Ride Hailing Service Demand Forecast by Integrating Convolutional and Recurrent Neural Networks

Zinat Ara

Department of Information Sciences and Technology
George Mason University
Fairfax, VA, USA
zara@gmu.edu

Mahdi Hashemi

Department of Information Sciences and Technology
George Mason University
Fairfax, VA, USA
mhashem2@gmu.edu, ORCID: 0000-0003-0212-0228

Abstract—**Ride hailing services, such as Uber, Lyft, and Grab have become a major transportation mode in the last decade. Current ride demand is one of the major factors in such services' pricing algorithm. Therefore, forecasting future travel demand for such services is essential to both drivers and riders. This study constructs a deep learning based model for ride hailing demand forecast aiming to achieve high accuracies in solving similar problems. This study attempts to address a limitation in existing ride hailing demand prediction models, where the area is divided into a rectangular grid and all travel demand forecasts are made between rectangular cells, rather than city neighborhood zones. The proposed model forecasts travel demand between city neighborhood zones. The forecast model integrates convolutional and recurrent neural networks and forecasts the demand for each pickup-destination pair for a particular hour, during the next day, by observing the demand over the past two weeks for that particular hour. Our experiments with a real-world hire vehicle dataset in New York City showed that the proposed model outperforms the CNN and LSTM models up to 18.41 % in RMSE and 22.65% in R^2 values.**

Keywords- travel demand prediction; deep learning; convolutional neural network; recurrent neural network

I. INTRODUCTION

Companies, such as Uber, Lyft, Juno, Gett, Grab that allow passengers to request rides from mobile applications are known as ride-hailing service companies. Online ride hailing services have rapidly grown in popularity in recent years. Receiving a correct approximation of travel demand at a certain time can help a passenger schedule their future trips more efficiently. On the other hand, drivers employed by the ride hailing services can choose to work at times with high demand to maximize their earnings. This paper proposes a deep learning method combining convolutional neural network and bidirectional long short term memory (biLSTM) for predicting travel demand of ride hailing services.

Space and time are the two indispensable dimensions of urban ride-hailing demand prediction. For instance, urban ride-hailing is always strongly linked between the public transport areas and the tourist regions, and between the residential and the industrial regions. In the peak hours of workdays morning and the peak hours of workdays evening, requests from residential and industrial regions increases significantly. In addition to the association between space and time, changes of urban ride-hailing demand are also affected by many other factors including weather, holidays, and unforeseen incidents such as sport events or national occasions. There have been extensive studies in traffic data prediction, including traffic volume, taxi pick-ups, and traffic in/out flow volume. With the fast developing of machine learning models, prediction models based on neural network is getting more attention. Existing methods on traffic prediction have only considered spatial relation (e.g., using CNN) or temporal relation (e.g., using LSTM) independently and state-of-art results are achieved in the work compared with the traditional statistical learning methods. The main advantage of CNNs that make them suited to forecasting time series is the ability to use filters to compute dilations between each cell. Considering the spatial relation which is the size of the space between each cell, allows the neural network to better understand the relationships between the different observations in the time series. In this paper, we are combining bidirectional LSTM with CNN for predicting ride hailing demand. Our work is different from the existing ones [1, 2, 3] as we are not using longitude-latitude to create grids for defining the location and the method does not require any graphical representation or image data as input. A rectangular division of a map can result in grids which fall into two different regions or have a majority part of it outside the land area. In either of these cases the demand calculation can suffer a level of inaccuracy. The study divides the total space into regions using the NYC Taxi Zones dataset which creates partitions based on NYC department of city planning's neighborhood tabulation areas. Pickup-destination pairs are created according to their regions and multiple one-dimensional arrays considering time period fit to be the input for our CNN. Applying filters, we convolve these one-dimensional matrices and then run biLSTM for each of them to predict ride hailing demand. Most of the studies have applied unidirectional LSTM to handle similar problems but bidirectional will run the inputs in two ways and can understand the prediction context better [4].

