CNN GRU for Short Term Transit Usage Prediction in Chicago

Mary Rose Fissinger
Massachusetts Institute of Technology
Cambridge, MA
mrf1@mit.edu

Abstract

This work uses deep learning to perform short-term prediction of public transit ridership in the city of Chicago. A CNN GRU model is applied to capture spatial and temporal dependencies among the data. The prediction performance for specific areas of the city is shown to be superior to classical statistical models. Investigations of the output lead to concerns about potential codification of discrimination by deep learning models in this context, and a gradient approximation technique reveals that lower demand in the most highly trafficked area of the city leads the model to assume an overall decrease in travel across the city compared to baseline, while inflated demand in this area leads the model to assume demand is transferred there from other parts of the city.

1. Introduction

The massive popularity of rideshare companies like Uber and Lyft has revealed some of the ways in which technology can transform mobility offerings, and the extent to which people desire convenience and real-time information from their transportation providers. The rideshare phenomenon has provided fuel among the public transit community for exploration of ways to incorporate technology into mass transit service so that it can offer some of the same real-time flexibility and information as rideshare companies. While mass transit cannot offer this in the form of door to door service, it can leverage the huge amounts of mobility data available today from smart transit card systems to better understand movement and public transit usage patterns and provide updates or service alterations that are more targeted to where they are most relevant. One key aspect of this is understanding where demand for public transit will be in the short term. The ability to predict future usage of a system enables better responses when service is disrupted, whether that is in the form of updates to users most likely to be affected, dispatching of additional vehicles, or both.

This paper aims to solve the short-term public transit usage prediction problem for the city of Chicago using deep learning methods. Deep learning methods, specifically neural networks, have shown impressive predictive power in nearly every field in which they have been applied, including transportation [9]. Over the past several years, computer science researchers have developed many new architectures for neural nets, some of which are particularly well-suited for transportation questions. Among these are architectures that can extract and use both spatial and temporal features of data. Convolutional Neural Nets (CNNs) are the primary deep learning tool for handling spatial data, as they take in images and consider each pixel in the context of those surrounding it and its position in the image. Recurrent Neural Nets (RNNs) are often used for modeling time-series data, as their structure retains information from the entire sequence of data as it ingests new inputs. In this paper, we combine these two structures into an architecture that takes in images, performs a convolution, flattens the output, and passes it through a Gated Recurrent Unit layer, which is a modification of the traditional RNN that accounts for the vanishing and exploding gradient issue. We investigate a few variations of this architecture structure, including a model with 2 layers instead of 1, and models that take in 3, 5, and 10 time steps as an input in order to predict demand at the next timestep. We also compare their predictive power to that of a classical statistical model for a given area of the city of Chicago.

The rest of this paper is organized as follows: Section 2 reviews the relevant literature on the topic, Section 3 presents the model architecture in more detail and gives background on the data and local context of Chicago, Section 4 presents the results and provides some analysis, and Section 5 offers concluding thoughts.

2. Related Studies

There is a deep literature on classical statistical models from the realm of spatial econometrics that attempt to capture the value of spatial information in temporal forecasting. These typically involve the inclusion of spatially lagged dependent variables, spatially lagged explanatory variables, or

spatially lagged error terms. Anselin's body of work on this topic spans many of the questions around how to think about modeling time-series data with spatial effects, including how to determine the nature of spatial effect (dependence vs heterogeneity) or the extent of the spatial dependencies (local vs. global) [2], and subsequently how these determinations should guide the modeling process. The models used for spatiotemporal data are largely variations on time-series models, notably Autoregressive (AR) models and Autoregressive Integrated Moving Average (ARIMA) models, adapted to incorporate spatial information. Kamarianakis and Prastacos [8] summarized a few key spatial variants of these, specifically Space-Time ARIMA and Bayesian Vector Autoregressive (BVAR) models, which have been used to model everything from traffic flow volumes to regional development. All of these models, however, require assumptions to be made about the distributions and relationships among error terms and, in many cases, included or dependent variables. These assumptions can limit the predictive power of the models and, if wrong, damage the interpretability of these models –which is their advantage over Machine Learning, and especially deep learning – by giving incorrect parameter estimates. It stands to reason that if we can glean some insight on the nature of the spatial dependence from models in which limiting assumptions are not imposed and predictions are closer to reality, we might be able to learn something about mobility patterns that is difficult to capture via traditional methods.

