Multi-Source Information Based Short-term Taxi Demand Prediction Using Deep-Learning Approaches

Ziyan Chen*
College of Transportation Engineering, Tongji University, Shanghai, China

*Corresponding author e-mail: chenziyan@tongji.edu.cn

Abstract. Taxi service is of great importance in public transportation for its flexibility and convenience. As the advent of smartphones, traditional taxi service is under revolution. Instead of randomly taking a taxi, people are able to call for a taxi pickup by the smartphone applications (e.g. Uber and Lyft). Once the nearest taxi driver receives the request, he can head to the pick-up location. While in most urban areas, the taxi service demand and supply are imbalanced, which indicates that sometimes passengers wait too long for taxis and drivers roam too long with vacant taxicabs. Short-term demand prediction is of great importance to the on-demand ride-hailing services. Predicted taxi demand information can facilitate efficient operations and rebalance both demand and supply sides. This paper proposes a multi-source information based spatiotemporal neural network (MSI-STNN) deep learning architecture to predict short-term taxi demand. The model fuses pick-up and drop-off time-series data, weather information, and location popularity data, using three deep-learning models, including stacked convolutional long short-term memory (ConvLSTM) model, stacked long short-term memory (LSTM) model and convolutional neural network (CNN) model. ConvLSTM captures the spatiotemporal features of pick-up and drop-off time series. LSTM and CNN extracts information of weather and popularity. Case studies were performed to predict short-term pick-up demand at zonal levels in 15 minutes using New York taxi data. The experiment results validate the accuracy performance of the proposed approach comparing with state-of-art time-series and deep learning approaches, including ARIMA, ConvLSTM, CNN, and LSTM.

Index Terms: Taxi demand, short-term prediction, deep learning, multi-source information.

1. Introduction
The twentieth century has witnessed the transition of travel mode preference from private to public. One is the advent of taxicab in 1898 in Paris. Taxi service is of great importance in public transportations for its flexibility and convenience. Taxi is an end-to-end travel mode. It is not the first choice for most of the commuters, though. Because it takes more money and waiting time compared to other modes. Moreover, it has high probability to encounter congestion problem. Under this circumstance, almost every taxi company commits themselves to exploring solutions to attract passengers. Currently, with the increasing popularity of taxi service platforms such as Uber and Lyft, people would prefer requesting a
pickup on their smartphone instead of randomly taking a taxi. While in most urban areas, taxi service demand and supply are imbalanced. For the passengers, there exists the situation when no taxicab is near their location. At the same time, some taxi drivers roam too long with vacant cars in other areas. Eventually, these areas are divided into over-demand and over-supply. On one hand, it leads to the profit loss for both drivers and taxi companies. On the other hand, it is a waste of time for passengers.

To address this imbalance between demand and supply, short-term taxi demand forecasting is a critical solution. If taxi companies acquire accurate demand prediction in advance, they can pre-allocate taxi fleet from over-supply to over-demand regions to meet the passenger demand and improve service performance. Predictive data analytic facilitates proactive decision support for both operators and travelers in transportation. Demand forecasting problem has been widely studied in the past decades, and vast amount of methods emerge successively, from model-based time series analysis, to machine learning techniques and model-free deep learning models.

Traditional time series analysis leverages statistical models, which can predict the future values by giving the successive historical records. For example, Bayesian Forecasting [1], the autoregressive integrated moving average (ARIMA) model [2] and Kalman filter [3] are the most classic ones. These algorithms have been applied in numerous advanced statistical models, such as the applications in supply chains [4] and traffic flow prediction [5-6]. However, these algorithms rely on a specific mathematical model, which should be linear and Gaussian distribution. This limitation is inconsistent with complicated characteristics of data from various sources. Typically, these statistical models have been proved to be efficient in simple stationary time series problems, which don’t take multi source data into account. And in this context, it is difficult to reach the high-precision requirement.

Therefore, machine learning methods were widely applied in demand forecasting in various domains. Some classic algorithms like artificial neural network [7] and random forest [8] have made great contributions to the time series analysis in transportation with the rapid development of computer science. For example, an improved K-nearest neighbors [9] and optimized artificial neural network [10] were adopted in traffic flow prediction. And the accuracy of these advanced models can be 90 percent or so, which satisfies the high-precision requirement. However, machine learning models are not always accessible. The prospective outcome heavily depends on the input, i.e., feature selection of input is necessary but problem-wise. While for the model, hyperparameters should be manually calibrated to yield the best prediction. Another shortcoming is that machine learning models should be implemented and trained on CPU, while the number of cores on GPU is more than that on CPU. This limits the capability of dealing with large datasets.

