

A Deep Spatio-Temporal Fuzzy Neural Network for Passenger Demand Prediction

Xiaoyuan Liang, Guiling Wang* Martin Renqiang Min[†] Yi Qi[‡] Zhu Han[§]

Abstract

In spite of its importance, passenger demand prediction is a highly challenging problem, because the demand is simultaneously influenced by the complex interactions among many spatial and temporal factors and other external factors such as weather. To address this problem, we propose a Spatio-TEmporal Fuzzy neural Network (STEF-Net) to accurately predict passenger demands incorporating the complex interactions of all known important factors. We design an end-to-end learning framework with different neural networks modeling different factors. Specifically, we propose to capture spatio-temporal feature interactions via a convolutional long short-term memory network and model external factors via a fuzzy neural network that handles data uncertainty significantly better than deterministic methods. To keep the temporal relations when fusing two networks and emphasize discriminative spatio-temporal feature interactions, we employ a novel feature fusion method with a convolution operation and an attention layer. As far as we know, our work is the first to fuse a deep recurrent neural network and a fuzzy neural network to model complex spatial-temporal feature interactions with additional uncertain input features for predictive learning. Experiments on a large-scale real-world dataset show that our model achieves more than 10% improvement over the state-of-the-art approaches.

1 Introduction

Accurate future passenger demand prediction is very important in the field of transportation. Knowing the future demands, a Transportation Network Company (TNC) can wisely pre-allocate resources (vehicles and drivers) to meet the demands, such that the best service can be provided to passengers with a minimum waiting time, and unnecessary driving around on road can be prevented, reducing energy consumption and traffic jam. However, passenger demand prediction is very challenging considering the future demands are simul-

taneously influenced by many factors, including continuous spatial and temporal factors, as well as many discrete external factors, such as weather and being daytime or nighttime. These factors have complex and non-linear interactions with future demands and capturing the interactions in one model to make prediction is very difficult. Moreover, data of external factors are often either inaccurate or too coarse due to data-collection sensors' sparse deployment and unavoidable errors.

To predict passenger demands in the near future, previous works have proposed various models. One of the most well-known methods uses the Auto-Regressive Integrated Moving Average (ARIMA) [1, 2], but it only considers the temporal feature interaction. Recent studies [3, 4] propose deep learning models considering both temporal and spatial feature interactions, which outperform previous methods considering only one type of factor. However, they handle the spatial and temporal feature interactions sequentially, resulting in information loss. In addition, they ignore other important discrete external factors, e.g., the weather. Although the model proposed in [5] considers the weather impact, it fails to consider inaccuracies within the collected data.

To tackle this challenging problem with desirable performance, we propose a deep Spatio-TEmporal Fuzzy neural Network (STEF-Net) to predict the passenger demands for a city area. In our network, we fuse all related factors to model the complex interactions among them, including spatial-temporal dependencies, external information and temporal relevance, and design an end-to-end learning framework with different neural networks modeling different types of feature interactions. Specifically, our model simultaneously captures the spatial and temporal dependencies via a Convolutional Long Short-Term Memory network (ConvLSTM). A ConvLSTM replaces the full connections in a traditional LSTM with convolutional operations such that the spatial feature interactions can be simultaneously captured and information loss can be avoided compared to sequential processing by stacking convolutional layers and LSTMs. Regarding the uncertain accuracy of external factors, we propose to model them via a fuzzy neural network. A fuzzy neural network, which combines the

*{xl367, gwang}@njit.edu, New Jersey Institute of Technology, Newark, NJ.

[†]renqiang@nec-labs.com, NEC Laboratories America, Inc., Princeton, NJ.

[‡]yi.qi@tsu.edu, Texas Southern University, Houston, TX.

[§]zhan2@uh.edu, University of Houston, Houston, TX.

fuzzy theory and neural networks, can learn the feature representation with high error tolerance and trainable rules. It shows significantly better performance than deterministic neural networks for this data type. We design a new feature fusion method using a convolution operation to connect the two separate networks without losing temporal information. We further propose to capture the temporal relevance of the high-level fused data via an attention layer, considering the future demands are unequally influenced by past ones. Our model is evaluated on the real data from Didi Chuxing, the biggest TNC in China, which is similar to Uber in the United States. The experimental results show that our model outperforms the state-of-the-art models.

In summary, our contributions include: (1) To the best of our knowledge, our work is the first to combine a fuzzy neural network and deep learning techniques to handle data uncertainty and learn complex interactions among multiple factors, which can achieve better performance than solely using deep learning. (2) We propose a new feature fusion method using a convolution operation, which can preserve the temporal relations of the outputs, capture the spatial information and achieve better performance than commonly used weighted addition. (3) We adopt an attention layer on every time step to provide explainable results on when the historical information influences most in the prediction and how the weather can influence the prediction. (4) Extensive experiments are conducted on real data to evaluate our model. Our model significantly outperforms state-of-the-art models in prediction accuracy.