This leads us to Machine Learning, and more specifically, neural network architectures, which, as mentioned, have been shown to outperform traditional econometric models in predictive power time and time again. Artificial Intelligence (AI) models that consider both spatial and temporal dimensions are key to such technologies like autonomous vehicles, which need to be able to take in video information and, like humans, have some confidence in what will occur next: Will the oncoming car turn? Will the pedestrian step into the road? As such, significant work has been done in this realm, with most hinging on some combination of RNNs, specifically GRUs and another variant, Long Short-Term Memory (LSTM) models, and CNNs. Many newer architectures combine elements of both. Examples include ConvLSTMs [1, 13], Quasi-Recurrent Neural Nets [4], and Dilated Recurrent Neural Nets [5]. Other research has attempted to adapt the internal structure of traditional CNNs to make it more suitable for sequence modeling, rather than introduce an RNN structure, which is hindered by the inability to do parallel processing [3].

In the past few years, deep learning architectures designed to model spatiotemporal data have been used to tackle a variety of long-standing transportation problems. The major areas of CNN and RNN model applications have been traffic flow or speed prediction on road networks [6, 11], mode assignment from smartphone traces, [14], individual path or next location prediction [10], and demand prediction [7, 12, 15]. The latter group of papers is the most relevant to this work, and researchers have employed complex architectures to capture patterns we know to exist in urban mobility patterns. Specifically, Ma et al. predict metro ridership in Beijing by combining a CNN with a bi-directional LSTM. This paper does not claim to offer an improvement on these methodologies, but rather adds to the still relatively small body of literature regarding deep learning methods applied to public transportation. While transportation is, as outlined, as clear application of many of these model structures, especially in the realm of mass transit, the data is often not in the hands of people whose priority is developing deep learning models. Thus, particularly in the United States, few studies exist applying deep learning to ublic transit applications. This paper implements a relatively straightforward method to show the applicability of deep learning to short term public transit ridership prediction in Chicago and illuminate some considerations when developing a model for such purposes. A worthwhile future exercise would be to replicate papers predicting transit ridership in other cities to understand the transferability of some of these models.

3. Experiment Setup

3.1. Models

Each of the deep learning models tested in this report has as its first layer a time distributed convolution layer, which enables the extraction of spatial demand relationships from a series of images that are understood to have a temporal relationship. The same filtering process is applied to each image in the input separately, and the temporal relationship is preserved. The output from this layer is then flattened and fed as the input to a GRU, which learns the temporal structure of the data. The GRU structure, like an LSTM model, is an improvement on the classical RNN structure that corrects for the vanishing or exploding gradient problem. As its name suggests, a GRU consists of gating units which control the flow of information by dictating how much of the previous state and how much of the candidate activation are considered when computing the new activation. The candidate activations, in turn, are computed by combining the new input with some portion of information from the previously computed state, modulated by a reset gate. The reset gate is computed as follows:

$$r_t^j = \sigma(W_r \boldsymbol{x}_t + U_r \boldsymbol{h}_{t-1})^j \tag{1}$$

Where $\boldsymbol{x}_t$ is the input for time t, and $\boldsymbol{h}_{t-1}$ is the activation for time $t-1$. This in turn modulates the previous activation via the Hadamard product in computing the candidate

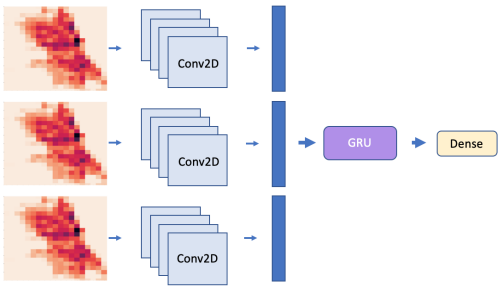

Figure 1. CNN-GRU architecture

activation for time t:

$$\tilde{h}_t^j = tanh(W\boldsymbol{x}_t + U(\boldsymbol{r_t} \odot \boldsymbol{h}_{t-1}))^j \quad (2)$$

The new activation is thus computed by

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \quad (3)$$

where z_t^j is the update gate and calculated similarly to the reset gate:

$$z_t^j = \sigma(W_z\boldsymbol{x}_t + U_r\boldsymbol{h}_{t-1})^j \quad (4)$$

Lastly, the output from the GRU is fed to a dense layer yielding a vector that can be reshaped to be a demand prediction image for the next timepoint.