Recently, researchers have seen the great breakthrough of implementing the deep learning in computer science, which inspires them to apply DL in the domain of transportation, especially in demand forecasting. Studies have shown the advantage of RNN and its variants, i.e., long short-term memory (LSTM) [11] and gated recurrent units (GRU) [12], in time series analysis. [13] employed long short-term memory neural network (LSTM NN) to capture temporal characteristics with the optimal time lags for the traffic speed prediction. [14] applied LSTM to describe temporal relations for traffic state prediction, and they also added autoencoder to extract some extreme situations, e.g., peak hour and traffic accidents. [15] considered sharp nonlinearities, e.g., transitions, breakdown, recovery and congestion, as major effects in performance of traffic flow prediction, and based on this, they put forward the combined \( \ell_1 \) regularization and a sequence of tanh layers to capture the nonlinearities. However, the models mentioned above fail to consider the spatial characteristics, which is an endogenous dependency in zonal based demand forecasting.

Meanwhile, CNN has shown its powerful capability in computer vision, which inspires the transportation researchers to employ CNN in traffic data prediction. Typically, the research area is seen as an image, and then we can extract local features using CNN. [16] proposed a CNN based architecture to predict the traffic speed. In that paper, they treat the traffic as images, and then CNN is applied to extract the features of a vehicle trajectory in each road segment, from which the vehicle speed can be estimated. However, these methods don’t pay much attention on temporal correlations, since they simply fuse the data extracted by CNN. Later, researchers found a suitable solution to combine the spatial and
temporal characteristics together, i.e., convolutional long short-term memory (ConvLSTM), which is first introduced by [17]. For example, [18] employed ConvLSTM to handle the data of travel time rate and demand intensity. And [19] applied ConvLSTM to extract the spatiotemporal features of crash risk and taxi trips. The experimental outcomes indicate that ConvLSTM is more reliable and efficient in terms of dealing with data that has both time dependencies and spatial discrepancies. But this method has not been widely adopted yet.

Short-term demand prediction is of great importance to the on-demand ride-hailing services. The predicted demand information can facilitate efficient operations (e.g., ride-matching, vehicle routing and rebalancing) and improve service performance (e.g., less waiting time and detour time). Several studies have proposed methods to predict taxi demand in short-term using deep learning techniques ([20][21][22]). However, most of the studies used only time-series demand data for prediction without considering important exogenous variables, such as weather or location popularity, etc. It is not trivial to fuse multi-source information to make real-time prediction using deep learning techniques.

To tackle the challenges above, we propose a multi-source information based spatiotemporal neural network (MSI-STNN) approach to fuse pick-up, drop-off and exogenous variables simultaneously. To be specific, after preparing the data of pick-up demand, drop-off demand, weather conditions and popularity, the stacked convolutional long short-term memory (ConvLSTM) is applied to extract the spatiotemporal features of pick-up and drop-off demand simultaneously rather than combining RNN and CNN to acquire spatial and temporal information separately. The stacked long short-term memory (LSTM) is adopted to predict the weather conditions, and CNN is used to extract zonal popularity feature among zones.

In summary, our contributions are listed as follow:

- POI
- ConvLSTM
- Multi source
- Outperform other baselines

The rest of this paper is organized as follow: Section 2 explicitly describes each explanatory variable and provides the structure and mathematical explanation of the proposed MSI-STNN. Section 3 gives the experimental outcomes and compares the performance between the proposed architecture and some baselines. Section 4 concludes the whole paper.

2. Methodology

2.1. Preliminaries

The short-term taxi demand forecasting is inherently a time series prediction problem, which implies that we can consider taxi demand from previous time as valuable information for future prediction. This paper focuses on the prediction of taxi pick-ups in the next 15 minutes. Conceptually, the number of drop-offs could influence taxi pick-up demand, since people may potentially take a taxi to a place for a particular activity (e.g., concert, party, work, etc.), and then return by using a taxi again. In addition, the weather conditions and location popularity could also impact the demand generation. Here, the definitions and notations of the variables used in this paper are described:

1. Taxi zone and time partition

   The urban area is partitioned into small grids with irregular shape, where each grid represents a taxi zone. For the time interval, we aggregate variables in every 15 minutes, which implies the time interval is 15 minutes. Based on this, the variables of pick up and drop off have same dimensions.

2. Pick-up and drop-off demand

   The pick-up demand at the tth time slot (e.g., 15 min) lying in zone i is defined as the number of pickups during this time interval within the zone, which is denoted by pt,i. And then the pick-up demand for all the zones in each time interval is defined as matrix \( P_t \), where the ith element is (\( P_t \))i = pt,i. Similar definition for drop-offs where dt,i denotes the number of drop-off records at the tth time slot.
(e.g., 15 min) lying in zone i. Again, the drop-off records for all the zones in each time interval is kept in matrix $D_t$, where the ith element is $(D_t)_i = dt,i$.