2 Literature Review

2.1 Deep learning and fuzzy learning Deep learning [6] has been successfully applied in many fields, such as computer vision and intelligent transportation [7]. Among all deep learning techniques, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are two popular models. CNNs [8] are often deployed to model data with spatial feature interactions while RNNs are often used to process data with temporal feature interactions. A special RNN called LSTM is widely adopted to overcome the vanishing gradient problem in traditional RNNs [9]. However, neither CNNs nor LSTMs are perfect models for addressing spatial-temporal problems.

To handle imperfect data, fuzzy learning is a powerful tool and shows better performance than deterministic methods [10]. Combining fuzzy theory and neural networks can improve complex data representation with probability distribution over cross-layer units [11]. Even though they have been widely applied in control systems [11] and portfolio management [12], no existing work ap-

plies fuzzy neural networks in demand prediction.

2.2 Passenger demand/traffic flow prediction

Passenger demand prediction is closely related to traffic flow prediction. Both have the same-format of data and are influenced by external factors. We review both in this section. Traditional approaches to predict future passenger demands only consider temporal information, such as ARIMA [1, 2] or ANN [13].

Recent advances in deep learning [6] motivate researchers to apply deep learning techniques for passenger demand and traffic flow prediction. Recent studies employ CNNs to capture complex spatial feature interactions [4] or RNNs (including LSTMs) to capture temporal feature interactions [14, 15]. Pioneering works combine CNNs and RNNs to capture both spatial and temporal feature interaction in the data recently. Yao et al. [16] propose a multi-view model, which employs a CNN and an LSTM to capture the spatial and temporal feature interactions sequentially, but not simultaneously, which can potentially lead to temporal information loss. The above works either captures only one of the spatial and temporal feature interactions or captures both sequentially. None of the methods fully captures spatial-temporal feature interactions simultaneously.

ConvLSTMs [17] are another deep learning model, which combines CNNs and LSTMs. A ConvLSTM can simultaneously capture the spatial and temporal feature interactions. It replaces the fully connected layer in the traditional LSTM with a convolutional layer, which shows better performance than the traditional LSTM in precipitation nowcasting. A follow-up work [5] uses ConvLSTMs to predict the passenger demands. The model is composed of a ConvLSTM and a LSTM to process the weather information, the travel time rate and demand intensity and simply fuses the results from the two networks. However, The model fails to consider the inaccuracy of external data and the inaccuracy of simple prediction result fusion. AttConLSTM, a multi-step model built upon the attention-based encoder-decoder framework for passenger demand prediction is proposed in [3]. However, it fails to consider external factors, which greatly influences passenger demands. Different from all previous models, our model captures temporal-spatial feature interactions simultaneously without information loss, employs fuzzy neural network to handle external data inaccuracy and includes a new and effective feature fusion method based on convolutions.

3 Problem Formulation

Being consistent with existing works, we make the following definitions. Based on the definitions, we present the problem statement of this paper.

DEFINITION 3.1. Region In this paper, we predict the passenger demands in different areas in a city. The whole city is partitioned into $W \times H$ equal-size grids. A grid is called a region, which is denoted by r . Let $r_{i,j}$ denote the region with the coordinate i, j , where $i \in [0, W)$ and $j \in [0, H)$.

DEFINITION 3.2. Service request A service request s_k made by a passenger is composed of the request ID, pick-up coordinates $s_{k,pc}$ (longitude and latitude), and pick-up time $s_{k,pt}$. (Note that we do not consider drop-off location and time for demand predictions.) A service request $s_k = \{s_{k,pc}, s_{k,pt}\}$. A valid request's pick-up location should be in the city. If it is outside the city, it is discarded. The total number of available legal requests is denoted by N .

DEFINITION 3.3. Passenger demands Time is divided into equal intervals. The t^{th} time interval, starting from 0, is the interval of $[t \times C, (t+1) \times C)$, where C is a constant representing the interval's time span. The passenger demand of a region $r_{i,j}$ is accumulated in the specific t^{th} time interval based on the requests' pick-up time. The passenger demand in region $r_{i,j}$ at the t^{th} time interval is denoted by $d_{i,j}^t$.

$$(3.1) \quad d_{i,j}^t = |\{k \in [0, N) : s_{k,pc} \in r_{i,j} \wedge s_{k,pt} \in [t \times C, (t+1) \times C)\}|.$$

The whole area's demands are denoted by D , which means

$$(3.2) \quad D^t = \{d_{i,j}^t | \forall i \in [0, W), \forall j \in [0, H)\}.$$

We can imagine that D^t is a demand snapshot of the whole area at the t^{th} time interval, where every pixel is the demand of that particular location.