II. RELATED WORK

Machine learning (ML) and prediction models are used by researchers to make transportation systems more intelligent [5, 6, 7, 8, 9, 10, 11, 12]. Lam and Liu [13] used the discrete choice model to analyze the correlation between dynamic pricing and waiting time in densely populated areas of New York. Gerte et al. [14] examined the demand for the ride hailing service using

a panel based random effects model in order to capture both heteroscedasticity and autocorrelation effects. The major challenge of statistical approaches is a lack of predictive accuracy, particularly under a complex data environment with different data sources.

Travel demand data has periodicity, so historical travel demand is used to predict future travel demand. Ma et al. [15] utilized CNN on images of traffic speed for the speed prediction problem. In these methods convolution network is applied on the entire city without any partitions and predictions were done based on all the regions. As a result, taking account of unrelated regions for the prediction of the target area may downgrade the performance. LSTM itself is widely used to process time series data and traffic prediction. Yu et al. [16] applied long short-term memory network and autoencoder to capture the temporal dependency for predicting the traffic volume particularly for peak-hour and post-accident scenarios. However, the spatial relation is not measured in these scenarios.

To capture both spatial and temporal dependences simultaneously in one end-to-end training model, researchers have made numerous attempts in recent years. Shi et al. [17] proposed the conv-LSTM network, which combined CNN and LSTM in one sequence to sequence learning framework and the results showed that the conv-LSTM outperformed fully-connected LSTM on the basis of learning the complex spatio-temporal features. A study by Ke et al. [18] applied the random forest framework to select the exogenic variables, ranking their significance. In addition, the image intensity was examined which was retrieved from the map sequences of travel time rates using CNN and LSTM tools for short time passenger demand forecasting. Rodrigues et al. [20] presented a deep learning architecture merging the text information with time-series data and applied the approach to the problem of taxi demand forecasting in event areas.

Previous methods are mainly designed to predict the taxi demand in a specific region and ignore the importance of movement direction between different areas [21]. Rare works have been done on the prediction considering all combination including destination and inter-region demands. Liu et al. [22] aims at predicting the taxi demand between all region pairs in a future time interval. An approach contextualized spatial-temporal network is proposed and proved to be effective in predicting taxi demands both in origin and destination. However, region partition is done by grid based method and the spatial and temporal information of taxi demands has not been fully taken into consideration in this case. Instead of using traditional matrix Chu et al. [23] developed and manipulated origin-destination (OD) based tensor to represent OD flows and applied convLSTM model to predict demand. Grid wise division is applied for measuring OD tensor. Guo [24] proposed an integrated CNN-BiLSTM-Attention based model to predict taxi demand. Pengfeng et al. [21] divided the urban area into H x W grids based on the longitude and latitude. While calculating travel demand, they have considered both pickup and destination location to create a demand matrix. All of these studies divide area's latitude and longitude into n×n geographical rectangles to consider the regions. But a region's map is not rectangular shape and dividing it as this can result in grids falling into two different regions or have a majority part of it outside the land area. In these cases, demand calculation will be inaccurate.

This paper is considering pickup-destination pairs according to their regions and create multiple one-dimensional arrays for different time periods fit to be the input for our CNN. Then adding bidirectional LSTM layers for each of them to predict ride hailing demand which will run the inputs in two ways to better understand the prediction context.

III. DATA DESCRIPTION

One of our primary datasets is collected from NYC Open Data source which provides a wide range of traffic datasets in different formats. We have chosen 2018 (January-December) for hire vehicles trip data. These records are generated from the Ride Hailing Services (RHS) trip record submissions made by different commuter vehicles. The RHS trip records include fields capturing the pick-up date, time, and taxi zone location ID, which correspond with the NYC taxi zones open dataset. Each row represents a single trip in a hired vehicle service [25]. Secondly, NYC taxi zones data corresponds to the pickup and drop-off zones, or location IDs, included in the yellow, green, and RHS trip records (Uber, Lyft) published to Open Data. The taxi zones are based on NYC Department of City Planning's Neighborhood Tabulation Areas (NTAs) and are meant to approximate neighborhoods, so one can visualize which neighborhood a passenger was picked up in, and which neighborhood they were dropped off in. This dataset provides the geolocations of neighborhoods where the output is multi polygons each representing different region with unique location ID.

