Network Level Travel Time Prediction on Urban Arterials Using Machine Learning Techniques

Hareesh Bahuleyan
CE10B024
Guide: Dr. V. Lelitha Devi

Final Review Presentation, Dual Degree Project

OUTLINE

- Introduction
- Literature Review
- Network and Data
- Link Travel Time Prediction
- Travel Time Prediction on Intersection Links
- Link to Path Projection
- Conclusions

Introduction

- Need for Real-time travel time information
- Challenges: Recurrent and Non-recurrent congestion
- Sources of Data
  - Loop Detector
  - GPS
  - Bluetooth Sensors
  - Video

Motivation

- Travel time prediction (TTP) on urban arterials is more challenging than on freeways

- In most studies for urban arterials, the route or corridor was fixed.
- Very few studies have addressed the problem of network TTP.
- Travel time information on each and every link in the network – User given the choice of origin, destination and route.

- Need for splitting certain links into intersection and midlink – to be analyzed separately

LITERATURE REVIEW

Techniques

Model Type	Authors	Remarks
Historic Data-Based	Hoffmann *et al.* (1990)	✓ Significant prediction errors in non-recurrent congestion scenarios
Linear Regression	Nikovski *et al.* (2005) ; Kwon *et al.* (2000)	✓ Coefficients and relationship between variables may not remain constant over space and time
Time Series	Guin (2006) Khoei *et al.* (2013)	✓ When current traffic conditions vary significantly from historic data - > Accuracy decreases
Kalman Filtering	Chien *et al.* (2003) Vanajakshi *et al.* (2009)	✓ No Historic data requirements ✓ Choosing initial values and subsequently updating error covariance terms
Machine Learning	Wu *et al.* (2004) Lee (2009)	✓ Possible to capture non-linear relationships ✓ Data-intensive and computationally expensive

Network Level Prediction

Authors	Remarks
Liu *et al.* (2005)	✓ Loop Detector data ✓ TT = link cruising time + intersection delays ✓ Intersection delays calculated with analytical formula
El Esawey and Sayed (2011)	✓ TT on inks with inadequate number of location-based sensors ✓ Predict with available data from neighbouring links and historic data
Lee *et al.* (2009)	✓ Linear combination of a historical predictor and a real-time predictor ✓ Weights assigned to the components were varied dynamically
Jenelius and Koutsopoulos (2013)	✓ Network model that used a correlation between TT on different links ✓ Observation model in which the mean and variance of link travel times were expressed as function of several variables
Hofleitner *et al.* (2012))	✓ Combination of machine learning framework with traffic flow theory ✓ Expectation Maximization (EM) to find probability distribution of link TTs

Intersection Queues and Delays

- **Video** (Cheek , 2007)
  - Video Imaging Vehicle Detection Systems and Image Processing;
  - Limited visibility of the vehicular queue
- **Loop Detectors** (Newell, 1965; Sharma et al., 2007; Anusha et al., 2013)
  - Simple Input-Output method, conservation of traffic flow
  - Limitation due to errors in volume counts
- **Shockwave Theory** (Wirasinghe, 1978; Liu et al., 2009)
  - Assumption of deterministic vehicle arrivals, no spillover between cycles
- **GPS Data**
  - Liu et al. (2013) analyzed the spatio-temporal distribution of turn delays at intersections using taxi cab data
  - Comert and Cetin (2009) – probabilistic model that required input as location of last probe vehicle
  - Liu et al. (2006) - sensitivity analysis of delay calculation to different GPS transmission frequencies

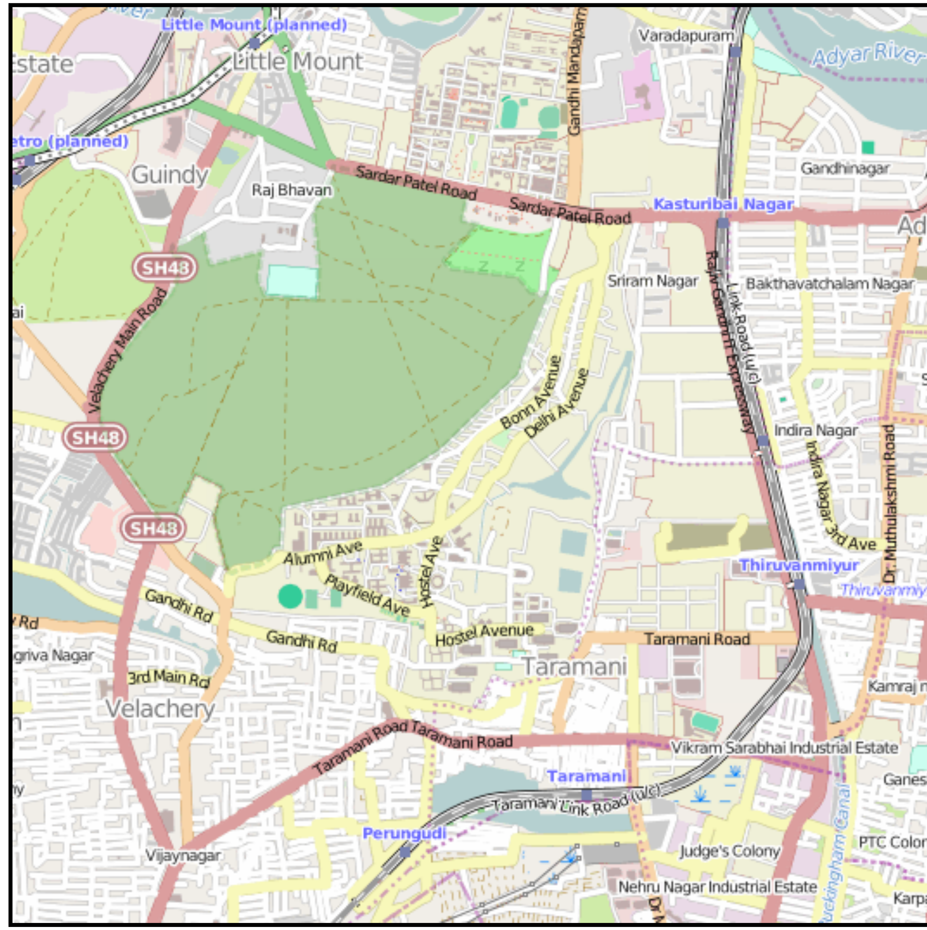
Research Objectives

- Develop a methodology to predict the travel time for an entire network through a link-by-link approach
  - Map-Matching
  - Exploratory Analysis
  - Link Travel Time Prediction
  - Intersection Link TTP with Spatial Data
  - Intersection Link TTP with Location Based Data
  - Link to Path Projection

NETWORK AND DATA

Network and Data

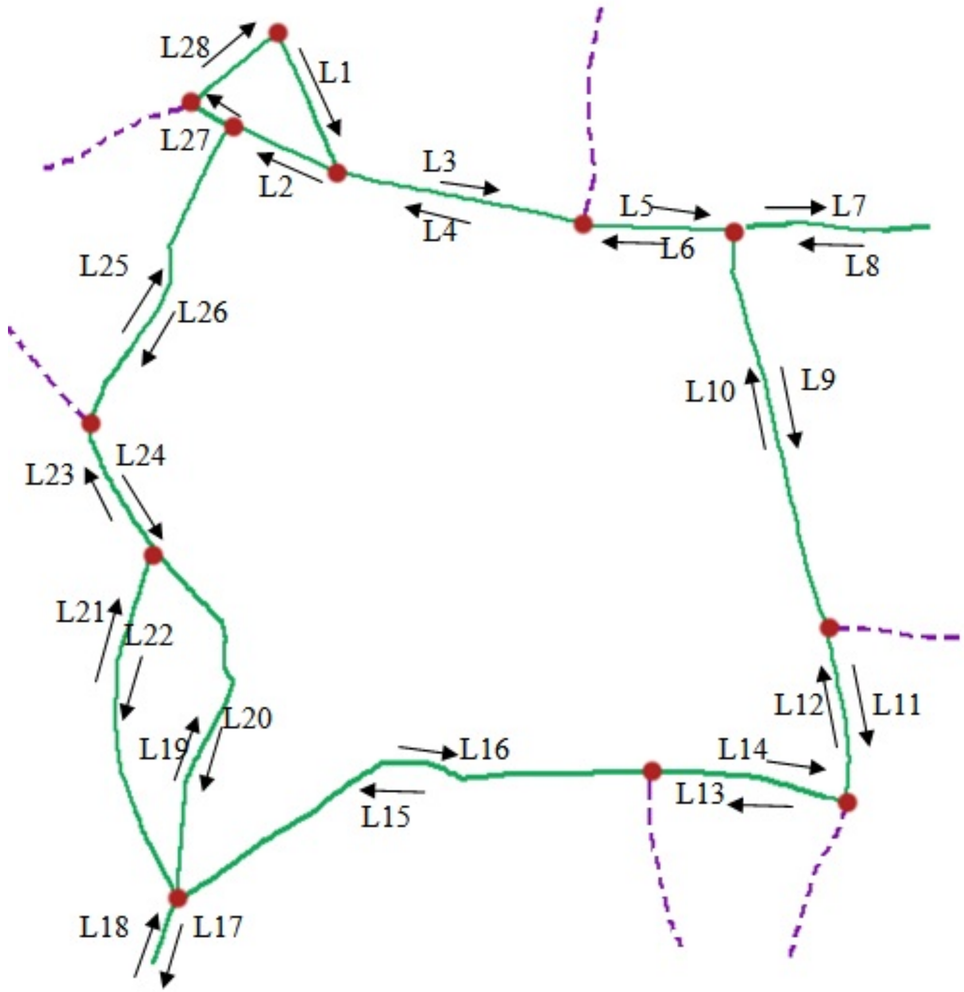
- Study Network around IITM
- Data Sources:
  - GPS data from Buses
  - 53 MTC buses
  - 6 months data
- GPS data has timestamp, latitude and longitude