3.2. Datasets

For this project, we employ 20 weeks' worth of weekday data taken from the Ventra database at the Chicago Transit Authority. The data consists of the origin tap location for every trip taken with a Ventra card (the smart card system of the CTA) between Friday, March 1, 2019, and Monday, July 17, 2019. Each tap was aggregated to a 15-minute time frame and one cell of a 19x22 grid of the city of a Chicago. Figure 2 shows the grid used for this analysis and gives a sense of the density of transit stations within each cell. Figures 3 and 4 give the spatial and temporal distribution of demand throughout the city for one day.

Public transit usage in Chicago reflects the structure of the rail network in that it is very radial, with people coming into the center city ("The Loop") in the morning for work, and heading back out to their homes in the evening. The morning peak period seen in Figure 4 is quite spatially diffuse, while the evening peak is heavily concentrated around the Loop, leading to an overall demand picture like the one seen in Figure 3, with a few cells seeing the bulk of the city's public transit demand.

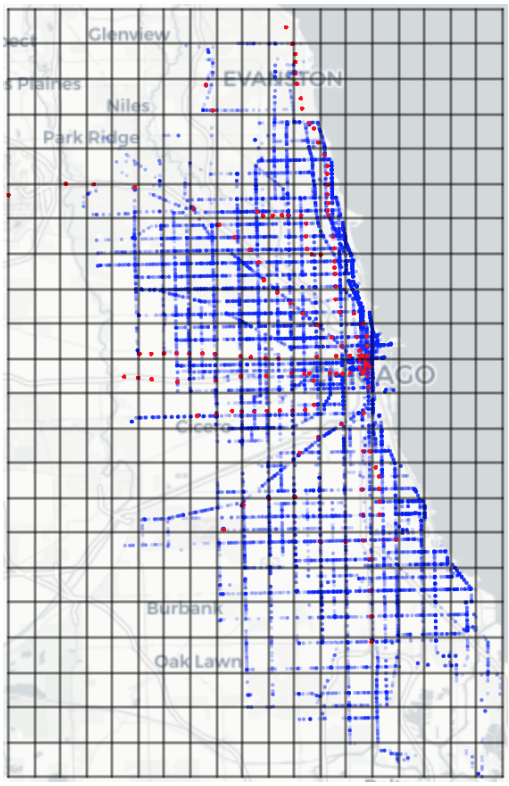

Figure 2. The grid used for this analysis of transit demand in Chicago. This image also depicts the location of every possible tap considered in this analysis.

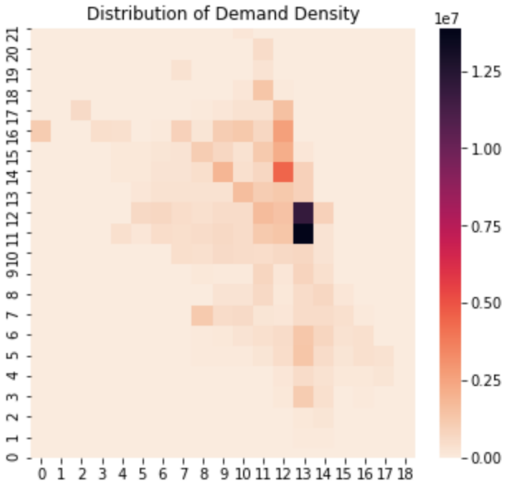

Figure 3. The total number of taps observed within each cell during the study period.

