



A Neighborhood-Augmented LSTM Model for Taxi-Passenger Demand Prediction

Tai Le Quy¹, Wolfgang Nejdl^{1(✉)}, Myra Spiliopoulou², and Eirini Ntoutsi¹

¹ L3S Research Center, LUH, Hannover, Germany
{tai,nejdl,ntoutsi}@l3s.de

² Otto-von-Guericke-University, Magdeburg, Germany
myra@ovgu.de

Abstract. Taxi is a convenient means of transportation worldwide. Accurately predicting the taxi-demand is crucial for taxi-companies to effectively allocate their fleet to taxi-stands and reduce the waiting time for passengers thus increasing their overall satisfaction and customer retention. Nowadays precise information about taxi-rides is available and can be used to infer the taxi-passenger demand across different locations and time-points. In this paper, we propose an approach for predicting the pick-demand of a given taxi-stand, that takes into account not only the demand-history of the particular stand but it also considers information from neighboring stands. Our model is an LSTM neural network augmented with information from the spatial neighborhood of the stands. Experiments with two versions of the taxi demand dataset from the city of Porto, Portugal show that our approach can provide better predictions comparing to approaches that do not exploit the neighborhood.

Keywords: Taxi-passenger demand · Time series prediction · LSTM · k -nearest neighbors · Deep learning · Neural networks

1 Introduction

Advances in sensor and wireless communication contribute to the development of intelligent transportation systems, which lead to the transformation of transportation domain. Taxi networks are the important means of transportation providing the convenient and direct services for passengers. Currently, many taxi vehicles are equipped with Global Positioning System (GPS) and wireless communication features that can generate a new source of rich spatial temporal information.

Intelligent online systems that plays a crucial role for real time taxi services scheduling, taxi sharing, fuel-saving routing, time-saving route finding are already developed to improve taxi service reliability [19]. Improving levels of passenger satisfaction and maximal profit for taxi providers are the main targets of taxi companies. Balancing the relationship between the passenger demand and the number of running taxi vehicles is the most efficient way to maximize the profit for taxi providers [19]. Knowledge on time and places that is emerged

the passenger demand can be an advantage for drivers even when there is no economic availability. The information regarding passenger demand is very useful for drivers in making decision moving to pick up passengers in a particular region in the city. GPS historical data are the main variable used in prediction models because it can reveal hidden mobility patterns.

Many researchers were attracted by the mobility data and proposed different approaches for taxi-passenger demand prediction. Among the investigated models are linear regression, ARIMA, feed-forward neural networks and more recently, deep neural networks. Most of these approaches focus exclusively on the information of the stand to predict passenger demand in the future. In this paper, we also exploit information from neighboring stands in an attempt to enrich the information provided to the model, in our case an LSTM neural network. Data augmentation [26] is a popular technique especially for data-insensitive models like Deep Neural Networks (DNNs); for example, the improved performance on ImageNet [3] was also attributed to image augmentation using different domain-specific augmentation techniques like image reflection, translation, cropping and changing the color palette. Unnikrishnan et al. [25] introduce an entity-centric stream classification approach that exploits the observation history of the particular entity and of entities similar to it. Similar entities are defined on the basis of static entity characteristics like gender and birthdate in case of patient data or product properties in case of product reviews. Our neighborhood-selection idea is similar as we also rely on the spatial neighborhood of the taxi stands rather than their demand histories. As our experiments with data from the taxi network in the city of Porto, Portugal spanning a period of one year show, such an augmentation is beneficial for the predictive performance of the model.

The rest of the paper is organized as follows: Sect. 2 overviews the related work. Our neighborhood-augmented LSTM approach is presented in Sect. 3. A detailed experimental evaluation is provided in Sect. 4. Finally, conclusions and outlook are summarized in Sect. 5.

2 Related Work

There is a large body of work on traffic-related data, from trajectory querying, to hotspot detection, clustering, trajectory prediction [1] etc. Hereafter, we focus mainly on existing works using taxi-data and related mainly to our demand prediction problem.

A taxi-sharing framework is proposed in [7] that returns the top- k taxi recommendations for a passenger request. They select the top- k candidate taxis for a specific location by considering its neighbors on the traffic network. For their experiments, they have used the New York city taxi dataset. Luca *et al.* [6] proposes a method to find the Nash equilibrium in a taxi sharing fare in case there are many passengers sharing one taxi in order to save money. For their experiments, they also use the New York city dataset.

