Traffic Flow Prediction

Arizona State University

Team Members

Name	ASURITE ID	Role
Rath, Prabin	prath4	Code, Research, Documentation
Cao, Richard	rscao	Code, Research, Documentation
Gade, Shreyash	sgade13	Research, Documentation
Tokala, Jaswanth Reddy	jtokala	Research, Documentation
Vasireddy, Vikas	vvasire2	Research, Documentation

Table of contents

Introduction

- In this project we present a deep regression model that can be used for predicting the traffic flow in urban regions and cities.
- Spatiotemporal data analysis is an emerging research area for the analysis of large vehicle trajectory databases.
- Trajectory data are collected from numerous sources such as GPS and infracture cameras.
- **Traffic Flow** is the number of vehicles passing a reference point per unit of time, vehicles per hour.

Problem Description

- Forecasting the traffic flows is important as this can help in traffic management and crowd control.
- There have been various incidents in the past which had major impact on the lives of people due to poor traffic management.
- For example, last month (Oct 31, 2022), at least 151 people were killed in a Halloween crowd stampede in South Korea.
- In order to reduce such incidents it is important to predict potential traffic congestions ahead of time.
- Apps such as Google/Apple Maps use traffic flow predictions for path planning and suggest alternate detours avoiding high traffic regions.

Literature Review

What are the challenges in Flow prediction?

- **Spatial dependencies** (How the flow at one region affects flow at other regions?)
- **Temporal dependencies** (Periodic properties of flow data such as 9am and 5pm traffic in office regions)
- **External influence** (Factors influencing psychology of the crowd such as festivals, and events)

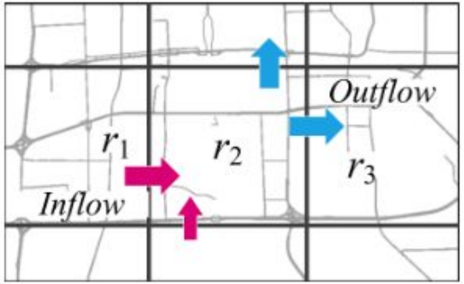
Literature Review

- One of the early methods used in the traffic flow prediction is Autoregressive Integrated Moving Average (ARIMA) technique for short horizon flow prediction.
- Statistical and Non-parametric methods such as Kalman Filters, K-NN, Fuzzy Inference, and ANN have also been previously employed for prediction.
- Shallow networks and other statistical or non-parametric methods are limited in terms of their ability to identify long temporal relationships within the flow data.
- Temporal learning models such as RNNs fail to capture spatial dependencies effectively.
- To overcome this limitation deep learning approaches can be used as they can capture long-term temporal and spatial relationships in flow data.

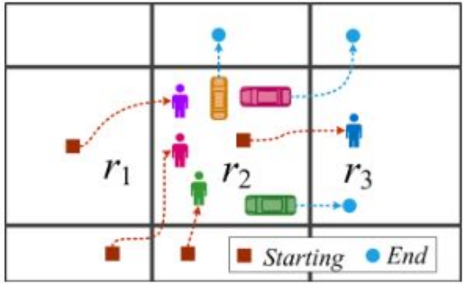
Literature Review

How to model spatial dependencies?

1. **Inflow** is the net flow entering a given region from other regions at a particular time.
2. **Outflow** is the net flow leaving a given region to other regions at a particular time.



(a) Inflow and outflow



(b) Measurement of flows

Methodology

How to model temporal dependencies?

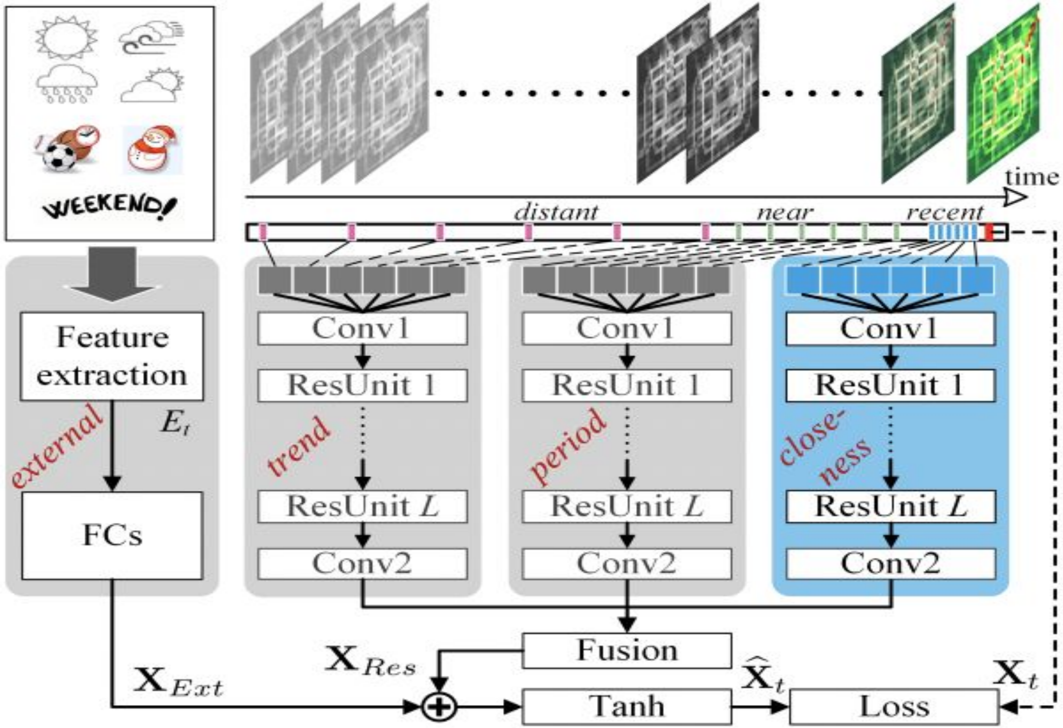
- **Closeness**: The recent traffic flow within the last few hours have significant influence on the current traffic flow at a region.
- **Period**: On consecutive weekdays at the same hour, the traffic flow might be comparable; for instance, the 9 a.m. traffic on Tuesday might be periodic with the 9 a.m. traffic on Wednesday, the next day.
- **Trend**: We need to monitor patterns in traffic flow over longer time periods. For instance, when winter approaches, traffic congestion in the morning hours may gradually move later since people tend to wake up later in the morning.

