Design and test a model which applies last inflows/outflows of a grid to predict the flows in next timeslot.

Group 3

JIANG Yuxin	16096336d
WANG Bokang	16097234d
YU Jing	16098537d
WANG Jiashuo	16096527d
CHENG Yiran	16098521d
HAO Shiqi	16098696d

Overview

Part 1: Problem Formulation

Part 2: Methodology

Part 3: Current Experiment Results

Part 1

Problem Formulation

1.1. Background and Question

Predicting the flows in a region can effectively ease traffic jams and greatly reduce traffic accidents, which is helpful for traffic management and public safety.

Design and test a model which applies last inflows/outflows of a grid to predict the flows in next timeslot. Using meteorological data may improve the prediction accuracy.

1.2. Notation Definition

- Region: (i, j)
  - a grid cell lies at the i-th row and j-th column in the map
  - 32*32 regions

- Inflow: $x_t^{in,i,j}$

  Outflow: $x_t^{out,i,j}$
  - inflow/outflow of the crowds at the time t for the grid (i, j)

1.3. Input

Raw Data:

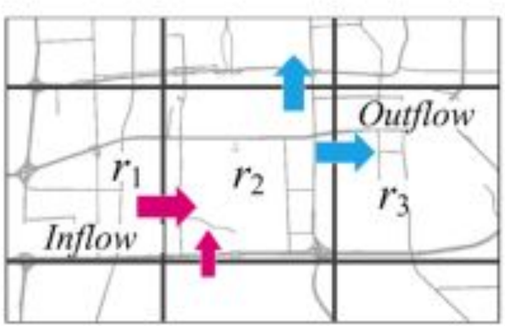
- Traffic
  - Date: 20151101--20160410 [date (7220,)]
  - Data: [data (7220, 2, 32, 32)]

- Weather
  - Wind speed/mph: 0--48.6
  - Temperature/◦C: −24.6--41.0
  - Weather conditions: 16 types

1.3. Input

Raw Data Features:

- Spatial dependencies
  - Nearby: The inflow of Region r2 is affected by outflows of nearby regions (e.g. r1). Likewise, the outflow of r2 would affect inflows of other regions (e.g. r3). The inflow of region r2 would affect its own outflow as well.



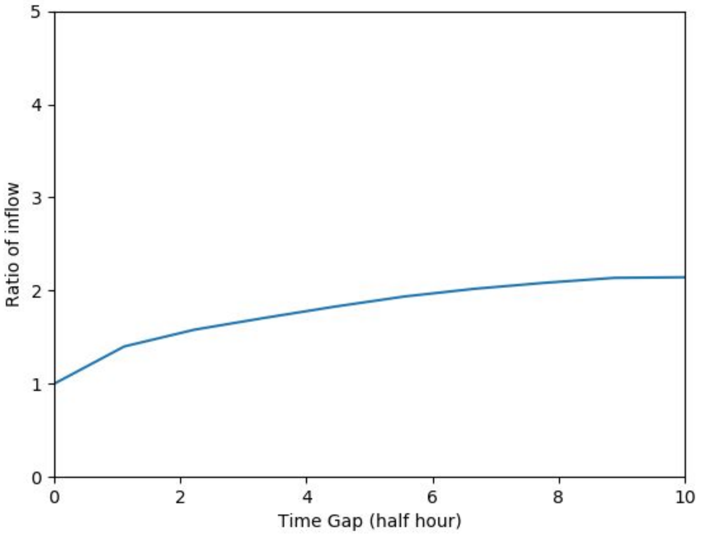
(a) Inflow and outflow

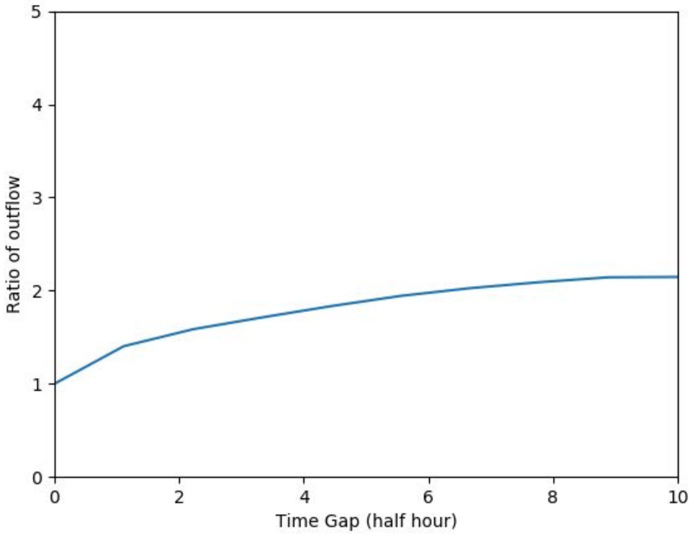
Ref: [Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., & Li, T. (2018). Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, *259*, 147-166.]

1.3. Input

Raw Data Features:

- Temporal dependencies
  - Closeness: The flow of crowds in a region is affected by recent time intervals.