The problem of taxi-passenger demand prediction has attracted the attention of many researchers recently and as result, several approaches have been proposed. Most of these approaches rely on well-known prediction models from the time-series forecasting domain [15]. Kaltenbrunner et al. [11] introduced an

auto-regressive moving average (ARMA) model approach to forecast the number of bicycles at a station from Barcelona’s bicycle network in order to increase the stations spatial deployment. Min and Wynter [17] applied another popular time-series prediction model, ARIMA (Auto-Regressive Integrated Moving Average) to predict the speed and volume of traffic in a road network. Luis *et al.* [18, 19] introduce an ensemble of experts to predict taxi demand, where each expert is specialized to a particular trend. In particular, their ensemble consists of a Time-Varying Poisson model, a Weighted Time-Varying Poisson model and a ARIMA model. The experiments were conducted on the Porto taxi dataset. Su *et al.* [27] predicts taxi-passenger demand in urban areas in Hong Kong using multiple features such as the number of vacant taxi on the roads, the waiting time, passenger demand, taxi fare as the input for a feed-forward neural network. Recently, TONG, Yongxin *et al.* [24] presented a multi-dimensional linear regression model to predict the taxi demand in Beijing and Hangzhou, China. Their multi-dimensional representation consists of temporal features, spatial features, meteorological features, and the combination of these features. Yao et al. [28] proposed a deep learning framework to model both spatial and temporal relations by using two neural network model CNN and LSTM to predict taxi demand in Guangzhou, China.

Contrary to most of the existing works that rely exclusively on taxi-stand’s own demand history we enrich the data representation of each stand using information from neighboring stands. Our intuition is that the demand of a taxi-stand might be indicative of the demand of some nearby stand as well. Such an augmentation is especially beneficial for data intensive models, our base model is an LSTM deep neural network model, in order to reduce over-fitting and eventually, generalization performance.

3 Neighborhood-Augmented Taxi Demand Prediction

3.1 Problem Definition

Let $S = \{s_1, s_2, \dots, s_N\}$ be the set of predefined N taxi-stands in a city. Consider $X_s = \{X_{s,0}, X_{s,1}, \dots, X_{s,t}\}$ to be a discrete time series (based on an aggregation period of P -minutes) that models the taxi-demand for stand s , that is, the number of pick-ups for each aggregation period P at s . We refer to this time series as the *demand history of stand s* . Our goal is to build a model which predicts the demand $X_{s,t+1}$ for the next time point $t + 1$ at taxi-stand s .

Traditional approaches rely solely on the demand history of the stand X_s for the prediction (we use such methods as baselines for our comparison, c.f., Sect. 4.3). In this work we propose to augment the stand’s demand history X_s with information from its neighborhood. The intuition behind this augmentation process is that nearby taxi-stands might display similar demands. Our dataset seems to justify our intuition: In Fig. 1 we show the spatial proximity of the different taxi-stands (left) vs their demand proximity (right). The demand proximity is evaluated using Pearson correlation and for efficiency reasons, only part of the history demand. Due to space, we show here only the information for the first 20 taxi-stands (IDs 1–20),

As we can see, when the pairwise spatial distances are high, an opposite trend is observed in the demand history correlation values. This can be observed for a variety of taxi-stands, for example, 4, 5 and 8.

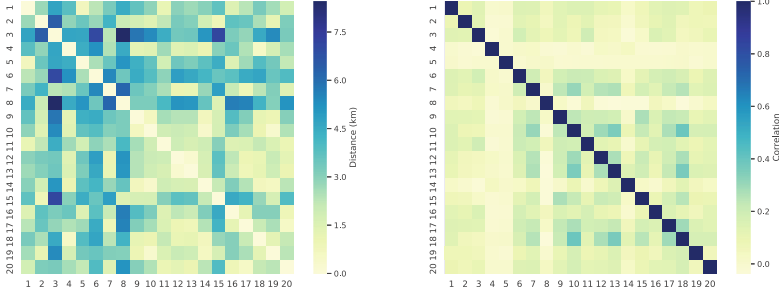


Fig. 1. Spatial proximity (left) vs pickup demand correlation (right) between taxi-stands (based on dataset D1).

Based on this motivation, we present hereafter the neighborhood-augmented LSTM model for predicting taxi-passenger demand of a taxi-stand.

3.2 Neighborhood-Augmented LSTM Model

Our model is an extension of the well known Long Short-Term Memory (LSTM) networks [8], a special kind of a recurrent neural network (RNN). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. There are several architectures of LSTM units. An LSTM cell takes an input and stores it for some period of time. Because the derivative of the identity function is constant, when an LSTM network is trained with back propagation through time, the gradient does not vanish. The activation function of the LSTM gates is often the logistic function. Intuitively, the input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections, which need to be learned during training, determine how the gates operate.

In our approach, we train an LSTM model for each taxi-stand s using not only its primary demand history X_s but also demand history information from its k -nearest neighbors. That is, the input to the LSTM is a $(k + 1)$ dimensional vector, X'_s . The actual demand values (ground truth) comes from taxi-stand s and therefore the goal is to fit the neighborhood-augmented LSTM model for predicting the demand values of taxi-stand s .

```

input : Taxi demand dataset; k-number of neighbors
output: Prediction model  $M_s$  for taxi-stand  $s$ 

1 //Data augmentation
2  $X_s$ : the demand history of taxi-stand  $s$  up to time  $t$ 
3  $X'_s \leftarrow X_s$  //extended representation
4  $\{Neighbors_s\}$ : the set of  $k$  nearest taxi-stands to  $s$ 
5 for  $i \leftarrow 1$  to  $|\{Neighbors_s\}|$  do
6 |    $X_i$ : the demand history of taxi-stand  $i$ 
7 |    $X'_s \leftarrow Extend(X'_s, X_i)$ 
8 end
9 Normalize features
10 //Train on the augmented data
11  $M_s \leftarrow LSTM(X'_s)$ 
Algorithm 1: Neighborhood-augmented LSTM model training
    
```

The pseudo code of the algorithm is shown in Algorithm 1. Each taxi -stand has its own LSTM model for training.

