Spatio-Temporal-History Model: A Deep learning model to predict traffic speed using public transport data

Student Name: Nikhil Gola

IIIT-D-MTech-CS-GEN-MT18129
JUNE, 2020

Indraprastha Institute of Information Technology
New Delhi

<u>Thesis Committee</u>
Dr. Vikram Goyal (Advisor)
Prof S.K. Gupta
Dr. Mukesh Mohania

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
in General Category

Keywords: bus transit system, GPS, Congestion, Spatio-temporal, Deep learning, CNN, LSTM, Traffic speed prediction.

Certificate

This is to certify that the thesis titled **"Spatio-Temporal-History Model: A Deep learning model to predict traffic speed using public transport data"** submitted by **Nikhil Gola** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by him under my guidance and supervision at Indraprastha Institute of Information Technology, Delhi. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Vikram Goyal

Department of Computer Science

Indraprastha Institute of Information Technology, New Delhi (IIIT Delhi)

Abstract

Traffic speed prediction is one of the challenging task and has many applications. Existing solutions either use crowd-sourced data or sophisticated technologies to perform the task, and hence are costly and unreliable. In this thesis, we propose a machine learning technique that uses the public transport movement data to predict the speed/congestion on a given road segment. Specifically, we use DIMTS buses movement data that comes in the form of GPS trajectories. The technique, we call as STH-Model (Spatio-Temporal-Historical Model), is based on CNN and LSTM models and captures the local spatial dynamics and temporal speed trends for its prediction task. We demonstrate the efficacy of our approach on real-time DIMTS data.

Acknowledgments

Its my privilege to express my sincere gratitude to my advisor Dr. Vikram Goyal at IIIT-Delhi for providing me the opportunity to work on this thesis. I would also like to thank him for his valuable guidance, suggestions, continuous support and encouragement throughout the thesis. I faced the problems and he always guided me in the right direction. I would like to thank my parents for the constant support throughout my entire life. I also thank Yogesh Pandey for giving me a valuable feedback on my work. At last I thank all my friends and college-mates for their support and motivation throughout my thesis.

Author

Nikhil Gola

Contents

List of Figures

Chapter 1

Introduction and Motivation

Traffic prediction means to estimate the speed of the traffic on a road segment in a particular time interval based on historical observations on the road segment. It has been used as the baseline for many traffic-related applications such as congestion prediction [21], estimation of arrival time [6] etc. Tackling traffic-related problems has always been a challenging problem whether it is a congestion detection [16, 18] or a travel time prediction. In recent decades, the increase in population in urban cities increased the need for more accurate traffic management systems. GPS traces data is one of the most used datasets in the study of traffic dynamics. This data is so much informative that it can be used for the various traffic related problems and it can be used for predicting speed more accurately.

Today existing technologies depend largely on the crowd-sourced data which may not be reliable all the time. Government agencies need to pay a hefty amount to third parties to do the same task for them. And this leads to user privacy breach, as these enterprises use user's information like there location, interests, etc, this data is further used for there other recommendation systems and advertisements. So we need a technique that does not impact user's privacy breach instead it uses the government installed device's data to get the information. We use Delhi transport data which gives us the GPS information or we can say the live location of the bus. Delhi transport already operating two applications called DIMTS and poochho which reports the live location of the bus. These app limits their functionality to live location of the bus only. Private players like Google, Bing, Here maps do the congestion detection and travel time estimation but charge a hefty cost for this.

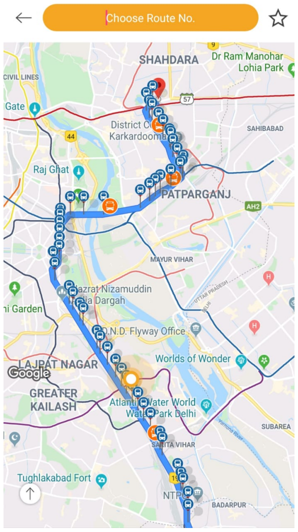

Figure 1.1: poochh-O app

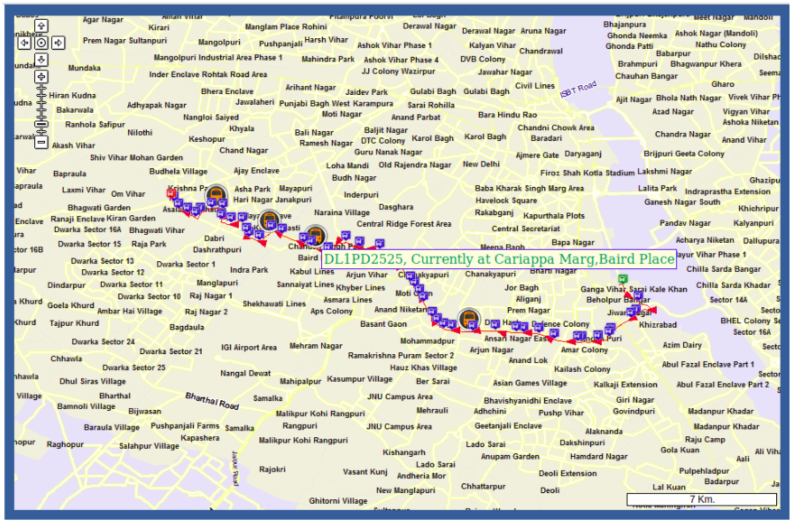

Figure 1.2: DIMTS webapp

Our technique helps the government agencies to upgrade their apps to do the same job at lower costs. In this regard, we propose a framework that includes data preparation and the machine learning model which is capable of predicting speed for future time-interval. For a road segment, the speed trends follow some temporal patterns like in normal time hours we can see the free-flowing speed in traffic and rush hour due to congestion the speed declines with time. And the speed trends also depends on the traffic flow from neighbouring roads, traffic signal etc. Our aim is to predict the traffic speed on road networks based on the spatial dynamics, temporal trends and past days/week speed trend. Hence we propose a Deep learning model STH-model(Spatio-Temporal-Historical model) that tries to solve above problems. We also compare our model with some pre-existing models through Mean Squared Error.

Chapter 2

Research Aim & Thesis outline

2.1 Research Aim

In the thesis, the aim is to address the following issues:

- Analysing the Spatio-temporal data for various feature extraction techniques like similarity basis, distance metrics, etc..
- Finding the best suitable technique for missing value problems.
- Getting the predicted speed value for the required time interval with the help of the machine learning model. We use the properties of CNN, LSTM, and ANN(Artificial Neural Nets) to prepare our Spatio-Temporal-Historical model.
- The model should learn the traffic pattern and its evolution during the time-intervals on the sparse dataset(Number of transmitting units is very low w.r.t number of roads).
- Comparing our model with some of the pre-existing models through Mean Squared Error.

2.2 Thesis Outline

The thesis is organized as follows first we discuss the related work in the field speed prediction by various Machine learning approaches, then the brief explanation of problem statement and continue with the methodology of the data preparation and model framework, At last, we conclude with the brief experimental methodology and the results.

Chapter 3

Related Work

The existing literature explains about the most used machine learning techniques to predict the speed values for the time-interval. The following content is divided based on their properties.

3.1 Linear Models

3.1.1 Support Vector Regression (SVR)

Support vector regression [5] [2] is the form of SVM(Support Vector Machine) and used for regression. The SVR uses the maximal marginal classifiers to predict the value and the loss function of SVR is called $\epsilon-$sensitive, which signifies the maximum distance between the actual value and predicted value should be ϵ else it adds penalty in the training loss function. The support vectors also depend on the ϵ, lower the value higher the support vectors. A kernel parameter γ which uses the kernel radial function [19],

$$K(x_i, x_j) = exp(-\gamma|x_i - x_j|^2)$$

here the γ is the inverse of the standard deviation. Gamma controls the radius of influenced support vectors. With the best parameters found in training, the model is used to predict the mean speed of the time interval. The SVR proposed in [4] does not consider the spatial dynamics of the road and predicts the traffic speed based on historical observations, they have done this on the small number of road segments. In our case, the data analysis shows we need a large number of roads to learn the variation in speed patterns. Training SVR on the large road network is a time-consuming process.