Network Extraction

- **Links:** Primary, Trunk and Secondary roads – links on top of the OSM hierarchy
- Network has 28 links

- **Nodes:** The end points of the above links



Network and GPS Data

Link ID	Link Name	Start Node		End Node	
		Latitude	Longitude	Latitude	Longitude
L3	Sardar Patel Road	13.00948	80.22814	13.00683	80.24022
L19	Velachery Main Road	13.00668	80.24742	12.98779	80.25139

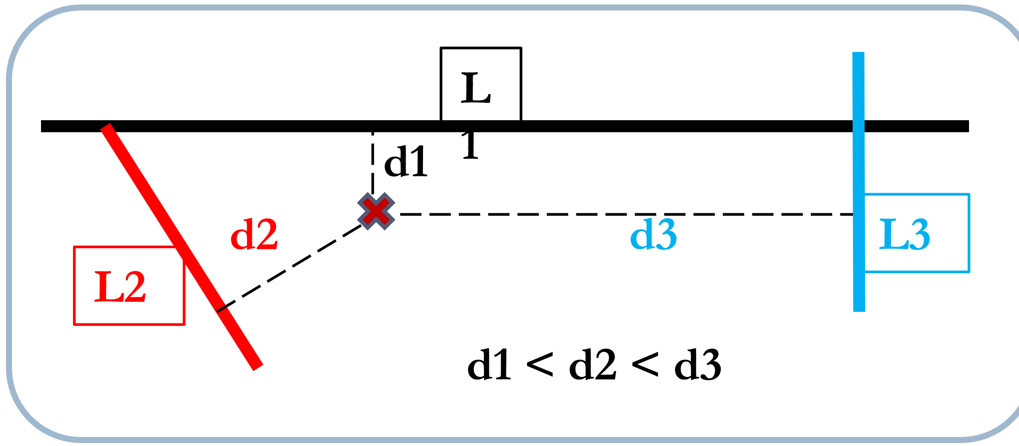
Network Data

GPS Log Data

Time Stamp	Latitude	Longitude
0:00:19	13.01434	80.22534
0:00:40	13.01435	80.22537

Map Matching

- Relate each GPS timestamp in Log data table to a unique link in Network Table



- **Algorithm:**

For all GPS timestamps

{

If it lies within the network

{ Find the distance between that point and all the links in our network }

{ Assign the shortest distance link to that GPS time-stamp }

If it lies outside the network

{ Assign NA }

}

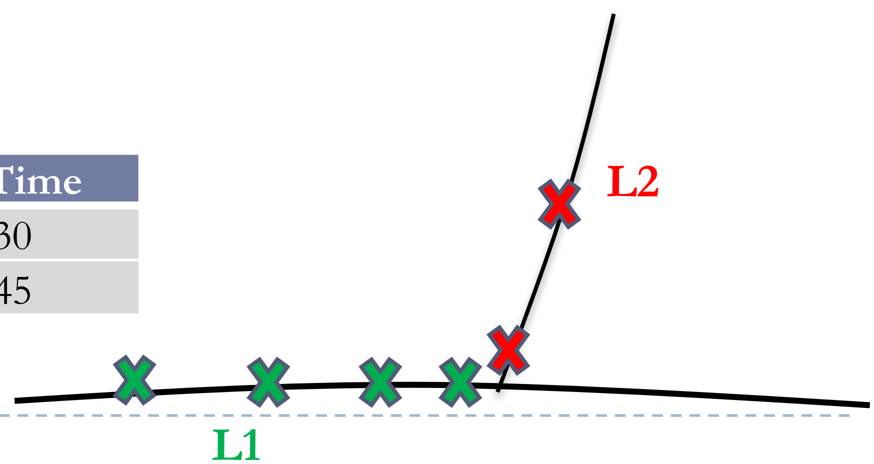
Map Matching Results

Bus enters link L1

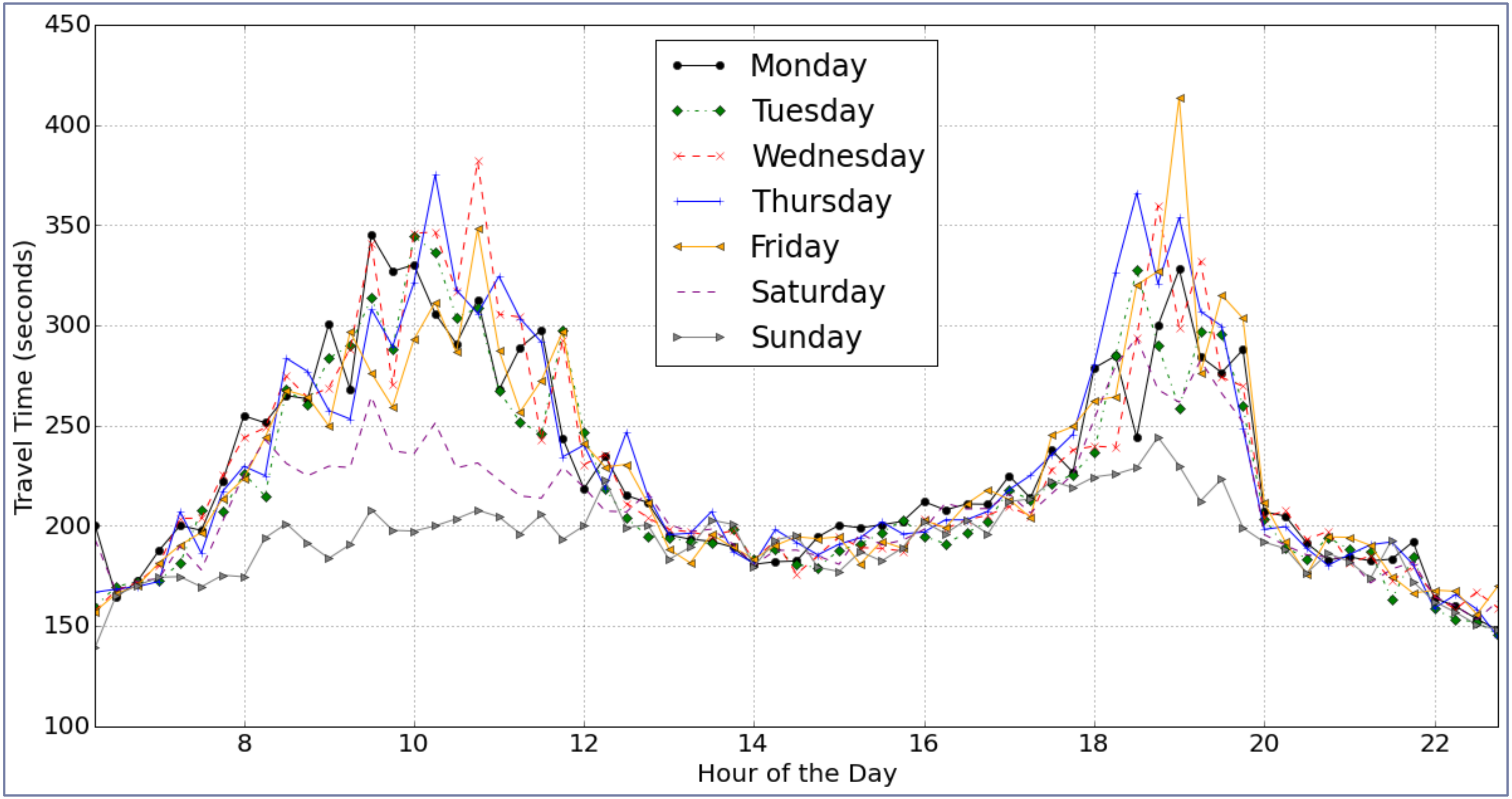
Time	Latitude	Longitude	Link ID
8:24:55	80.25276	12.98018	L1
8:25:05	80.25276	12.98013	L1
8:25:15	80.25265	12.97992	L1
8:25:25	80.25240	12.97993	L1
8:25:35	80.25233	12.97996	L2
8:25:45	80.25228	12.97998	L2