In the above algorithm, the normalization step aims to normalize all features in the [0–1] range. This is an important step for LSTM convergence [13]. In particular, we use min-max normalization.

The structure of our LSTM network is shown in Fig. 2 and explained hereafter. In this model, time series of stand X with its $k - neighbors$ are used as the input of the first LSTM layer, followed by a hidden layer before a dropout unit. Predicted time series Y is the result of our model. The tuning of the hyper-parameters is discussed in detail in Sect. 4.4 but the selected values are mentioned here as well:

1. Input (X'_s , the extended description of stand s ; look back value = 5 (see Sect. 4.4.))
2. LSTM (N=200, optimizer = ‘Adamax’, Activation function = ‘tanh’, loss= ‘mean squared error’, batch size = 100 (see Sect. 4.4.))
3. Full connected LSTM (N=200, Activation function = ‘tanh’)
4. Dropout = 0.7 (see Sect. 4.4.)
5. Dense (Activation function = ‘tanh’)

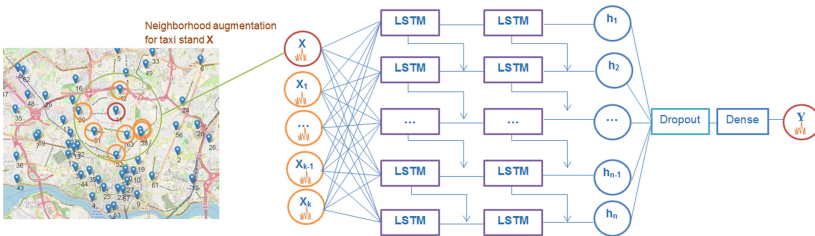


Fig. 2. The architecture of the neighborhood-augmented LSTM.

4 Experimental Evaluation

We evaluate our approach on the publicly available dataset on taxi-demand from Porto, Portugal (Sect. 4.1). The experimental setup and evaluation criteria are discussed in Sect. 4.2. The goal of our experiments is to evaluate the impact of the neighborhood-based augmentation on prediction quality (Sect. 4.5) as well as to study how the “quality” of the neighborhood as evaluated by the average distance of neighboring stands from the reference stand affects the predictions (Sect. 4.6).

Our LSTM approach was implemented using Keras and Tensorflow, whereas for the other approaches we use the available implementations in Python.¹

4.1 Dataset

We use the dataset from [19] that contains information on taxi trips organized by a taxi company in the city of Porto in Portugal and was part of the ECML 2015 challenge². The dataset spans over a period of one year (from July 2013 to June 2014) and contains 1.710.670 records. Each record corresponds to a completed taxi trip, described in terms of 9 features: (1) TRIP_ID: a unique identifier for each trip; (2) CALL_TYPE: the way to use the taxi service and contain one of three possible values: A (the trip is assigned from the call central), B (the trip is departed from a specific stand) or C (passengers are pickup from a random street); (3) ORIGIN_CALL; (4) ORIGIN_STAND: a unique identifier for the taxi-stand; (5) TAXI_ID; (6) TIMESTAMP: Unix Timestamp (in seconds); (7) DAYTYPE: the daytype of the trip’s start (holiday or any other special day); (8) MISSING_DATA; (9) POLYLINE: the trajectory of trip. In addition, the dataset also provides the information of all 63 taxi-stands with their name and GPS coordinates.

Figure 3 depicts the spatial distribution of the taxi-stands in Porto, each stand is assigned a unique ID from 1 to 63. As one can see, the stands are not randomly distributed rather their spatial density reflects the demand with most stands located close to the city center. Moreover, we can see that despite the aforementioned mandatory regulation there are trips that do not start at the location of the taxi-stand. The intensity of the color in Fig. 3 shows the density of the starting points; in many cases taxi-stands have the highest local density but not all trips start at some taxi-stand.

We preprocess the data as follows: Firstly, we sort all records by timestamp in ascending order. We remove features MISSING_DATA and POLYLINE and we add two new features: LATITUDE and LONGITUDE extracted from the POLYLINE attribute and describing the coordinates of the starting trip location. After, we remove instances that have no both taxi-stand ID starting location. This results in a clean dataset of 1.706.572 completed taxi trips. Contrary to previous work [18, 19], we create two versions of the dataset for the experiments.

¹ The code is available at: <https://github.com/quytai3985/PortoTaxiPrediction>.

² <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data>.