Methodology

- Our methodology divides the city into grids based on longitude and latitude.
- Each grid represents a region within the city where flow data is recorded.
- Our prediction model is based on the paper Deep Spatio Temporal Residual Network (**ST-ResNet**) that collectively predicts the inflow and outflow of crowds in every region of the city.
- Flow is characterised with four components: **closeness**, **trend**, **period** and **external influence**.
- A convolution-based residual network has been used to model nearby and distant spatial dependencies.

Model Architecture (High Level)

Our model takes flow grid for the four components and external influence as input.

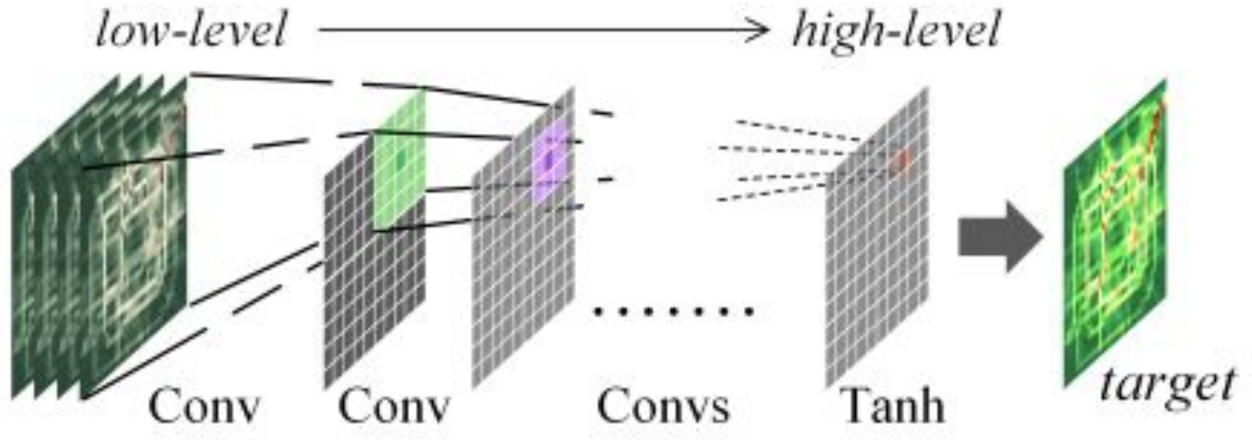


Model Architecture (Why CNN?)

- Flow of various regions are affected by neighbouring regions as well as by distant regions.
- Such data can be effectively handled using the convolutional neural networks, as it has shown ability to hierarchically capture the spatial structure information.
- CNN captures the flow dynamics that is essential for predicting the flow ahead of time.
- Different kernel sizes represent the neighbourhood radius that will be considered for learning the flow dynamics.

Model Architecture (Why CNN?)

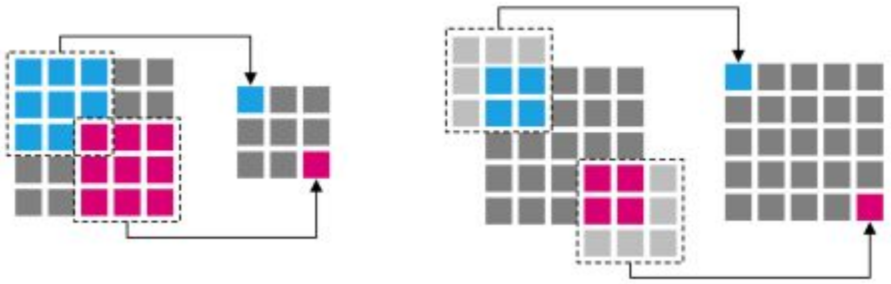
It can be observed that each convolution captures the nearby spatial dependencies and the stacked layers of convolutions capture the distant spatial dependencies.



(a) Convolutions

Model Architecture (Why CNN?)

Our model employs **same convolution**, i.e it does not use any pooling layers for feature extraction. So, the input and output dimensions of the gridmaps remain unchanged throughout the model pipeline. Narrow convolution is useful for classification and representation learning but traffic flow prediction is a regression task. Same convolution **avoids data loss** which is essential for minimizing the MSE loss during training.



