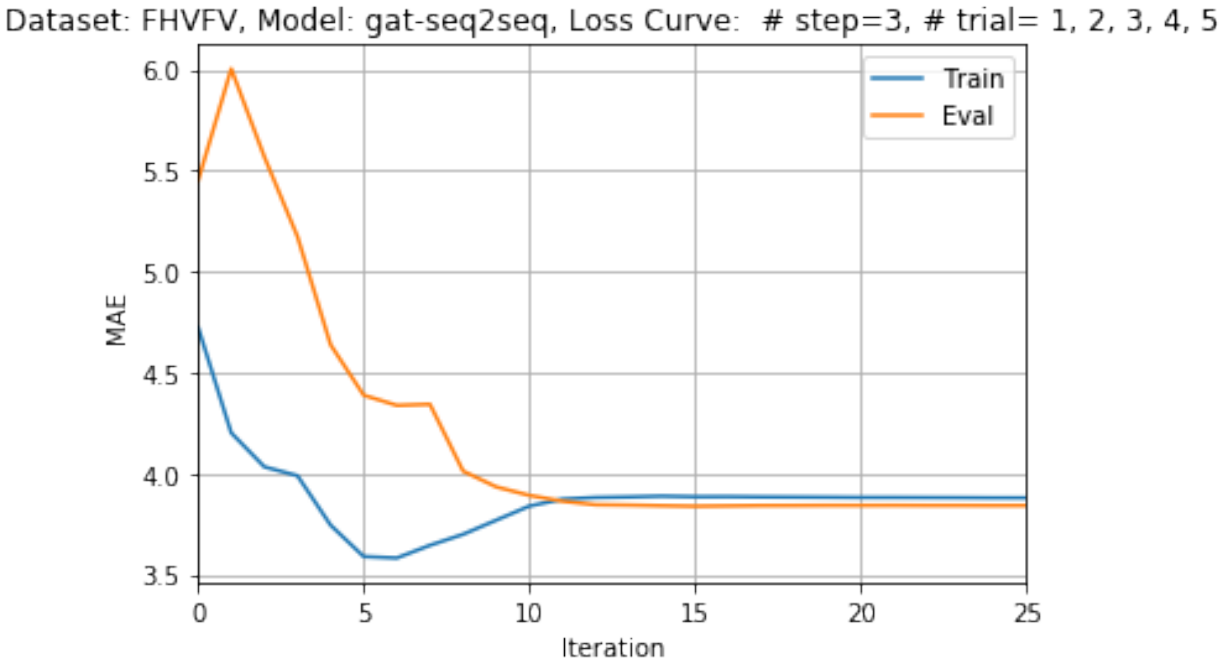
B.24: MAE loss curve by Seq2Seq model using FHVFV dataset for future 30 minutes prediction for 5 times.