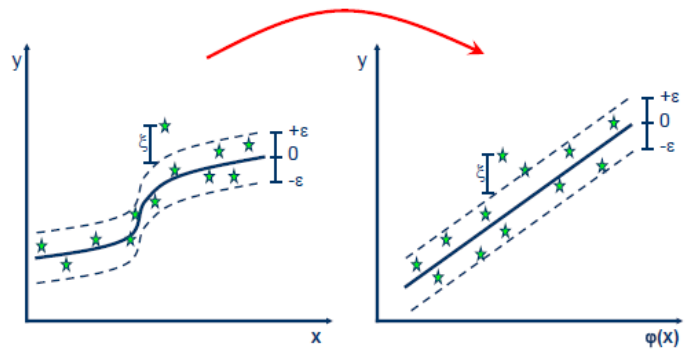

Figure 3.1: Support Vector Regression

3.1.2 Multiple Linear Regression

Linear Regression [15] is a parametric model of supervised learning class. Linear regression predicts the value of the various linear combination of input features. Linear Regression tries to fit the line which fits the linearly dependent and independent features.

$$y = W^T * X + b$$

here X is the features, W is the learnable weights and b is the error component. The Linear regression can be fit on two types of data set either on the historical speed trends or the current day trends. The linear regression model predicts the mean speed for the time slot based on the features provided [1]. These features are mean speed, the standard deviation of average speed, and many more, These are the statistical features to represent traffic status. But these statistical feature needs to be computed every time as the traffic dynamics is very noisy and it changes with the different scenarios(like rain, roadblock, etc) on the road. But our approach, on the other hand, consider the time series of the traffic so it tries to learn the traffic dynamics by Spatial and Temporal learning. Hence our approach can build higher-dimensional solution space to take account of solution space.

3.2 Non-Linear Model

3.2.1 Random Forrest

A Random Forrest [1] is a regression approach that predicts the value by growing different decision trees for different values. The different decision tree takes the various input features and predicts the speed values and then these predicted are averaged out gives the final mean speed. The Random forest depends on the decision making on based on the features hence it does not capture all the dynamics of the road same problem explained above.

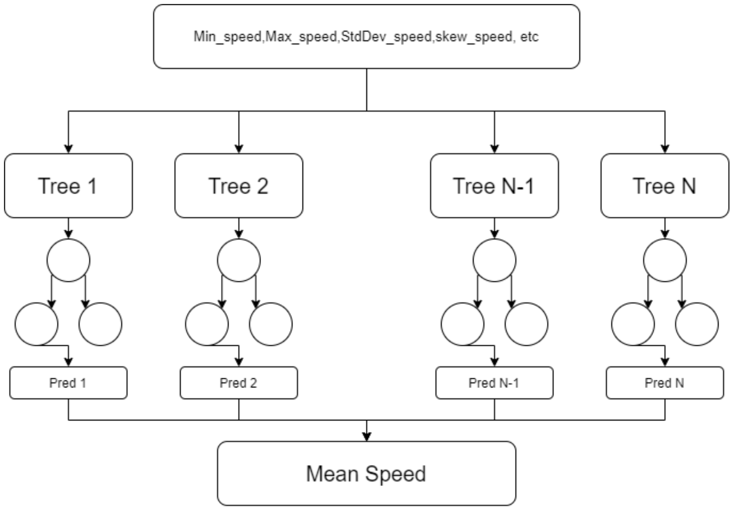

Figure 3.2: Random Forrest for mean speed prediction.

3.2.2 Multilayer Perceptron

Artificial Neural networks [13] is the most powerful tool of machine learning of a supervised class of learning it can deal with the non-linear boundaries. It consists of an input layer, a set of hidden layers, and output layers. Input features contain the various features of the road segment for the time-interval and predict the mean speed for the time slot. The features [1] used contain mean speed, std-dev in speed, skewness in speed, etc. These statistical features make learning to learn traffic speed statistical patterns. These statistical patterns need to be learned for every event that occurs in road traffic but sometimes some rare events can cause it to fail. Meanwhile, our approach is capable of adapting these events.

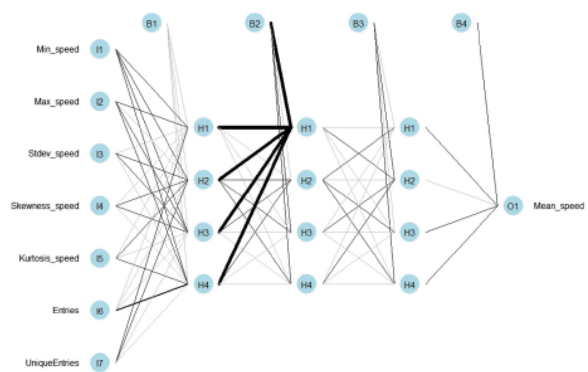

Figure 3.3: Artificial Neural network architecture explaining input features, hidden states and output. source: [1]

3.2.3 CNN

Convolution Neural Networks [7] is the most powerful tool used in capturing the spatial features among the pixels, it consist of several convolution layers, pooling layers and Fully connected layers. In recent times CNN proves to be most favourite tool among the researchers for various kind of image dataset. In traffic speed prediction [7] or congestion detection [8] [17] cnn also proves its capability. There are two different ways to do traffic speed prediction using CNN either providing the demographic pixel like image of traffic speed [7] or providing matrix of speed patterns of the speed which can be the current timestamp speed or a series of several previous time-intervals. CNNs are capable of learning patterns among neighbours, this property ,makes them predicting more accurately if neighbouring roads has some significance. CNN are capable of learning spatial features but fails to capture time series pattern, we used CNN and introduce some more learning tools like LSTM and Historical learning we make it more accurate.

3.2.4 RNN

Recurrent Neural Networks [12] is another powerful tool of supervised learning which performs exceptionally on time series data. Because of this nature, RNN is quite popular among researchers. LSTM and GRU overcome RNN's gradient vanishing problem for long series of data. Encoder-Decoder [9] model is quite popular when it comes to speed prediction. In the encoder-decoder model, the time series which contains the speed is encoded and decoded for the next time intervals. [11] creates a matrix of speed trends and for the speed trend vector they form it to a Sequence to Sequence problem, the main problem is that the sequence performs better in time series problems but when we have the impact of spatial dynamics then it results in some increase in error. Hence we used LSTMs to learn these time series but with the inclusion of learning of spatial trends with the help of CNN.

Chapter 4

Problem Statement

In this section we briefly formulate the problem of road based speed prediction. Here we define our road network and prediction problem.
We denote our Road network as a graph $G \in (V, E)$ where each $v \in V$ represents the intersection of road segments and $e \in E$ represents the road segments. Road network shown in figure 4.1.

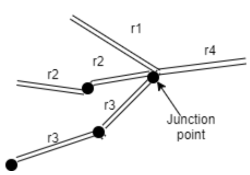

Figure 4.1: Road graph explained with edges represented as road ID.

t	t+1	t+2	t+3	t+4	t+5
23.5	24.36	18.67	19.84	21.685	22.369

Figure 4.2: Speed vector of road r which contains average speed for various timeslots.

Given a time-slot $\mathbf{t}$(t is the index representing the time divided in into the window of w minutes e.g t=12 in w=15 mins represents 3:00am) for a road segment r_i ($r_i \in E$), where $s_t^{r_i}$ represents the average speed of road r_i on time-slot t. For every road segment we create a speed vector denoted by $S_t^r = [s_0^r, s_1^r, .., s_{t-1}^r]$. S_r represents the average speed vector(shown in fig4.2) which contain average speed for all time-slots from 12:00 am to the current time interval $\mathbf{t}$. Max size of S_r is total time slots in a day.

Problem: Given historical observations $\{S_t^r | r = 0, 1....|E-1|\}$, We aims to predict $Y_{t+1}^r = \{S_r^{t+1}\}$ where Y_{t+1}^r is the predicted speed for t+1 slot and t+1 represents the next time-interval for road r.

Chapter 5

Methodology

The following chapter describes the whole methodology to implement STH-model and its perquisites. First we focus on the **Dataset** and then we cover the **Model** overview in depth.