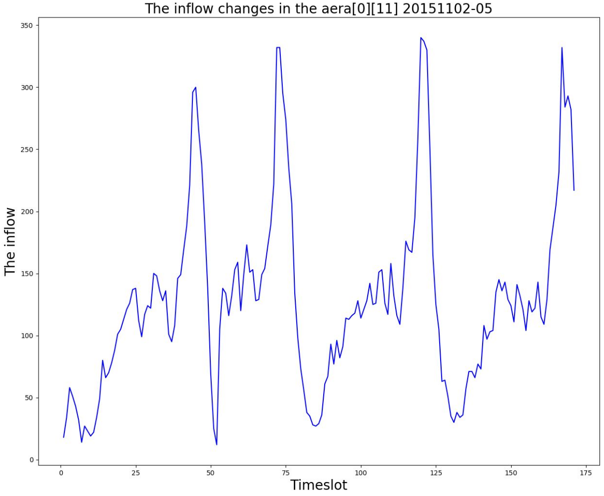


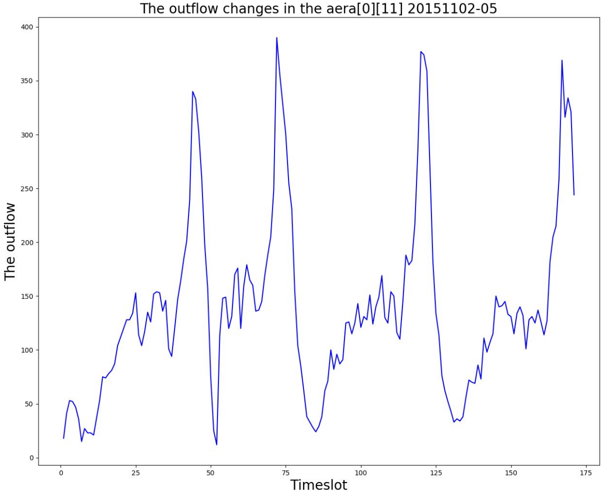


1.3. Input

Raw Data Features:

- Temporal dependencies
  - Period: Traffic conditions may be similar on consecutive weekdays (every 24 hours).





1.3. Input

Raw Data Features:

- External influence.
  Some external factors, such as weather conditions and events may change the flow of crowds tremendously in different regions of a city.

1.3. Input

Input Data:

Crowd inflows/outflows: the inflow and outflow of the crowds are the total number of taxis that arrive and leave this grid during the time slot.

Meteorological data: , temperature, wind speed of each day

1.3. Input

Dimensions:（3*2+1+1+1*2）*32*32 = 10*32*32

Meaning of each dimension：

- The inflow and outflow data of 32*32 area: three timeslot periods before the predicted timeslot……………………………………………………………………(3*2)*32*32
- Temperature (expand the scale to 32*32 by ourselves): predict timeslot……...1 *32*32
- Wind Speed of (expand the scale to 32*32 by ourselves): predict timeslot…...1 *32*32
- The inflow and outflow data of 32*32 area ：The same timeslot of the day before the predict day ...…………………………………………………………….(1*2)*32*32

1.4. Output

Predict: inflow and outflow of crowds of all area in next timeslot

MSE: $$\frac{\sum_{i=1}^{n}(\hat{y}_t - y_t)^2}{n}$$

RMSE: $$\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$

L1loss: $$\frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Part 2

Methodology

2.1. Chosen Method / Model

Convolutional Neural Network (CNN)

- with convolutional layer and activating layer (ReLU)
- has a powerful ability to hierarchically capture the spatial structural information
- good to deal with inputs with a high dimensionality

Why choose CNN?

- the crowd flows are spatial dependencies and temporal dependencies
    - CNN can can capture spatial structures

2.2. Train the Model

Data Used:

- Inflows and outflows of each cell
- Meteorology data including temperature and wind speed (weather)

Model:

- A CNN with 2 convolutional layers 2 activing layers

Training:

- Data will be divided in goups, each of which has a size of batch_size
- For every epoch, MSE is used to do gradient descent
- Improve the model according to the result

2.3. Loss Function

Mean Square Error (MSE)

- the loss function in our model
- the sum of squared distances between our predicted value and the real value
- sensitive to error bigger than 1
- constrained function’s constringency

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2.$$

2.4. Train/Test Data and Result Format

Train / Test Data

- Train 80%, Test 20%

Output

- a 2*32*32 matrix containing inflows and outflows in all locations in the certain time slot

2.5. Model Evaluation

Time Complexity: The real running time on the machine (seconds).

Accuracy of prediction:

- L1 Loss
- Mean Squared Error (MSE)

2.6. Optimization (minimize the loss)

We test the model with different parameters including:

- input data
  - Whether to use the weather information
  - How many previous time slots to be used
  - How many previous days will be referred
- number of filters
- number of hidden layers
- learning rate
- dropout rate

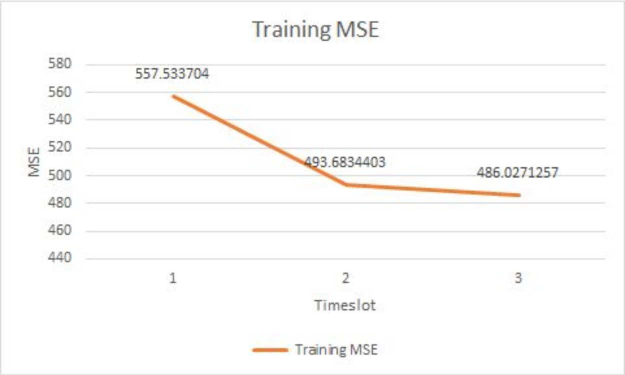
Will be introduced later ...