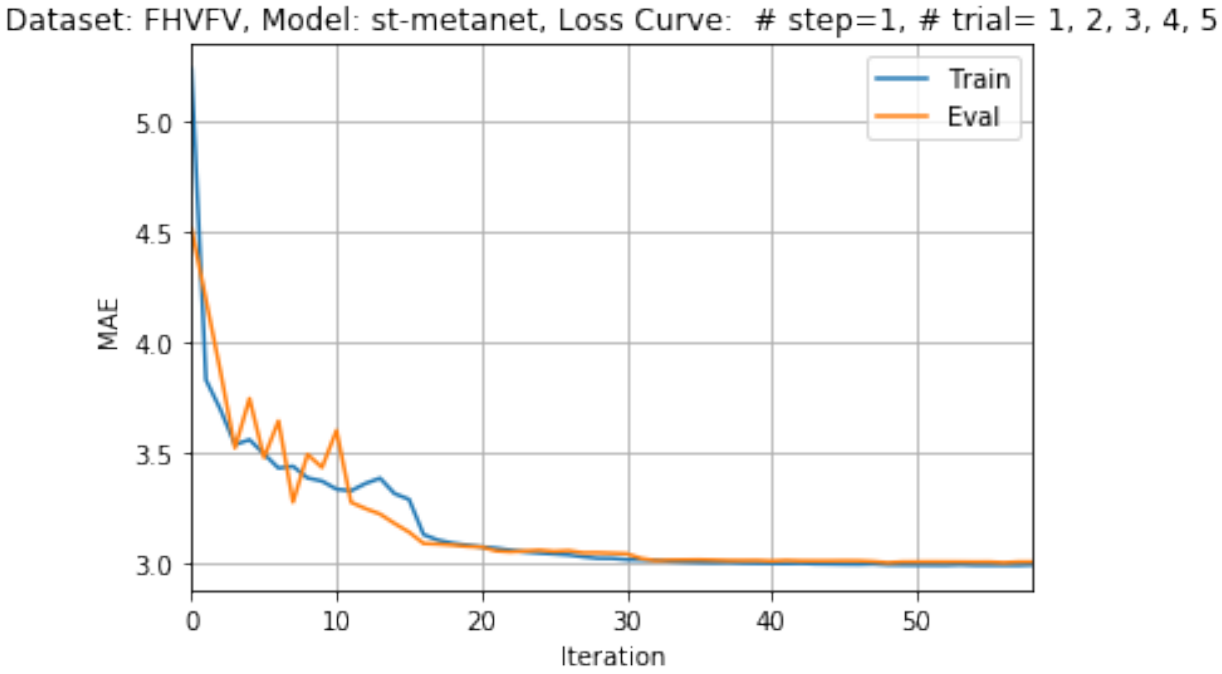
B.24: MAE loss curve by GAT-Seq2Seq model using FHVFV dataset for future 10 minutes prediction for 5 times.



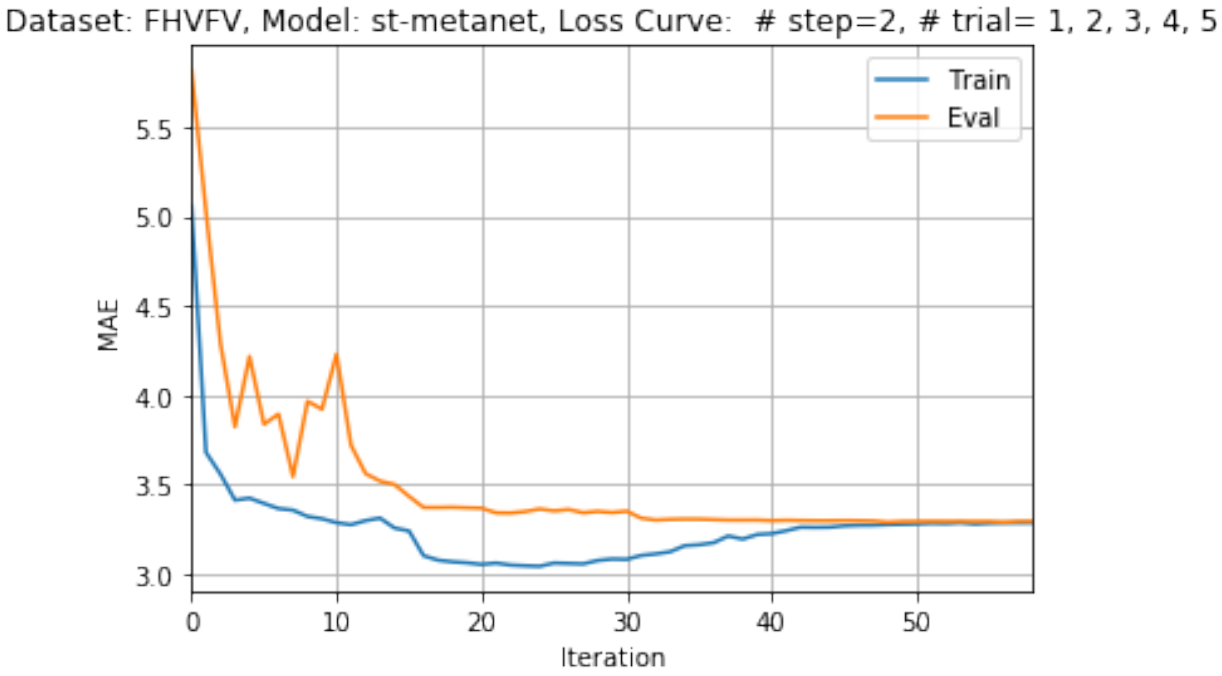
B.25: MAE loss curve by GAT-Seq2Seq model using FHVFV dataset for future 20 minutes prediction for 5 times.