(a) *narrow* convolution (b) *same* convolution

Model Architecture (Residual Learning)

- In order to model the entire citywide dependencies, upto 15 layers deep convolutional units are required. Generally such deep neural networks tend to have the problem of **vanishing gradients**.
- This can be tackled using **residual learning**, which has demonstrated to be very effective for training super deep neural networks.
- In our model, we have **stacked residual units**. A unit is made up of two convolutional layers, each preceded by a ReLU activation.
- Within each residual unit, **batch normalization** is applied on the data before feeding it to the convolution layer. BN has shown significant improvements in terms of training time reduction.

Model Architecture (External Influence)

- External features are encoded as on-hot feature vectors.
- In our implementation, we have considered weather, and metadata like day of the week, weekday/weekend.
- We stack 2 fully-connected layers each followed by ReLU activation
  - 1st layer is an embedding layer for each sub feature.
  - 2nd layer is for upscaling to high dimension, in order to have the same dimensions as the original gridmap.

Model Architecture (Fusion)

- In this stage, we fuse the outputs from first 3 components (CNN+Residual Units), using a **parametric matrix fusion** approach.

$$\mathbf{X}_{Res} = \mathbf{W}_c \circ \mathbf{X}_c^{(L+2)} + \mathbf{W}_p \circ \mathbf{X}_p^{(L+2)} + \mathbf{W}_q \circ \mathbf{X}_q^{(L+2)}$$

- As shown in the above equation, we have taken sum of the **Hadamard product** between the weights (Wc,Wp,Wq) and component outputs (Xc,Xp,Xq).
- The fused result is directly added with the reshaped external component output.
- Finally, the merged output is activated with a **tanh** function to generate the flow.

$$\widehat{\mathbf{X}}_t = \tanh(\mathbf{X}_{Res} + \mathbf{X}_{Ext})$$

Dataset Description

Dataset	**TaxiBJ**	**BikeNYC**
DataType	Trajectory data from taxicab GPS data and meteorological data in Beijing, China.	Trajectory data from the NYC bike rental system.
Location	Beijing	New York
Time periods	4 time periods ● 7/1/2013-10/30/2013 ● 3/1/2013-6/30/2014 ● 3/1/2015-6/30/2015 ● 11/1/2015-4/1/2016	1 time period ● 4/1/2014-9/30/2014
Time Interval	30 minutes	1hr
Grid Map Size	32x32	21x12

BikeNYC

- This dataset consists of inflow and outflow of the bike journeys from January 2015 to June 2017, sampled at 1hr.
- The **Fig1** shows the flow of traffic in different days of the week during normal days and **Fig2** shows the traffic in different days of the week during a holiday week (Sept 1 Labor Day).
- From the **Fig1**, we can observe that weekdays generally have higher flow than weekends.
- From the **Fig2**, we can observe that flow is higher during the weekends compared to the weekdays.

Fig1

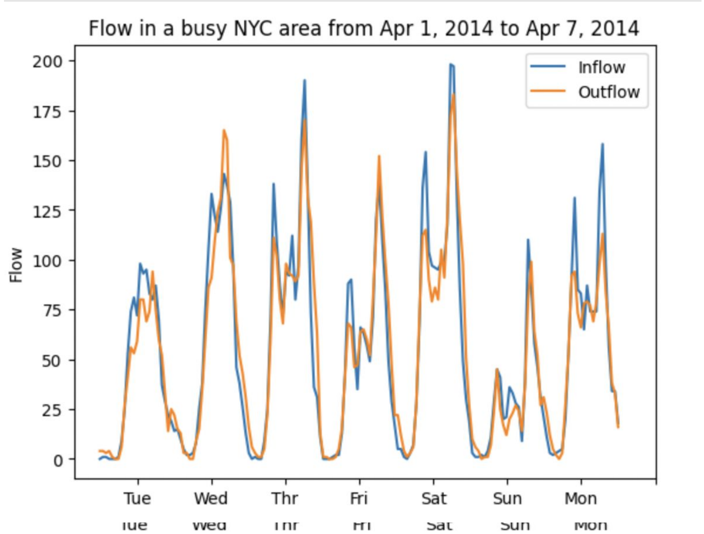
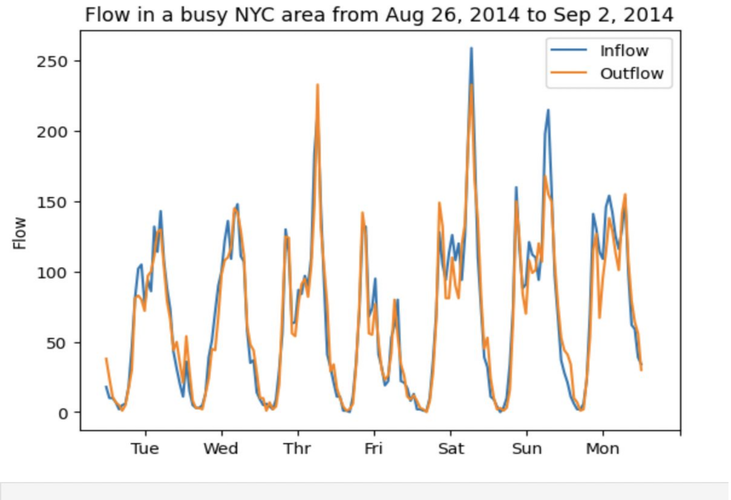


Fig2



TaxiBJ

- The data is made up of trajectory data from taxicab GPS data and meteorological data in Beijing.
- Data was collected in four different time periods and they are sampled at 30 mins.
- The **Fig1** depicts the flow of traffic during the entire week.
- The **Fig2** shows the change in temperature with time in Beijing.
- The **Fig3** shows the change in wind speed with time in Beijing.