4. Results

4.1. Prediction

The key hyperparameters in the CNN-GRU models are the number of filters and kernel size in each convolution layer and the number of units in the GRU layer. Figure 7 in the appendix shows the MSE for a set of models in which each of these was varied systematically. In short, the number of GRU units had the most impact on model ac-

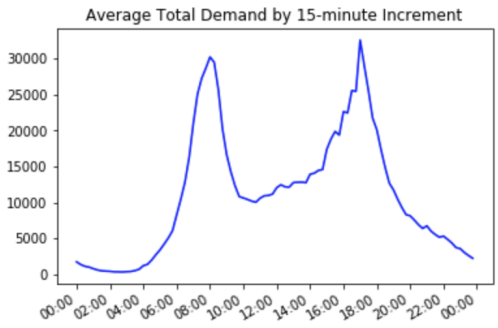

Figure 4. The average number of taps recorded during each 15-minute period of a day throughout the study period.

Model	Training MAE	Training MSE
Basic	$3.94 \cdot 10^{-4}$	$1.67 \cdot 10^{-6}$
2 Layer	$3.84 \cdot 10^{-4}$	$1.78 \cdot 10^{-6}$
5 LB	$4.31 \cdot 10^{-4}$	$2.51 \cdot 10^{-6}$
10 LB	$3.83 \cdot 10^{-4}$	$1.50 \cdot 10^{-6}$

Model	Test MAE	Test MSE
Basic	$-2.79 \cdot 10^{-2}$	$1.34 \cdot 10^{-3}$
2 Layer	$-3.84 \cdot 10^{-2}$	$1.34 \cdot 10^{-3}$
5 LB	$-8.67 \cdot 10^{-2}$	$1.46 \cdot 10^{-3}$
10 LB	$-5.24 \cdot 10^{-2}$	$1.16 \cdot 10^{-3}$

Table 1. Mean Absolute Error and Mean Squared Error for each model, reported in terms of the normalized input data.

curacy when holding other hyperparamters constant, with MSE dropping substantially between 32 and 64 GRU units. For our base model, we used a CNN-GRU with 32 filters in the convolution layer and a 4x4 filter, and a 96-unit GRU. We then investigated the impact of adding another convolutional layer (this time adding a Max Pooling layer after each convolution layer), as well as increasing the size of each input from the 3 previous time steps to 5- and 10 timestep look backs. We found that adding another convolutional layer did not improve the MSE of the prediction, while increasing the look back size did.

We also compared the prediction accuracy of the basic model and the 10 step look back models for a few individual cells to a benchmark statistical model applied to those cells and not taking any spatial information into account. Specifically, we compared the prediction in the highest demand cell and a representative mid-level demand cell for these models to the prediction from an autoregressive model trained individually on the two cells. We find that the deep learning models substantially outperform the AR(p) models in both cases. The MSE on the test set in each of these cases is shown in Table 2.

Model	Mid-Level MSE	High MSE
Basic	83.3	73, 106
10 LB	80.9	55, 672
AR(p)	95.2	224, 286

Table 2. Mean Squared Error (in scaled up demand numbers) for the basic CNN GRU and 10 stop look back CNN GRU as well as an AR(p) model trained individually on just the time-series for each of these cells.

4.2. Analysis

Figures 8 to 11 in the appendix show the residuals for the highest demand cell and a representative mid-level demand cell after running the basic model and the 10 step look back model were run on the test data. We see that for the high demand cell, the magnitude of the error for the 10 step look back model is consistently lower than for the basic model, but the opposite is true for the mid-level demand cell, where the errors are smaller in magnitude and more centered around zero for the basic model. This may suggest a tradeoff between the two: the 10 step look back model better predicts demand in the cells that contain most of the ridership while overestimating demand elsewhere, leading to a smaller MSE than the basic model, which misses the target by a wider margin when the target is large, but fits the mid-level demand cell well.

All of the CNN GRU models predict uniformly 0 demand in cells that see very low levels of demand, or demand that is typically in the single digits per 15-minute increment. This may be a function of the normalization process, which was an unsophisticated one: We simply divided all demand values by the maximum demand ever observed in one 15-minute increment in one cell over the course of the study period. Refining this normalization process may improve the prediction in low-demand cells. The error from these cells, however, is a small portion of the overall error, and thus not a major concern.