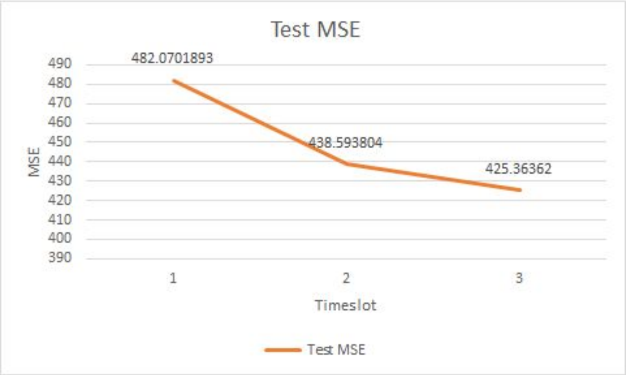
Part 3

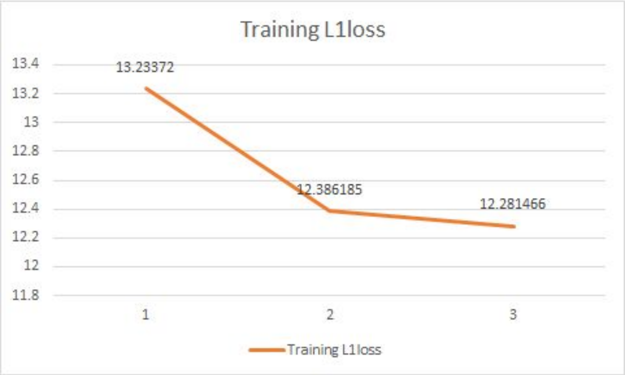
Current Experiment Results

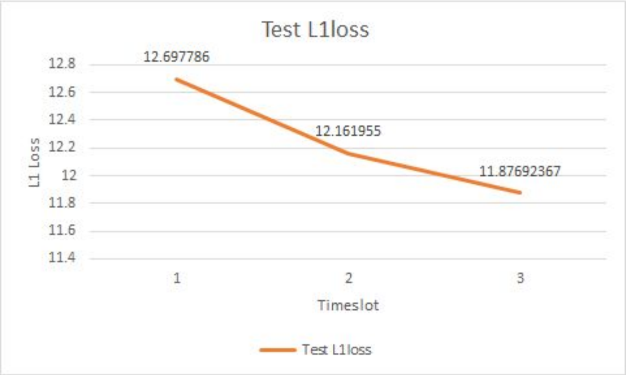
(All the data below are the average value after running the model 5 times)