DEFINITION 3.4. External information Let e^t denote the external information set at the t^{th} time interval. The external factors impacting the passenger demands considered in the paper includes the weather, the day in a week and being daytime or nighttime. The process details of the external information will be given in next section.

Problem statement The problem is defined as follows. Suppose the current time interval is t . Given the historical passenger demands and the external information at the t^{th} time interval, our goal is to predict the passenger demands in all regions in the city at the $(t+1)^{\text{th}}$ time interval. Specifically, in our problem, we take the historical data, demands and external information in the last k time intervals as input and the output is the predicted passenger demands at the $(t+1)^{\text{th}}$ time

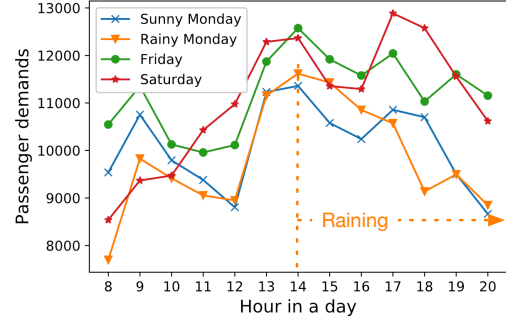


Figure 1: Different hourly demands in different days

interval. Let \hat{D}^{t+1} denote the predicted passenger demands at the $(t+1)^{\text{th}}$ time interval. \hat{D}^{t+1} is a function f of the previous k time intervals' data.

$$(3.3) \quad \hat{D}^{t+1} = f(D^{t-k}, D^{t-k+1}, \dots, D^t, e^{t-k}, e^{t-k+1}, \dots, e^t).$$

Our goal is to minimize the difference between \hat{D}^{t+1} and the true passenger demands D^{t+1} .

4 STEF-Net

4.1 Preliminary analysis In this section, we conduct a preliminary data analysis to provide some intuition on how passenger demands are influenced by different factors. We use a dataset from Didi Chuxing, China. The data contains over 5.24 million service requests from 11/01/2016 to 11/30/2016. Fig. 1 shows the total passenger demands over different hours in different days. We pick two Mondays and a Friday as representatives of weekdays and a Saturday as that of weekends. We select a rainy Monday and a sunny Monday to show the impact of weather on passenger demands. Except the rainy Monday, all the other days are sunny. In the figure, the x axis is the hour of the day and the y axis is the passenger demands during the hour. We can see the passenger demands have different patterns in different days, at different time of the day, and under different weathers. For example, at the noon time, the demand drops on Monday and Friday but it increases on Saturday. About the weather factor, on the rainy Monday, it starts to rain at 2pm. Comparing the sunny Monday with the rainy Monday, we can see that the patterns of passenger demands on the two days before rain are similar while the patterns become different after rain comes. Specifically, the number of passenger demands keeps decreasing on rainy Monday while there is an increase on sunny Monday. The figure shows the passenger demands are determined by complex interactions among many factors.