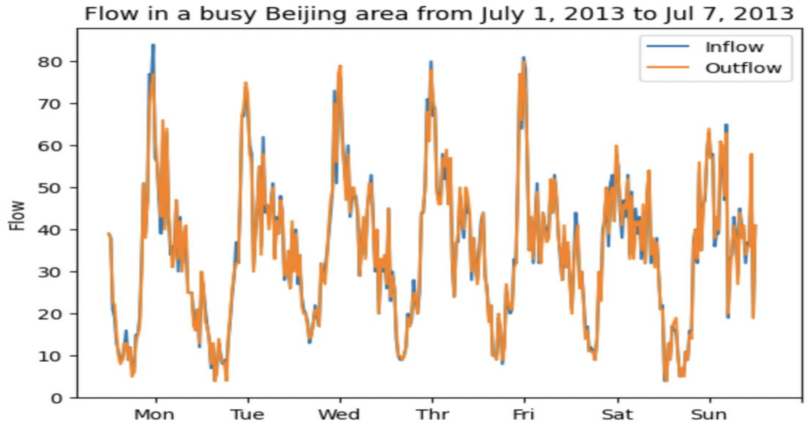


Fig1

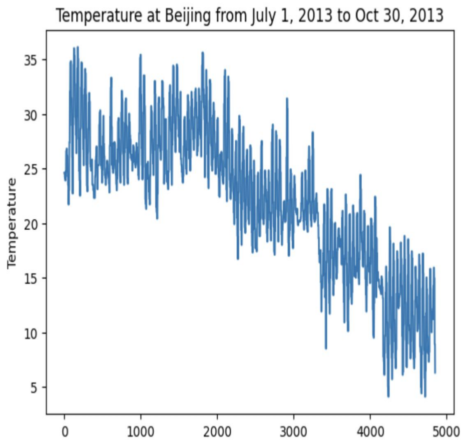


Fig2

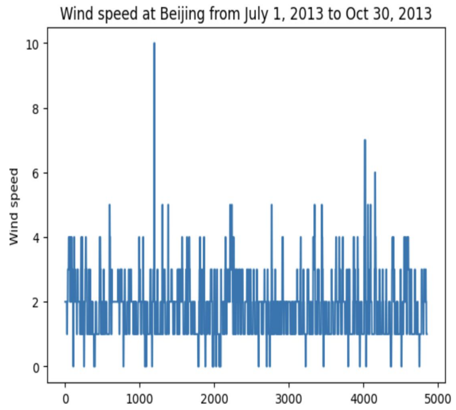


Fig2

Model Changes

Original Model **Implemented Model**

This inclusion of the ReLU layer has improved the training time of our model.

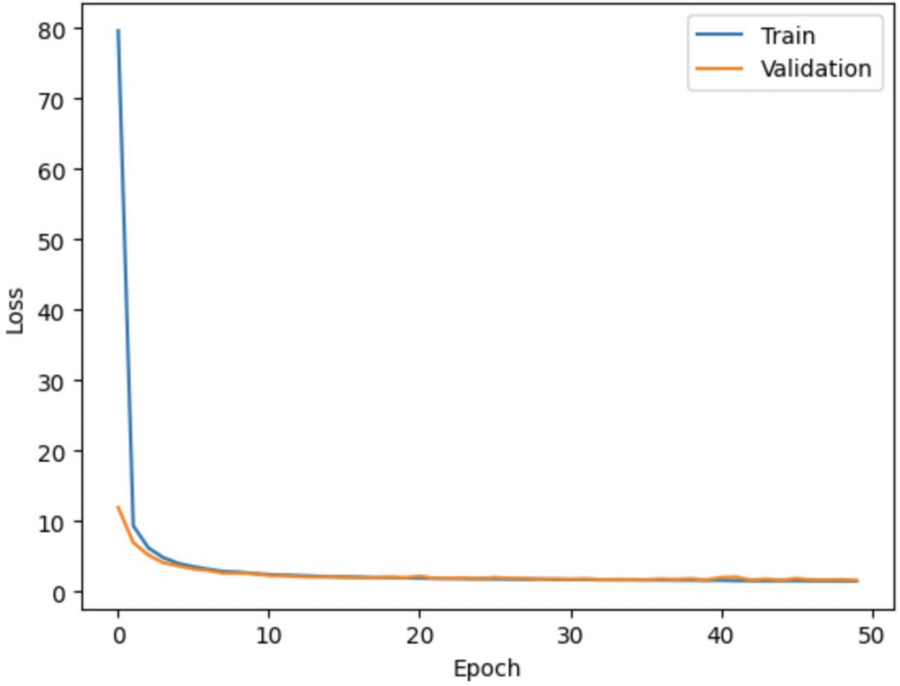
Training Parameters

- N_closeness = 4
- N_period = 1
- N_trend = 1
- Batch_size = 32
- Num_of_epochs = 50
- Test_train_split = 9:1
- Adam_initial_learning_rate = 0.0001

Flow data, Temperature and Wind speed are normalized in the range of (0 to 1) during the training process.