5.1 Dataset

This section mainly focuses on the data extraction and preparation of the spatio-temporal data for model training and for various data analysis.

5.1.1 Data Extraction

The data is extracted from the Open Transit System of Delhi transport with the help of an API provided. The refresh rate of data is 10 seconds the GPS values with ID are transmitted by the DIMTS buses. Every 10th second, we fetch the data from the server. The data contains the GPS trajectories of more than 1500 buses running on the Delhi road network. For a single day, we fetch more than a million GPS trajectories for various trips done by buses.

Busid	Routeid	Latitude	Longitude	Timestamp

Table 5.1: Data points fetched from every API call.

These trajectories contain unique bus ID, route ID (which route is bus currently operating), Latitude, and Longitude of the bus along with the timestamp. These points explain the position of the bus at a particular time. These trajectories are not so useful if they are not mapped to a road. For the road information, the Delhi shapefile provided by OSM(Open Street Maps) is being used. This shapefile gives us road geometry information inline string format. This line string format for a road is a collection of various GPS points that together can form a road when plotting on map. Hence these points can be used as the base road information.

5.1.2 Data preparation

This section is divided into two parts first to explain the preparation of the road network and from second part it is all about the preparation of the dataset(collection of dataunits) used for STH-Model training.

Road Network

From shapefile, we get all the GPS points of their respective roads. For all these points we form the connected edges of (Lat, Longs) with the help of Geometry line strings given in shapefile. A unique road ID is generated for each road segment. For this, we used a DFS algorithm to find the junction points (A junction point is defined as the point where more than two points meet). A similar road ID is assigned to all the edges lie between two junction points. Every edge from a junction point(shown in fig5.1.2) has the unique road id. Assigned road ID signifies the uniqueness of each road in the road network, with the help of junction point and road ID we generate a graph $G \in (V, E)$. In graph G, $r \in E$ denotes the road segment with road ID **r** and $v \in V$ represents the junction point **v**.

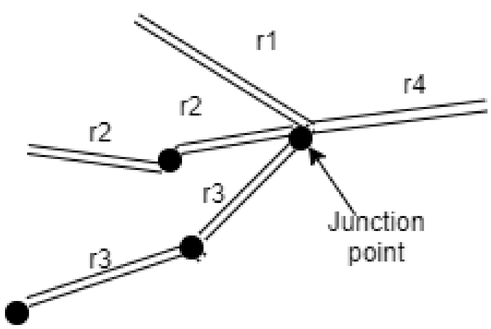

Figure 5.1: Road Network Sample with road ID assigned and a junction point.

Speed Dataset

The GPS traces need to be mapped on the road segment, assigning a road ID to a bus for a particular timestamp. The Delhi map dimensions are divided into the Grid of the equal-sized cell (To reduce the computation time only grid size has no impact on the results here), Every point(Latitude and Longitude) of the road is hashed among these cells. For a GPS point **p** which lies inside Delhi's defined grid and near to road segment, we fetch the Road ID **r**.

Algorithm 1 Fetching Road ID for GPS point p

Result: Road ID r

Input: GPS point p

cellid = $hashtocell(p)$

allcells = cellid + neighbourcells //This is a simple append

visitedcells=allcells

MinimumDistance=MAXDOUBLE

while *visitedcells is not empty* **do**

cell = extractcell(visitedcells) //also remove cell from list

For every road point **rp** in cell:

mindist=calculateMinimumProjectedDistance(p,rp)

MinimumDistance = min(mindist,MinimumDistance)

end

r = assignRoadId(MinimumDistance)

From algorithm 1 we get the road **r** for the GPS point **p**, in algorithm 1 the projected distance or the normal distance is the minimum distance of the point from the edge. The problem of co-linearity is also handled (distance becomes 0 if the point is at 0degree, 180degree to the line), the problem is solved with the help of vector calculus.

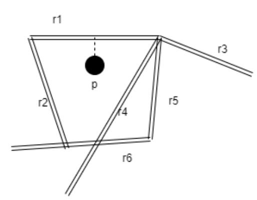

Figure 5.2: Displays the normal distance of the point p from road r1 is minimum it will assign road r1 to it.

Road ID assignment to each GPS trajectory will give the road location of each bus. For each bus, trace speed calculation is done by taking the edge distance between two known points. Algorithm 2 explains the computation of the total distance covered by bus between two locations. The full length of first and last edge is not taken instead the distance between the second edge from the current location is taken and distance of second last edge to the last location is taken, this makes distance more accurate can be seen in figure 5.1.2. The length of the edge is the haversine distance between two endpoints of the edge. The speed of the bus is calculated with this total distance and the time difference between these two known points. The speed is considered as the average speed of the bus and assumed that this speed is uniform on all the in-between edges/road.

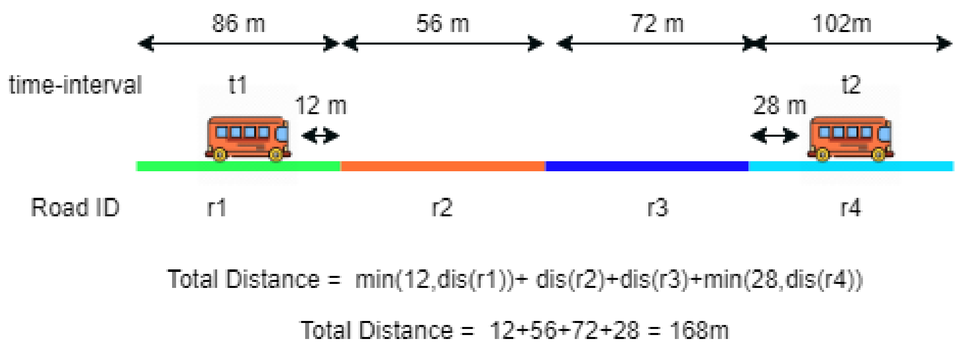

Figure 5.3: Displaying the distance calculation of the bus with respect to the roads.

Algorithm 2 Computing the total distance covered by bus between two time-intervals

Result: Total Distance d

Input: Location l1,l2

```
path = computeshortestpath(l1,l2)
initialedge = path[0]
finaledge = path[-1] //last edge
totaldistance = distance(l1,path[1]) // Computing the distance of bus from l1 to edge1
totaldistance += distance(path[-2],l2)// Computing the distance of bus from second last edge
while path is not empty & path!=initialedge or finaledge do
    ed = path.dequeue()
    totaldistance+= length(ed)
end
return totaldistance
```

After calculating the speed of every bus on the particular road, the average speed of each road for the timeslot **t** where timeslot is of time interval size **w** (where w is in minutes e.g. 5 mins, 10 mins,15 mins and so on. For example if **w** is 15 mins then we have 4 slots for every hour.).The calculation results in missing values for some time slots on some roads, this is due to the data sparsity problem which is because of less number of readings(as 1500 buses are running on the Delhi road network which itself contain the huge number of roads).

The missing values can lead to improper learning of trends and can result in a high error rate. [20] explains the technique to fill the Spatio-temporal data missing values, we will explain the missing values problem in later sections. We fill all the missing time slots with the speed values. After filling up missing values, the data will look like a vector for road r where $r \in setofRoadIDs$ containing $24 * 60/\mathbf{w}$(for w =15 there are 96-speed slots) speed values where each index represent time slot (e.g. If timeslot is of 15 mins then index 20 will represent 5:00 am). This data act as the base dataset for our further data-preparation techniques.

Missing values

The most challenging part is the missing values in data. The missing values here can be defined of two types first when a GPS trajectory from the bus is expected at every 10th second but due to some internal (Server error, API failure, etc) or external (bus is in no network zone, device failure, etc) factor no value is received. This leads to no value and causes the missing value problem. The second type of missing value is when we calculate the speed for each road for every time-slot but some of the time-slots do not record any speed in a particular time-slot due to data sparsity as the number of buses transmitting data is very less compared to the number of roads. This leads empty slots in for roads at various time intervals. We have accounted for both the situations and handled both of them an overcome the problem. To handle the first situation we find the points where the data transmission is lost and where it is regained, we then fill these two points with the most used path by the bus and filled all the in-between roads with avg speed at the calculated time based on speed and distance. This results in the complete path that can be seen in image 6.2 part c and d.