3) Popularity

A point of interest (POI) represents a specific point location that people may find useful and interesting. So, the point of interests in a region implies the popularity of this region. We assume that the popularity is zonal-based attribute. The popularity is defined as number of reviews of entities within a zone. The average review number, median review number, minimum review number, maximum review number and standard deviation in zone i are defined as the average value, median value, minimum value, maximum value and standard deviation of the popularity, respectively, which are denoted by $r^i_{avg}$, $r^i_{med}$, $r^i_{min}$, $r^i_{max}$ and $r^i_{std}$, respectively.

4) Weather

We consider 7 categories of weather variables, including maximum temperature (measured by Fahrenheit degree), minimum temperature (measured by Fahrenheit degree), precipitation (measured by millimeters), average wind speed (measured by meters per second), snowfall (measured by inches), smoke or haze (dummy variable), heavy fog or heavy freezing fog (dummy variable).

All of the aforementioned variables have hourly values (i.e., variables are taken average per hour). And we assume that weather variables only have temporal dependencies (i.e., variables have same values across zones).

We denote these variables in different ways, since we have both numerical and dummy variables. Thus, maximum temperature, minimum temperature, precipitation, average wind speed and snowfall at tth time slot are denoted as $w_t,t_{max}$, $w_t,t_{min}$, $w_t,prcp$, $w_t,awnd$, $w_t,snow$, respectively.

And then, we introduce dummy variable $w_t,skhz$ to characterize the attribute of smoke or haze, given by:

$$w_{t,skhz} = \begin{cases} 
1, & \text{if smoke or haze happens at } t^{th} \text{ time slot} \\
0, & \text{otherwise}
\end{cases}$$

We also denote another dummy variable $w_t,hfog$ to be heavy fog:

$$w_{t,hfog} = \begin{cases} 
1, & \text{if heavy fog happens at } t^{th} \text{ time slot} \\
0, & \text{otherwise}
\end{cases}$$

5) Problem formulation

The prediction problem is to predict pick-up demand in next time interval (15 min) at each taxi zone using pick-ups, drop-offs, weather and popularity information. It can be formulated as:

$$\hat{P}_{t+1} = (P_s | s = t, t - 1, \ldots, t - m; D_s | s = t, t - 1, \ldots, t - m; R_s | z = 1, 2 \ldots, 63; W_s | s = t, t - 1, \ldots, t - m). (1)$$

Where $\hat{P}_{t+1}$ is the taxi pick-up demand prediction in next 15 minutes. m is the look-back time window. And $P$, $D$, $R$, $W$ represent pick-up demand, drop-off demand, popularity and weather, respectively.

2.2 Multi-source Information Based Spatiotemporal Neural Network (MSI-STNN)

This section presents the proposed deep learning architecture, i.e., MSI-STNN, to integrate spatiotemporal variables (i.e., pick-up and drop-off records) and exogenous information (i.e., weather and popularity) for short-term taxi demand forecasting. Specifically, the method is composed of 3 deep learning models: convolutional long short-term memory (ConvLSTM), long short-term memory (LSTM) and convolutional neural network (CNN), which are utilized to capture different characteristics. And then we concatenate these extracted features to get the prediction. We first give a brief introduction of CNN and LSTM, then present the adopted ConvLSTM, and finally illustrate the proposed architecture and training algorithm of MSI-STNN.

CNN and LSTM Models

A typical CNN architecture is shown in Fig. 1(a). Unlike a fully connected neural network in which the hidden activation H is computed by multiplying the entire input V and weights W, the CNN leverage convolution kernels to multiply a small local input (i.e., [v1, v2, v3]) against the weights W. And then the kernel moves to next local input (i.e., [v2, v3, v4]), which means the kernel is fixed and the weights...
W are shared across the entire input V. After computing the hidden units, a maxpooling layer with filters of pooling size (e.g., $2 \times 2$) outputs the maximum of activations in each filter, which means every MAX operation discards 75% of the activations if filter size is $2 \times 2$. Compared with traditional fully connected neural network, CNN can progressively reduce the number of parameters and avoid overfitting problem.