Training Metrics

- The figure here shows the graph between loss and epoch during our model training.
- As the number of epochs increase there is a decrease in the loss for both training data and validation data.
- For every epoch, the training loss and the test loss are very close and therefore, there is no overfitting or underfitting.
- During training we observed that batch normalization helps in attaining quick convergence.
- Note that the loss presented here is before denormalization.



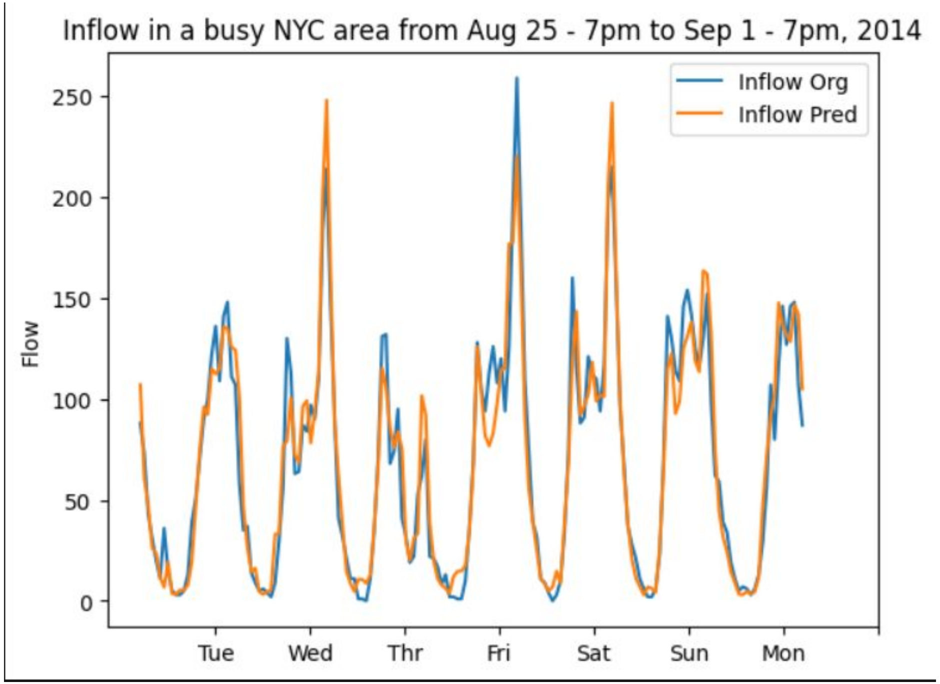
Training Metrics - Grid Root Mean Square Loss

- In case of BikeNYC dataset, the difference can be attributed to the fact that our implementation does not include an external unit.
- In case of Taxi BJ dataset, the rmse loss is almost similar, the slight difference is because in our implementation we have not taken holidays into consideration.
- The loss presented here are after denormalization.

	Paper Implementation (RMSE)	**Our Implementation (RMSE)**
BikeNYC Dataset	6.33	9.517
Taxi BJ Dataset	16.89	17.904

Results

This figure compares our model's predictions (orange) for a normal week at an office area in NYC with the real inflow.

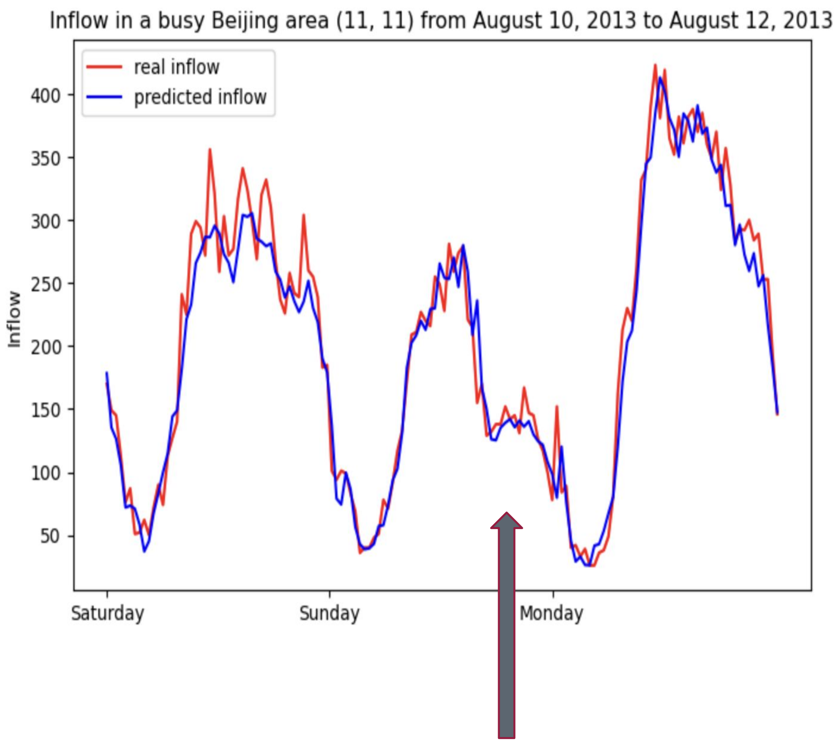


Results

The figure shows the comparison between the real inflow and the predicted inflow in a busy region in Beijing area.