3.1. Closeness - How many previous time slots to be used

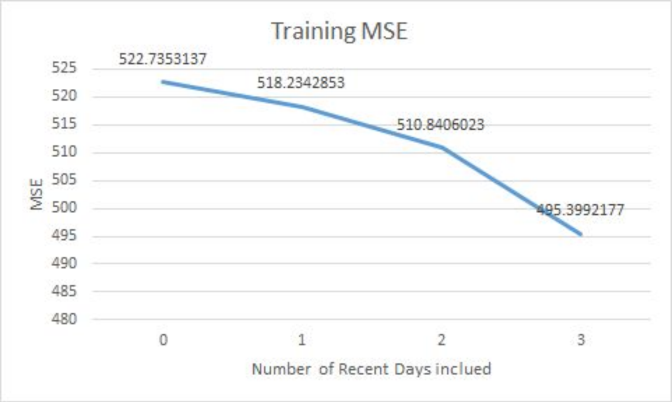


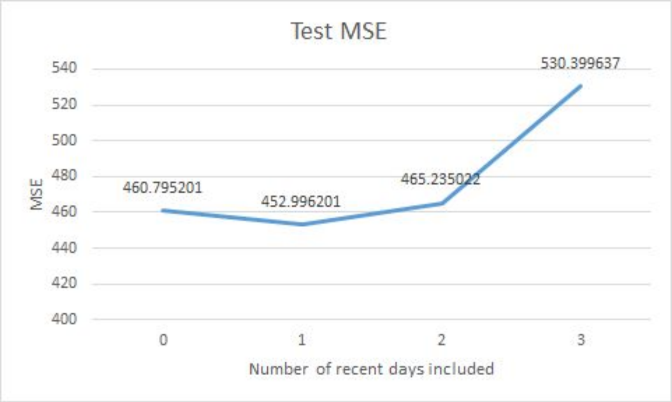


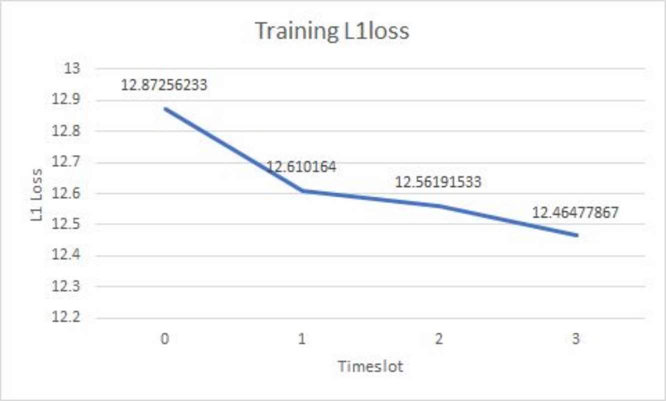


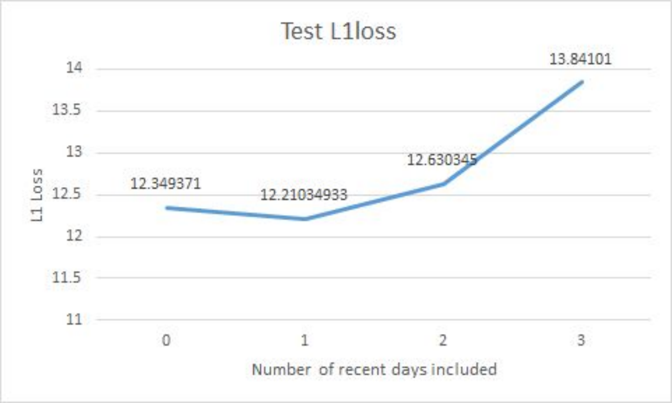


3.2. Period - How many previous days will be referred

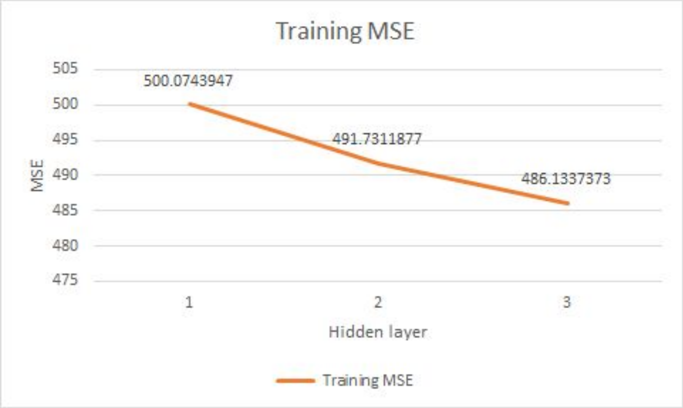