Figure 1   Pickup demand pattern in NYC in year 2018

Figure 1 shows the number of requests along Y axis in 2018 for RHS in New York city from January to February based on pickup datetimes (X axis). From the pattern of the distribution with respect to time it is certain that passenger requests are higher in weekdays and comparatively lower in weekends. The features which have been considered for training in this study are timestamp, hour, demand, and weekend. These features were chosen for their higher impact over demand prediction and availability.

IV. METHODOLOGY

A. Non-grid partition

Most of the related studies have [21, 3, 22, 23, 24] used grid wise rectangular division method where they selected latitude and longitude value of a city and then divided into n×n geographical squares. For simplicity we define each square block unit as pixel. The overall travel demand about one city is reflected by each pixel's demand. The grid wise division in geographical area is shown in Figure 2. But this method has few problems in following cases:

1. First problem will be the resolution; a pixel may contain regions with high demand and regions with low demand so they will combine in the same pixel and that pixel will basically show their average demand. The larger the pixel gets this problem becomes more severe (Figure 2).

2. A pixel may contain only regions with water, no land area. Demand will be zero in that pixel. These types of pixels cannot be disregarded in grid wise division method.

3. If a pixel includes both land and water area, that pixel will not provide the actual demand of the region. There is a high chance that demand counts might decreased for considering non-regional areas into the same pixel.

A geographical map is not a perfect geometric rectangular shaped object, so partitioning it based on grid is less realistic option. To avoid this problem geographic information of an area can be applied to define the boundaries where the area longitude latitude values are formatted as polygons. This represents each area boundary precisely and will be useful to solve this problem. We are utilizing the NYC taxi zones which is based on city planning's neighborhood tabulation area and where each region is partitioned into polygon shape to its approximate neighborhoods. Instead of using grid based division, partitioning regions based on its actual geolocation information is more effective in terms of accuracy for demand calculation. Figure 3 plots the map using geolocations which represents the taxi zones of NY city and each region has a unique identification number. In our dataset each trip information for RHS contains pickup and destination location ID which comes from the polygons. We are calculating pickup-destination pairs according to their regions and create multiple one-dimensional arrays considering time period fit to be the input for our CNN. The raw data is processed and features are extracted. Then transformed into a matrix as the format of input data of the prediction framework. r is the number of inter area hire vehicle request from a pickup location i to destination j during a time period t which is represented by r_{ij}^t. Similarly, for each pickup-destination pair, we will calculate the corresponding r at time t. Finally, the total number of requests in the city R^t at period t which will consider each pair count and can be represented in form a matrix:

$$R^t = \begin{bmatrix} r_{11}^t & \cdots & r_{n1}^t \\ \vdots & \ddots & \vdots \\ r_{n1}^t & \cdots & r_{nn}^t \end{bmatrix}$$

where n is the number of total area polygons or regions. The demand prediction problem aims to predict the demand at time
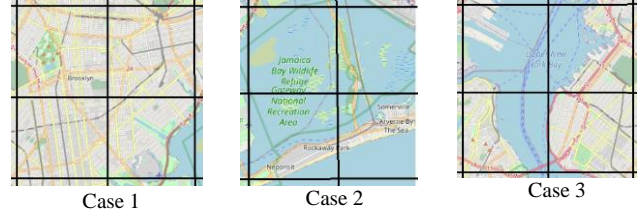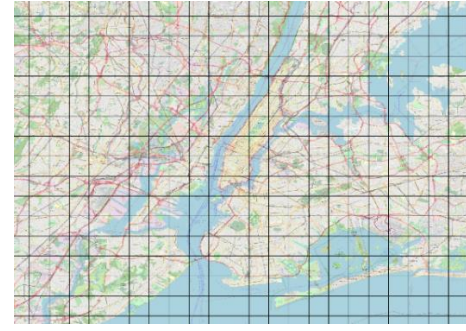