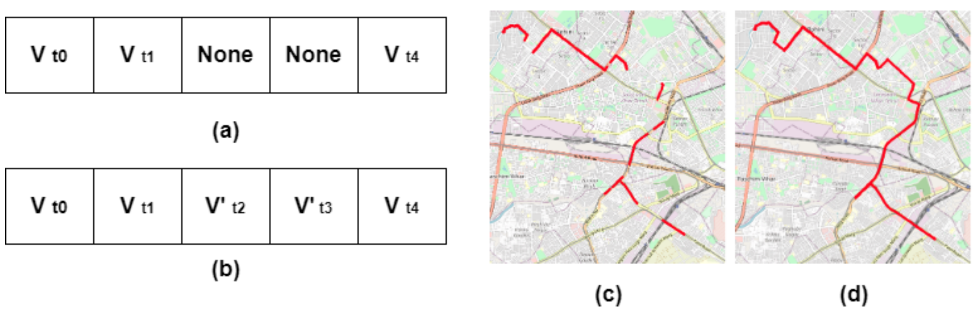

Figure 5.4: (a) vector of Road r contain missing value displays no speed is recorded in that slot,(b) displays filling the missing values in the vector, (c) displays bus did not transmit data at different locations,(d) completion of the path by our Route Maker module.

For the second situation there are many ways in which we can fill the missing values.Simplest one is the linear interpolation between two known values but this technique make data more linear and the linear trends create the model biased towards it. But in real scenario we rarely face these real trends as data is always have some noise in it. Inspired with the techniques in [20] and [14] to fill the spatio-temporal data filling technique. We have used three domains to fill our data Spatial Domain, Temporal Domain and Historical speeds. In spatial domain we took **K** neighbours of road **r** and than calculate the spatial velocity with this formula.

$$SV_{r,t} = \sum_{k}^{K}(V_{k,t} * d^{-\alpha} * sim(r,k))/(\sum_{k}^{K}(d^{-\alpha} * sim(r,k)))k \in K \tag{5.1}$$

here t is the timeslot for which we wants to fill and d is the distance between road r and road k , α is the decay rate(in our case 1) more the road is far less significance it has. sim is the similarity function in both the roads it basically tells how these two roads are similar on scale of 0 to 1. In $V_{k,t}$ k represents road and t represent timeslot For the temporal domain we took the current days speed trends on the road starting from 00:00 am. The velocity for temporal domain given by

$$TV_{r,t} = \sum_{i=0}^{t-1}(V_{r,t-i} * \beta^{t-i}) / \sum_{i=0}^{t-1}(\beta^{t-i}) \tag{5.2}$$

here β is the smoothing factor(in our case 0.5). For final missing value we take the weighted sum of all these with history speed of the road.

$$V'_{r,t} = w0 * SV_{r,t} + w1 * TV_{r,t} + w2 * History_{r,t} \tag{5.3}$$

hence $V'_{r,t}$ is the final value to be filled for road r on time-slot t.

5.1.3 Data Formation

Filling missing values to the base data complete the dataset for every timeslot. The main task in this is that we need to form our data in such a way that we can extract features of spatial as well as temporal from that. The Neighbourhood size **N** and Temporal domain **T** are the two parameters used to define the data. Neighborhood size is the number of neighbors (neighbors can be adjacent as well as one-hop neighbors because some time neighbor road segments are very small that bus does not record data on them.) to be considered with the road r like if N=3 we will consider 2 similar(based on similarity and distance) neighbors of r in the dataset. The temporal domain is the number of previous time-slots to be considered of road r and its N-1 neighbors to predict the speed of next time-slot if T=4 then we consider 4 timeslots and 5th one as the target value of road r. So the data is formed with the help of **N** and **T**. The figure 5.1.3 will help you to understand better in this context. The data unit **D** looks like a small matrix with dimension **T X N**, which is also the input data for predicting speed for time-slot t+1 for road r.

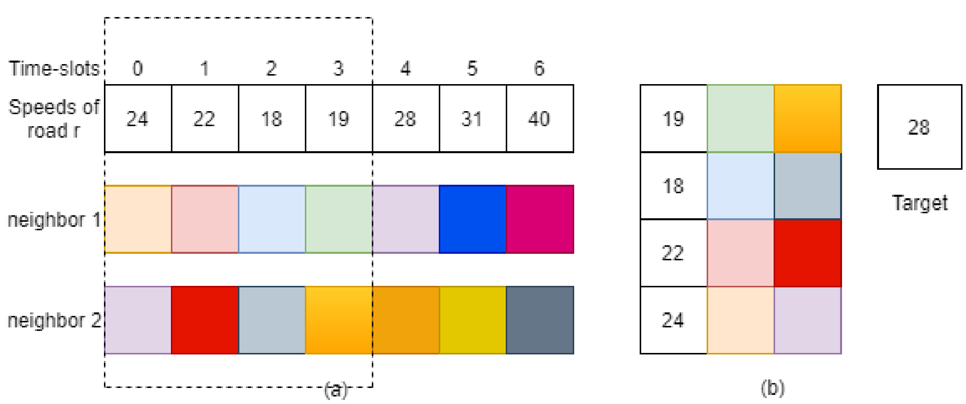

Figure 5.5: (a) is the speed vector of road r and its neighbors (colors also represent the speed), (b) is the data sample for T=4 and N=3.

5.2 Model Overview

5.2.1 Overview of Spatio-Temporal-History Model

Figure in section 5.2.2 represents the architecture of the STH(Spatio-Temporal-History) model, which is comprised of the CNN, LSTM, and Fully Connected layers. Where the general idea is to capture the spatial dynamics of the traffic flow from its neighbor roads with the help of CNN and also taking an account of the speed patterns on the current day at previous time intervals using LSTM which performs well on time series data. At last, fully connected layers are used for feature extraction and introduce the historical trends to the prediction. The introduction of the historical trend boosts speed prediction accuracy.

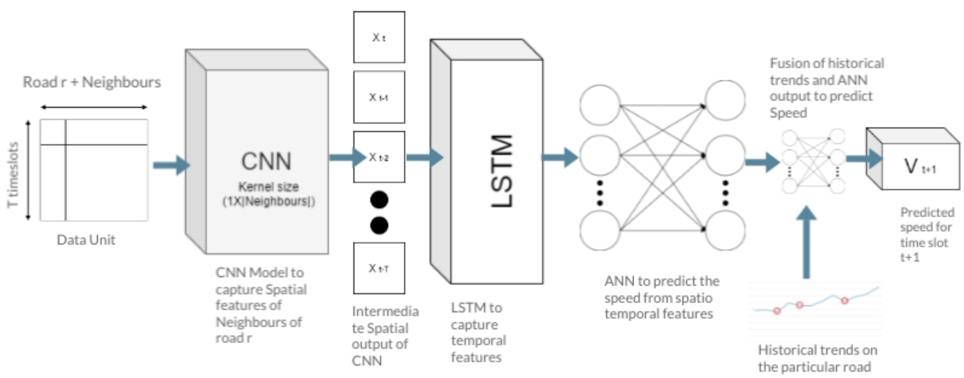

Figure 5.6: STH-Model block diagram,CNN : Convolution Neural Network, LSTM: Long short term memory,

5.2.2 CNN

CNN is the most favorite tool of Deep learning when we have image data or textual data. CNN is widely used to capture the spatial relationship between the adjacent pixels. Most of the researchers use this property of CNN for traffic speed or congestion prediction when they have image data [10] or sensor data [7]. So we also decided to use this property of CNN and modeled our data in such a way explained in above sections.