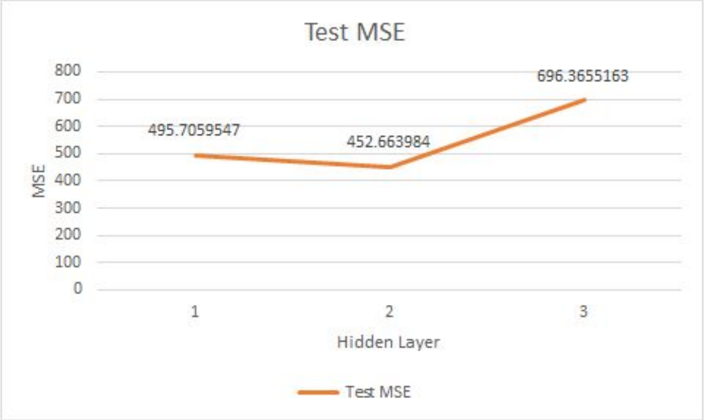


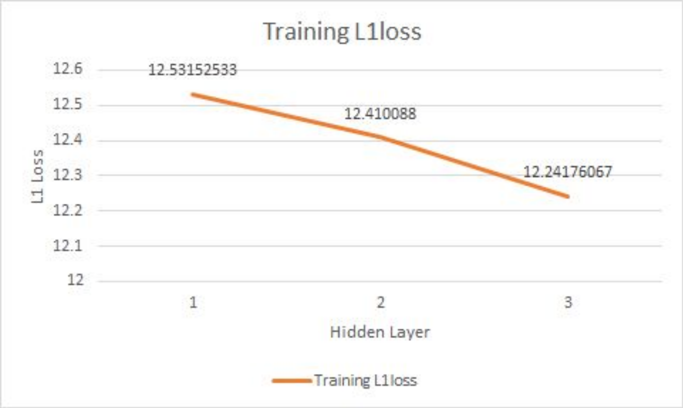


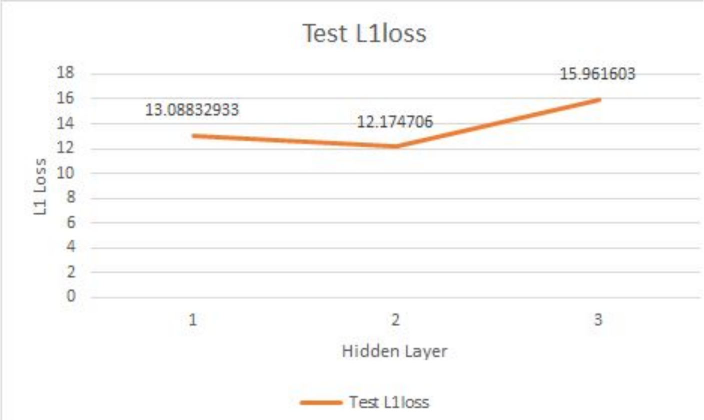


3.3. Number of Hidden Layers

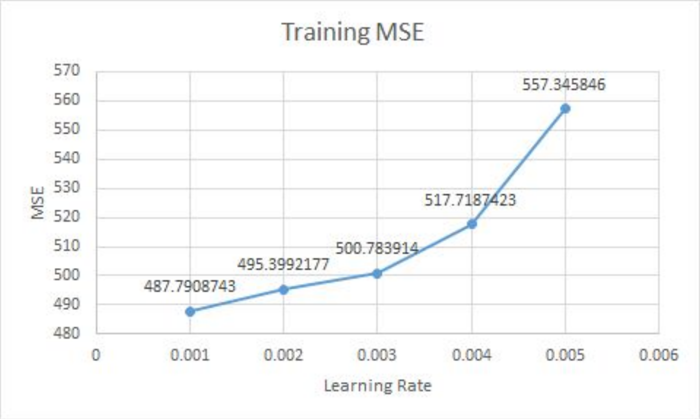


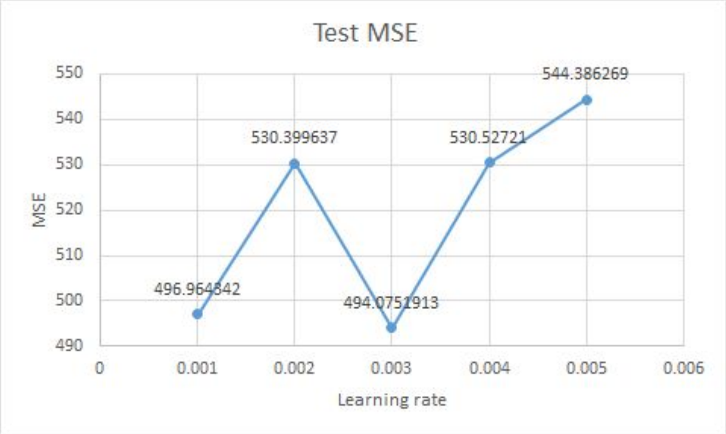


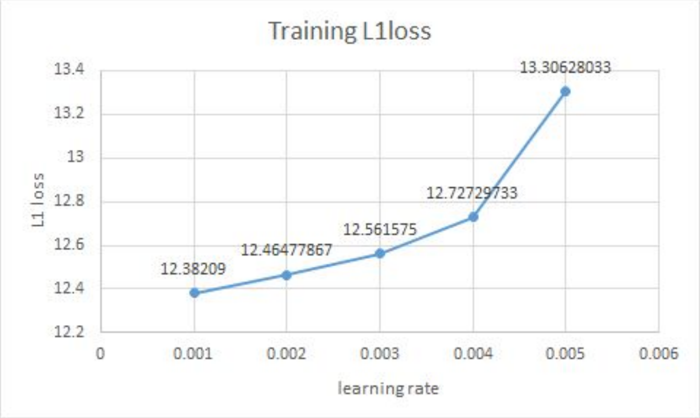


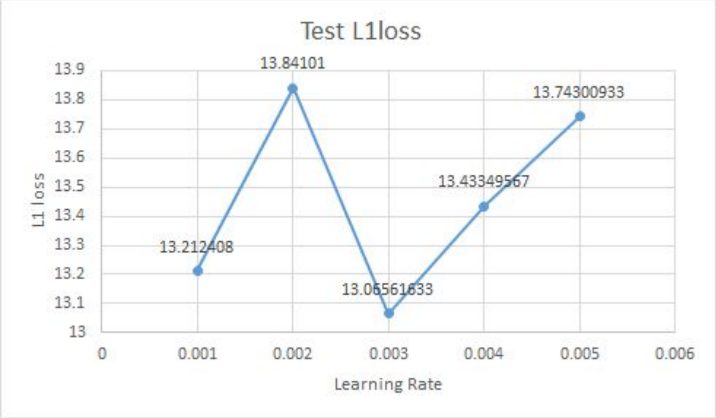


3.4. Learning Rate