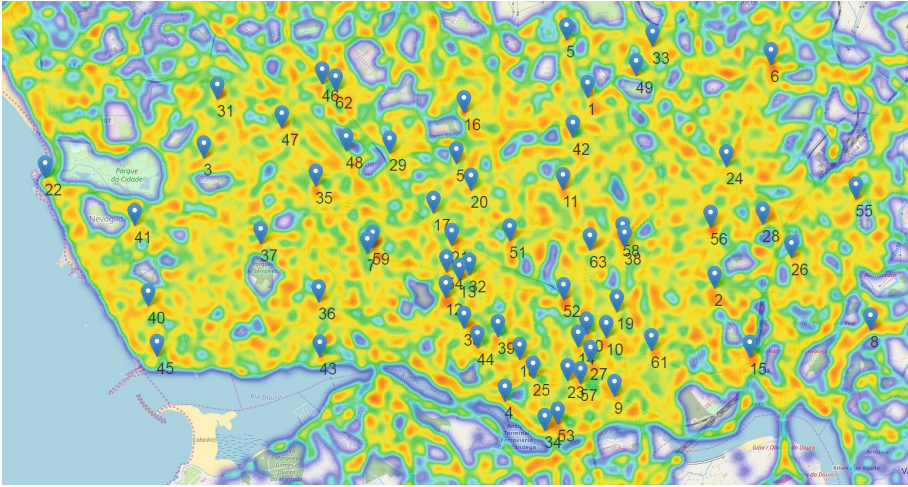


Fig. 3. Spatial distribution of the taxi-stands. Numbers 1–63 indicate the IDs of the stands.

The first dataset (D1) has only the taxi trips with `CALL_TYPE` equal to ‘B’, i.e., all trips that are departing from some taxi-stand. This dataset contains 817.861 instances and can be used for building a prediction model that can forecast the short term demand for specific taxi-stands.

However, as already mentioned not all trips start from a taxi-stand (i.e., the initial location does not match the location of some taxi-stand). Due to the amount of these trips (888.711 trips, 52.07% of the overall dataset), this information cannot be easily omitted, rather these trips might play an important role for the predictions and one needs to consider them for the forecasting. Therefore, in the second version (D2), we use all records of the clean dataset. For the trips that do not start from a taxi-stand (i.e., those with a `CALL_TYPE` equal to ‘A’ or ‘C’), we assign them to their closest taxi-stand based on distance between the starting location of the trip and the location of the taxi-stand. Intuitively, we consider the taxi-stand as covering some region in the city with the its coordination being the center of this region.

The distribution of the trips on the different taxi-stands for the (D1), (D2) datasets is shown in Figs. 4 and 5, respectively. For each dataset, we also display the mean and median demand values. It is easy to observe that there are a large number of taxi demand in several stands. For example, the most popular stand is stand 15, which corresponds to the main train station. The top 10 most crowded stands in **D1** account for approximately 46.5% of the total 817.861 passenger demand. In dataset **D2**, that proportion is around 36.3% of 1.7M pickups. A closer look at the top 10 stands via Google Maps reveals that they are all close to the main train station and the city center with many historical sites, shops and hotels.

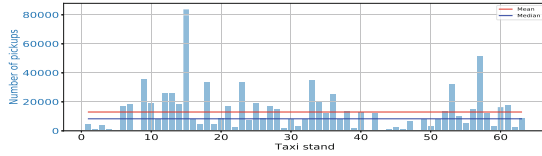


Fig. 4. Pickup distribution per taxi-stand on D1.

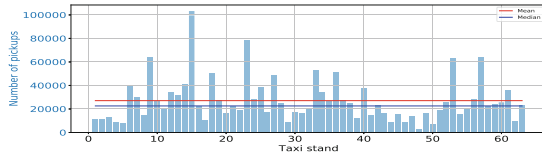


Fig. 5. Pickup distribution per taxi-stand on D2.

4.2 Experimental Setup and Evaluation Measures

We set the aggregation period at 30 min based on the average waiting time at a taxi-stand as in [19]. We generate the demand history at each taxi-stand by aggregating the number of pickups every 30 min.

Demand history examples are presented in Fig. 6 for taxi-stands 1 and its spatial neighbor - taxi-stand 49 during one week (from 01 Jul 2013 to 07 Jun 2013). Similarly, Fig. 7 describes the demand history of taxi-stand 15 and its spatial neighbor, taxi-stand 61. In both figures, we use D1 as the data source. As we can see, the pickup demand series are different and depended on the location of the taxi-stand. For example, taxi-stand 1 is located far from the city center (around 5 km) whereas taxi-stand 15 is close to the main train station. As a result, the demand on stand 15 is much higher with over 83,000 yearly pickups whereas the demand for taxi-stand 1 is only 4,500. Similarly, the taxi demand on stand 61 (close to stand 15) is around 17,000 pickups whereas the demand for taxi-stand 49 (close to stand 1) is only 8,000. Except for the differences in the amplitude of the demand, we can also see differences w.r.t. the temporality of the demand. For example, both stands 15 and 61 have around-the-clock demand, but this is not the case for stands 1 and 49. This behavior comprises our motivation behind the proposed neighborhood-augmented demand prediction model.