Case 1   Case 2   Case 3

Figure 2   Grid based partitioning and problems associated thereof

interval $t + k$, where t is current timestamp and k is the lag size. In addition to historical demand data, we also incorporated influential features such as time of day, day of week, holiday, weekend and areas of pickup and destination.
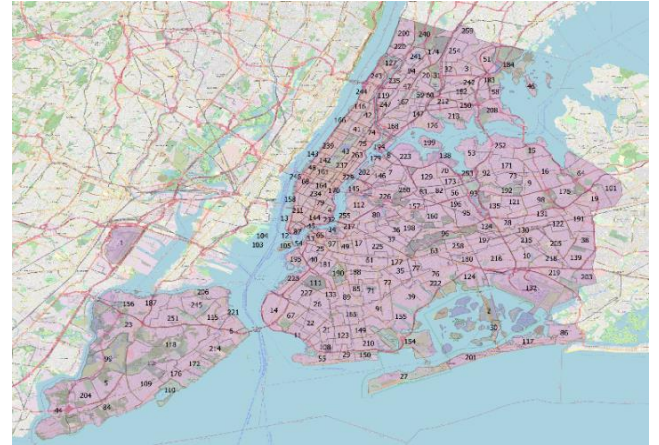

Figure 3   Partitions using New York city neighborhood zone

Following are the primary steps of our approach for demand prediction:

1. Extract pair-wise demand for region based pickup-destination locations.

2. Create 2D matrix for each time interval's RHS demand so that each matrix is a temporal snapshot of demand for all pairs.

3. Each temporal snapshot will be given input to 1D multivariate CNN model.

4. Compute feature information using CNN, store feature output in each subsequent array.

5. Use each subsequence with feature information from CNN as a new feature for biLSTM, memorize long-sequence features from each subsequence, use to predict the RHS demand for input pairs in future time interval.

B. One dimensional convolution network

A modified version of 2D CNNs called 1D Convolutional Neural Networks (1D CNNs) has recently been developed [3, 21]. Rather than matrix operations, forward and backward propagation in 1D CNNs require simple array operations. For this reason, the computational complexity of 1D CNNs is significantly lower than 2D CNNs. 1D CNNs with relatively shallow architectures (i.e., small number of hidden layers and neurons) are able to learn challenging tasks faster.

Main components of 1D CNN model used in this study are 1) convolution layers where both 1 dimension convolutions and sub sampling or pooling occur, and 2) fully connected layers that are indistinguishable to the layers of a typical Multi-layer Perceptron (MLP). CNN layer first performs a series of convolutions, the sum of this operation is passed through the activation function, f, followed by the pooling operation. This is the main difference between 1D and 2D CNNs, where 1D arrays replace 2D matrices for both kernels and feature maps. As a next step, the CNN layers process the raw 1D data and learn to extract such features which are used in the prediction task performed by the MLP-layers. The detailed explanation about the forward and backward propagation functions can be found in [26].

C. Bidirectional LSTM

Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction and the other in a backwards direction. biLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm. The model we are using in the experiment consists of an output layer, a dense layer and a biLSTM layers. The input of the model requires a fixed-length vector. The dense layer is used to compress the dimension of the output vector of the biLSTM layer. The output layer collects the vector from the dense layer and outputs the desired regression value.