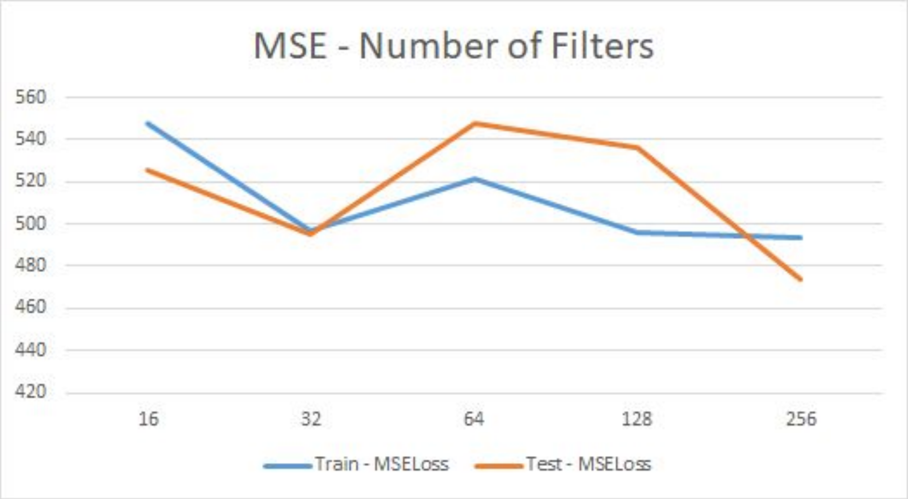


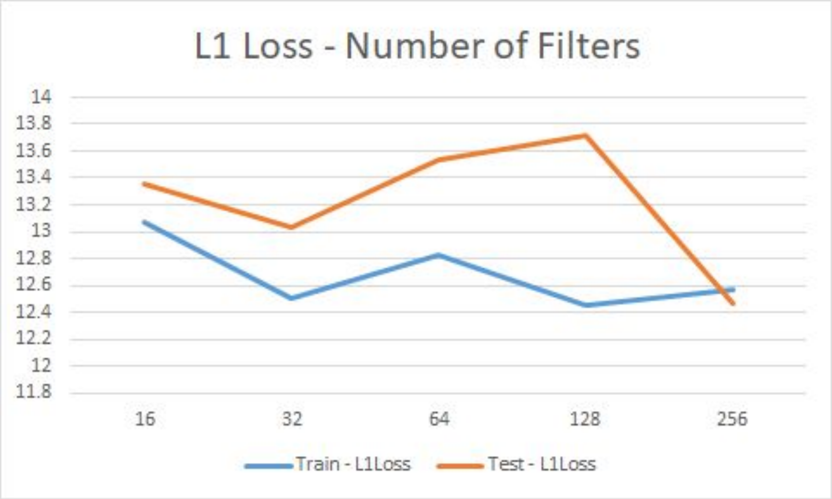




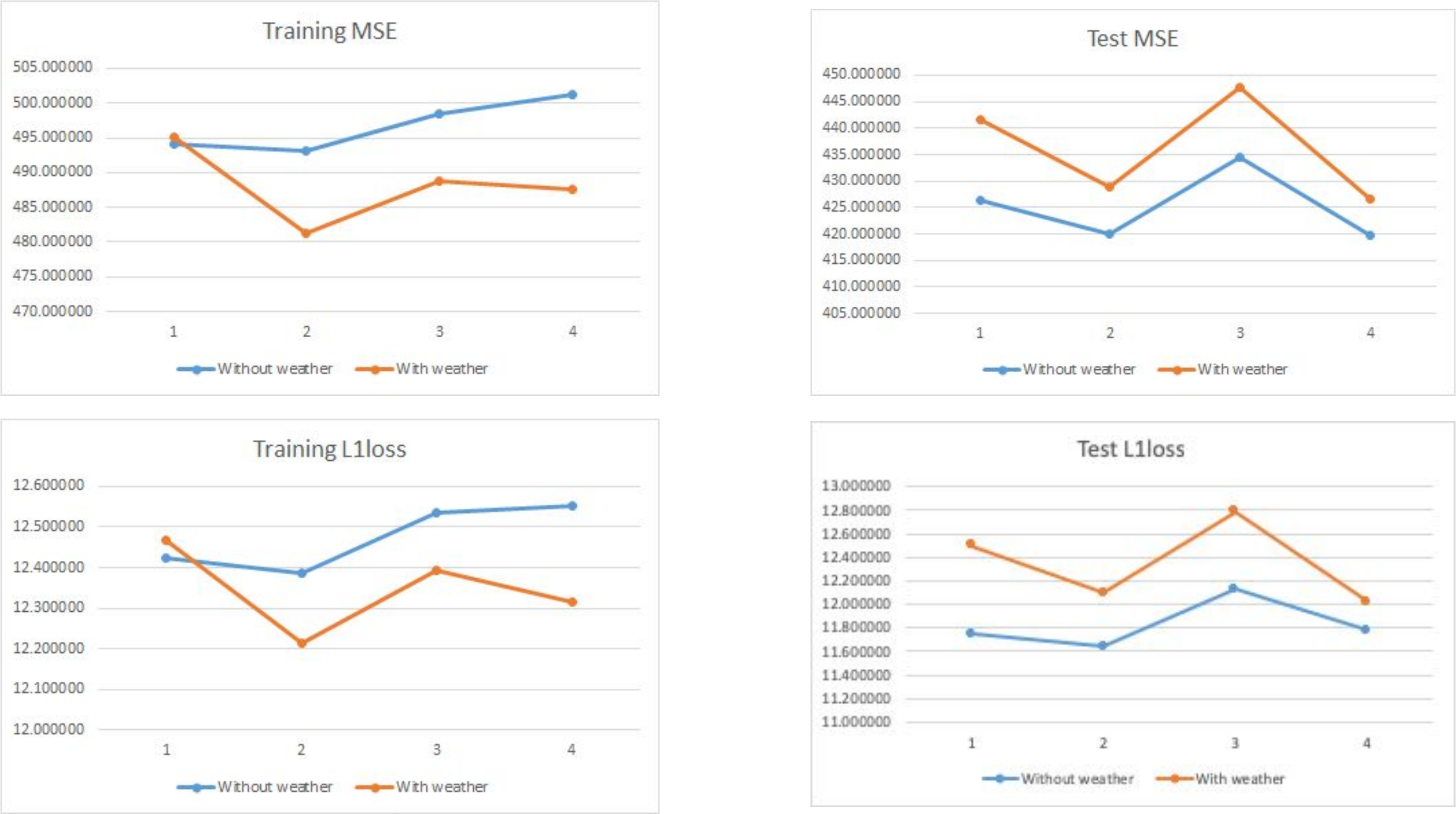


3.5. Number of Filters

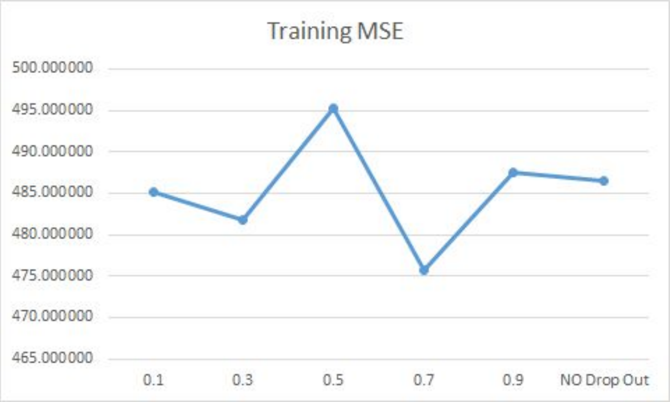




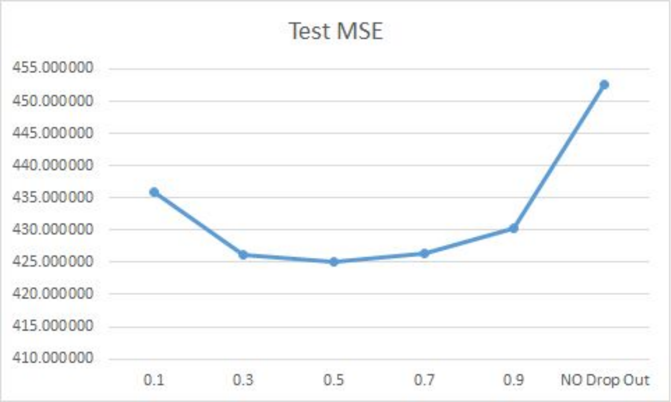
3.6. External Influence (Weather)



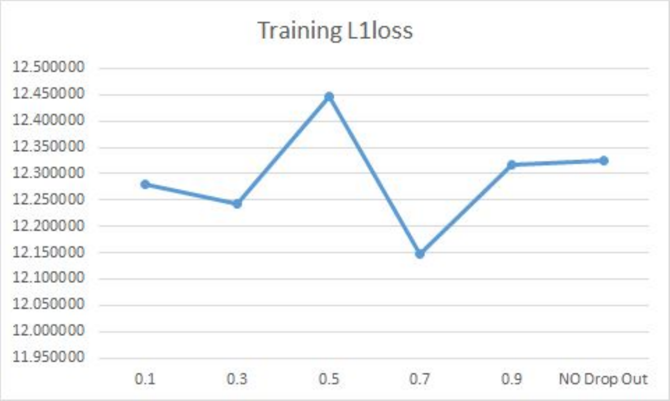
3.7. Dropout