Bus exits link L2

Entry Time	Exit Time	Travel Time
8:24:55	8:25:25	0:00:30
9:10:31	9:11:16	0:00:45



Travel Time Variation on SP Road



LINK LEVEL TT PREDICTION

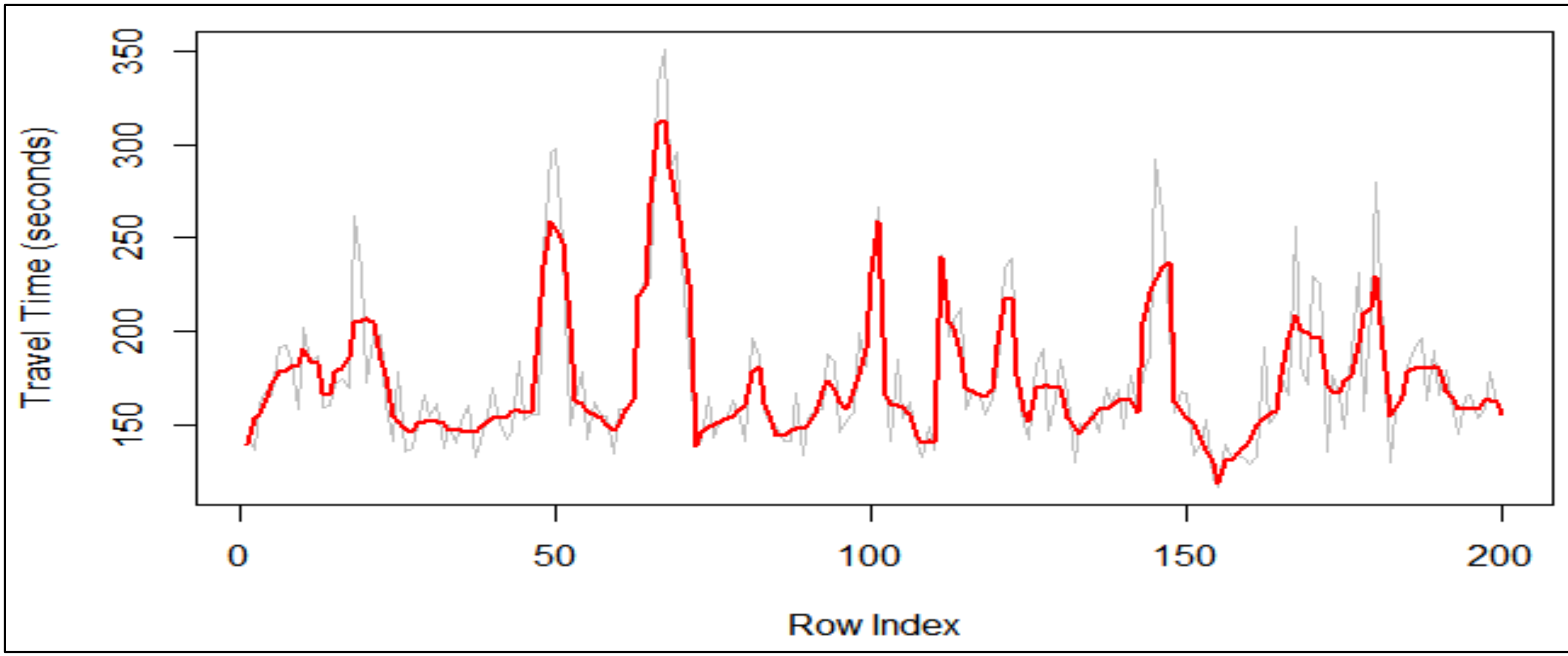
Travel Time Prediction Models

- Most widely used tools in Traffic Literature:
  - Historic Data Based Models
  - Linear Regression
  - Time Series Forecasting
  - Machine Learning
  - Kalman Filtering

- Measurement of Performance
  - $MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_{i,actual}-y_{i,predicted}|}{y_{i,actual}}$

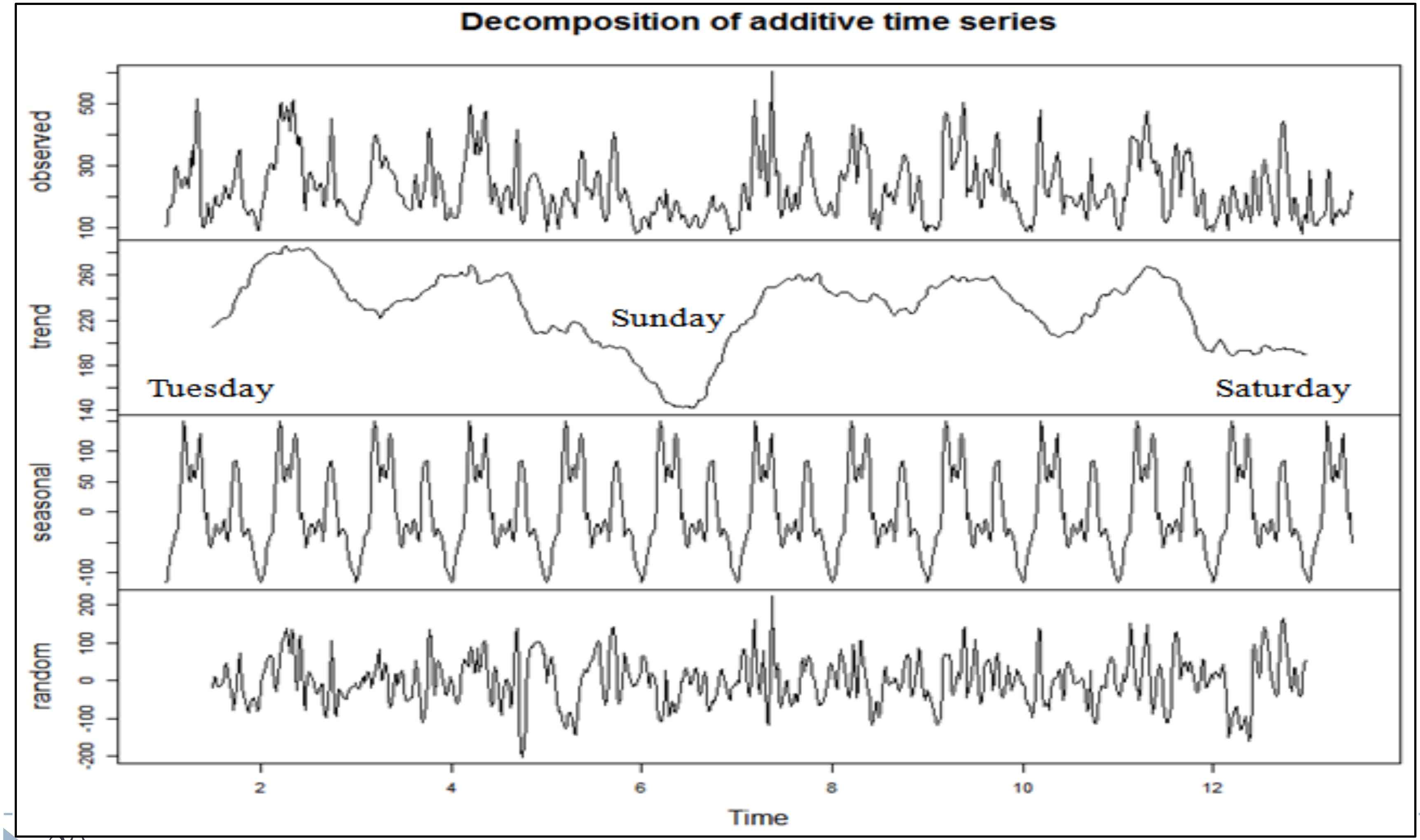
Time Series

- Non-constant sampling interval-difficult to work with (especially with respect to model fitting and future forecasting)
- Smoothen curve and extract TT at 15-min interval