D. Proposed Model: CNN-biLSTM

The convolution layer in proposed model extracts the correlation between the input features and captures spatial dependency. Maximum pooling layer calculates the maximum value in each patch of each feature map. We flatten the output of the convolutional layers to create a single long feature vector. Next this tensor is passed through a bidirectional LSTM layer, which interprets the context from both directions. And it is connected to the final layer, which is known as dense layer. The network solves a regression task and final outcome of the model is number of demands for specific regions at different timestamps. Figure 5 illustrates the whole structure of the proposed model. As a desirable neural network to deal with the long-term dependencies in time series, LSTM is designed to overcome the vanishing gradients through a special gating mechanism [24]. LSTM avoids long term dependency problems
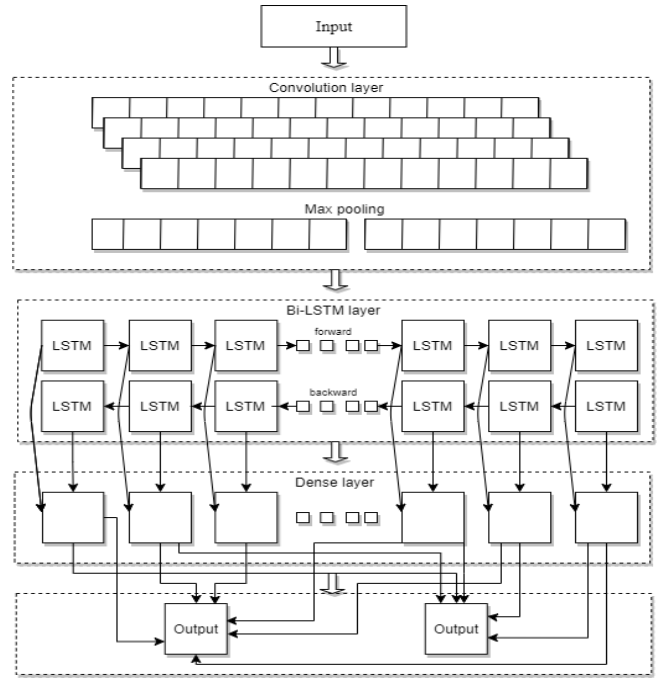

Figure 5   CNN-biLSTM architecture

by bringing the cell state, gate, and other schemes. The basic units of the LSTM network consist of three doors (input gate, output gate, and forget gate) and two memories (long-term memory and short-term memory). The input gate aims at selecting the needed new information and adds it to the cell state. The forget gate tends to remove the information that is no longer required by the memory cell, while the output gate decides what kind of necessary information in the cell should be output. Generally, the gating mechanisms can ensure the cells in LSTM network to store and update the essential information over long periods of time.
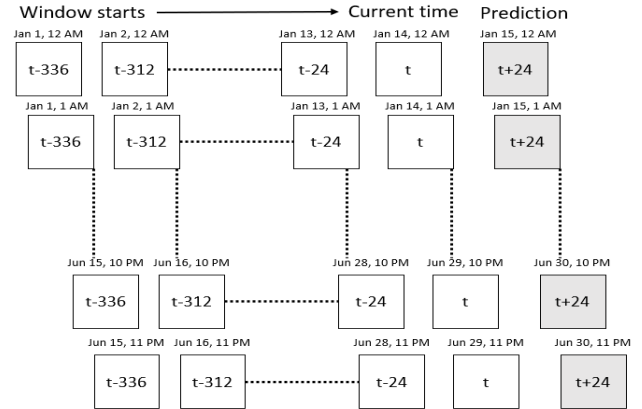

Figure 6   An example of the temporal window structure

Figure 6 describes the temporal window structure this paper uses for predicting the travel demand between a particular regional pair. Current time is noted by t. The training starts from 2 weeks ago, which in hour count is 24*14=336. Therefore, the first node is labeled t-336 and the hour count is incremented by 1 day or 24 hours for each consecutive node. The next training

row starts by shifting the entire window by one hour which results in the prediction of the next hour. The diagram shows the prediction window starting from January 1, 12:00 AM leading up to predicting the demand for January 15, 12:00 AM horizontally and vertically shifting 1 hour from above row up to June 15 11:00 PM to predict demand for June 30 11:00 PM. The paper also implements some baseline models for comparison. A typical LSTM model and single one dimensional convolutional network consisting single CNN, max pool, flatten and dense layers. Another baseline model is combining CNN and LSTM models together.