We also note that the spatial distribution of the error differs slightly among the models. Notably, as shown in Figure 4.2, while both tend to underestimate demand in the southern part of the city, the basic model simultaneously overestimates demand in the northern part of the city. While it is unclear why the different model structures lead to spatially distinct error patterns, it is important to note that these differences exist. It is especially important in this context, as the southern part of the city is home to most of Chicago's poor and African-American population and has historically been under-served by public transit and other public resources. If a deep learning model were to be adopted by the CTA to direct public transit resources, it would be essential that the model not systematically project higher demand in the wealthier parts of the city and lower demand in the poorer parts. This illustrates a way in which deep learn-

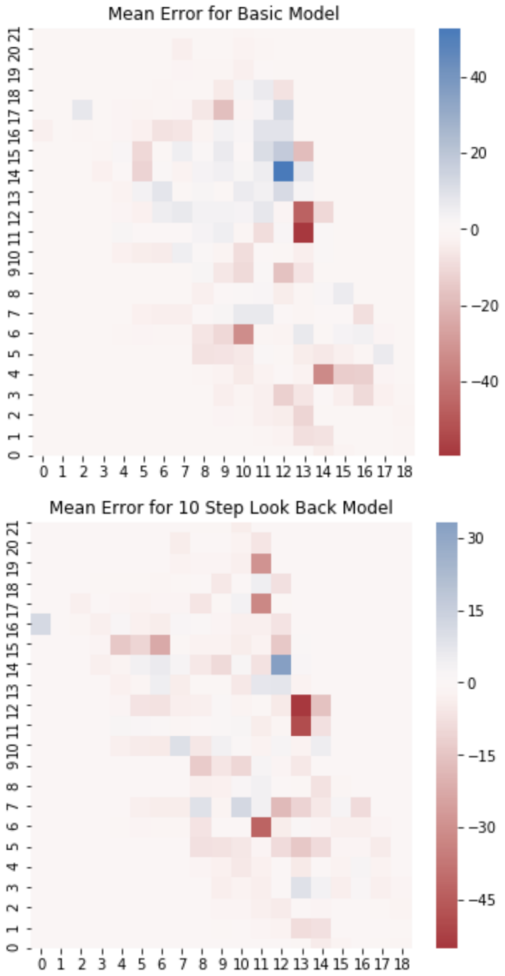

Figure 5. Average error across test data per cell of the city for the Basic CNN GRU model (top) and 10 Step Look Back model.

ing models may codify discrimination that exists in the real world, thus perpetuating the subjugation of certain populations.

4.3. Gradients

DNNs are well-known for being difficult to interpret. While it is certainly true that the parameters of the trained model carry little meaning in and of themselves, unlike classical econometric models, it is still possible to gain some insight into what the model has learned about the spatial and temporal relationship among variables. Here we specifically investigate the impact that the demand level of a single cell has on subsequent demand estimations, according to the basic model. We do that by choosing a slice of real data, in this case one 3 timestep input, and systematically varying the demand of a given cell by the same amount in all 3 timesteps to produce an input that is identical to the real input, but with demand that is suppressed or inflated in the cell of interest. We perform the same adjustment to each of the 3 timestep inputs to ensure that the 3 timeframes are consistent with one another. We can then run the model on the sequence of altered inputs to see how the predicted demand across the city changed as a result.

We do this to investigate the impact on citywide demand prediction of changing the demand in the highest demand cell. We choose one specific set of inputs on which to focus. Namely, we limit our analysis to the input corresponding to the demand prediction for the 10:45-11:00AM time period on the first day of test data. The input consists of three images depicting demand in each cell of the city from 10:00-10:15AM, 10:15-10:30AM, and 10:30-10:45AM.

Figure 6 shows the resulting change in predicted demand, compared with the predicted demand under the true value. We see that the bulk of the change occurs in the cell whose demand we changed, which is unsurprising. It does have a significant effect on a few surrounding cells as well, however. Interestingly, decreasing the demand in the cell of interest leads to decreases in the affected cells, while an increase in the demand in the cell of interest leads to an increase in projects demand for the adjacent cell of similarly regular high demand, but a slight decrease in projected demand for the cells to the northwest. This may suggest that lower demand in the key cell typically signals lower overall travel, leading the model to predict decreased or unchanged demand in the surrounding areas. Higher demand in the key cell, however, seems to signal not overall increased travel, but a shift of demand that might have occurred elsewhere to the downtown area. More investigation is required here.