Moreover, the implemented model was able to predict the flow during a thunderstorm, evidenced by the steep drop in the flow during that time.

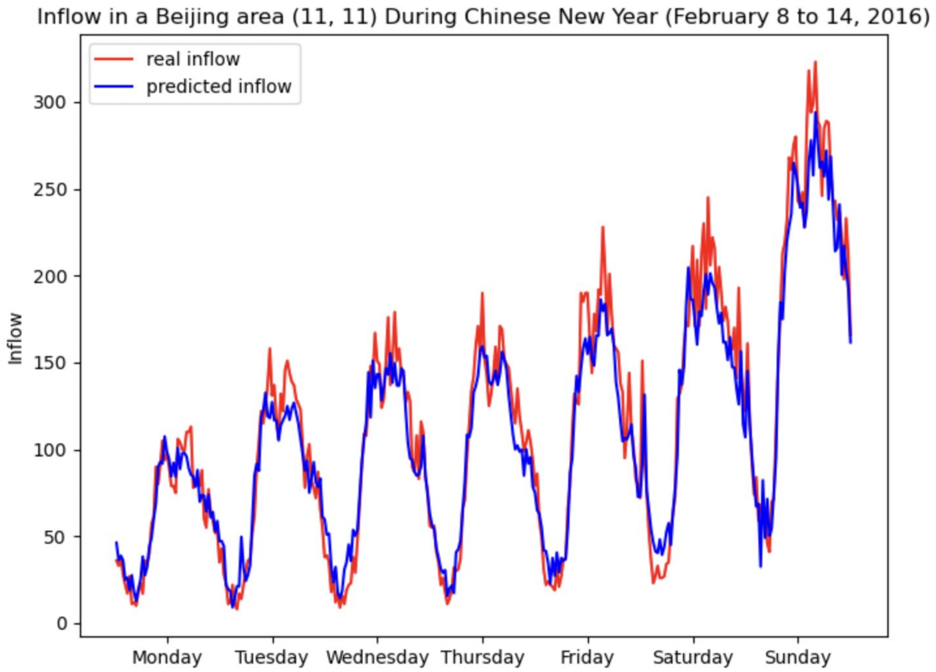
The prediction aligns very well with the real inflow which illustrates the model's predictive power.



Thunderstorm

Results

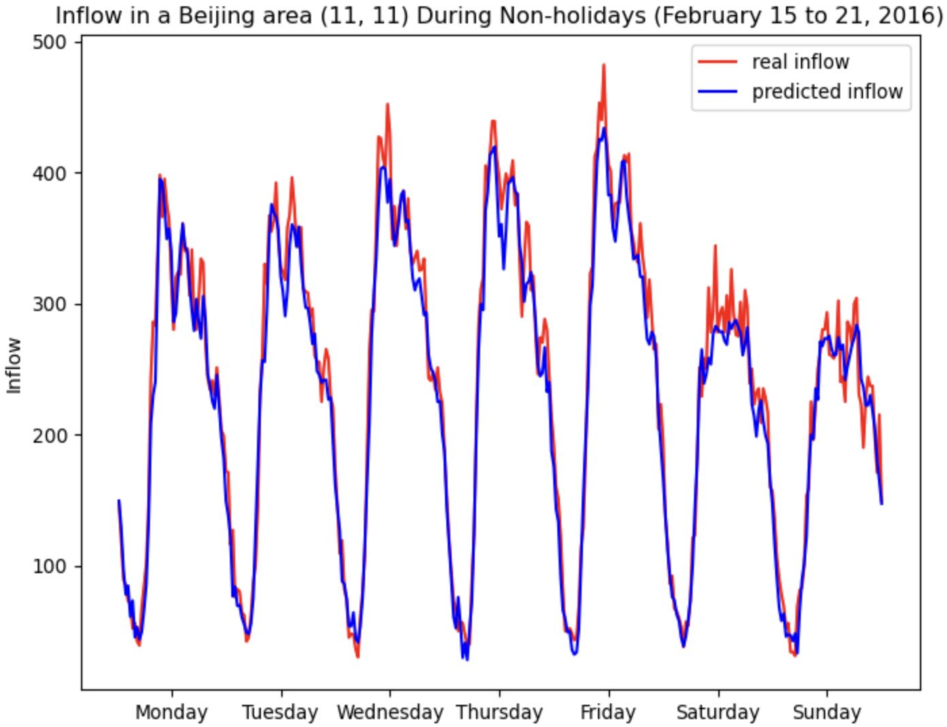
Our model predictions (blue) for a week during Chinese New Year. Predictions are accurate with respect to the real inflow despite the stark difference in the flow pattern compared to normal days.



Results

For the week after Chinese New Year (non-holidays/normal days), our model's predictions are also accurate with respect to the real inflow.

In the previous graph, the scale of the inflow during Chinese New Year ranged from 0 to 300, whereas during normal days, the inflow scales from 0 to 500, indicating that more people stay indoors during Chinese New Year.