LSTM is actually a special recurrent neural network (RNN) architecture. Unfortunately, RNNs are not always capable of handling long-term dependencies, which is why the LSTM is widely used. LSTMs also have this chain-like structure, but the LSTM cell is quite special. In standard RNNs, each cell has only one single neural network layer, while there are 4 in each LSTM cell, shown in Fig. 1(b). As demonstrated in Eqs. (2)-(7), $f_t$, $i_t$, $C_t$, $o_t$ represent forget gate, input gate, memory cell and output gate, respectively, sharing the same dimension with $h_t$.

$$f_t = \sigma \left( W_f \cdot [h_{t-1}, x_t] + W_{cf} \odot C_{t-1} + b_f \right).$$  \hspace{1cm} (2) \\
$$i_t = \sigma \left( W_i \cdot [h_{t-1}, x_t] + W_{ci} \odot C_{t-1} + b_i \right).$$  \hspace{1cm} (3) \\
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$  \hspace{1cm} (4) \\
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t.$$  \hspace{1cm} (5) \\
$$o_t = \sigma \left( W_o \cdot [h_{t-1}, x_t] + W_{co} \odot C_{t-1} + b_o \right).$$  \hspace{1cm} (6) \\
$$h_t = o_t \odot \tanh(C_t).$$  \hspace{1cm} (7)

where $W_f$, $W_i$, $W_C$, $W_o$, $W_{cf}$, $W_{ci}$, $W_{co}$ are parameter matrices of weights which conduct a linear transformation, while $b_f$, $b_i$, $b_C$, $b_o$ are parameters of bias. It is noteworthy that $C_t$ is a parameter matrix which acts as an accumulator of the state information. Every time a new input comes, the information will be accumulated into the memory cell $C_t$ once the input gate $i_t$ is activated. Also, the past cell status $C_{t-1}$ could be forgotten if the forget gate $f_t$ is on. Whether the latest cell output $C_t$ will be propagated to the final state is further determined by the output gate $o_t$. The operator $\odot$ refers to the Hadamard product that conducts element-wise multiplication operation. $\sigma$ and $\tanh$ are two non-linear activation functions given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$  \hspace{1cm} (8) \\
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$  \hspace{1cm} (9)

Although LSTM layer has been proven to be quite suitable to handle the data with temporal characteristics, it lacks the ability to extract spatial information. To address this problem, we adopt the convolutional LSTM (ConvLSTM) which is an extension of LSTM. In order to capture the spatial information, all elements in LSTM, including input, output, hidden state, memory cells, input gate, output gate and forget gate, will be resized to 3D tensors in ConvLSTM whose last two dimensions (i.e., height and width) represent the spatial characteristics. We assume input as an “image” in each timestep. And then the spatiotemporal information of image flows in ConvLSTM cells. The future output is determined by both output and input from previous timestep in ConvLSTM. And this process can be achieved by leveraging convolution operator in the state-to-state and input-to-state transitions. So, we just need to replace the dot product operator (“ $\cdot$ ”) in LSTM to convolution operator (“ $\ast$ ”). The key equations are given in Eqs. (10)-(15) below:

$$f_t = \sigma \left( W_f \ast [h_{t-1}, x_t] + W_{cf} \odot C_{t-1} + b_f \right).$$  \hspace{1cm} (10) \\
$$i_t = \sigma \left( W_i \ast [h_{t-1}, x_t] + W_{ci} \odot C_{t-1} + b_i \right).$$  \hspace{1cm} (11) \\
$$\tilde{C}_t = \tanh(W_C \ast [h_{t-1}, x_t] + b_C).$$  \hspace{1cm} (12) \\
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t.$$  \hspace{1cm} (13) \\
$$o_t = \sigma \left( W_o \ast [h_{t-1}, x_t] + W_{co} \odot C_{t-1} + b_o \right).$$  \hspace{1cm} (14)
\[ h_t = \sigma_t \odot \tanh(C_t). \] (15)

The forget gate tensors, input gate tensors, memory cell tensors, output gate tensors, hidden state tensors, input tensors at timestep are denoted as \( f_t, i_t, c_t, \sigma_t, h_t, x_t \in \mathbb{R}^{N \times H \times W} \), respectively, where \( H \) and \( W \) (i.e., height and width) stand for spatial dimensions. And all the weights tensors, including \( W_f, W_i, W_c, W_o, W_{cf}, W_{ci}, W_{co} \), are fixed for each convolution kernel, which implies that the weights are shared when the kernel moves. We can extract the different spatial features (e.g., congestions and crowds) by using multi kernels. To make sure that the states have the same size of height and width as inputs, we fill none values with zeros when the kernel moves to the boundary, which is called the zero-padding.

![Fig. 1 illustration of typical CNN and LSTM. (a) Typical CNN architecture. (b) Standard LSTM cell.](image)

**MSI-STNN Model**

We propose multi-source information based spatiotemporal neural network (MSI-STNN) model to integrate multi sources data into our deep learning architecture. The structure of MSI-STNN is shown in Fig. 2. The stacked ConvLSTM layers is used to handle the spatiotemporal variables (i.e., pick-up and drop-off demand). And LSTM layers are implemented to deal with the temporal variables (i.e., weather), while CNN layers are adopted to extract the feature of popularity in each zone. After getting the encoded information from different sources, they are concatenated and two fully connected layers (i.e., dense layer) are used as decoder. Finally, the predicted pick-ups are obtained in next timestep (15 min).