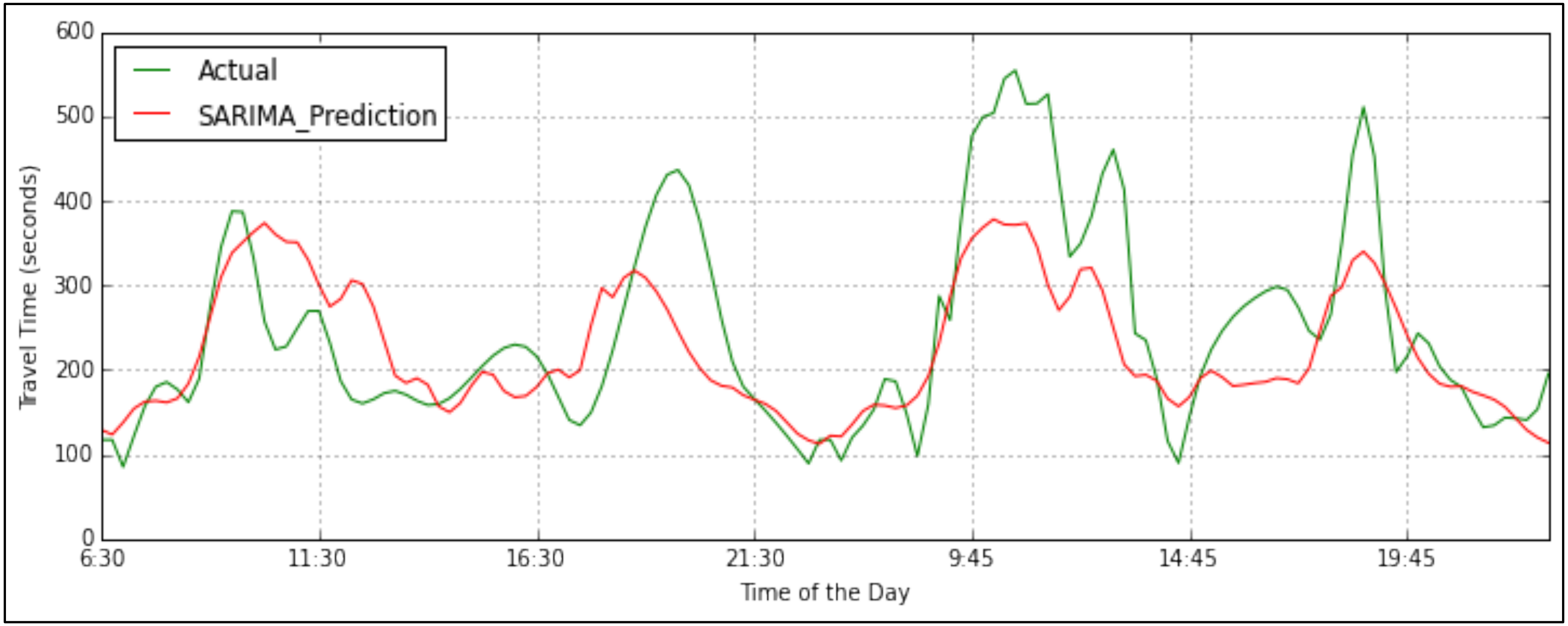
Time Series Decomposition

- Observed TT = Trend + Seasonal Component + Random Component



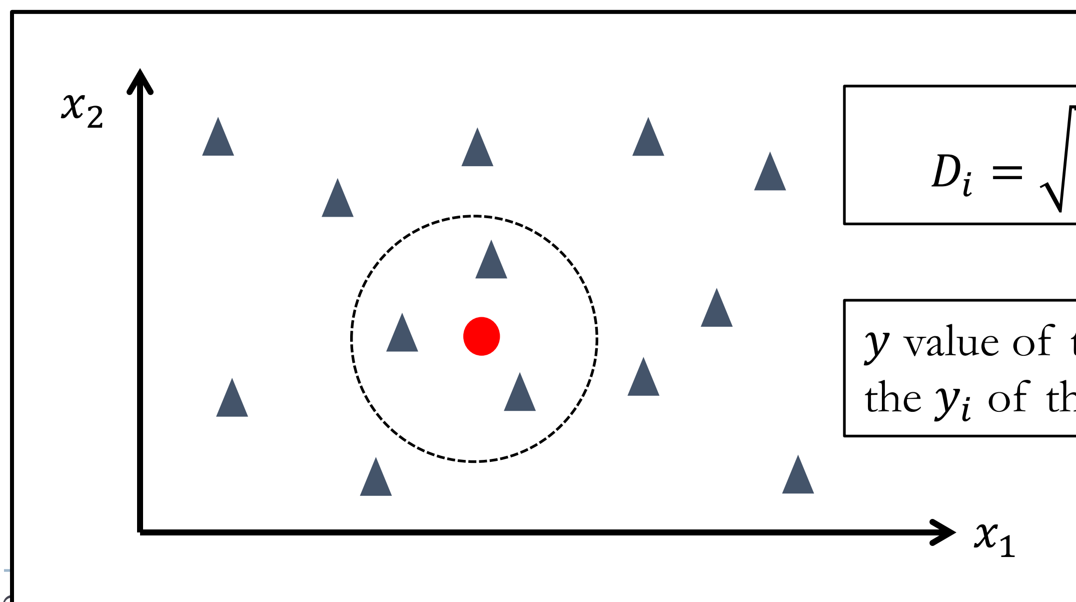
SARIMA Prediction

- Monday data for model parameter estimation and forecasting
- MAPE = 32 %



K-Nearest Neighbour

- Machine Learning Algorithm
- Target Variable: y_i
- Features: $(x_{1i}, x_{2i}, \dots, x_{mi})$
- Search for 'most similar' record in the training data set
- Euclidean Distance is the similarity measure chosen:

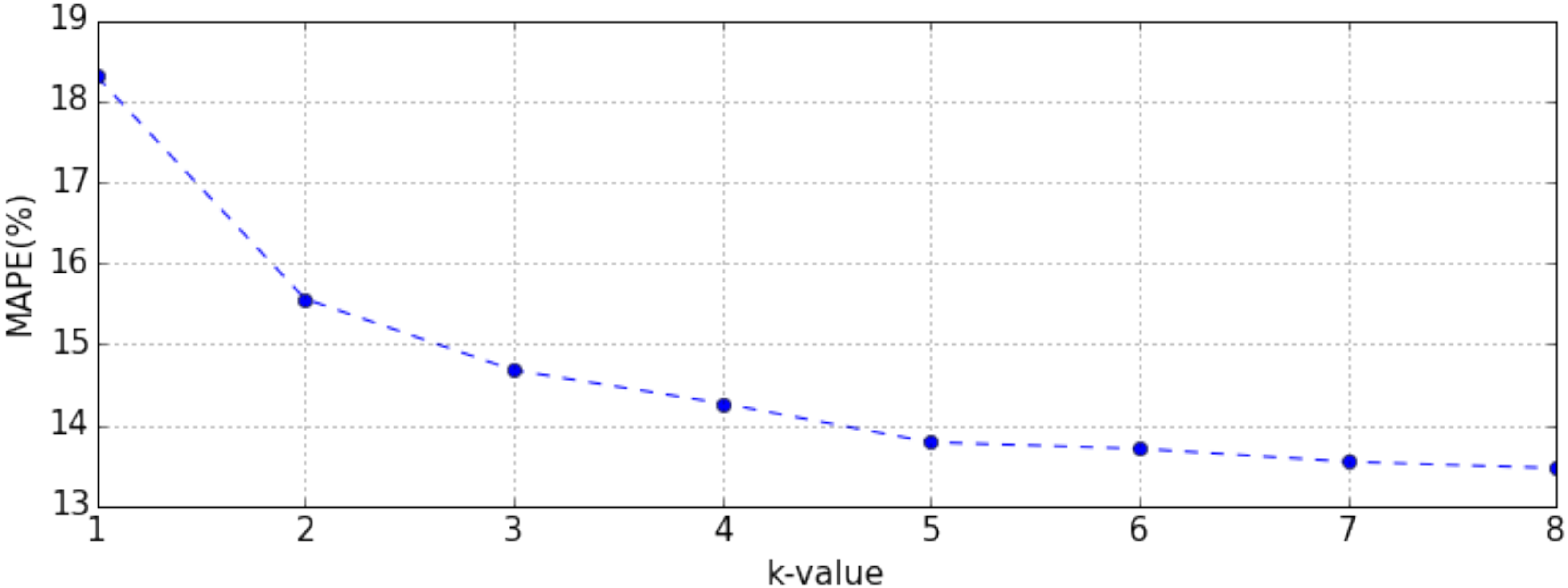


$$D_i = \sqrt{\left(x_{1i} - x_{1j}\right)^2 + \left(x_{2i} - x_{2j}\right)^2}$$

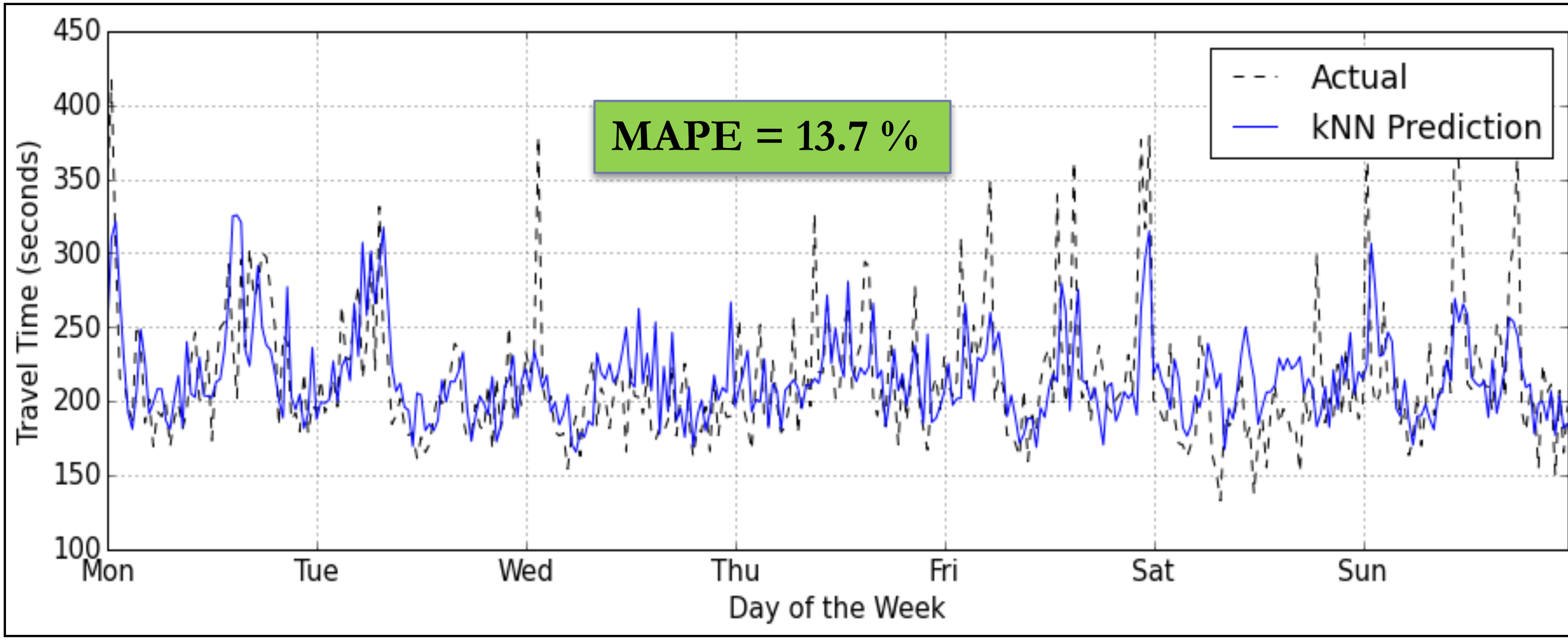
y value of the new record is the average of the y_i of the k nearest neighbors

kNN Parameters

- **Features:**
  - *TT (t-1), TT (t-2), TT (t-3), TT(yesterday), TT(last week)*
- **Sensitivity of Error with k-value:**



kNN Results



Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
MAPE(%)	14.0	14.1	12.2	14.6	13.1	12.9	15.7