5. Conclusion

We have shown that CNN GRU models have significant predictive power when used to forecast short term public transit usage in Chicago. When the MSE for a single grid cell is compared with the MSE for an AR(p) model trained on the historical demand for just that grid cell, the deep learning models vastly outperform the AR(p)model. We also uncovered some interesting differences between the basic model with a 3 time step input and the model with a 10 step input. First, the 10 step look back model better predicts demand in the cell with the highest demand levels in the city. In contrast, the basic model largely matches the pattern of demand here but underestimates it, while performing better in cells with middling demand. The 10 step look back model overestimates demand in the mid-level demand cell. Additionally, the basic model tends to underestimate demand in the south part of the city and overestimate demand in the north, while the 10 step model more consistently underestimates demand. This is an interesting manifestation of potentially discriminatory artificial intelligence: if this model were to dictate the allocation of resources, it would likely direct resources to the northern, more wealthy part of the city, at

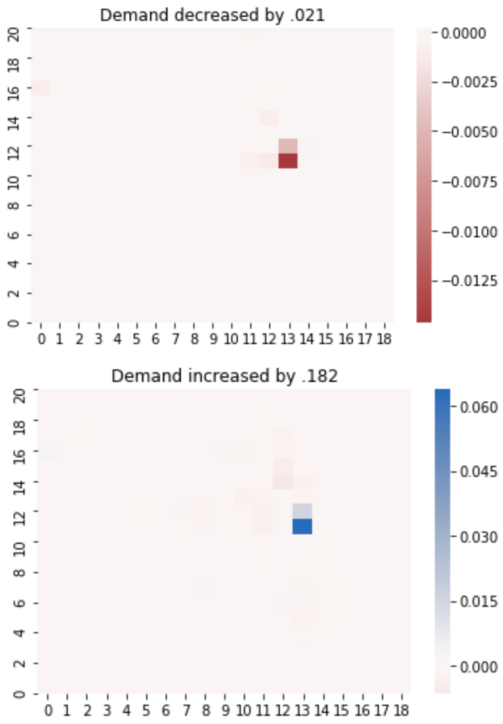

Figure 6. The change in predicted demand for each cell after subtracting 0.021 from (top) or adding 0.182 to (bottom) the scaled demand for the highest demand cell (13,11) in each of the three input time steps (scaled demand in each of these was roughly 0.085), while leaving demand in the other cells unchanged.

the expense of the poorer southern part. This suggests an interesting area for further exploration.

Lastly, using the simplest of the models, we demonstrated the potential to glean information about how demand in one part of the city affects demand elsewhere by systematically varying the demand levels in the highest demand cell to understand its impact on usage prediction in the rest of the city. We saw that suppressing demand in the key cell led to uniformly lower or unchanged demand predictions, while inflating it led to inflated usage predictions in the adjacent high-demand cell but suppressed demand predictions elsewhere, suggesting that the model views inflated demand downtown to be an indication that ridership that might have occurred elsewhere is downtown instead.

Areas for further work include investigation into the ways in which potential spatial bias in these models might be manifested and subsequently counteracted, how the addition of more feature data might improve performance, and the impacts of a more refined grid or more sophisticated normalization technique.

References

[1] DeepRain: ConvLSTM Network for Precipitation Prediction using Multichannel Radar Data.

[2] L. Anselin. Spatial Externalities, Spatial Multipliers, and Spatial Econometrics. *International Regional Science Review*, 26(2):153–166, Apr. 2003.

[3] S. Bai, J. Z. Kolter, and V. Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271 [cs]*, Apr. 2018. arXiv: 1803.01271.

[4] J. Bradbury, S. Merity, C. Xiong, and R. Socher. Quasi-Recurrent Neural Networks. *arXiv:1611.01576 [cs]*, Nov. 2016. arXiv: 1611.01576.

[5] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. Hasegawa-Johnson, and T. S. Huang. Dilated Recurrent Neural Networks. *arXiv:1710.02224 [cs]*, Nov. 2017. arXiv: 1710.02224.