1) **Structure of spatiotemporal variables**

The spatiotemporal variables (i.e., pick-up and drop-off demand) share the same training structure and they are handled separately. And each one is composed of stacked ConvLSTM layers, one batch normalization layer (BN) and one dropout layer (DO). The formulation of the architecture for spatiotemporal variables are given in Eq. (16) – (21).

\[
h_t = (p_{t,1}, ..., p_{t,i}).
\] (16)

\[
d_t = (d_{t,1}, ..., d_{t,i}).
\] (17)

\[
(H_{t-k}^{LP}, ..., H_{t-1}^{LP}) = \mathcal{F}_{LP} \circ \mathcal{F}_{BN} \circ \mathcal{F}_{ConvLSTM} \cdots \mathcal{F}_1(h_{t-k}, ..., h_{t-1}).
\] (18)

\[
\hat{v}_t^p = \sigma(w_H \cdot H_{t-1}^{LP} + b_H).
\] (19)

\[
(D_{t-k}^{LD}, ..., D_{t-1}^{LD}) = \mathcal{F}_{LD} \circ \mathcal{F}_{BN} \circ \mathcal{F}_{ConvLSTM} \cdots \mathcal{F}_1(d_{t-k}, ..., d_{t-1}).
\] (20)

\[
\hat{v}_t^d = \sigma(w_D \cdot D_{t-1}^{LD} + b_D).
\] (21)

where \((h_{t-k}, ..., h_{t-1}), (d_{t-k}, ..., d_{t-1})\) are the input vectors of pickup and dropoff, respectively. And \(\hat{v}_t^p, \hat{v}_t^d\) are the encoded output vector of pickup and dropoff, respectively. \(k\) is the look-back time window. \(i\) is the number of zones. \(LP, LD\) are the last layer of ConvLSTM for pickup and
and dropout, respectively. And \( w_H, w_D, b_H, b_D \) are the weights and bias parameters. \( \sigma \) is the sigmoid function which has been defined in Eq. (8).

(2) Structure of temporal variable

The weather condition has no difference in the research area, which implies weather is a non-spatial variable. But it changes as the time goes by, which means it’s a temporal variable. The change of weather conditions, especially some extreme weather conditions (e.g., snowstorm and rainstorm), would affect the demand of pickup. A sequence of vector is defined: \( p_t = (w_{t,max}, w_{t,min}, w_{t,prcp}, w_{t,awnd}, w_{t,snow}, w_{t,skhz}, w_{t,hfog}) \). This vector is fed into the stacked LSTM architecture, which is transferred to the encoded weather information \( \tilde{V}_t^{wa} \).

\[
\begin{align*}
(p_l^{LSTM}_{t-k}, ..., p_l^{LSTM}_{t-1}) = \mathcal{F}_{1}^{LSTM}(p_l^{LSTM}, ..., p_l^{LSTM}).
\end{align*}
\]

Where \( k \) is the look-back time window, while \( L_w \) is the last layer of stacked LSTM architecture. \( p_l^{LSTM}, ..., p_l^{LSTM} \) stands for the output of the last layer. \( w_p, b_p \) are the parameters of weights and bias.

(3) Structure of spatial variable

The point of interests (POI) in the studied area is the spatial variable. The reason is that the number of POIs reflects the popularity in the zone. And a CNN architecture is applied to extract the information of popularity. The formulation is given below:

\[
\begin{align*}
\mathcal{F}_{2}^{CONV} \mathcal{F}_{1}^{MAXPOOL} \mathcal{F}_{2}^{CONV} \mathcal{F}_{1}^{MAXPOOL} (r_i^1, ..., r_i^l).
\end{align*}
\]

Where \( r_i^1, r_i^2, r_i^3, r_i^4, r_i^5 \) are the average value, median value, minimum value, maximum value and standard deviation of the popularity in zone \( i \), respectively. And \( w_R, b_R \) are the weights and bias, respectively.

(4) Information fusion

The encoded information from different sources are concatenated together, and then decoded by deploying two dense layers to get the ultimate output. The prediction of pick-up demand at \( t \)th time interval is:

\[
\begin{align*}
\tilde{V}_t &= (\tilde{V}_t^p, \tilde{V}_t^wa, \tilde{V}_t^w). \\
\tilde{V}_{t, ultimate} &= \mathcal{F}_{2}^{DENSE} \mathcal{F}_{1}^{DENSE} (\tilde{V}_t).
\end{align*}
\]

where \( \tilde{V}_t \) is the concatenated vector, while \( \tilde{V}_{t, ultimate} \) is the ultimate prediction at \( t \)th time interval.