Kalman Filtering

- Estimation of variables, based on observations and prior knowledge about the process

- State Equation: $x_n = A_n x_{n-1} + w_{n-1}$
- Measurement Equation: $z_n = Hx_n + v_n$

Time Update
(Prediction)

A priori estimate of travel time

$$\hat{x}_n^- = A_n \hat{x}_{n-1} + w_{n-1}$$

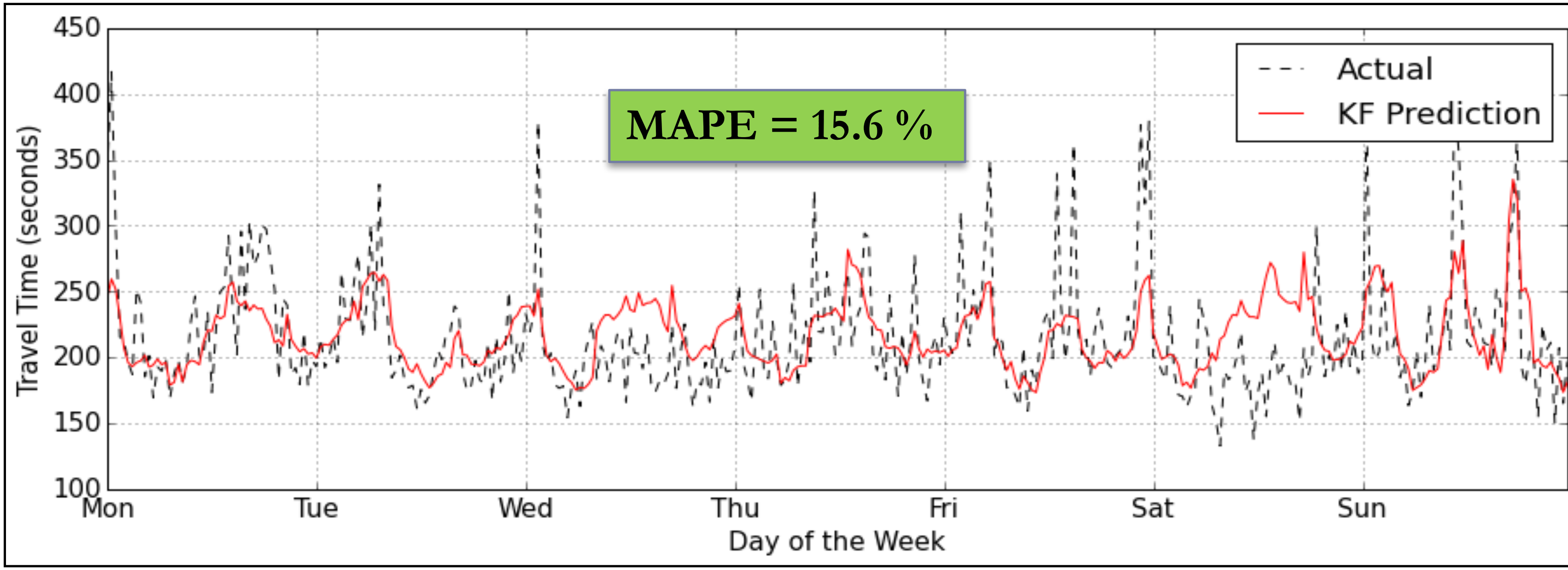
Measurement Update
(Correction)

A posteriori estimate of travel time

$$\hat{x}_n = \hat{x}_n^- + K[z_n - \hat{x}_n^-]$$

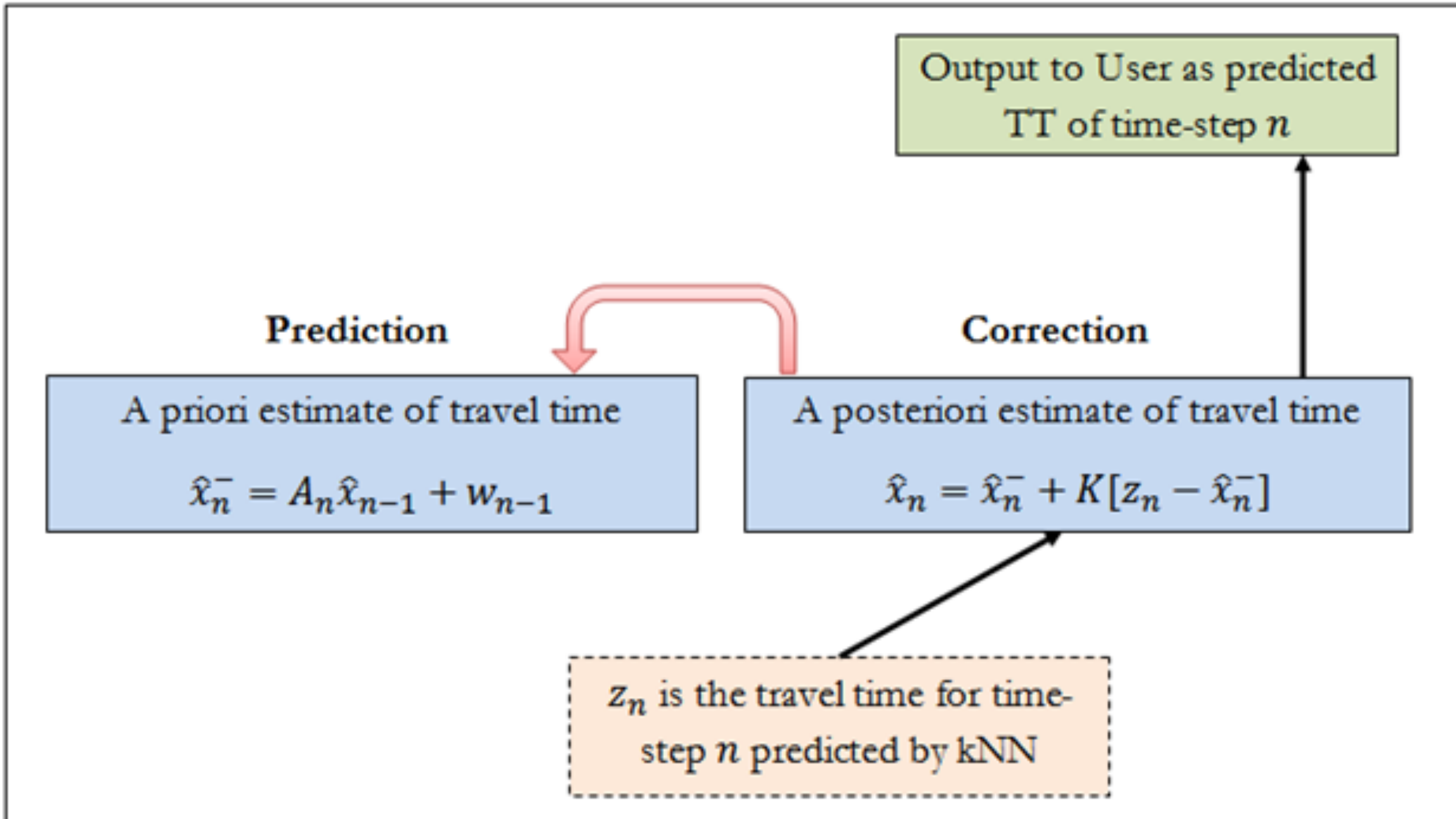
KF Results

- A_n = *Ratio of TT(t-1) to TT(t-2)*
- z_n = *Average of TT(yesterday) to TT(last week)*