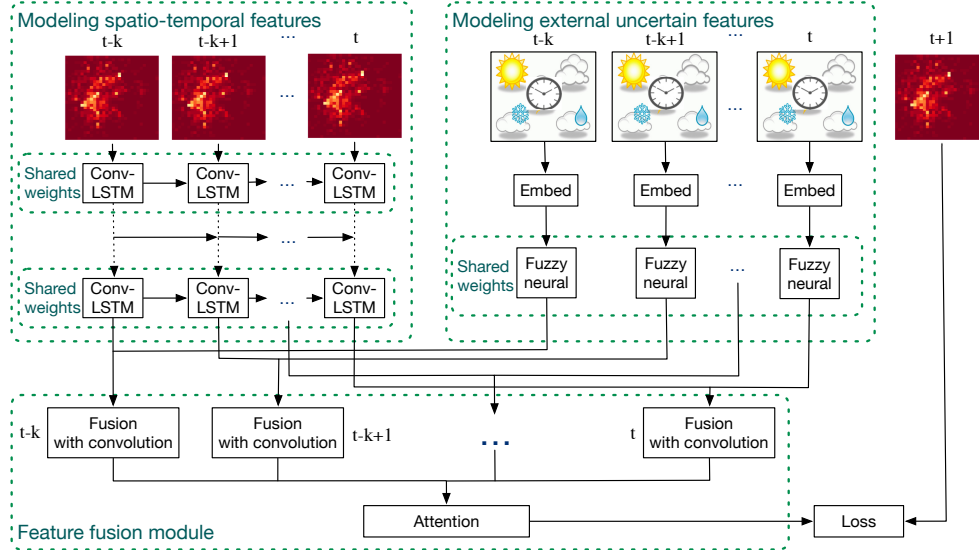


Figure 2: Our deep learning model STEF-Net

4.2 Overview of our deep learning model We propose a deep learning model, STEF-Net, to predict the passenger demands incorporating the complex interactions between various factors. Our model is illustrated in Fig. 2, which is mainly composed of four components. (1) As shown in the left side of the figure, we employ a stacking ConvLSTM to capture the *spatio-temporal feature interaction* with the passenger demands. The input is the historical passenger demands with location and time information, and the output is the prediction using only spatial-temporal information. (2) As shown in the right side of the figure, in parallel, we employ a fuzzy neural network to capture the *external information's interactions* on passenger demands. The input is the data about external information and the output is the prediction using external information. (3) As shown on the bottom of the figure, the outputs from the stacked ConvLSTMs and the fuzzy neural network are fused into one network to generate the final output. We propose a new feature fusion method using convolution on the data from the same time period, which keeps the temporal relation of the outputs from the two networks. Considering the passenger demands are unequally influenced by different time intervals, we further propose to adapt an attention layer on the high-level fused data to capture the temporal relevance. The data are then reshaped into the output format, which matches the regions in a city. (4) As shown in the right corner of the figure, we employ a loss function to measure the difference between the predicted value and true value. A neural network's goal is to minimize the loss defined by an objective function. In the following, every

component is presented in detail.

4.3 Modeling spatio-temporal features We propose to simultaneously capture the deep spatial and temporal dependencies in passenger demands by stacking ConvLSTMs. A ConvLSTM is a neural network model that combines convolutional operation and LSTM units, where an LSTM is known to well handle temporal feature interaction without the vanishing gradient problem, while convolutional networks are known to gracefully handle spatial feature interactions. A ConvLSTM uses the convolution operation to replace the full connections in traditional LSTMs. In an LSTM, all the elements are 1D tensors, which accepts the input from $T \times L$ dimensions and generates outputs into $T \times L'$ dimensions. T is the length of the time sequences, L is the length of one input vector, and L' is the length of one output vector. A ConvLSTM transfers all the inputs, memory cell values, hidden states, and various gates in an LSTM into 3D tensors, where the first two dimensions are considered as the spatial information, rows and columns and the last dimension is the channels.

To adapt ConvLSTM in our problem, we treat the first two dimensions in the passenger demand data as rows and columns at one time interval. The ConvLSTM in our problem can be considered as a function $R^{T \times W \times H \times L} \rightarrow R^{T \times W \times H \times L'}$, where T , L and L' are the same as those in the traditional LSTM, and W and H are the length of rows and columns, corresponding to the width and height of the grids in the city-wide area in our problem. Several ConvLSTM layers are stacked in our neural network. In the last ConvLSTM layer, the

length of the output vector is set 1, which means the output is a 4D tensor with the size of $T \times W \times H \times 1$. This is equivalent to a 3D tensor with the size of $T \times W \times H$.

4.4 Modeling external uncertain features Passenger demands are influenced by many external factors in addition to location and time which are handled in the previous section. We need to identify the factors and obtain corresponding data to train our model. Note that existing data about external factors are likely inaccurate or of coarse granularity, and thus we propose for the first time to employ a fuzzy neural network to model the data. In the section, we first present our factor selection and data pre-processing, and then we present our fuzzy neural network modeling.

4.4.1 Factor selection and data pre-processing

Our model aims to include all highly correlated factors and chooses (1) the weather, (2) the day in a week, and (3) being daytime or nighttime. Obviously, weather greatly influences a passenger’s choice between taking uber and walking, waiting and then taking a bus. When it rains heavily or when it is very cold or very hot, people tend to take a more comfortable way of transportation. In our model, the weather is represented by temperature, dew point, humidity, pressure, wind speed, and weather condition. The first five variables are numerical variables. The last one, weather condition, is represented by ten different categories: clear, partly cloudy, scattered cloud, mostly cloudy, haze, light rain, shower, mist, patches of fog, and fog. The categories are indexed from 1 to 10. The numbers are embed using one-hot vectors with 10 dummy variables. The second factor, the day in a week, refers to the seven days in a week, which also affluences people’s daily transportation behavior. This factor is a categorical variable and is represented by a one-hot vector with 7 dummy variables. The third factor is whether it is dark outside and it is differentiated by the sunrise and sunset time of the day. In summary, to represent all the three factors, we choose 24 variables in total in our model. The external information is 1D tensor.