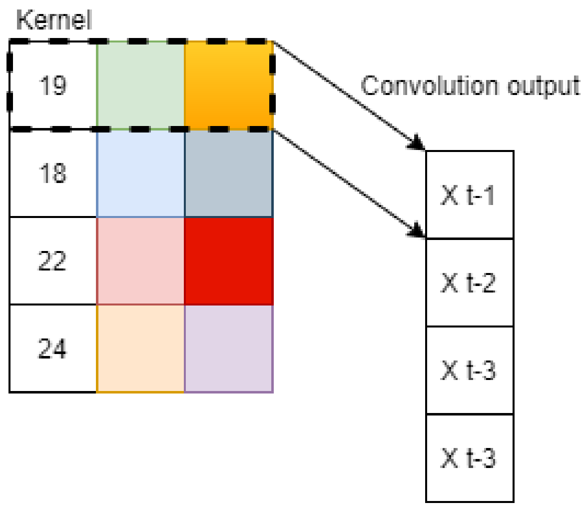

Figure 5.7: Convolution operation in CNN

The data unit **D** convolve with filter **f**. We deliberately choose a filter of size **1 X N** (Neighbourhood size) because we want out CNN to just extract spatial features of the same time-interval. We get the following **X** as the output

$$X = [X_0, X_1, ..., X_{T-1}]$$
$$X_t = relu(D[t, :] * W^T + b)$$

D[t,:] is the extraction of the row for the data unit. We fetch this intermediate result and reshape it to feed it to LSTM it extracts the temporal features from it. We use h X N filter size where h=1 to get descriptive features where h>1 gives us diversified features, we have found out that both are giving similar results.

5.2.3 Recurrent Neural Network / LSTMs

Over past time RNN proves to be a good Deep learning tool over time series data [3]. CNN results in spatial features and these features do not contain any temporal information. So we feed the output to the RNN, but RNN suffers from gradient vanishing problem. To overcome this problem we have LSTMs that use various gates for it. Result X from obtained CNN is reshaped and given to the LSTM with the initial hidden state,

$$V_h = LSTM(X, h_i)$$

here h_i is the hidden state. and V_h is the output hidden state of the LSTM which contains the Spatio-temporal features of the road r and these features are used by ANN(Artificial Neural Network) to predict speed.

5.2.4 Fully connected layer and Historical trends

For the **Vh** we get from the CNN-LSTM we forward this information to an ANN(Artificial Neural Network) which applies certain weights on it to predict the speed from it. The predicted speed is based on the Spatio-temporal data and there is no information regarding the historical speeds of the road. Let's consider a situation: From time-slot, t on-road r due to congestion average speed decline with the increase in time-slot and this is a daily trend but the model will not account for these changes on the basis of just Spatio-temporal. The predicted value from the output of the list is,

$$Y_p = V_h * W_p^T + b_p$$

Y_p is the predicted value from V_h after applying W_p weights and b_p bias. The historical trends of the road are as follows,

$$Y_h = H * W_h^T + b_h$$

Y_h is the historical features extracted from **H** historical trends of road r. Historical trends are the speed trends of a road r at time slot **t**, if we are doing prediction for slot **t**.

The predicted speed Y_p is predicting close to the actual speed. To make it more accurate we introduce historical trends to it. Hence we decided to fuse both Y_p and Y_h together to get more accurate result,

$$Y_t^r = (W_0 * Y_p^r + W_1 * Y_h^r)$$

here r is the road, t is the time slot we are predicting and W_0 and W_1 are the learn-able weights. This equation looks like similar to $y = x * W^T + b$, so we assign a small ANN to do the Task.

$$Y_t^r = [Y_p, Y_h] * W^T + b$$

where Y_t^r is the predicted speed value for road r at time-slot t. We used back-propagation method on MSE(Mean Squared Error) to learn the weights. The Loss is defined as,

$$L(W) = \|Y - X\|_2^2$$

here W is the learnable parameters, Y is the predicted speed, and X is the actual speed. The introduction of historical trend boost accuracy and reduce the error.

Chapter 6

Experimental Results

We mainly experiment on the road network of Delhi, where the road network is considered as the graph and each edge represents the road segment from one junction (where two or more roads meet) to another junction and nodes represent the junctions. The GPS trajectory data is being collected from Jan 20, 2020, to March 10, 2020. There are more than 15000 road segments that are used by buses every day. The length of these road segments may vary from 30 meters to 1.8 Km long. We get around 2 million GPS traces every day from these buses.

6.1 Data Preparation

For all these traces we calculated the speed of the corresponding buses on the particular road, these speeds are further divided into time slot buckets lets say a bucket of 10min or 15 min. The average speed of a road for a particular time slot is calculated and filled with the missing value. We have used the 1-month data for training and the 5 weekdays data for testing as the training is also done on weekdays data, not on the weekend. The data-set is the collection speeds from time-slot t,t-1,t-2,t-T (Here T is the time domain to be considered for training), and the target value if the predicted speed for time-slot t+1 of road r along with n neighbors. Then the data for road r for particular time-slot **t** will look like a matrix of speed d[T][n]. the value of t lies from 5:00 am to 11:30 pm as there are very fewer entries before and after these timings mainly the data is from depot stations. In the pre-processing step, we use Min-Max normalization to scale the data elements in the range of 0 to 1 and re-scaled the output to the normal form. For training purposes the following hyper-parameters values were used batch-size(2 56),1X$\|Neighbours + 1\|$ filter size, learning rate 0.001 with Adam optimizer. 80% is used for training and 20% used for validation from training dataset.

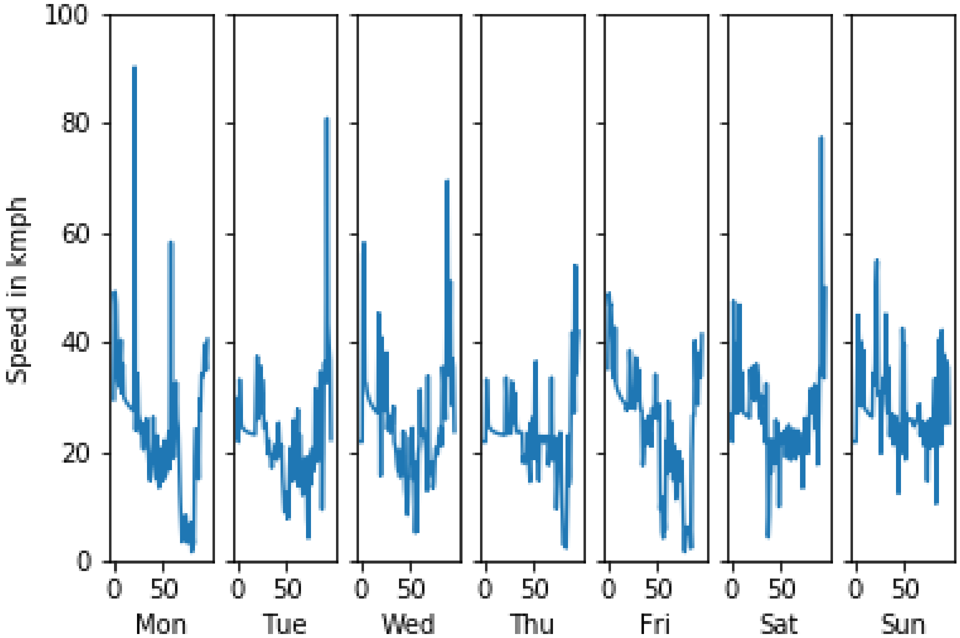

Figure 6.1: Displaying the week speed trends of the Okhla NSIC to Okhla Mandi road, At the end of the day it can be noted impact of congestion during weekdays not on weekends.

6.2 Model Learning

In this section we see the results by varying the data, like varying neighbour hood size, temporal domain, time-slot size.

6.2.1 Effect of time-slot size

The time-slot is the time window size in minutes where we record the speed readings and average out for each of these slots and this speed is the average speed of the road the corresponding time-slot example: if time window is 15 minutes then time-slot 60 will represent 3:00 pm. The various time-slots does have varying missing values, shown in figure 6.2.