In time series prediction, the measurement symmetric Mean Absolute Percentage Error (sMAPE) [16] is more meaningful than other measurement, such as MSE, RMSE. One reason is the proportion values are more comprehensive than squared errors [21]. As a consequence, in our experiment, we evaluate the prediction quality of the models for each taxi-stand by comparing the forecast values with the original ones using sMAPE. However, we still report our results on MSE measurement as a reference one. In particular, let the true demand for a taxi-stand s be: $X_s = \{X_{s,0}, X_{s,1}, \dots, X_{s,t}\}$ and the predicted demand:

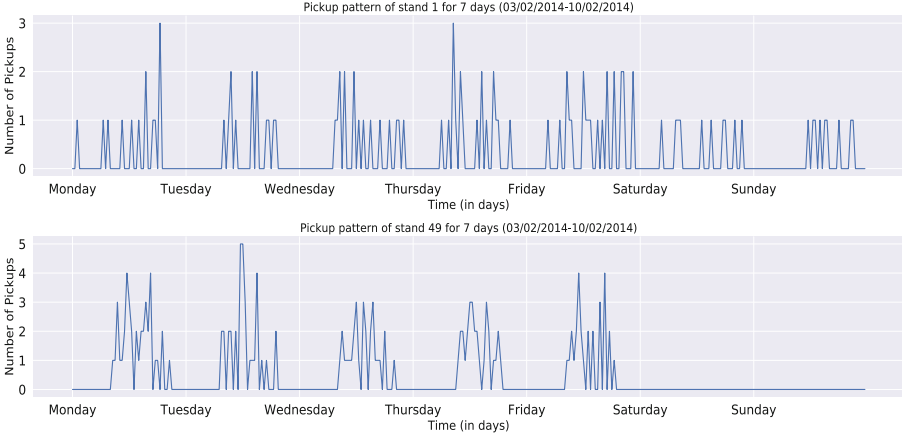


Fig. 6. Pickup demand history for nearby taxi-stands 1 and 49.

$\hat{X}_s = \{\hat{X}_{s,0}, \hat{X}_{s,1}, \dots, \hat{X}_{s,t}\}$. Then $sMAPE_s$ is given by:

$$sMAPE_s = \frac{100\%}{t} \sum_{i=1}^t \frac{|X_{s,i} - \hat{X}_{s,i}|}{(X_{s,i} + \hat{X}_{s,i})/2} \quad (1)$$

In Eq. 1, sMAPE values fluctuate between -200% and 200% [16]. Flores [5] claims that a percentage error between 0% and 100% is much easier to interpret, therefore we omit factor 2 in the denominator. Furthermore, due to possible prediction of negative demand values, we use absolute values in the denominator of Eq. 1. Additionally, Eq. 1 can result in a high error if the real demand is 0 and the predicted one is non-zero; in such a case, the error would be 100% . To deal with this issue, we use *Laplace correction* [10] by adding a constant c to the denominator. Finally, the modified $sMAPE_s$ that is used for our evaluation is given by:

$$sMAPE_s = \frac{100\%}{t} \sum_{i=1}^t \frac{|X_{s,i} - \hat{X}_{s,i}|}{|X_{s,i}| + |\hat{X}_{s,i}| + c} \quad (2)$$

The constant c is user-defined. In our experiments, we use the corrected sMAPE version (Eq. 2) with $c = 1$. The aforementioned formulas refer to the error at each stand, we aggregate the error over all taxi-stands as follows:

$$sMAPE = \frac{\sum_{i=1}^N sMAPE_i}{N} \quad (3)$$

where N is the number of taxi-stands.

4.3 Baselines and Method Parameter Settings

We compare our approach against well-known prediction methods, described hereafter together with their parameter tuning.

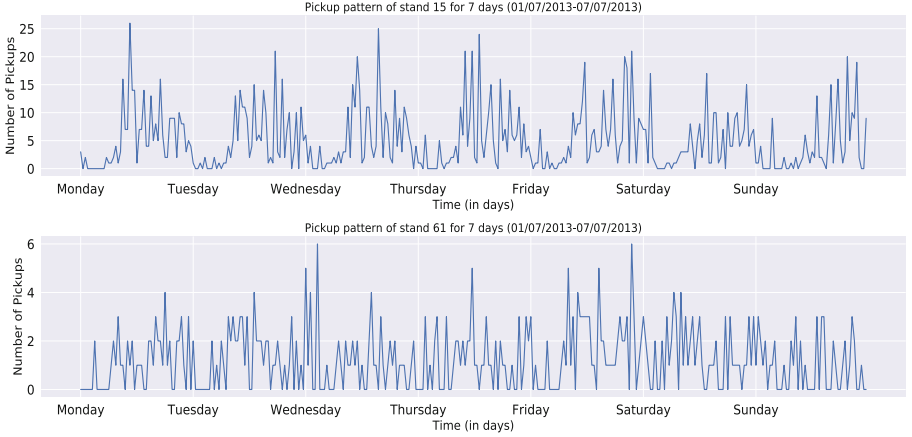


Fig. 7. Pickup demand history for nearby taxi-stands 15 and 61.