4.4.2 Fuzzy neural network modeling

We adopt a fuzzy neural network to learn the representation of the external information. The fuzzy neural network is composed of two hidden layers, membership function layer and logic rule layer. The architecture is shown in Fig. 3. The membership function layer calculates the degree that an input node belongs to a certain fuzzy set. Let x_i denote the i^{th} element in the input. In the membership function layer, every element is

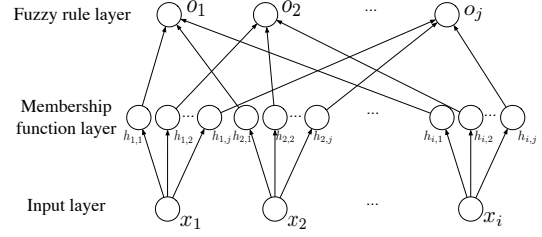


Figure 3: Illustration of the fuzzy neural network

split by multiple Gaussian distributions. Let j index the Gaussian distribution for the i^{th} element. One distribution is denoted by $(\mu_{i,j}, \delta_{i,j})$. The membership function layer’s output is calculated as follows,

$$(4.4) \quad h_{i,j} = e^{-\frac{(x_i - \mu_{i,j})^2}{\delta_{i,j}^2}}$$

The logic rule layer performs the “AND” fuzzy logic operation as follows,

$$(4.5) \quad o_j = \prod_i h_{i,j}$$

Through the rule layer, the output can present the probability that it is related to every unit. In the fuzzy neural network, all time intervals’ external information shares the same member function layer and logic rule layer.

The output from the fuzzy neural network of one interval’s external information is reshaped into two dimensions to match the passenger demands in a whole city, which is $W \times H$. The outputs from the past T time intervals’ external information are a 3D tensor with the shape size of $T \times W \times H$.

4.5 Feature fusion module In the above two sections, the passenger demand data and external information are mapped into the same feature representation format from two separate networks. They need to be combined to predict the next time interval’s demand.

Previous works employ weighted addition [5, 4] to fuse the two components. In this paper, we propose to use convolutional operation to fuse the outputs from two networks. In addition, to keep the temporal feature interaction, we first fuse the data from the same time interval and then apply an attention layer to generate the final output.

The outputs of the passenger demand and external information at the t^{th} are denoted by $\mathbf{O}_{t,p}$ and $\mathbf{O}_{t,e}$, respectively. Let $\mathbf{O}_{t,f}$ denote the output after fusion with convolution denoted by \oplus . The illustration of fusion with convolution is shown in Fig. 4. The calculation can be presented by the following equation,

$$(4.6) \quad \mathbf{O}_{t,f} = \mathbf{O}_{t,p} \oplus \mathbf{O}_{t,e}$$

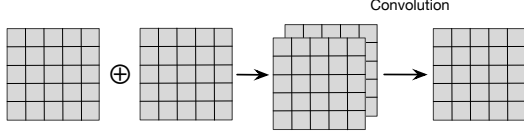


Figure 4: The fusion method with convolution

We concatenate the two by adding a new dimension, which can be imaged as the channel in a CNN. After concatenation, $\mathbf{O}_{t,f} \in R^{W \times H \times 2}$. To make the output's dimension consistent, we apply a convolutional operation with window size $w \times h$ ($w \ll W$ & $h \ll H$) with 1 channel, which outputs a 2D tensor with the size of $W \times H$. In this way, our fusion with convolution method only needs $w \times h$ parameters while the weighted addition requires $2 \times W \times H$ parameters in previous works [5, 4]. In addition, the convolutional operation can further learn the spatial information on the fused data.

After fusing the data, we use a bidirectional LSTM and attention to further capture the temporal relevance. In the bidirectional LSTM, the data are flattened into one dimension and fed into LSTMs:

$$(4.7) \quad \vec{\mathbf{h}}_t = \overrightarrow{LSTM}(\mathbf{w}_t, \vec{\mathbf{h}}_{t-1}),$$

$$(4.8) \quad \overleftarrow{\mathbf{h}}_t = \overleftarrow{LSTM}(\mathbf{w}'_t, \overleftarrow{\mathbf{h}}_{t+1}).$$

w_t and w'_t are the weights in the forward and backward LSTMs, respectively. The outputs from the forward and backward are added in an element-wise way:

$$(4.9) \quad \mathbf{h}_t = \vec{\mathbf{h}}_t + \overleftarrow{\mathbf{h}}_t.$$

We keep the number of units in the LSTM being same as the number of grids, which is equal to $W \times H$. All the time steps are concatenated into a matrix \mathbf{H} :

$$(4.10) \quad \mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_t).$$

For all time steps' values in every grid, we use a linear transformation and a *softmax* activation function to get the attention weights on the time step domain:

$$(4.11) \quad \mathbf{a} = \text{softmax}(\mathbf{WH}).$$

The outputs are the weighted sum of the hidden states and the attention weights in an element-wise way. The passenger prediction is the outputs after being weighted by attention.