Results

T-step Look Ahead Predictions

- Implemented for or relevant for the closeness component
- The previous T inflow predictions are utilized to predict the next inflow value
- A smaller value of T has better prediction power
- A larger value of T is more prone to erroneous predictions

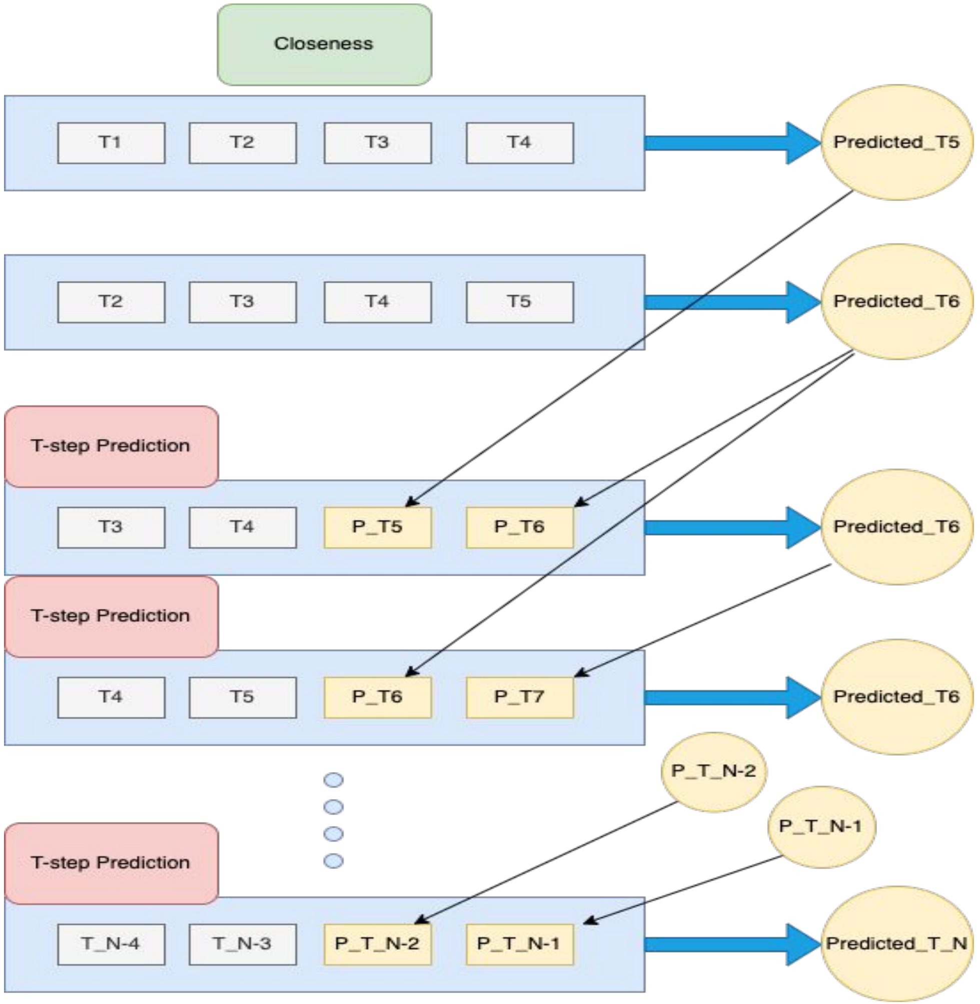
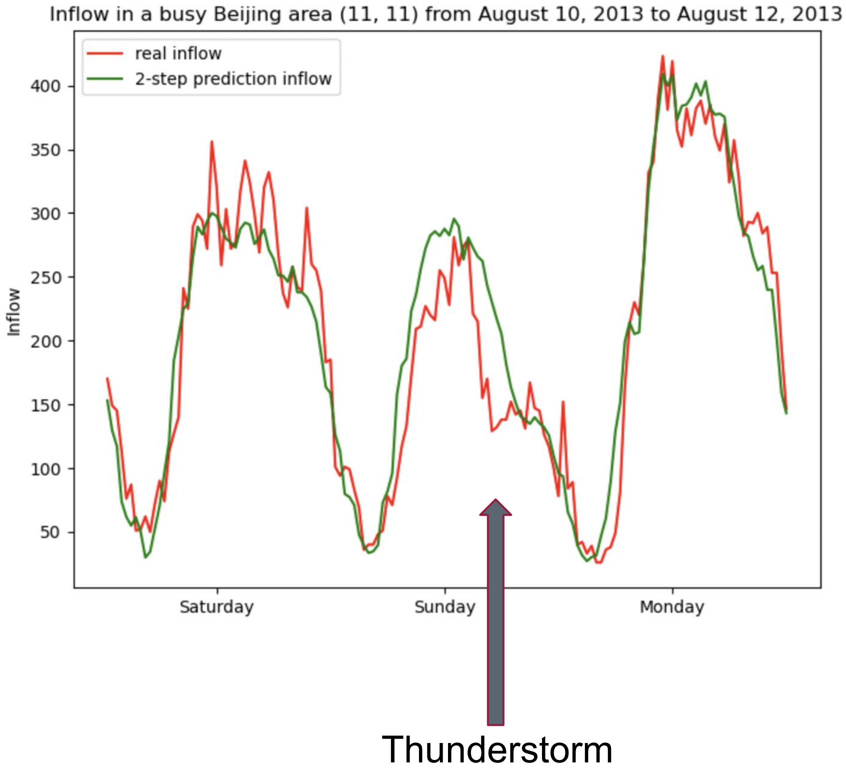


Figure: Scheme for the 2-step look ahead prediction

Results

This figure compares the real inflow with *the 2-step look ahead* predicted inflow.