Fig. 2 Framework of the proposed MSI-STNN model.
(5) Objective function

For the training process, mean square error (MSE) between prediction and actual demand is used as loss function, which is given by:

\[
\text{loss} = \frac{1}{n} \sum_{i=1}^{n} (\hat{V}_{t, \text{ultimate}} - V_{t, \text{actual}})^2.
\]  \hfill (29)

And then error can be reduced by learning the weights and bias through back propagation. The training steps in Algorithm 1.

Algorithm 1. MSI-STNN training

**Input**
- Observations of pick-up demand \{P1, ..., Pn\} in training set
- Observations of drop-off demand \{D1, ..., Dn\} in training set
- Observations of weather \{W1, ..., Wn\} in training set
- Observations of popularity \{R1, ..., Rn\} in training set
- Input time step: K + 1
- Input zone id: i

**Output**
- MSI-STNN with learnt parameters

**Procedure**

1. Initialize a null set: V \(\leftarrow \emptyset\)
2. for all available time intervals t (1 ≤ t ≤ n) do
3. \(\mathcal{P}_t^p = [h_{t-k}, ..., h_{t-1}]\)
4. \(\mathcal{P}_t^d = [d_{t-k}, ..., d_{t-1}]\)
5. \(\mathcal{P}_t^w = [p_{t-k}, ..., p_{t-1}]\), where \(p_t = (w_{t, \text{tmax}}, w_{t, \text{tmin}}, w_{t, \text{precP}}, w_{t, \text{awnd}}, w_{t, \text{snow}}, w_{t, \text{skh}}, w_{t, \text{hfo}})\)
6. \(\mathcal{P}_t^r = [r^i, ..., r^j]\), where \( r^i = (r_{\text{avg}}, r_{\text{med}}, r_{\text{min}}, r_{\text{max}}, r_{\text{std}})\)
7. A training sample \((\mathcal{P}_t^p, \mathcal{P}_t^d, \mathcal{P}_t^w, \mathcal{P}_t^r)\) is put into V
8. end for
9. Initialize all the weights and bias parameters
10. repeat
11. Randomly extract a batch of samples \(V^b\) from V
12. Update the parameters by minimizing the objective function shown in Eq. (29) within \(V^b\)
13. until convergence criterion met
14. end procedure

### 3. Case Study

#### 3.1. Study site and Dataset

The study site is located in Manhattan, NY, which has been partitioned into 63 taxi zones. The zones with ID: 103, 104, 105, 153, 194 and 202 were not considered, where the taxi demands are most zeros (shown in Fig. 3). The datasets utilized in this study are collected from multi sources, which are described explicitly as follow:

1. Pick-up and drop-off requests

The taxi pick-up and drop-off requests of these 63 zones are extracted from NYC Taxi & Limousine Commission (TLC) during the period from January 1st, 2017 to March 31st, 2018. The dataset contains about millions of yellow taxi requests in total, and each of them includes pick-up travel time, drop-off travel time, pick-up location ID and drop-off location ID. We first aggregate the dataset by each time interval (i.e., 15 minutes), and then the demand data in a zone in each time interval was obtained by summing up the requests from that zone during that time interval. This dataset was further separated
into pick-up and drop-off datasets. To avoid using future information, both pick-up and drop-off datasets is partitioned into 80% training set comprised of requests between January 1st, 2017 and December 31st, 2017, and 20% testing set containing the rest of observations from January 1st, 2018 to March 31st, 2018. Fig. 4(a) shows the daily pick-up records in January 2018, which is clear that the pick-up demand in January 4th, 2018 is extremely low, and the reason is that there was a snowstorm during that day, which may have a great impact on the transportation. Fig. 4(b) shows the spatial distribution of pick-up demand on January 26th, 2018 between 6 p.m. and 7 p.m., from which we can see that most of the pick-up requests increased in the middle part of Manhattan. These spatiotemporal characteristics is a great challenge for the taxi short-term demand forecasting, which inspires us to introduce POI and weather datasets to obtain a better prediction.

(2) Popularity

The yelp Fusion API is used to extract the number of historical reviews in each point of interest. In total, 11,760 POIs were extracted covering most of the business areas in Manhattan. Then the review counts were merged in each taxi zone to obtain the average, median, maximum, minimum and standard error values to represent the popularity of each zone. The popularity is assumed to be stationary during the studied period.

(3) Weather

The weather data is collected from National Oceanic and Atmospheric Administration (NOAA) website which provide hourly aggregated weather information from Central Park, which is the best station to obtain the weather data in Manhattan. The information includes hourly maximum temperature, minimum temperature, precipitation, average wind speed and snowfall.
Fig. 4 samples of temporal and spatial characteristics of taxi demand. (a) Daily pick-up demand in January 2018. (b) Spatial distribution of pick-up demand during 6-7 pm on Jan. 26th, 2018.