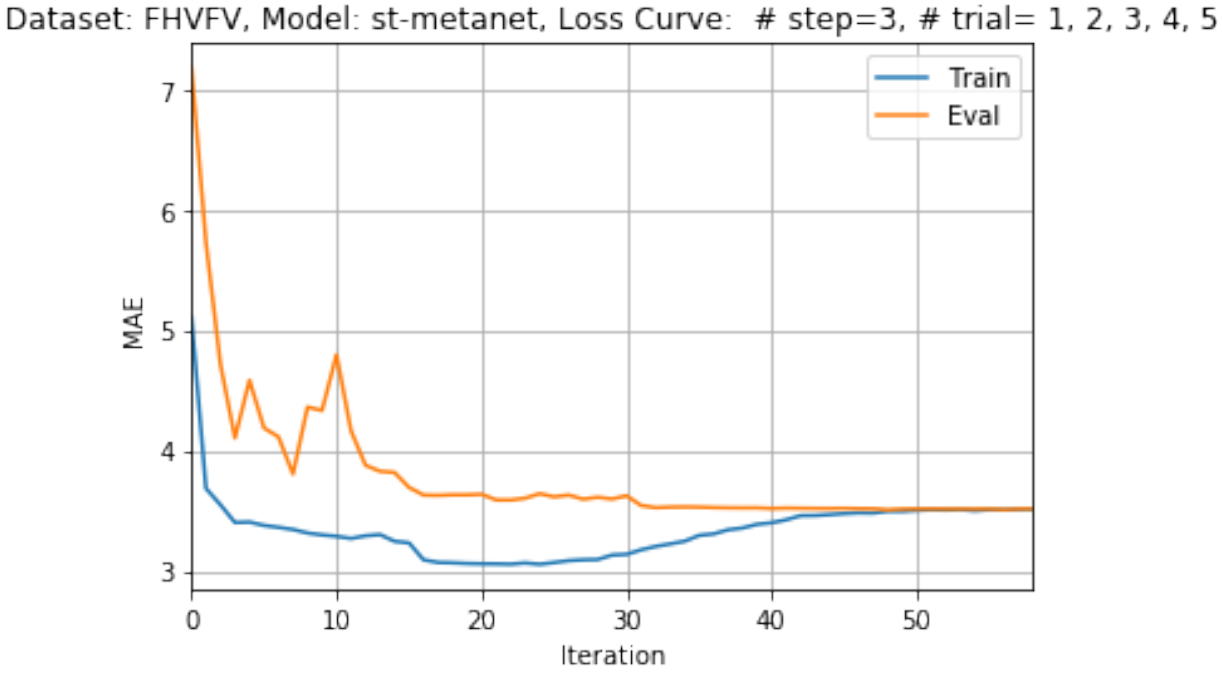
B.26: MAE loss curve by GAT-Seq2Seq model using FHVFV dataset for future 30 minutes prediction for 5 times.



B.27 MAE loss curve by ST-MetaNet model using FHVFV dataset for future 10 minutes prediction for 5 times.



B.28: MAE loss curve by ST-MetaNet model using FHVFV dataset for future 20 minutes prediction for 5 times.



B.29: MAE loss curve by ST-MetaNet model using FHVFV dataset for future 30 minutes prediction for 5 times.