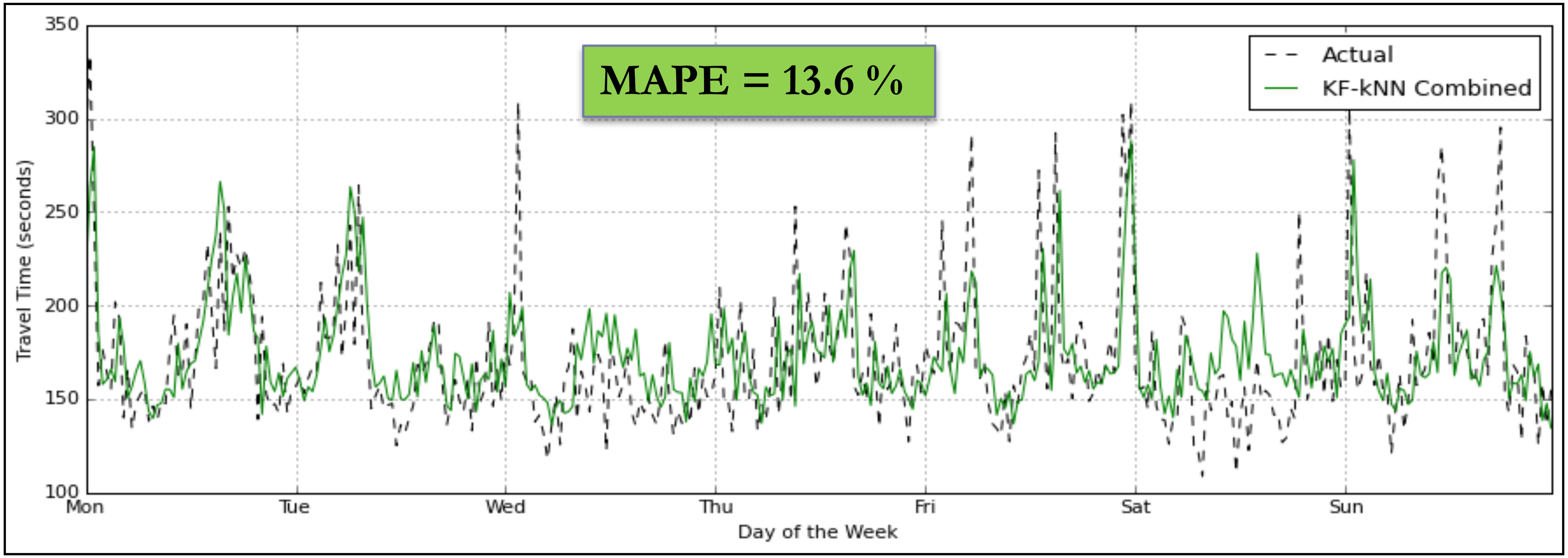


Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
MAPE(%)	14.5	14.7	12.7	20.4	16.2	13.0	18.8

Fusion KF-KNN Framework



Fusion KF-KNN Results



Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
MAPE(%)	13.3	13.7	12.6	14.7	13.7	12.2	15.6

Link Level TTP Conclusions

- kNN alone and the fusion model had lower MAPEs than the KF alone model, which was significant at the 5% level.
- Performance of the fusion model and the kNN were at par, with not much difference in MAPE (No significant difference at the 5% level)
- One could use the k-nearest neighbour method for the link level travel time prediction rather than the fusion.

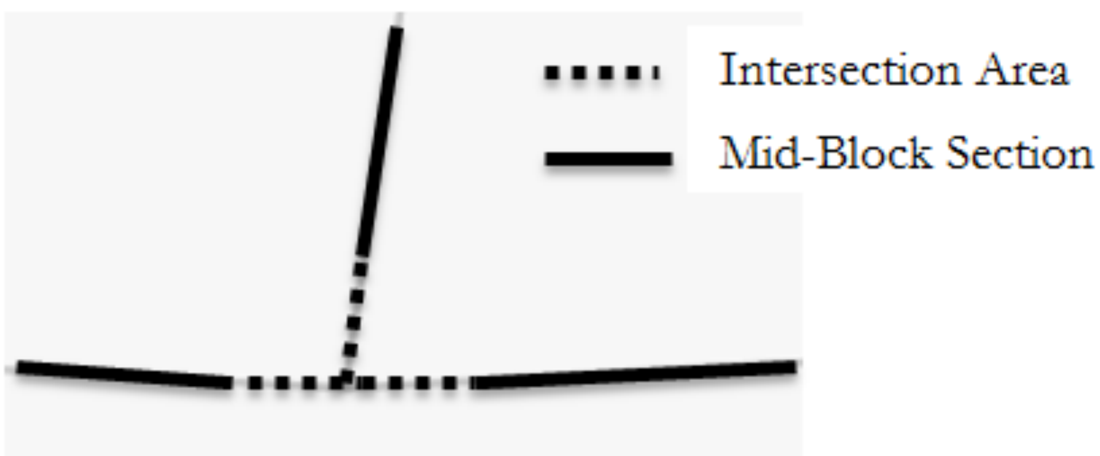
TTP ON INTERSECTION LINKS

Intersection Links

- The MAPE was within 20% for most of the links.
- Links which had > 20% MAPE were identified to be the ones that were connected to major intersections.
- During peak hours, the congestion at these intersections is significantly high resulting in long queues and large delays.
- Such full-links would be split into a mid-link and an intersection link and analyzed separately

Sources of Delay

- Mid-blocks: Lower Variability
- Intersections: Higher Variability



Length of Intersection Link

- Should be based on the usual length of queues formed at the intersection.
- In the absence of prior knowledge about network, the proposed method that uses GPS data – trajectory and speed
- Determine at what distance from the intersection the speed of the bus becomes less than 10kmph for the first time
- Indicative of when the bus slows down and joins back of queue
- Average the above distance taken during the trips made during the peak traffic hours
- Only a static value but one could use a dynamic value changing it based on the time of the day.

Special Cases

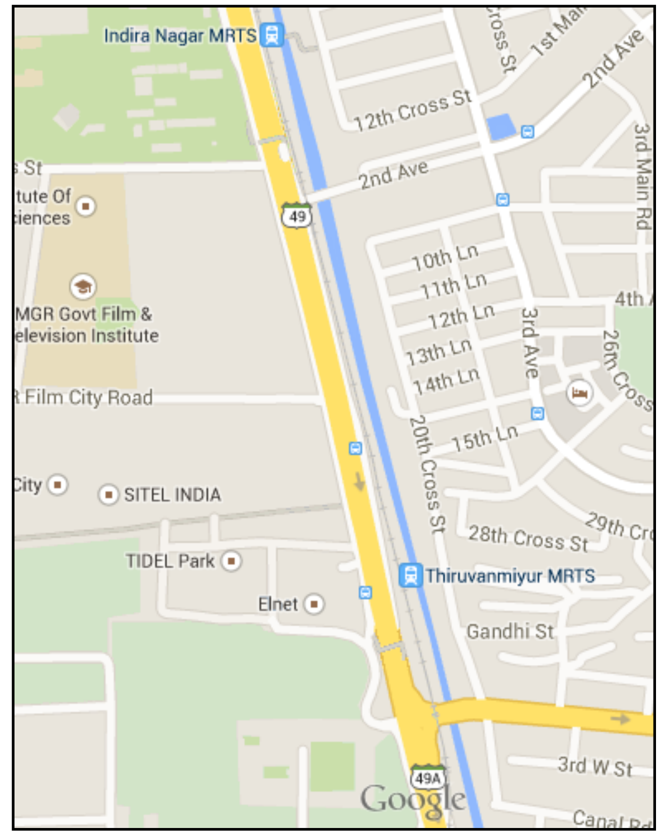
- Split is not made if:
  - **Case 1:** The "queue distance" extents to a large portion of the link.
  - **Case 2:** The "queue distance" is relatively small compared to the overall link size.
  - **Case 3:** The link is too small for it to be split into mid link and intersection area.

- Criteria for split: The queue distance must be:
  - greater than 200m,
  - less than 600m, and
  - within the range of 10 to 50% of link length

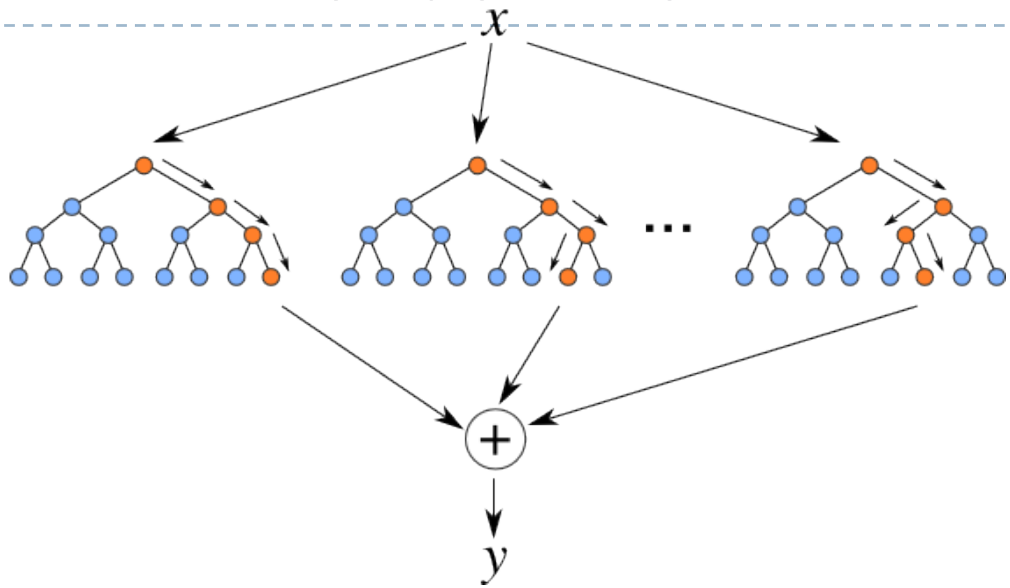
Prediction at Intersection Link

- The first method requires spatial data from probe vehicles alone and uses the random forest technique.