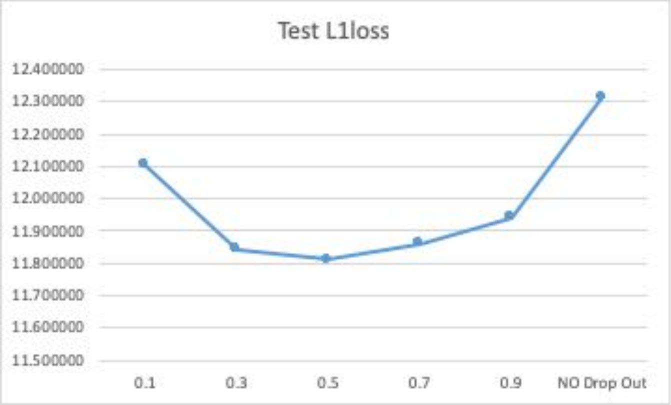
Training MSE
500.000000
495.000000
490.000000
485.000000
480.000000
475.000000
470.000000
465.000000
0.1
0.3
0.5
0.7
0.9
NO Drop Out



Test MSE
455.000000
450.000000
445.000000
440.000000
435.000000
430.000000
425.000000
420.000000
415.000000
410.000000
0.1
0.3
0.5
0.7
0.9
NO Drop Out