V. RESULTS AND DISCUSSION

In order to compare our framework with the other baseline models, the experiments are conducted on a real dataset. No. of parameters for LSTM, CNN layers are 128, for dense layer 97. 32 filters and kernel size 3 is set for convolution layers. New layer has been added step by step with its previous layer, parameters are tuned and result is observed with new mode. In order to evaluate the predicted performance, the root mean square error (RMSE) and coefficient of determination, or R^2 are used. Root mean square error measures how much error there is between two data sets. The smaller an RMSE value, the closer predicted and observed values are:

$$Y_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{pred}(i) - X_{act}(i))^2} \tag{10}$$

The coefficient of determination, or R^2, is a measure that provides information about the goodness of fit of a model.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \tag{11}$$

Table 3 demonstrates the comparison among all models for 2018 NYC RHS dataset. Multiple pickup-destination pairs are compared and result for sample 5 pairs are shown. We considered same region based pair where pickup and destination ids are same and also different id based pairs. It is noted that CNN-biLSTM model outperforms the baseline models in terms of RMSE and R^2 for any pair. R^2 scores vary for different pairs but it is always closer to 1 which indicates our model works as expected and is a good fit. The table shows that for some pairs sequential model LSTM achieved better result than standalone CNN model and vice versa, however combining them both always showing a better prediction.

TABLE 3 RMSE AND R^2 VALUE COMPARISON AMONG ALL MODELS FOR 2018 NYC RHS DATA

Regional pair	Model	R^2 value	RMSE value
Pair 1	LSTM	0.717	76.414
	CNN	0.713	45.646
	CNN-LSTM	0.718	33.837
	CNN-biLSTM	**0.720**	**33.705**
Pair 2	LSTM	0.722	17.838
	CNN	0.596	21.501
	CNN-LSTM	0.714	18.088
	CNN-biLSTM	**0.731**	**17.542**
Pair 3	LSTM	0.834	10.957
	CNN	0.831	11.039
	CNN-LSTM	0.838	10.822
	CNN-biLSTM	**0.840**	**10.744**
Pair 4	LSTM	0.674	24.916
	CNN	0.704	23.737
	CNN-LSTM	0.728	22.768
	CNN-biLSTM	**0.729**	**22.706**
Pair 5	LSTM	0.736	20.095
	CNN	0.667	22.577
	CNN-LSTM	0.740	19.947
	CNN-biLSTM	**0.742**	**19.866**

The proposed model CNN-biLSTM outperforms all models which indicates this model is able to capture spatial and temporal correlations successfully.
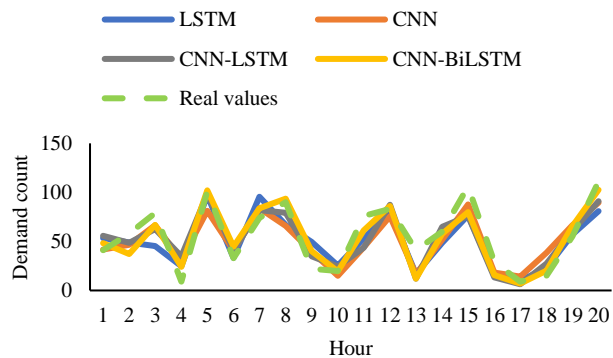

Figure 7 Comparing travel demand in one day for a specific origin-destination pair

Figure 7 demonstrates the travel demand prediction curve for one day for a specific region pair in 2018. Prediction curves of all models are compared with the real value. From the illustration it is noted that CNN-LSTM and CNN-biLSTM models exhibit a trend similar to real data. Figure 8 depicts the percentage decrease in RMSE for out proposed model comparing to baseline models. This measurement is performed on 2018 RHS dataset. For pair 2 our model achieved the best accuracy which is 18.41% against the CNN model. Similarly, for this pair the proposed model achieved 22.65% improvement over the R2 value. Combining CNN and LSTM has achieved better performance than standalone CNN and LSTM models, applying bidirectional recurrent network (biLSTM) further improves the performance.