- The second method proposed requires data from loop detectors to predict queue length and from that delay on a cycle-by-cycle basis.



Random Forests



- Ensemble Learning based on a pre-specified number of decision trees.
- Each decision tree is trained separately and the prediction from each tree, is taken into consideration while making the final prediction.
- For each decision tree
  - Bagging – creates a new training subset by random sampling with replacement from the original set of *n* training examples.
  - Random subset of features

RF Parameters and Features

- **Features:**
  - *TT (t-1), TT (t-2), TT (t-3)*
  - *TT(yesterday)*
  - *TT(last week)*
  - *hour*
  - *day*
- **Parameters:**
  - *No. of Decision Trees = 1000*
  - *No. of features for each decision tree: 3 out of 7*
  - *Training set size for each decision tree: 0.632 of entire training set*
- Final output TT is the median of the predictions from individual decision trees.

Smoothing

- Prior to applying the RF predictor, a data smoothing operation was carried out.
- Spline:
  - kth order spline is a piecewise polynomial function of degree k

- Minimize the spline objective function for cubic spline

$$\mathcal{L}(m, \lambda) \equiv \frac{1}{n} \sum_{i=1}^{n} (y_i - m(x_i))^2 + \lambda \int dx (m''(x))^2$$

Smooth Spline

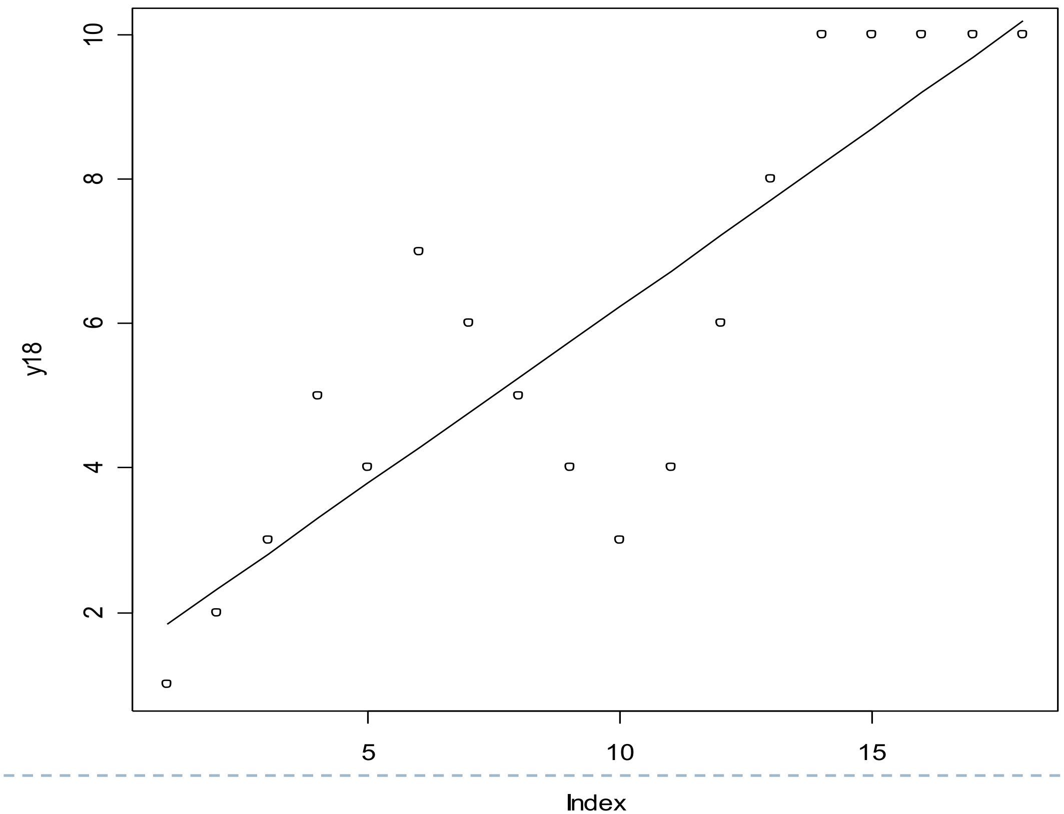
$$\mathcal{L}(m, \lambda) \equiv \frac{1}{n} \sum_{i=1}^{n} (y_i - m(x_i))^2 + \lambda \int dx (m''(x))^2$$

- 1st Term: Mean Squared Error (MSE)
- 2nd Term: m'' is the second derivative -> curvature at different values of x

- Minimize both least squared error and the average curvature of the curve
- For same MSE, the fit with the smaller average curvature is preferred

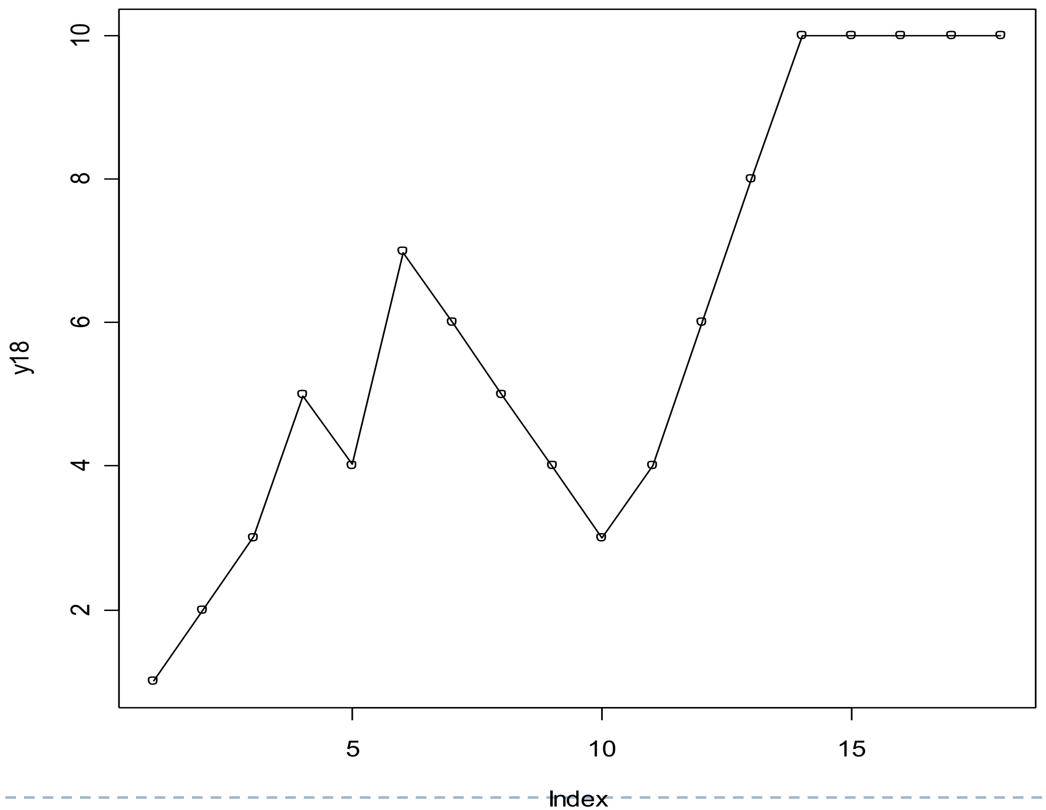
$$\mathcal{L}(m,\lambda) \equiv \frac{1}{n}\sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda \int dx (m''(x))^2$$

- If λ is set to a large value, any curvature is penalized. So linear (which has zero curvature) would be the best fit.

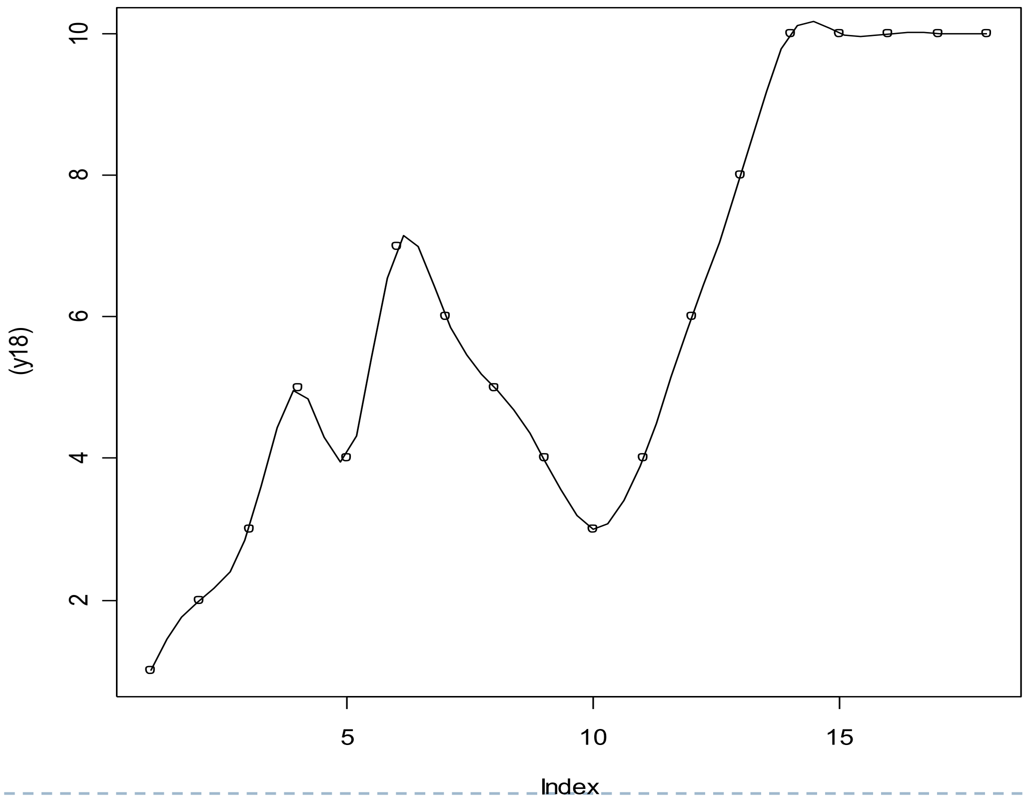


$$\mathcal{L}(m, \lambda) \equiv \frac{1}{n}\sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda \int dx(m''(x))^2$$