Simple Moving Average: A simple moving average (SMA) [9] is an arithmetic moving average calculated by averaging the observed values of a time series in the calculation period. Given a calculation period of q timepoints, the prediction $X_{s,t+1}$ for the next time point $t + 1$ is given by:

$$X_{s,t+1} = \frac{1}{q+1} \sum_{j=0}^q X_{s,t-j} \quad (4)$$

The number of periods q should be set; when $q = 0$ this is simply the value of the last observation. For our experiments, we choose $q = 20$ using grid search. We set the range of q from 2 to 24 (equals to 1–12 h) with step 1. The selection of q was based on taxi-stand 1 and dataset D1. Taxi-stand 1 is chosen as the representative stand for tuning as its location is far from the places that concentrate a huge amount of vehicles, such as the main station or city center. Moreover, in our experiments the performance of the different models on this taxi-stand was close to the average values.

Linear Regression: In a linear regression model [22] the future value of a variable is assumed to be a linear function of its past q values, where q defines the amount of past values contributing to the computation.

$$X_{s,t+1} = \beta_0 + \beta_1 X_{s,t-1} + \beta_2 X_{s,t-2} + \dots + \beta_q X_{s,t-q} \quad (5)$$

For our experiments, we choose $q = 15$ using grid search, similarly to parameter selection for SMA. We apply $q = 15$ for all 63 different models/taxi-stands. However, the parameter β_0 is adapted to each taxi-stand using grid search with β_0 in the range 10^{-16} to 10^6 and step 100.

Random Forest Regression: Random forest [14] is an ensemble technique averaging the forecasting of a large number of decorrelated decision trees. Random forests are built on two main ideas - bagging to build each tree on a different

bootstrap sample of the training data, and random feature selection to decorrelate the trees. During the forecasting for time point $t+1$, each tree B_j ($j = 1..m$) provides a prediction $X_{s,t+1,j}$. The final prediction of the random forest is the majority vote of the m trees.

For the experiments, we set the number of trees m per taxi-stand using grid search in the range 10 – 800 and step 40.

XGBoost Regression: XGBoost [2] is an implementation of gradient boosting decision trees designed for efficiency. For the experiment, we use grid search to select the number of trees of the ensemble (in the range 40 to 600 trees with step 40) as well as the maximum tree depth (in the range 1 to 4 with step 1). Parameter selection is done per taxi-stand.

4.4 LSTM Parameter Settings

The number of neighbors k is selected by grid search based on the representative taxi-stand 1; the result is a value of $k = 15$. A similar process is followed for the rest of the parameters, i.e., they were set using grid search over the data from the representative stand 1. In particular, the *look back value* parameter is select from a range of 2 to 24 (corresponding to 1 to 12 h in the history) with step 1. The best *look back value* = 5 is chosen as it raises best value of *sMAPE*. *AdamMax* and *tanh* are selected for the gradient descent optimization algorithm and activation function, respectively as they cause the best *sMAPE* values compared to other functions. Additionally, a list of possible candidates (10, 15, 20, 25, 50, 100, 200, 500 1000) is investigated to find the optimal *epoch* and *batch size* number. The best results were obtained with *epoch*=25 and *batch size* = 100. Furthermore, the range from 10 to 300 with step 10 and the range from 1 to 4 with step 1 were explored to find the best number neurons per layer and the number of hidden layers, respectively. According to the results, we construct our model with 1 hidden layer and $N = 200$ neurons. Besides, to prevent our LSTM model from *overfitting* we use the *dropout* technique that randomly drop units (along with their connections) from the neural network during training in order to avoid co-adapting too much [23]. The dropout rate was set to 0.7, base on our experiments with a range of dropout values fro 0.1 to 0.9 with step 0.1.

4.5 Taxi-Demand Prediction Quality Results

Table 1 summarizes the prediction quality of the different models for dataset D1, containing trips starting from an actual taxi-stand. In this table, neighborhood-augmented LSTM is experimented with $k = 15$. Table 2 summarizes the results for dataset D2., containing all trips from the cleaned dataset and after mapping the trips that do not start from a stand to their closest stand. $k = 25$ is the number of neighbors used in Neighborhood-augmented LSTM architecture.

As we can see, our approach, Neighborhood-augmented LSTM, results in the smallest *sMAPE* errors, followed by vanilla LSTM. Moreover, the LSTM models outperform traditional prediction models with linear regression models

Table 1. Prediction quality of the different models on D1.

Model	Training		Testing	
	sMAPE (%)	MSE	sMAPE (%)	MSE
Simple moving average			23.34	1.721
Linear regression	24.37	1.61	24.52	1.765
Random forest regression	16.83	0.383	24.25	1.660
XGBoost regression	23.90	1.391	23.91	1.585
LSTM	18.37	1.659	18.54	1.839
Neighborhood-augmented LSTM	17.32	1.465	17.63	1.682

Table 2. Prediction quality of the different models on D2

Model	Training		Testing	
	sMAPE (%)	MSE	sMAPE (%)	MSE
Simple Moving Average			30.33	5.369
Linear Regression	30.78	4.206	31.23	5.988
Random Forest Regression	18.49	0.715	31.03	5.503
XGBoost Regression	30.466	3.605	30.51	5.449
LSTM	27.03	4.16	27.22	6.660
Neighborhood-augmented LSTM	25.88	3.84	26.07	6.444

performing worse in both datasets. The improvement rates are higher for dataset D1 comparing to dataset D2. A possible reason is the assignment of the trips to their closest taxi-stands, a process that might have introduced errors. We plan to investigate alternative assignments in our future work, for example some weight decay approach based on the distance of the pick-up from its closest taxi-stand or soft assignments to multiply nearby taxi-stands.