3.2. Performance Diagnostics

The proposed MSI-STNN with multi-source data is trained on the training set and validated on the testing set. The training process of this model is illustrated in Algorithm 1. In our study, the proposed model includes four ConvLSTM layers for spatiotemporal features extraction (i.e., pick-up and drop-off requests), three CNN layers for spatial feature extraction (i.e., POI) and three LSTM layers for temporal feature extraction (i.e., weather). The number of training epoch is set to be 100, while the batch size is set to be 32.

Fig. 5(a) shows the temporal comparison of first 1,000 timesteps in the test dataset between ground truth pick-up demand and predicted results by MSI-STNN, where the red color represents ground truth pick-up demand and green color implies prediction. It is obvious that the prediction results match the actual data very well. Moreover, the accuracy of demand forecasting exists great discrepancy across zones, which is represented by the heat map of error distribution (i.e., subtraction of prediction and ground truth), where deeper color implies a larger error as shown in Fig. 5(b).
To illustrate the value of multi-source information in prediction, we developed three models for comparison: the LSTM model with only pick-up request data, CNN model with only pick-up request data and traditional ARIMA method. We evaluate the models via two measurements of effectiveness: mean squared error (MSE) and mean absolute error (MAE), formulated as:

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2. \]  \hspace{1cm} (30)

\[ MAE = \frac{1}{N} \sum_{i=1}^{N} |Y_i - \hat{Y}_i|. \]  \hspace{1cm} (31)

Where \( Y_i \) and \( \hat{Y}_i \) are the \( i \)th ground truth and predicted values for all the zones, respectively. And \( N \) is the size of testing set.

Table 1 lists the predictive performance of the proposed model and two benchmark models on the testing set. It can be found that the proposed MSI-STNN outperforms the benchmarks in the three measurements of predictive performance, which validates the importance of multi-source data in demand prediction.

![Fig. 5 Prediction results by the MSI-STNN model. (a) Temporal prediction outcome of first 1,000 timesteps. (b) average error distribution across zones.](image-url)
Table 1. Predictive Performance Comparison

| Model    | MSE    | MAE    |
|----------|--------|--------|
| ARIMA    | 147.230| 12.603 |
| CNN      | 121.357| 7.534  |
| LSTM     | 126.304| 12.202 |
| MSI-STNN | 118.320| 6.668  |

3.3. Sensitivity Analysis

Sensitivity analysis of the MSI-STNN was performed to further validate the model performance. Three parameters are investigated, including training epoch, layers of structure and look-back time window.

Firstly, to investigate the impact of the training epoch on the predictive performance, the epoch is increased from 10 to 100 and the validation error (MSE) after each training epoch is recorded for the proposed model. From Fig. 6(a), we can see that the MSE of MSI-STNN decreases slowly in the initial 60 epochs and then drops sharply to around 130, which indicates that the MSI-STNN model has a low learning speed since the model structure is complicated (i.e., four parallel submodules) and the dataset is large (i.e., one-year training set and three months testing set). In the training process, it takes around 100 seconds per epoch. The MSE reduces slightly from epoch 60 and epoch 100, changing from around 130 to 118, while the computation time increases by 4,000 seconds. Considering about the trade-off between performance and computation time, the best training epoch for this model is around 60, which sacrifices a little predictive performance to obtain high-efficient computation resources.

Once obtaining the best training epoch, various layers of structures were tested. The ConvLSTM layers of pick-up and drop-off submodules were changed simultaneously. Fig. 6(b) shows that the MSE drops and then rises as the increase of number of ConvLSTM layers. This result is intuitive: the MSI-STNN with a higher network depth could face the overfitting problem, which cause the model performance decrease when the number of ConvLSTM layers is greater than 3.

Different look-back time windows are tested in the model as shown in Fig. 6(c). It shows that the MSE reduces slightly as the increase of look-back time window, and it fluctuates after a certain look-back time window. This could be explained that a longer look-back time window can achieve a better predictive performance while it may also cover useless information that deteriorate the model performance.

Fig. 6 Sensitivity analysis of MSI-STNN. (a) Training epoch. (b) ConvLSTM layers. (c) look-back time window.
4. Conclusion
This paper proposes a novel deep learning approach, multi-source information based spatiotemporal neural network (MSI-STNN), to predict short-term taxi pick-up demand. The architecture has three submodules, including convolutional long short-term memory (ConvLSTM), long short-term memory (LSTM) and convolutional neural network (CNN). ConvLSTM captures spatiotemporal characteristics of pick-up and drop-off demand, while CNN and LSTM extracts spatial and temporal information of zonal popularity and weather information.