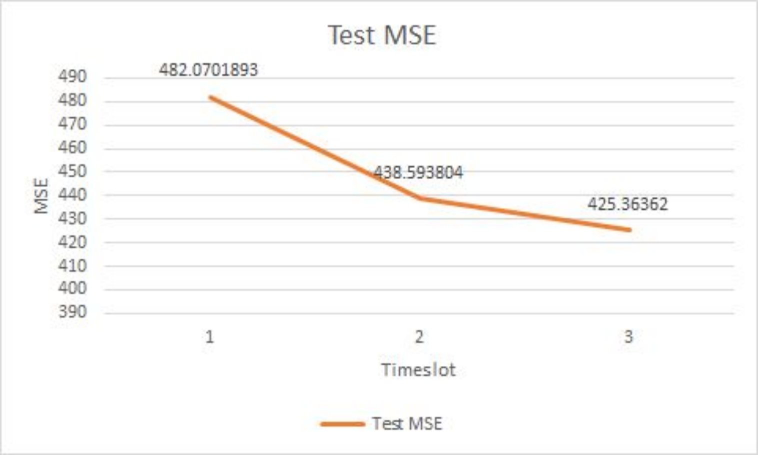


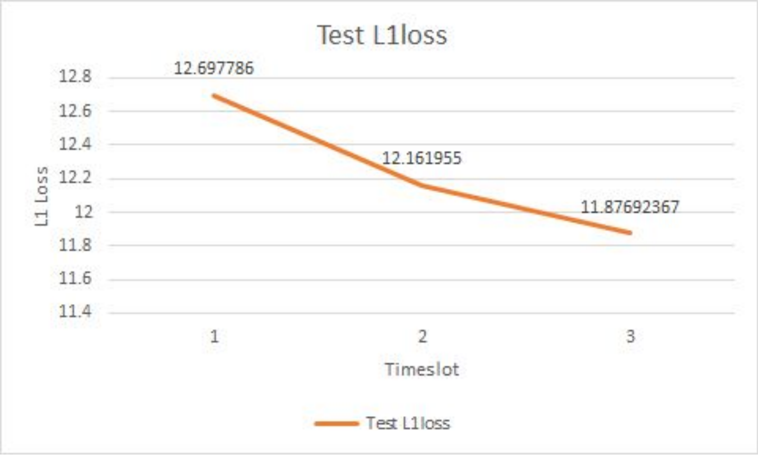
Training L1loss
12.500000
12.450000
12.400000
12.350000
12.300000
12.250000
12.200000
12.150000
12.100000
12.050000
12.000000
11.950000
0.1
0.3
0.5
0.7
0.9
NO Drop Out



Test L1loss
12.400000
12.300000
12.200000
12.100000
12.000000
11.900000
11.800000
11.700000
11.600000
11.500000
0.1
0.3
0.5
0.7
0.9
NO Drop Out

3.8. Which Input Data Contributes Most





Closeness contributes most
Possible Reason:Most of regions in the city have more or less closeness

3.9. Which Model Performs Best (Time Efficiency)

Running Parameters:

Kernel Size = 3

Epoches = 50

Batch Size = 64

Number of Filters = 256

Dropout = 0.5

Learning Rate=0.001

Hidden Layers = 1

Performance:

Train MSE: 456.259246

Train RMSE: 21.36022579468672

Train L1 Loss: 11.8522027

Test MSE: 407.9730357

Test RMSE: 20.19834239898578

Test L1 Loss: 11.4629663

Training Time: 11.4 s/epoch * 50 epoch = 570 seconds

Hardware Environment: GPU GTX 1080Ti, RAM 64G, CPU i9-7900X, Windows10

3.10. Which Model Performs Best (Better Performance)

Running Parameters:

Kernel Size = 3	Epoches = 50	Batch Size = 64
Number of Filters = 256	Dropout = 0.7	Learning Rate=0.001
Hidden Layers = 2		

Performance:

Train MSE:	428.087264	Train RMSE: 20.69027	Train L1 Loss: 11.517178
Test MSE:	372.346	Test RMSE: 19.296269	Test L1 Loss: 11.001296

Training Time: 98.3 s/epoch * 50 epoch = 4915 seconds

Hardware Environment: GPU GTX 1080Ti, RAM 64G, CPU i9-7900X, Windows10

3.11. Comparison with Different Models

Model	RMSE
	TaxiBY
HA	57.69
ARIMA	22.78
SARIMA	26.88
VAR	22.88
ST-ANN	19.57
DeepST	18.18
RNN-3	26.68 ± 3.41
RNN-6	30.03 ± 1.60
RNN-12	45.51 ± 2.01
RNN-24	51.12 ± 1.99
RNN-48	43.42 ± 1.20
RNN-336	39.61 ± 0.77
LSTM-3	26.81 ± 2.80
LSTM-6	26.07 ± 1.87
LSTM-12	27.59 ± 3.69
LSTM-24	25.69 ± 2.25
LSTM-48	27.80 ± 2.87
LSTM-336	40.68 ± 1.08
GRU-3	22.97 ± 1.11
GRU-6	23.64 ± 1.14
GRU-12	27.40 ± 3.72
GRU-24	27.01 ± 1.58
GRU-48	28.56 ± 3.71
GRU-336	40.27 ± 2.30

Our Test RMSE result:
Time efficiency model: 20.19834239898578
Better Performance model:19.296269

Ref: [Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., & Li, T. (2018). Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, *259*, 147-166.]

Q&A