[6] A. Ermagun and D. M. Levinson. Spatiotemporal Traffic Forecasting: Review and Proposed Directions. Working Paper, Aug. 2016.

[7] C. Heghedus, A. Chakravorty, and C. Rong. Neural Network Frameworks. Comparison on Public Transportation Prediction. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 842–849, May 2019. ISSN: null.

[8] Y. Kamarianakis and P. Prastacos. Spatial Time-Series Modeling: A review of the proposed methodologies. page 10.

[9] M. G. Karlaftis and E. I. Vlahogianni. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399, June 2011.

[10] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*, Feb. 2016.

[11] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, 17(4):818, Apr. 2017.

[12] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun. Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2278–2288, June 2019.

[13] 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 C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

[14] X. Song, H. Kanasugi, and R. Shibasaki. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level. In *IJCAI*, 2016.

[15] 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. page 8.

Appendices

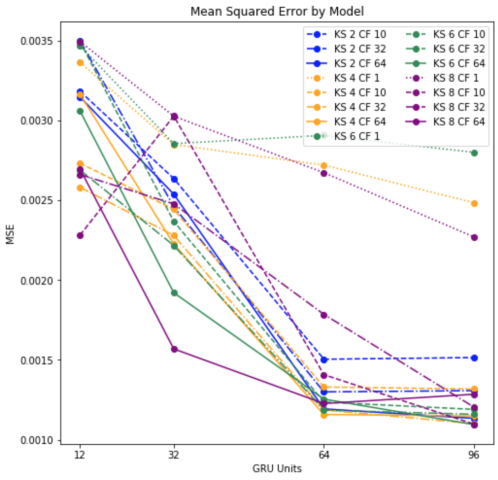

Figure 7. The Mean Squared Error of models with one convolution layer and one GRU layer. Each model is named according to the kernel size (KS) and number of filters in the convolution later (CF) and plotted with GRU units along the x-axis.

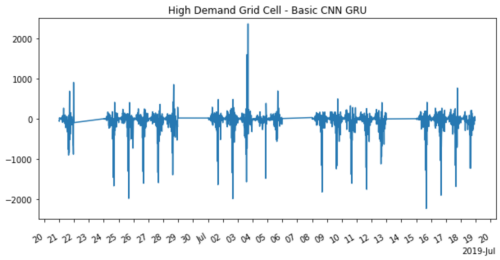

Figure 8. Residuals for the highest demand cell when the basic model is run on the test data.

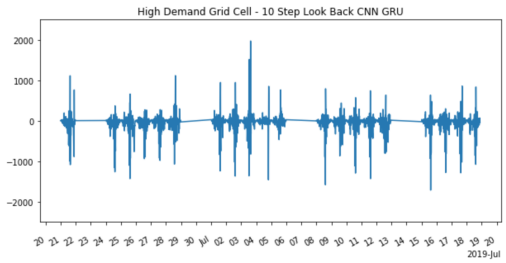

Figure 9. Residuals for the highest demand cell when the 10 step look back model is run on the test data.

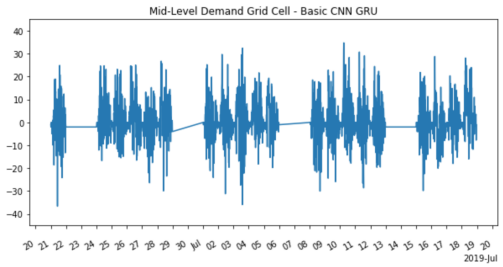

Figure 10. Residuals for a representative mid-level demand cell when the basic model is run on the test data.

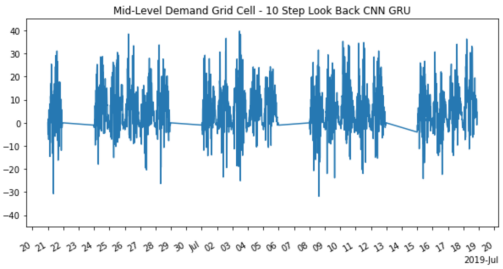

Figure 11. Residuals for a representative mid-level demand cell when the 10 step look back model is run on the test data.