4.6 Objective function In our model, the objective function is defined as the mean squared error between the true passenger demands and predicted passenger

demands. The model is trained based on mini-batches. Suppose there are m samples in a mini-batch and every sample is indexed by i , the objective function $L(\theta)$ with trainable parameters θ is defined as follows,

$$(4.12) \quad L(\theta) = \frac{1}{m} \sum_{i=1}^m \|\hat{\mathbf{D}}_i^{(t+1)} - \mathbf{D}_i^{(t+1)}\|^2,$$

It is the mean squared error between the predicted and true passenger demands in a mini-batch. The optimization algorithm in our model is the ADaptive Moment estimation (Adam) [18], which adaptively changes the effective learning rate during training.

5 Evaluation

5.1 Evaluation objectives and metrics We evaluate STEF-Net by comparing it with state-of-the-art models on real data with regarding to the accuracy in passenger demands prediction. The accuracy is measured by two metrics, Mean Absolute Error (MAE) and Rooted Mean Squared Error (RMSE). The MAE and RMSE are two widely employed metrics to evaluate the performance of a prediction system [4].

5.2 Dataset The dataset used for training and testing is from Didi Chuxing. The data contains over 5.24 million non-duplicating service requests from 11/01/2016 to 11/30/2016 in Chengdu City, China. In the dataset, every service request record is composed of the request ID, pick-up time, pick-up coordinates, drop-off time and drop-off coordinates. (Note that we do not need drop-off time and drop-off coordinates here.) The data of the first 23 days is used for training and that of the last 7 days (one week) for testing. The area is about $14.41\text{km} \times 14.39\text{km}$. We divide the whole area into 20×20 same-size grids. The length and width of every grid are both about 700 meters. The time interval is set half an hour. Same as previous studies [4, 3], the passenger demands are scaled into $[0,1]$ using max-min scaling. In the final step, the demand values are recovered by the inverse of max-min scaling.

Regarding the external information, the day of a week is extracted from the pick-up time. The weather information and sunrise/sunset information are crawled from the Weather Underground website [19] using Python. The website provides historical weather information in Chengdu. As presented in previous section, we extract 24 features about the external information.

5.3 Hyperparameters and development environment In our network, 3 ConvLSTMs are stacked and all the ConvLSTMs use 64 filters of size 3×3 . A convolutional layer with one filter is added after the

stacked ConvLSTMs to convert the data into a 20×20 tensor. The membership layer in the fuzzy neural network is composed of 24×400 units, and the fuzzy rule layer has 400 units. The outputs from the fuzzy neural network are reshaped into 20×20 and the following convolutional layer has one filter with the kernel size of 3×3 . In the fusion part, the convolutional layer has one filter with the kernel size of 3×3 . In the fully connected layers, there are three layers with 100 units, 200 units, and 400 units, respectively. The output is reshaped into 20×20 to match the prediction in the whole city. The parameters in the fuzzy neural network are uniformly initialized from 0 to 1. All the other parameters are uniformly initialized.

In our model, to be consistent with the previous studies we compare with [5, 3], we by default use previous 8 time intervals' data to predict the current time interval's passenger demands, It means the current time interval's passenger demands are predicted based on the historical 4 hours' data. We also use 2 hours' data to predict the demands to see how the time length can influence the prediction accuracy.

Our model is developed on the top of Keras [20] with the backend of Tensorflow [21]. The model is running on a desktop with an Intel Xeon 3.10GHz \times 4 CPU and a GeForce GTX 1050 Ti GPU. The model is trained by mini-batches. Every mini-batch has 16 samples. Every model is trained using 50 epochs and the results are generated after that.

5.4 Baselines For a thorough comparison with existing methods, we compare our model with three categories of methods, time-series (ARIMA), regression-based (Ridge and XGBoost) and neural network-based methods (ST-ResNet, AttConLSTM and FCL-Net). The methods are presented as follows:

- **ARIMA [2]:** ARIMA uses both moving average and autoregressive to predict the next time interval's passenger demands.
- **Ridge linear regression [22]:** Ridge linear regression uses a linear equation to model the relationship between historical features and future passenger demand. We reshape all features in this paper to a vector and feed vectors into the linear regression.
- **XGBoost [23]:** XGBoost (2016) is a widely used boosting method with a tree structure. All features are also reshaped into vectors to feed the XGBoost model.
- **AttConLSTM[3]:** AttConLSTM (2018) fuses attention and ConvLSTM into an auto-encoder model. It stacks CNNs and ConvLSTMs to en-