Among the state of the art models for ride hail service demand prediction, Liu et al. [22] achieved an RMSE of 19.85 in predicting only the origins and an RMSE of 1.32 in predicting demand between origin-destination pairs on NYC-TOD dataset. Chu et al. [23] achieved an RMSE of 1.015 in predicting taxi demand in NYC using a MultiConvLSTM. Shu et al. [21] applied CNN-LSTM model on historical data in Haikou (China) and obtained lower RMSE values than LSTM. The dataset applied in all these models is presented for grids and not neighborhood. On the other hand, the dataset used in our study is based on city neighborhood zones. There is no method to convert the data from grid format to neighborhood format, or vice versa, without making unrealistic assumptions and significantly lowering the data accuracy. Therefore, a

meaningful comparison between the accuracy of the grid-based versus neighborhood-based models would not be possible.
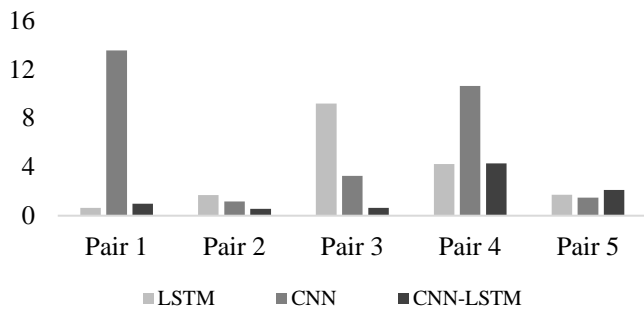

Figure 8   Percentage-wise RMSE decrease (improvement) in the proposed model (CNN-biLSTM) versus LSTM, CNN and CNN-LSTM.

VI.   CONCLUSION AND FUTURE WORK

Travel demand modeling is an inherent part of smart transportation system. Forecasting travel demand can help us manage the hot spot of passenger demand in the next period, balance supply and demand and schedule vehicle resources for passengers. In this paper, a convolutional and recurrent network based deep learning model for ride hailing service demand prediction is proposed that takes advantage of both temporal and spatial properties on areal dataset. Proposed models' performances are significantly beyond baseline models, confirming that it is better and more flexible for the travel demand prediction. In addition, we found that rectangular grid based partition method has several issues to calculate demand. Therefore, choosing a suitable partitioning method to predict travel demand is vital and has significant impact on increasing accuracies. In future, we will further improve the model and test with different pattern data sets. We want to extend the idea of utilizing the geojson information for partitioning method while demand calculation. For calculation, including data from adjacent cities can be used to verify model suitability.

REFERENCES

[1] T. Kim, S. Sharda, X. Zhou and R. M. Pendyala, "A stepwise interpretable machine learning framework using linear regression (LR) and long short-term memory (LSTM): City-wide demand-side prediction of yellow taxi and for-hire vehicle (FHV) service," *Transportation Research Part C,* 2020.

[2] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye and Z. Li, "Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction," *machine learning applications for shared mobility,* 2018.

[3] D. Wang, Y. Yang and S. Ning, "DeepSTCL: A Deep Spatio-temporal ConvLSTM for Travel Demand Prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazi, 2018.

[4] Y. Bao, Z. Huang, L. Li, Y. Wang and Y. Liu, "A BiLSTM-CNN model for predicting users' next locations based on geotagged social media," *International Journal of Geographical Information Science,* pp. 1-22, 2020.

[5] M. Hashemi, "Emergency evacuation of people with disabilities: A survey of drills, simulations, and accessibility," *Cogent Engineering,* vol. 5, no. 1, p. 1506304, 2018.