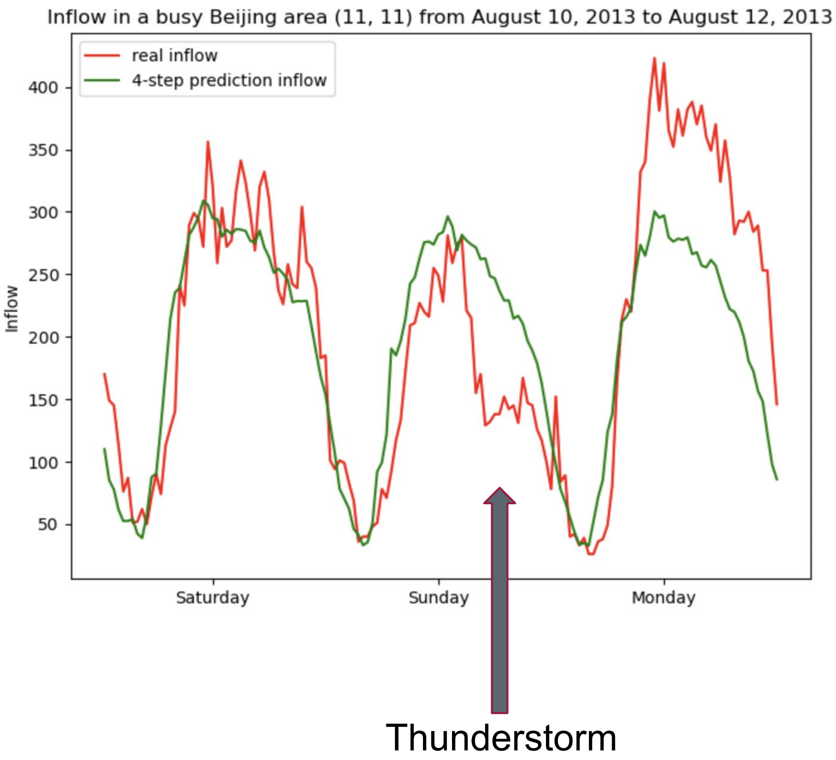
It is evident that the predicted inflow is not captured as well as in the previous case (T=1). However, this prediction is still quite meaningful because it captures the general trend of the traffic flow.



Results

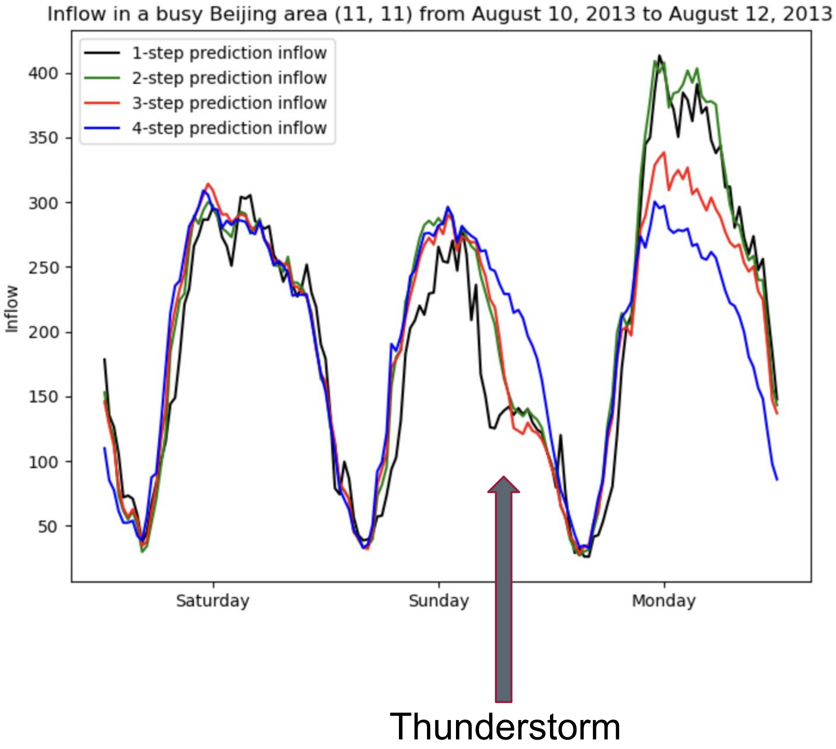
This figure compares the real inflow with *the 4-step look ahead* predicted inflow.

Increasing the T-step look ahead value will increase the degree of error or introduce more loss of data. This is expected because as T increases, each prediction increasingly relies on more of the previous predictions.



Results

This figure superimposes T = 1 to T = 4 step look ahead predictions. It can be observed that the fit worsens from 1 to 4. Our current efforts are aligned on a model to improve these lookahead prediction accuracy.



Conclusions and Future Work

- We implemented the original ST-ResNet model with few customizations and achieved very accurate results
- Both the datasets were analyzed comprehensively and results of analysis were presented
- Flaws with the look ahead predictions were highlighted and efforts were made to understand the ways to improve the results
- We are planning to improve T-Step look ahead using RNNs on top of ST-ResNet
- Since the model is an ensemble neural network, its intricate architecture can always be modified to improve the results

Thank You