Model name	2 hours		4 hours	
	MAE	RMSE	MAE	RMSE
ARIMA	3.61	6.42	2.85	4.91
Ridge linear	3.50	6.32	3.47	6.25
XGBoost	3.48	6.18	3.29	5.87
ST-ResNet	2.90	5.15	2.86	5.02
AttConLSTM	2.63	4.58	2.60	4.55
FCL-Net	2.58	4.46	2.52	4.39
STEF-Net	2.31	4.05	2.27	3.89

code and decode the passenger demands and extracts passenger demands patterns as references in an attention network.

- **FCL-Net [5]:** FCL-Net (2017) employs ConvLSTM and LSTM to extract information from demands, time and weather. It fuses the outputs from two networks by addition.
- **ST-ResNet [4]:** ST-ResNet (2017) uses ResNet to capture the spatial and temporal information on demands from three categories, recent, near and distant. Only weather information at the current time interval is considered. It fuses data by addition. Because the distant demands require at least three weeks for one sample, we only take the recent and near categories.

All the models follow the settings in their original papers, and all of them are trained 50 epochs.

5.5 Results

5.5.1 Comparisons with baselines We present the comparison results with baselines in Table 5.5.1. From this table, we can see our model outperforms all the others regarding to both metrics. When the historical data is 4 hours, our model can achieve 3.89 in RMSE and 2.27 in MAE. The results at least 9.9% better in MAE and 11.3% better in RMSE than the best one among all baseline methods and about 34.6% better in MAE and 37.8% better in RMSE than the worst among all baselines. Ridge linear regression performs worst because it only considers the linear relation among features. Note that after the model is trained, we use 2 hour data and 4 hour data to predict future demands, respectively. When more historical data is utilized to predict, the performance improves.

5.5.2 Qualitative results In this section, we illustrate how the passenger demands are influenced by the historical data. Using heatmaps, we plot the attention weights under different weather conditions (sunny or rainy) at 8pm on two different days in Fig. 5. In each day, we plot the attention weights of the last time step and the second to the last time step, which repre-

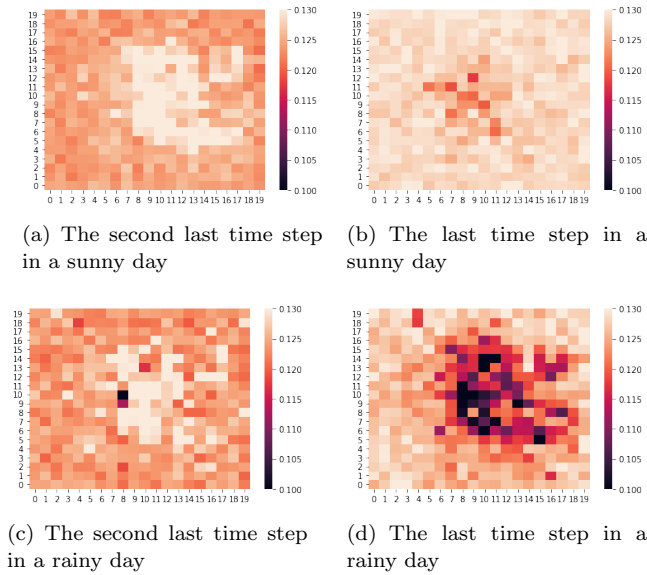


Figure 5: The attention probabilities of the historical time steps that influence the current passenger demands under different weather conditions

sent the intermediate history and next intermediate history. Note that there are 20×20 grids, and thus in the heatmaps, every small square represent one geographic grid. Center grids represent the downtown area of the city and border grids represent suburban areas. Comparing Fig. 5(a) with Fig. 5(b), and comparing Fig. 5(c) with Fig. 5(c), we can see that the current passenger demands in the city center are more influenced by the second to the last time step than by the last one while those in the border area are more influenced by the last time step regardless of the weather. Comparing figures in the sunny day and rainy day in Fig. 5, we can also see that the weather greatly influences the passenger demands. When the outside is rainy, the passenger demands are less influenced by the last time step in the center of the city (downtown). The rainy weather may incur passengers in the downtown area to change their travel plan by pre-scheduling their activities, e.g., reducing the outdoor activities, while the users in the suburbs are not influenced much.