[6] M. Hashemi, "Automatic inference of road and pedestrian networks from spatial-temporal trajectories," *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, no. 12, p. 4604–4620, 2019.

[7] M. Hashemi, "Dynamic, stream-balancing, turn-minimizing, accessible wayfinding for emergency evacuation of people who use a wheelchair," *Fire Technology,* vol. 54, no. 5, p. 1195–1217, 2018.

[8] M. Hashemi, "Reusability of the output of map-matching algorithms across space and time through machine learning," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 11, p. 3017–3026, 2017.

[9] M. Hashemi, "A testbed for evaluating network construction algorithms from GPS traces," *Computers, Environment and Urban Systems,* vol. 66, p. 96–109, 2017.

[10] M. Hashemi, "Intelligent GPS trace management for human mobility pattern detection," *Cogent Engineering,* vol. 4, no. 1390813, 2017.

[11] M. Hashemi and H. A. Karimi, "Collaborative personalized multi-criteria wayfinding for wheelchair users in outdoors," *Transactions in GIS,* vol. 21, no. 4, p. 782–795, 2017.

[12] M. Hashemi and H. A. Karimi, "Indoor spatial model and accessibility index for emergency evacuation of people with disabilities," *Journal of Computing in Civil Engineering,* vol. 30, no. 4, p. 04015056, 2016.

[13] C. Lam and M. Liu, "Toward Inclusive Mobility: Ridesharing Mitigates Geographical Disparity in Transportation," *SSRN 2997190,* 2020.

[14] R. J. Gerte, K. C. Konduri and N. Eluru, "Is There a Limit to Adoption of Dynamic Ridesharing Systems? Evidence from Analysis of Uber Demand Data from New York City," *Transportation Research Record Journal of the Transportation Research Board,* 2018.

[15] X. Ma, Z. Dai, Z. He, J. Na, Y. Wang and Y. Wang, "Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction," *Sensors,* 2017.

[16] R. Yu, Y. Li, C. Shahabi, U. Demiryurek and Y. Liu, "Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting," in *Proceedings of SIAM International Conference on Data Mining*, 2017.

[17] X. Shi, Z. Chen, H. Wang and D.-Y. Yeung, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Advances in Neural Information Processing Systems,* pp. 802-810, 2015.

[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: Emerging Technologies,* vol. 85, pp. 591-608, 2017.

[19] T. Kim, S. Sharda, X. Zhou and R. M. Pendyala, "A stepwise interpretable machine learning framework using linear regression (LR) and long short-term memory (LSTM): City-wide demand-side prediction of yellow taxi and for-hire vehicle (FHV) service," *Transportation Research Part C,* 2020.

[20] F. Rodrigues, . I. Markou and F. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach," *Information Fusion,* vol. 49, pp. 120-129, 2019.

[21] P. Shu, Y. Sun, Y. Zhao and G. Xu, "Spatial-Temporal Taxi Demand Prediction Using LSTM-CNN," in *IEEE International Conference on Automation Science and Engineering*, 2020.

[22] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang and L. Lin, "Contextualized Spatial-Temporal Network for Taxi Origin-Destination Demand Prediction," *IEEE Transactions on Intelligent Transportation Systems,* pp. 1-13, 2019.

[23] K.-F. Chu, A. Y. S. Lam and V. O. K. Li, "Deep Multi-Scale Convolutional LSTM Network for Travel Demand and Origin-Destination Predictions," *INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 21, no. 8, pp. 3219-3232, 2020.

[24] X. Guo, "Prediction of Taxi Demand Based on CNN-BiLSTM-Attention Neural Network," in *International Conference on Neural Information Processing*, 2020.

[25] "https://data.cityofnewyork.us/Transportation/2020-For-Hire-Vehicles-Trip-Data-January-June-/m3yx-mvk4/data," [Online].