# Traffic Flow Prediction using Deep Spatio-Temporal Residual Networks


Arpith Reddy Singareddy
1217133827
asingar1@asu.edu

Sharad Saxena
1216924566
ssaxen18@asu.edu

Manish Aakaram
1217852896
maakaram@asu.edu

Nishant Washisth
1217130460
nwashist@asu.edu

Abhay Shrinivas Saraswathula
1217205626
asarasw2@asu.edu

CSE 575: Statistical Machine Learning
Computing, Informatics, and Decision Systems Engineering
Arizona State University

Under the guidance of
Prof. Yingzhen Yang



## Abstract

In contemporary times, major metropolitan cities all around the world from New York to London, from Abu Dhabi to Beijing, all experience a significant amount of traffic on their roads. Considering the economic contribution and impact of these hubs, it becomes crucial to understand and manage the traffic flow of these cities. We are living in the era of Big Data and the Internet of Things, where sensors are recording nearly all aspects of our lives, and all activities leave a digital footprint. It is our goal to leverage this data in conjunction with state of the art machine learning techniques to develop a system that is capable of predicting the flow of traffic, and modelling the potential impact of unforeseen events such as storms, power outages, and pandemics.

## 1. Introduction

Accurate and timely traffic flow information is currently strongly needed for individual travellers, and government agencies [1]. It is important for managing the crowds during special events and incidents. Sudden massive crowds can lead to many issues such as stampedes. Examples of such incidents occur everywhere around the world. Stampedes are more evident in populous areas and are obvious during events such as Kumbh Mela in India, or Football Derbies or the Occurrence of augmented rare Pokémon when playing the game of 'Pokémon Go'. If such events involving high amounts of traffic flow can be predicted, one can make necessary adjustments to prevent dangerous situations from happening by issuing warnings, deploying more traffic control, or evacuating people in advance. Traffic flow prediction has also been a prominent player in the development of Intelligent Transportation Systems (ITSs). There have been attempts to predict Traffic flow in the past but were not much successful due to the use of hand-designed features and lack of robust prediction models. More success has been found in recent years due to the advent of Deep Learning and Big Data.

## 2. Problem Statement

The goal of the project can be defined as follows: Given historical observations of traffic flow at a given region for a set of sequential time intervals, the traffic flow for the next time interval must be predicted. External factors such as weather changes, special events can also be provided as parameters to improve the accuracy of prediction of Traffic flow. The generated model should be able to capture the different types of dependencies such as Spatial, Temporal, and effects of any external factors that determine the Traffic flow. The Traffic flow of a region can be considered to consist of Inflow and Outflow (Zhang et al. 2016). Inflow can be defined as the amount of traffic entering a region and Outflow as the amount of traffic leaving a region.

In this project, we implement an approach using Deep Spatio-Temporal Residual Network [2] to solve the problem of Traffic flow prediction. Deep Spatio-Temporal Residual Network or ST-ResNet is a combination of a series of Convolutional Layers and Residual Units [3]. The ST-ResNet primarily attempts to capture the Spatial and Temporal closeness, period, and trend properties along with the effects of external influence.

## 3. Preliminary Approach

This section discusses the preliminary approach for solving the problem of Traffic flow prediction briefly. As mentioned previously ST-ResNet is being used to address our problem. As per the problem statement, the prediction model must be able to capture Spatial as well as temporal dependencies of the traffic. Expanding on this, spatial dependency can be explained with a simple example of vehicles travelling from one region to another. There is a dependency between the two spatial regions where the vehicle travels from and the region where the vehicle travels to.

Regarding Temporal dependencies, one can say from general knowledge that the traffic on the streets during weekdays is higher during the morning and evening periods when people travel to work and back to home compared to other times. There is also a spatial aspect to this, a region which might contain a workplace or office will experience higher traffic during weekdays whereas a residential area during a similar time experiences minimum traffic. We continue to discuss the preliminary implementation details of the solution.

### 3.1 Data and pre-processing

For our implementation, we need data that describes the inflow and outflow of a particular region along with the time period. As a part of the solution, the entire data for a given region is divided into a grid-like structure and assigned to the grid cells. Each cell in the grid has details regarding the inflow and outflow of traffic. Therefore, the entire data for a given region or city can be represented as a 2-channel image with height and width the same as the grid. Hence, Convolutional Neural Networks can be used to capture different relations between the data.