The model performance is validated using Taxi data in Manhattan, New York by comparing with the state-of-art time series and deep learning approaches, including ARIMA, LSTM and CNN. The proposed MSI-STNN outperforms the benchmark algorithms in the measurements of MSE and MAE, which implies the value of multi-source information in demand prediction. The sensitivity analysis highlights that the proposed model is robust against the change of hyperparameters. Future work will explore advanced deep learning architectures to fuse multi-source information as well as mutually constraints among zonal pick-ups and drop-offs.

Acknowledgments
This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, “This work was supported in part by the U.S. Department of Commerce under Grant BS123456.”

The next few paragraphs should contain the authors’ current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrnim.go.jp).

References
[1] P.J. Harrison and C.F. Stevens. “Bayesian Forecasting,” Journal of the Royal Statistical Society. Series B (Methodological), Vol. 38, No. 3, 1976, pp. 205-247.
[2] M.S. Ahmed and A.R. Cook. “Analysis of freeway traffic time-series data by using box-jenkins techniques,” 1979.
[3] I. Okutani and Y.J. Stephanedes. “Dynamic prediction of traffic volume through Kalman filtering theory,” Transportation Research Part B, Vol. 18, No. 1, 1984, pp. 1–11.
[4] P.M. Yelland. “Bayesian forecasting of parts demand,” International Journal of Forecasting, 2010, pp. 374-396.
[5] B.M. Williams and L.A. Hoel. “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results,” J. Transp. Eng. Vol. 129, No. 6, 2003, pp. 664–672.
[6] J. Guo, W. Huang and B.M. Williams. “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” Transportation Research Part C: Emerging Technologies. Vol. 43, No. 1, 2014, pp. 50–64.
[7] W.S. McCulloch and W.H. Pitts. “A logical calculus of the ideas immanent in nervous activity,” The Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp. 115-133.
[8] L. Breiman. “Random Forest,” Machine Learning, Vol. 45, No. 1, 2001, pp. 5-32.
[9] L. Zhang, Q. Liu, W. Yang, N. Wei and D. Dong. “An Improved K-nearest Neighbor Model for short-term Traffic Flow Prediction,” Procedia-Social and Behavioral Sciences, Vol. 9, No. 6,
2013, pp. 653-662.

[10] E.I. Vlahogianni, M.G. Karlaftis and J.C. Golias. “Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach,” Transportation Research Part C, Vol. 13, No. 3, 2005, pp. 211–234.

[11] S. Hochreiter and J. Schmidhuber. “Long Short-term Memory,” Neural Computation, Vol. 9, No. 8, 1997, pp. 1735-1780.

[12] J. Chung, C. Gulcehre, K.H. Cho and Y. Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” 2014.

[13] X. Ma, Z. Tao, Y. Wang, H. Yu and Y. Wang. “Long Short-term Memory Neural Network for Traffic Speed Prediction using Remote Microwave Sensor Data,” Transportation Research Part C: Emerging Technologies, Vol. 54, 2015, pp. 187-197.

[14] H. Yu, Z. Wu, S. Wang, Y. Wang and X. Ma. “Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks,” Sensors, Vol. 17, No. 7, 2017, pp. 1501.

[15] N.G. Polson and V.O. Sokolov. “Deep Learning for Short-term Traffic Flow Prediction,” Transportation Research Part C: Emerging Technologies, Vol. 79, 2017, pp. 1-17.

[16] X. Shi, Z. Chen, H. Wang and D.Y. Yeung. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” 2015.

[17] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang and Y. Wang. “Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction,” Sensors, Vol. 17, No. 4, 2017, pp. 818.

[18] J. Ke, H. Zheng, H. Yang and X. Chen. “Short-term Forecasting of Passenger Demand Under On-demand Ride Services: A Spatio-temporal Deep Learning Approach,” Transportation Research Part C, Vol. 85, 2017, pp. 591-608.

[19] J. Bao, P. Liu and S.V. Ukkusuri. “A Spatiotemporal Deep Learning Approach for Citywide Short-term Crash Risk Prediction with Multi-source Data,” Accident Analysis and Prevention, Vol. 122, 2019, pp. 239-254.

[20] K. Zhang, Z. Liu and L. Zheng. “Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning,” IEEE Transactions on Intelligent Transportation Systems, 2019, pp. 1-11.

[21] S. Liao, L. Zhou, X. Di, B. Yuan and J. Xiong. “Large-Scale Short-Term Urban Taxi Demand Forecasting using Deep Learning,” 23rd Asia and South Pacific Design Automation Conference, 2018, pp. 428-433.

[22] N. Mukai and N. Yoden. “Taxi Demand Forecasting Based on Taxi Probe Data by Neural Network,” Intelligent Interactive Multimedia: Systems and Services, SIST 14, 2012, pp. 589-597.