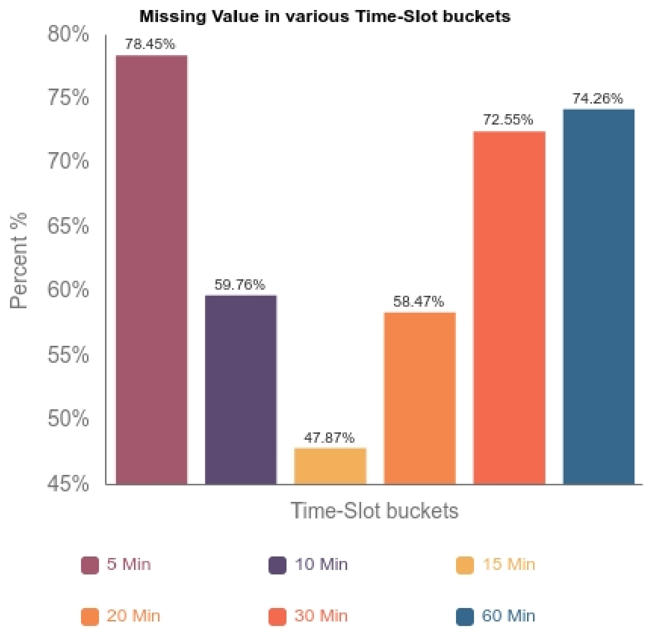

Figure 6.2: For different time-slot size the percentage of the Missing values in the data.

The 15 min slot is having less percent of missing values and we are having more sensor data to operate as compared to others.

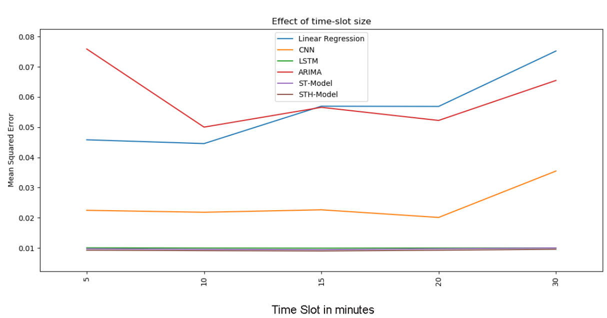

Figure 6.3: Performance of all benchmarks vs STH-model on various time-slot

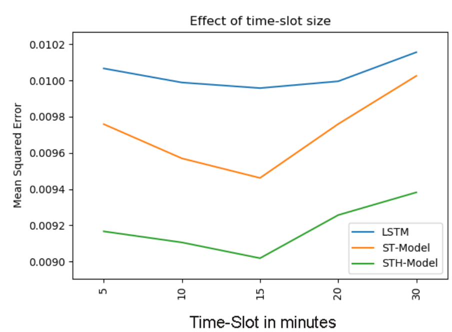

Figure 6.4: Performance of LSTM and STH-model on various time-slot

The above figure shows the effect of time-slot, STH-model shows better performance on every slot but is more accurate on 15min slot. This is due to less missing values and we have more actual reading. The T=10 and N=8 for our ST-Model and STH-Model. LSTM also shows the better performance than other benchmarks. Thus we do not conclude that the 15 min widow size is optimal but for our data points it is showing good results.

6.2.2 Spatial,Temporal and Historical learning

STH-model accounts for all the factors spatial, temporal, and historical. In this part, we are showing the performance of these three individually. Spatial learning is basically a CNN, FC-layer to learn only the spatial features, while temporal learning is an LSTM which fed with a time series data. In historical learning, we create a regression model based on historical data.

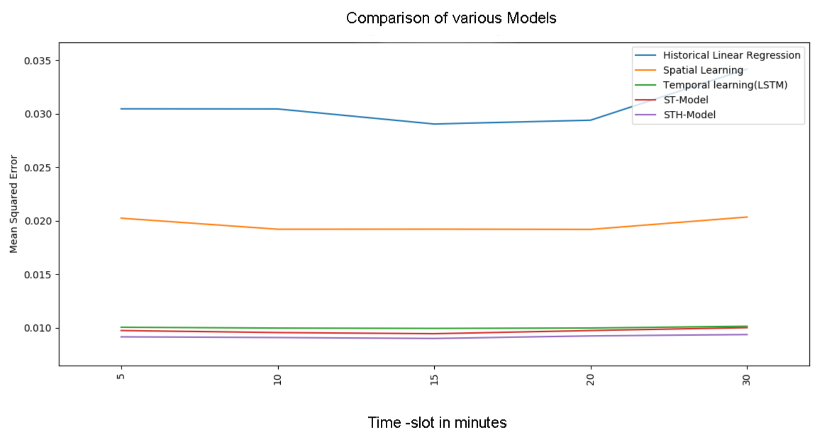

Figure 6.5: Comparing spatial learning, temporal learning, historical learning with our STH-Model

The spatial and historical learning is not performing as good as the STH-Model. The MSE is very low for the STH model if all three were combined even lower than our ST-Model(without historical trends). This shows that STH-Model account for all these three factors and shows improve accuracy. From the above figure we can observer that the temporal trends play major role in understanding the traffic dynamics. And the inclusion of the all STH trends we improves the prediction accuracy and gives the more accurate result.

6.2.3 Effect of Neighbor-hood size(N)

This experiment shows the performance of model on different sizes of **N**, As seen in figure with increase in size of N the error decreases.

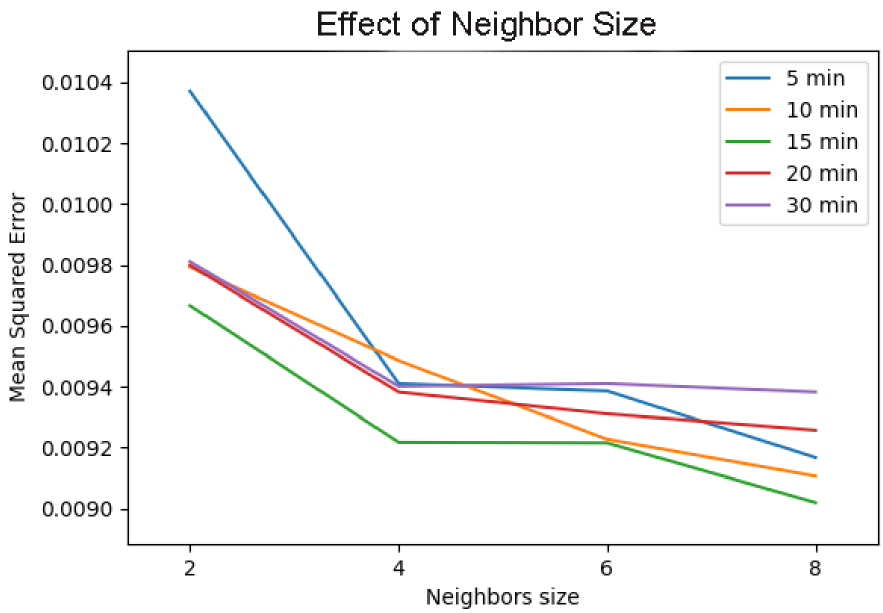

Figure 6.6: STH-Model performance on different Neighbor-hood sizes

T is fixed to 10, for N=8 we can see the 15 min time slot is giving the least error this shows that the model gives the best output when it considers more neighbors. If the neighbor size is very low the model does not capture the spatial traffic dynamics. N=8 is set as the upper limit as it is the average number of neighbors for the road r. Increasing the N>8 creates more influence of other neighbors which are not even show similarity to road r. The neighbors are selected on the basis of the similarity index between the road r and neighbor road r_n.

6.2.4 Effect of Temporal domain(T)

The temporal domain T specifies the number of time series points needed to predict the next slot. The more temporal trends to capture make the model more accurate and precise. Here we fix the value of N=8 and vary the T and found out that for the T=10 the model is most accurate when used of 15-minute time-slot. It for T>10 the error is increasing for timeslot 5,10,15 but is less for timeslot 20,30. But they are not showing as good performance than T=10 on 15 min slot data. Here T<10 shows that the number of temporal points is low to capture time-series patterns. For 15 min slot size T=10 is found to be the best temporal window domain in simple words we can say we are recording the previous 2.5 hrs traffic dynamics of the road to predict next 15 minutes.