In this project, we operate with the NYC taxicab data provided by NYC Taxi and Limousine Corporation (TLC) [4]. The data consists of different trips taken in the city of New York over a period of time. Each row describes a trip and it contains details of the trip such as the Pickup coordinates and Drop-off coordinates, start time, duration of the trip which are relevant to our problem. These details must be pre-processed to calculate the inflow and outflow of each cell on the grid based on the pickup and drop-off points. We can consider the time period to be around 30 or 60 minutes.

### 3.2 Model Architecture

To capture all the temporal dependencies of closeness, trend, and period, ST-ResNet is comprised of 3 similar components of a series of Convolutional layers and Residual units. The convolutional layers are very well known for handling the spatial dependencies and aspects of the image data. We make sure that the convolutional layers are deep enough to capture the relation of the entire grid data over the image to the output of a particular cell. Vanishing gradients is a problem that arises when the number of layers increases in the neural network. To address this, we can use the residual units which again consist of Convolutional layers paired with ReLU [5] activations.

The outputs from the convolutional layers of all the 3 components are then merged using parametric-matrix-based fusion which again contains learnable parameters. The inputs for all the 3 components comprise of a sequence of images over time as mentioned previously. Before passing the input, all the sequences of images will be merged to form a single image with 2*sequence length channel image. The output generated will be a 2-channel image that depicts the inflow and outflow of the cells in the grid for the future time period.

To incorporate external parameters, we extract weather data for the required time period and this data is passed to fully connected layers to reduce the dimensions. The output is merged with the image produced from the previous step and then passed to a Tanh activation function which generates output for each cell in the range of [-1, 1]. Adam optimizer [6] is used to calculate the loss which is used to adjust the weights of the convolutional layer filters using backpropagation for a certain number of epochs.

### 3.3 Training and Testing

We use the NYC taxi trip data [4] as mentioned in the previous section to perform training of the model and testing it. We use 80-20 train-test split mechanism to create the training data and testing data. Since there are many training examples in the data, the loss calculation can be very slow leading to slow learning. To mitigate this, we are going to use Stochastic Gradient descent [7] where the entire train data will be split into batches randomly and during the training, backpropagation is done for each batch without waiting for the entire training data to be iterated.

### 3.4 Metrics

When the testing is done, we compare the results of our predicted model with that of the ground truth. Various metrics such as RMS error [9], Mean Absolute percentage error (MAPE) [8] can be calculated to get a better understanding of the performance of the trained model.

## 4. Milestones

Below is a rough schedule of various tasks and is open to modification

- 09/15/2020 – Group Formation
- 09/30/2020 – Project Proposal Submission
- 10/15/2020 – Preparation of the Dataset
- 10/20/2020 – Developing the model
- 10/25/2020 – Training the model
- 11/1/2020 – Ablation studies and fine tuning the model
- 11/12/2020 – Project presentation
- 12/10/2020 – Project Final report submission

## 5. Roles of Team Members

Below is rough breakdown of tasks of each team member. However, all the team members are expected to contribute when necessary and read up on topics that benefit the project.

- Nishant: Data pre-processing
- Abhay: Feature Engineering
- Arpith: Model selection
- Manish: Model deployment
- Sharad: Ablation studies, fine tuning of model

## References


[1] N. Zhang, F.-Y. Wang, F. Zhu, D. Zhao, and S. Tang, “DynaCAS: Computational experiments and decision support for ITS,” IEEE Intell. Syst., vol. 23, no. 6, pp. 19–23, Nov./Dec. 2008.

[2] Junbo Zhang, Yu Zheng, Dekang Qi, “Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction” Jan 2017 https://arxiv.org/pdf/1610.00081.pdf

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[4] TLC Trip Record Data URL: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[5] Abien Fred Agarap, “Deep Learning using Rectified Linear Units (ReLU)” 2019 https://arxiv.org/abs/1803.08375

[6] Kingma, Diederik P. and Ba, Jimmy Adam: A Method for Stochastic Optimization. (2014). , cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 .

[7] Ruder, S. (2016). An overview of gradient descent optimization algorithms. ArXiv Preprint ArXiv:1609.04747.

[8] de Myttenaere, B Golden, B Le Grand, F Rossi (2015). "Mean absolute percentage error for regression models", Neurocomputing 2016 arXiv:1605.02541

[9] Wikipedia contributors, "Root-mean-square deviation," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=941256353 (accessed September 30, 2020)