A closer look at the performance of our approach vs the original demand for the different taxi-stands is presented in Figs. 8 and 9 for datasets D1, D2, respectively. As we can see, two different patterns of performance are shown. In dataset D1, the performance of the model has large variation, probably due to the large deviation of pickups among taxi-stand. The picture is different in dataset D2, where the actual number of pickups appears more balanced across the taxi-stands.

The variation in the performance of the different prediction models over the different taxi-stands is demonstrated more clearly in Fig. 10, where each boxplot corresponds to one prediction method and summarizes the sMAPE error over all stands. As we can see, there is large variation in D1 for all methods. Moreover, traditional approaches like MSA, LR, RF and XGBoost display skewed performance whereas the LSTM approaches are symmetric so the error over the different stands follows a normal distribution. Interestingly, and despite the

lower performance of the methods on dataset D2 comparing to D1, the spread of the error across the taxi-stands is very small for all methods, although there exist outliers. In case of LSTM-based models, most of the outliers correspond to stands with better predictions (lower sMAPE). Another interesting observation is that the model performs best when the number of pickups is close to average demand. As an extreme case, the most popular stand, stand 15 corresponding to the main train station, has the highest error on both datasets D1 and D2. A possible explanation is that such a stand is very difficult to model with a single model and one might need to consider different models for different contexts (e.g. season based, weekdays vs weekends etc). We leave this as our future work.

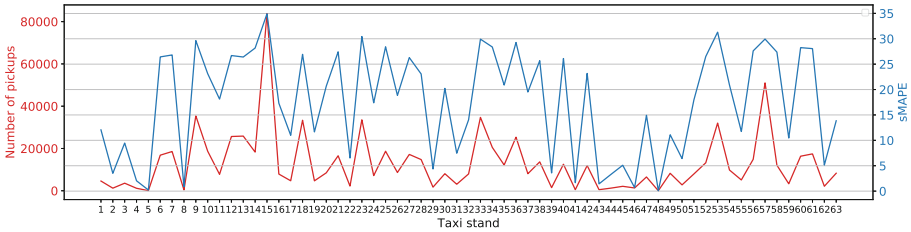


Fig. 8. Real demand vs neighborhood-augmented LSTM error across different taxi-stands for dataset D1.

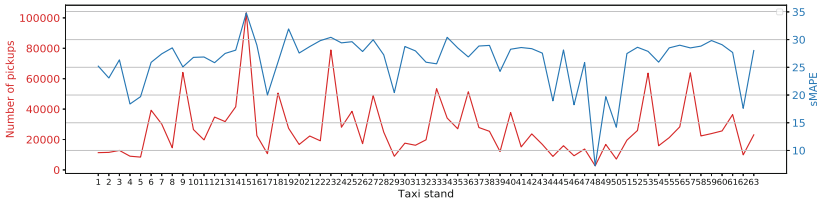


Fig. 9. Real demand vs neighborhood-augmented LSTM error across different taxi-stands for dataset D2.

4.6 Impact of Neighborhood

Our augmentation approach is based on the number of neighbors k parameter. We evaluate the impact of k on the predictive performance within a range of k from 1 to 61 and step 4. The results for both datasets D1 and D2 are shown in Figs. 11 and 12, respectively. The effect of k is more pronounced when testing with dataset D1. On dataset D1, when k is greater than 15 or the average distance from a specific taxi-stand to its neighbors is farther than 1 km, the performance of LSTM has a light fluctuation. While these values in dataset D2 are 25 and approximately 1.7 km, respectively. This shows that the proximity taxi-stands have a great influence on the prediction ability of the model. This is understandable because passengers in remote locations will be difficult to access the current pick-up stand for a short time.

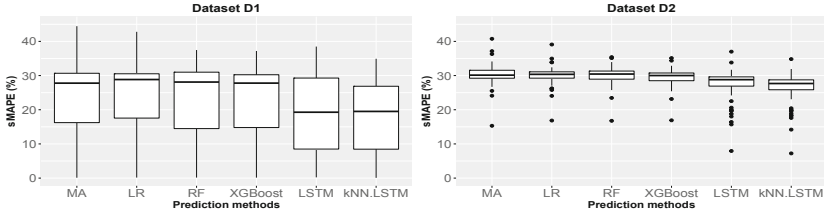


Fig. 10. Comparing error distributions for different prediction methods for dataset D1 (left) and D2 (right).