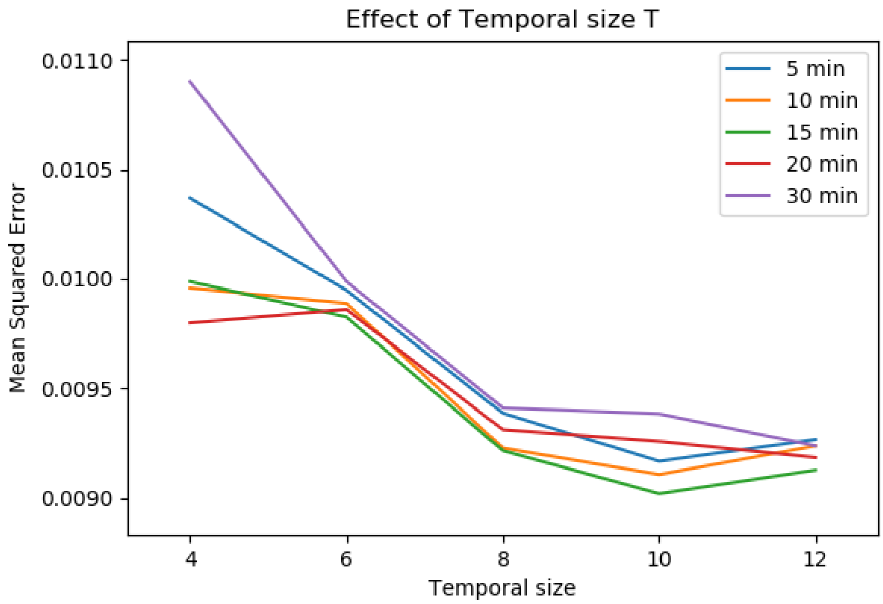

Figure 6.7: STH-Model performance on different Temporal Domain sizes

6.3 Performance

We used MSE(Mean squared error) for comparing the performance of different models and their variations. MSE represents the squared difference between actual value and its predicted value and the error always a positive number.

6.3.1 Performance comparison with other benchmarks

There are various benchmark techniques to compare the model, so we choose the very well known one.

- **Linear Regression:** As speed prediction is nothing but can be formed as the regression problem. Linear regression is one of the very well known Machine learning approaches used for prediction.
- **CNN:** CNN is one of the most powerful deep learning tools to account for the spatial features of the data. It uses Convolution layers in which with the help of filter the spatial featured are learned.
- **LSTM:** LSTMs are the variants of RNN capable of learning long term dependencies, we can use this property of LSTMs to learn the temporal features.

- **ARIMA:** Auto-Regressive Integrated Moving Average are the models which are used to predict the forecast value based on the previous past values

Methods	MSE
Linear Regression	0.74269268
CNN	0.22654723
LSTM	0.00995786357
ARIMA	0.56575127
ST-Model	0.009461862627
ST-Model-fuse-History	**0.009018740617**

Table 6.1: Comparison of different models

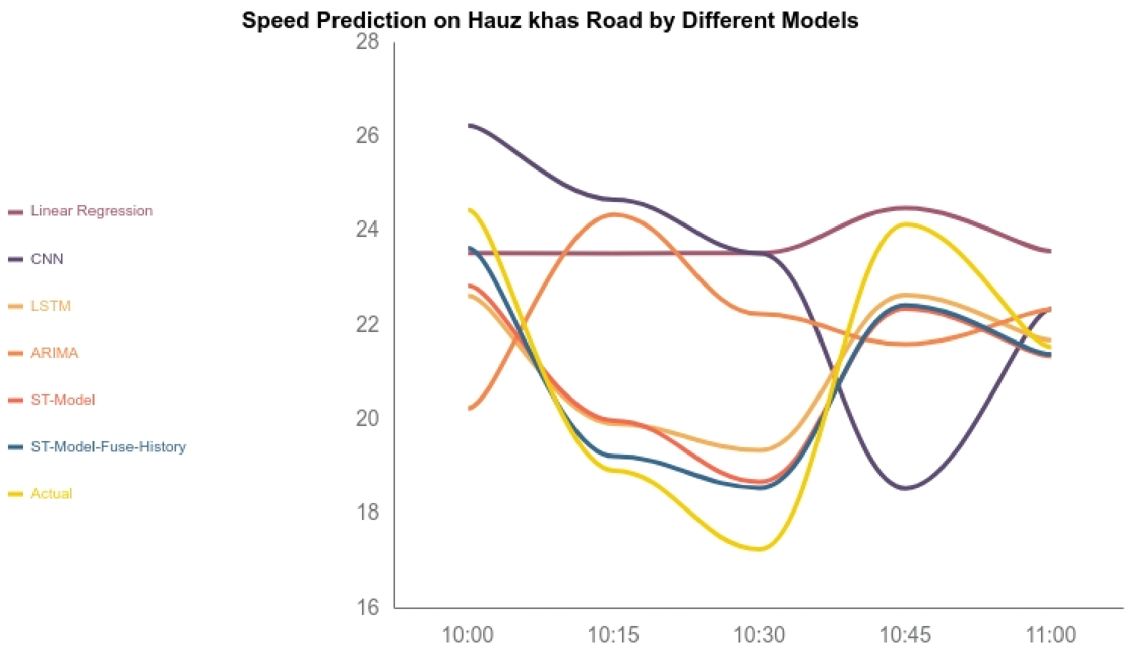

Figure 6.8: Comparison of speed Predictions with different Models

LSTM model is also having low error as a compared model other than ST-Model because it accounts for the temporal factors in the dataset. On the other hand, ST-Model accounts for both the factors spatial as well as temporal. So the error is very low as compared to others. When the historical speed trends are fused with the ST-model then we can note a little drop in the error again.

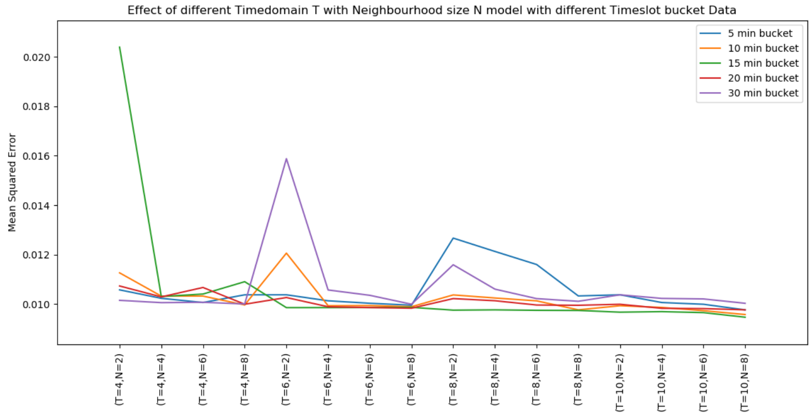

Figure 6.9: Effect of different values of T(temporal domain), N(Neighbourhood size) and different timeslot buckets on Model.

6.3.2 Performance comparison of variants of our Model

Various factors can affect the learning of the model, like Temporal domain T, Neighbourhood size N, Various time-slot buckets, introduction to historical speed trends of the road. With taking all these factors in the account after several training experiments we have found out that the 15 mins time-slot buckets are giving the best result when historical trends were not introduced.

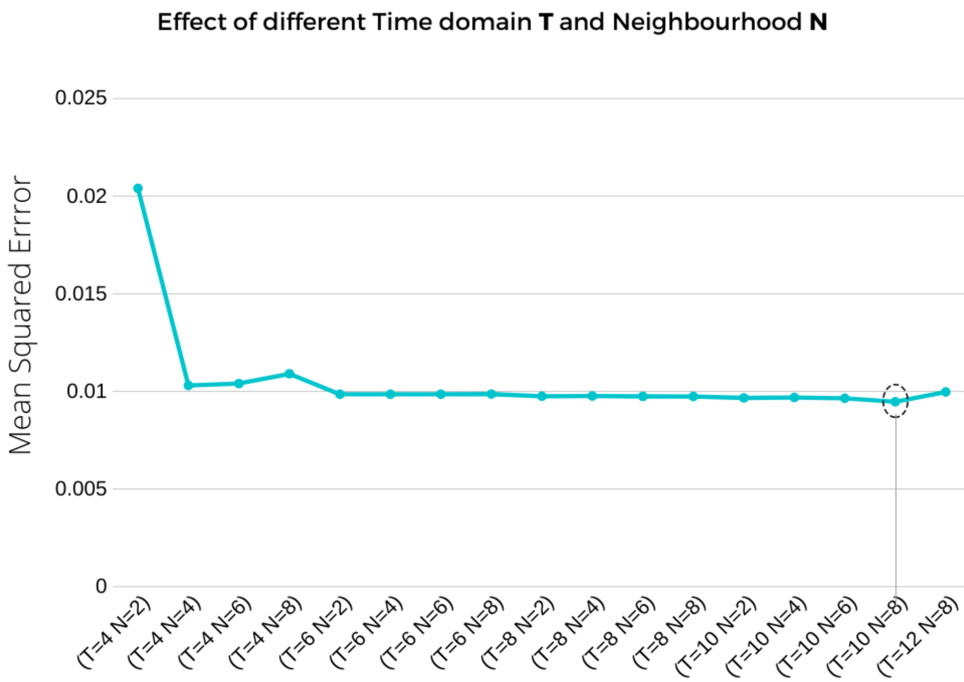

Figure 6.10: Effect of different values of T(temporal domain), N(Neighbourhood size) and 15 timeslot buckets on Model. Optimal value can be seen on T=10,N=8.