5.5.3 Ablation studies: comparisons with variants of STEF-Net In our evaluation, we also compare STEF-Net with its variants to explore how different components influence the prediction performance. We explored the following variants,

- **ConvLSTM \rightarrow CNN&LSTM** : 3 convolutional layers with the 64 windows of size 3×3 and 3 LSTM layers with 64 units in each layer are stacked to

Model name	MAE	RMSE
ConvLSTM \rightarrow CNN&LSTM	2.32	4.01
ConvLSTM \rightarrow LSTM	2.36	4.09
Fuzzy \rightarrow LSTM	2.58	4.51
No attention layers	2.59	4.35
No external data	2.40	4.24
Weighted addition	2.38	4.09
STEF-Net	2.27	3.89

replace ConvLSTMs for the demand information processing.

- **ConvLSTM \rightarrow LSTM**: 3 LSTM layers with 64 units in each layer are stacked to replace ConvLSTMs for the demand information processing.
- **Fuzzy \rightarrow LSTM**: 3 LSTM layers with 64 units in each layer are stacked to replace the fuzzy neural network for the external information processing.
- **No attention layers**: no attention layers are used after the outputs from the ConvLSTM and the fuzzy neural network are fused.
- **No external information**: no external information is used in the model. Features from the ConvLSTM are directly fed to the attention layer.
- **Weighted addition**: weighted addition is used to replace our convolutional operation in data fusion.

The results are shown in Table 5.5.3. We can see that our model, STEF-Net, has the best performance among all variants. Specifically, our model can achieve the smallest RMSE and MAE compared to its variants. Comparing to these variants, we can see that the ConvLSTM can effectively capture the spatio-temporal information, which performs better than both CNN&LSTM and LSTM, and the fuzzy neural network can outperform the LSTM in processing the external information. In addition, we can see that attention layers can further capture the temporal relevance and improve the performance. Among all the components, fuzzy neural network can significantly improve the performance comparing to the LSTM. The feature fusion method can further capture the temporal feature interaction and the weighted addition performs worse than the fusion with convolution. Comparing the results between our model with the model without external information, we can see that the external information is important in predicting the passenger demands, which matches our intuition of fusing it into our model.

6 Conclusion

In this paper, we propose a Spatio-TEmporal Fuzzy neural Network (STEF-Net) to accurately predict passenger demands in the near future. Our model can ef-

fectively capture complex input dependencies, including spatial, temporal and external factors, which may influence future passenger demands. In the proposed approach, we combine deep learning with a fuzzy neural network to model spatio-temporal and external information, respectively. We employ a new feature fusion method with convolution followed by an attention layer, to fuse two neural networks into one and keep temporal relations for further temporal relevance modeling. Extensive experiments on real-world dataset show that, our model outperforms the state-of-the-art approaches with over 10% improvement in RMSE.

References

- [1] S. Shekhar and B. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2024, pp. 116–125, December 2008.
- [2] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, September 2013.
- [3] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. Marina Del Rey, CA, USA: ACM, February 2018, pp. 736–744.
- [4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA: AAAI, February 2017, pp. 1655–1661.
- [5] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, December 2017.
- [6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [7] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1803.11115*, 2018.
- [8] X. Liang and G. Wang, "A convolutional neural network for transportation mode detection based on smartphone platform," in *2017 IEEE MASS*. IEEE, October 2017, pp. 338–342.
- [9] S. Hochreiter, Y. Bengio, and P. Frasconi, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *Field Guide to Dynamical Recurrent Networks*, J. Kolen and S. Kremer, Eds. USA: IEEE Press, January 2001.
- [10] L. A. Zadeh, "Fuzzy sets," in *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*. NJ, USA: World Scientific, May 1996, pp. 394–432.
- [11] Y.-J. Zheng, W.-G. Sheng, X.-M. Sun, and S.-Y. Chen, "Airline passenger profiling based on fuzzy deep machine learning," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 2911–2923, December 2017.
- [12] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 1006–1012, August 2017.
- [13] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, June 2012.
- [14] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, January 2017.
- [15] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. Houston, TX: SIAM, June 2017, pp. 777–785.
- [16] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, and J. Ye, "Deep multi-view spatial-temporal network for taxi demand prediction," *arXiv preprint arXiv*, vol. 1802.08714, pp. 1–9, February 2018.
- [17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*. Montreal, Canada: Neural Information Processing Systems Foundation, December 2015, pp. 802–810.
- [18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv*, vol. 1412.6980, pp. 1–15, December 2014.
- [19] "Weather underground historical weather," <https://www.wunderground.com/history/>, April 2018.
- [20] F. Chollet *et al.*, "Keras," <https://github.com/fchollet/keras>, April 2015, accessed April, 2018.
- [21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv*, vol. 1603.04467, pp. 1–19, March 2016.
- [22] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, February 1970.
- [23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd sigkdd*. San Francisco, CA: ACM, August 2016, pp. 785–794.