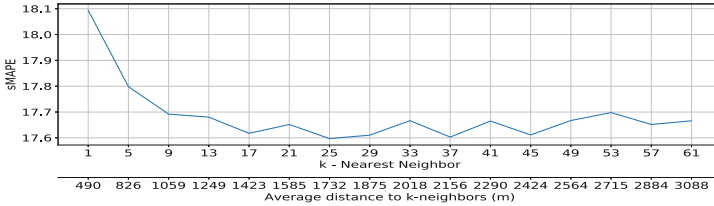


Fig. 11. Evaluating the impact of neighborhood on the predictive performance of neighborhood-augmented LSTM model on: D1

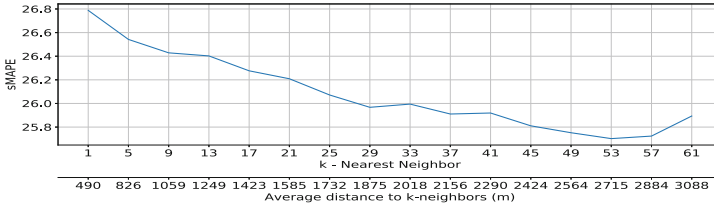


Fig. 12. Evaluating the impact of neighborhood on the predictive performance of neighborhood-augmented LSTM model on: D2

5 Conclusions and Outlook

In this paper we propose a neighborhood-augmented LSTM model for predicting the pick-demand of a given taxi-stand. Our experiments show that such an augmentation benefits the predictive performance of the model comparing to an LSTM approach that exploits strictly the demand history of the taxi-stand as well as to traditional prediction methods like SMA and regression.

There are several extension possibilities. In this work, we have considered a global neighborhood threshold k for all taxi-stands. However a more careful selection of the neighborhood and eventually a stand-tuned k would be more appropriate in order to account for different demand densities and taxi-stand densities in the city. Such a tuning could also take into account the data sparsity in the taxi-stand and grow the neighborhood progressively in order to cope with

the high demand of data-intensive models like LSTM neural networks and their potential overfitting. Another direction is to extend our approach by including other sources of information regarding the mobility demand in a city, for example, points of interest, event mentions from social networks [4], traffic patterns [20] as well as weather conditions [12].

Acknowledgement. The work was inspired by the German Research Foundation (DFG) project OS-CAR (Opinion Stream Classification with Ensembles and Active learners) for which the last two authors are Principal Investigators.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971 (2016)
2. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
4. Fafalios, P., Iosifidis, V., Stefanidis, K., Ntoutsi, E.: Tracking the history and evolution of entities: entity-centric temporal analysis of large social media archives. *Int. J. Digital Libr.* 1–13 (2018)
5. Flores, B.E.: A pragmatic view of accuracy measurement in forecasting. *Omega* **14**(2), 93–98 (1986)
6. Foti, L., Lin, J., Wolfson, O., Rishe, N.D.: The nash equilibrium among taxi ridesharing partners. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 72. ACM (2017)
7. Fu, X., Huang, J., Lu, H., Xu, J., Li, Y.: Top- k taxi recommendation in realtime social-aware ridesharing services. In: Gertz, M., et al. (eds.) SSTD 2017. LNCS, vol. 10411, pp. 221–241. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64367-0_12
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
9. Hyndman, R.J.: Moving averages. *International encyclopedia of statistical science* pp. 866–869 (2011)
10. Jaynes, E.T.: *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge (2003)
11. Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., Banchs, R.: Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system. *Pervasive Mob. Comput.* **6**(4), 455–466 (2010)
12. Koutroumanis, N., Santipantakis, G.M., Glenis, A., Doulkeridis, C., Vouros, G.A.: Integration of mobility data with weather information (2019)
13. Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., Bengio, Y.: Batch normalized recurrent neural networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2657–2661. IEEE (2016)

14. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. *R News* **2**(3), 18–22 (2002)
15. Makridakis, S.: A survey of time series. *Int. Stat. Rev./Revue Internationale de Statistique* **44**, 29–70 (1976)
16. Makridakis, S., Hibon, M.: The m3-competition: results, conclusions and implications. *Int. J. Forecast.* **16**(4), 451–476 (2000)
17. Min, W., Wynter, L.: Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C: Emerg. Technol.* **19**(4), 606–616 (2011)
18. Moreira-Matias, L., Gama, J., Ferreira, M., Damas, L.: A predictive model for the passenger demand on a taxi network. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems, pp. 1014–1019. IEEE (2012)
19. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi-passenger demand using streaming data. *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1393–1402 (2013)
20. Ntoutsis, E., Mitsou, N., Marketos, G.: Traffic mining in a road-network: how does the traffic flow? *Int. J. Bus. Intell. Data Min.* **3**(1), 82–98 (2008)
21. Sanders, N.R.: Measuring forecast accuracy: some practical suggestions. *Prod. Inventory Manage. J.* **38**(1), 43 (1997)
22. Seltman, H.: Simple linear regression. Chapter 9, 217–240 (2015)
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
24. Tong, Y., et al.: The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662. ACM (2017)
25. Unnikrishnan, V., et al.: Entity-level stream classification: exploiting entity similarity to label the future observations referring to an entity. *Int. J. Data Sci. Anal.* 1–15 (2019). <https://doi.org/10.1007/s41060-019-00177-1>
26. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp? In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–6. IEEE (2016)
27. Xu, J., Wong, S., Yang, H., Tong, C.O.: Modeling level of urban taxi services using neural network. *J. Transp. Eng.* **125**(3), 216–223 (1999)
28. Yao, H., et al.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