We can say that the model is giving the best results on 15 mins time-slot bucket when we increase the Spatio-temporal domain. Because it is accounting more of these features to predict the speed.

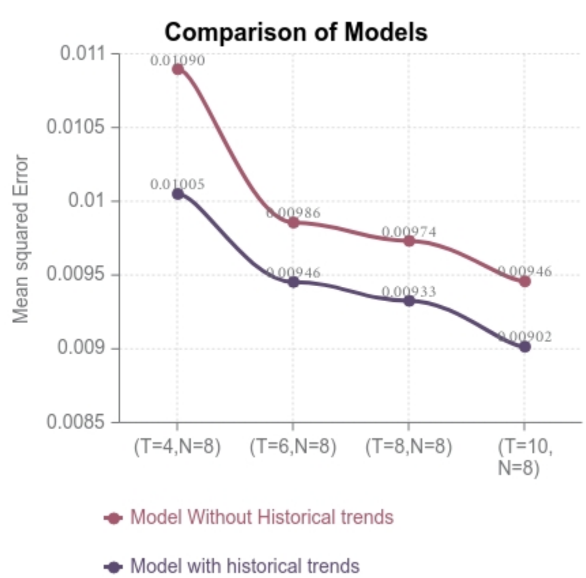

Figure 6.11: Model without consideration of historical trends and with historical trends fusion

The introduction of the historical trends to the model also helps the model to move towards the optimal value, As these trends also help the model to understand the history of the road in the particular time-slot. This can be noted that the model in which historical speeds are fused shows better performance than the model without historical trends.

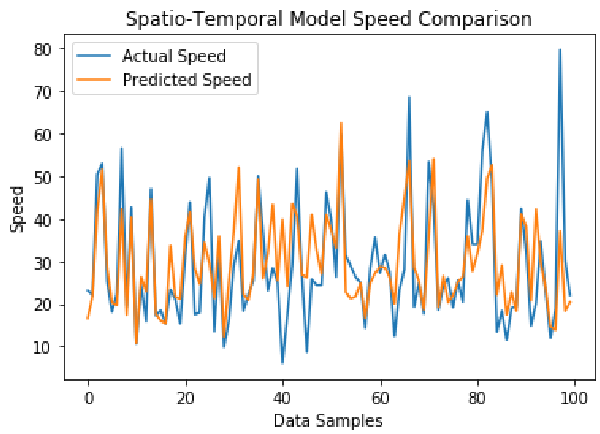

Figure 6.12: Predicted speed and actual speed plot on test data.

Chapter 7

Conclusion, Limitations, Future Work

7.1 Conclusion

In this paper, we propose a deep learning STH-model which predicts the traffic speed. Our model not only integrates the functionality of both spatial and temporal features but also take the benefit of the historical trends of the road. This gives more accurate results and also improves the accuracy of the model. The model's can be benefited for applications like Estimated Arrival time of the bus on the next stop and many more. The prediction can be used by the traffic personnel to identify the congestion, so they can pro-actively respond to it. It can also be integrated with applications like the poochoo-app and DIMTS app to get the arrival time of the bus and help government agencies and end-user to monitor buses.

7.2 Limitation

Data preparation is the time-consuming process if the time-slots size is less then the time for computation increase significantly. The model does not preserve any context of a specific road it just tries to learn the pattern from the data.

7.3 Future Work

The future work consists of the Estimated arrival time prediction of the bus, this information is very helpful for the users waiting for the bus on the stop. Future work also consists of learning weather trends, as we notice that Delhi takes advantage of all four seasons and during the rainy season the average speed decreases. Delhi also faces the problem of pollution due to which the government enforces the ODD-EVEN scheme on road, we also want our model to learn these external factors also.

Bibliography

[1] Charalampos Bratsas, Kleanthis Koupidis, Josep Maria Salanova Grau, Konstantinos Giannakopoulos, Aristides Kaloudis, and Georgia Aifadopoulou. A comparison of machine learning methods for the prediction of traffic speed in urban places. *Sustainability*, 12:142, 12 2019.

[2] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.

[3] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[4] G. Gopi, J. Dauwels, M. T. Asif, S. Ashwin, N. Mitrovic, U. Rasheed, and P. Jaillet. Bayesian support vector regression for traffic speed prediction with error bars. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 136–141, 2013.

[5] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.

[6] Sonia Khetarpaul, S. K. Gupta, and L. Venkata Subramaniam. Analyzing travel patterns for scheduling in a dynamic environment. In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl, and Lida Xu, editors, *Availability, Reliability, and Security in Information Systems and HCI*, pages 304–318, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[7] Youngjoo Kim, Peng Wang, Yifei Zhu, and Lyudmila Mihaylova. A capsule network for traffic speed prediction in complex road networks. 10 2018.

[8] Jason Kurniawan, Sensa G.S. Syahra, Chandra K. Dewa, and Afiahayati. Traffic congestion detection: Learning from cctv monitoring images using convolutional neural network. *Procedia Computer Science*, 144:291 – 297, 2018. INNS Conference on Big Data and Deep Learning.

[9] Duanyang Liu, Longfeng Tang, Guojiang Shen, and Xiao Han. Traffic speed prediction: An attention-based method. *Sensors (Basel, Switzerland)*, 19(18):3836, Sep 2019. 31491921[pmid].

[10] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Na, Yong Wang, and Yunpeng Wang. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction, 2017.

[11] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-scale transportation network congestion evolution prediction using deep learning theory. *PLOS ONE*, 10(3):1–17, 03 2015.

[12] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[13] Bharti Sharma, Sachin Kumar, Prayag Tiwari, Pranay Yadav, and Marina I. Nezhurina. Ann based short-term traffic flow forecasting in undivided two lane highway. *Journal of Big Data*, 5(1):48, Dec 2018.

[14] Antti Sorjamaa, Amaury Lendasse, Cornet Yves, and Eric Deleersnijder. An improved methodology for filling missing values in spatiotemporal climate data set. *Computational Geosciences*, 14:55–64, 02 2010.

[15] Joseph P Tarris, Christopher M Poe, John M Mason Jr, and Konstadinos G Goulias. Predicting operating speeds on low-speed urban streets: Regression and panel analysis approaches. *Transportation Research Record*, 1523(1):46–54, 1996.

[16] L. Wang and Z. Zhou. Congestion prediction for urban areas by spatiotemporal data mining. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 290–297, 2017.

[17] P. Wang, L. Li, Y. Jin, and G. Wang. Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a cnn-architecture trafficnet. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1134–1139, 2018.

[18] Li Wei and Dai Hong-ying. Real-time road congestion detection based on image texture analysis. *Procedia Engineering*, 137:196 – 201, 2016. Green Intelligent Transportation System and Safety.

[19] Chun-Hsin Wu, Chia-Chen Wei, Ming-Hua Chang, and Jan-Ming Ho. Travel time prediction with support vector regression. pages 1438 – 1442 vol.2, 11 2003.

[20] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. St-mvl: filling missing values in geo-sensory time series data. 2016.

[21] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, page 99–108, New York, NY, USA, 2010. Association for Computing Machinery.