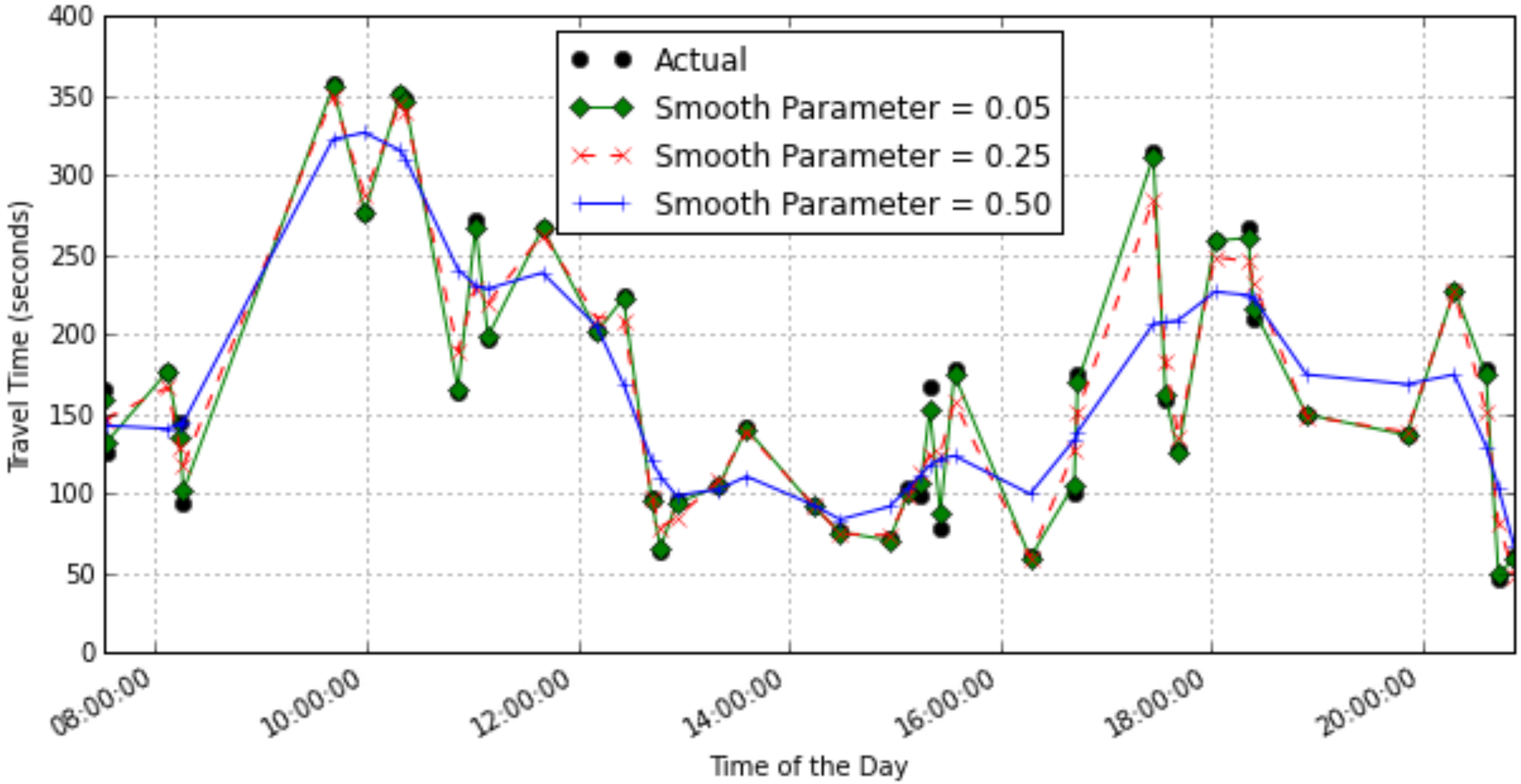
- If λ is set to zero, it means we are fine with any curvature, so 1st term is minimized.



- Larger λ means not representative of the data.
- Smaller λ means over-fitting
- Ideally λ should be in between 0 and 1

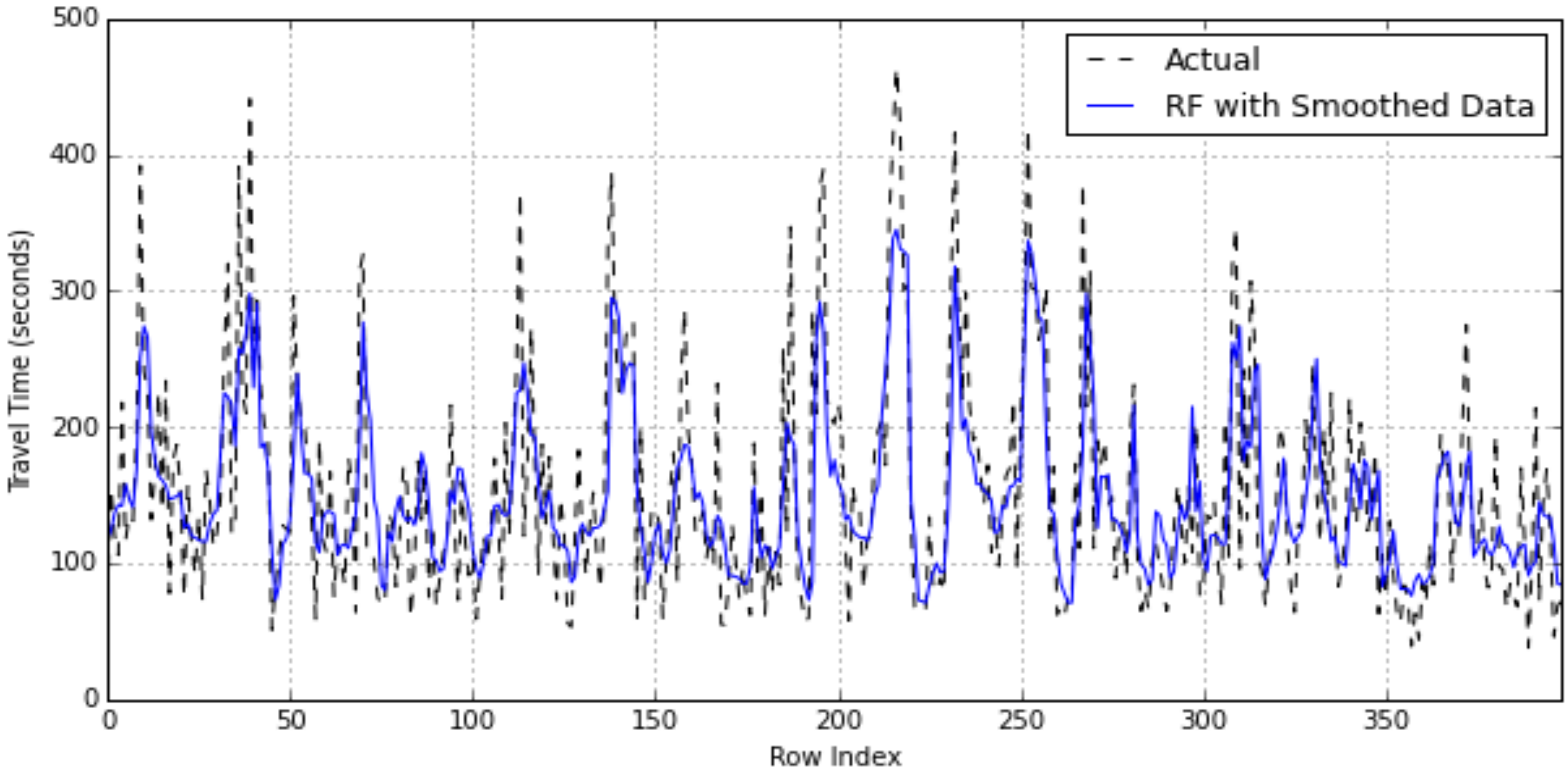


Smoothing on Sample TT Data

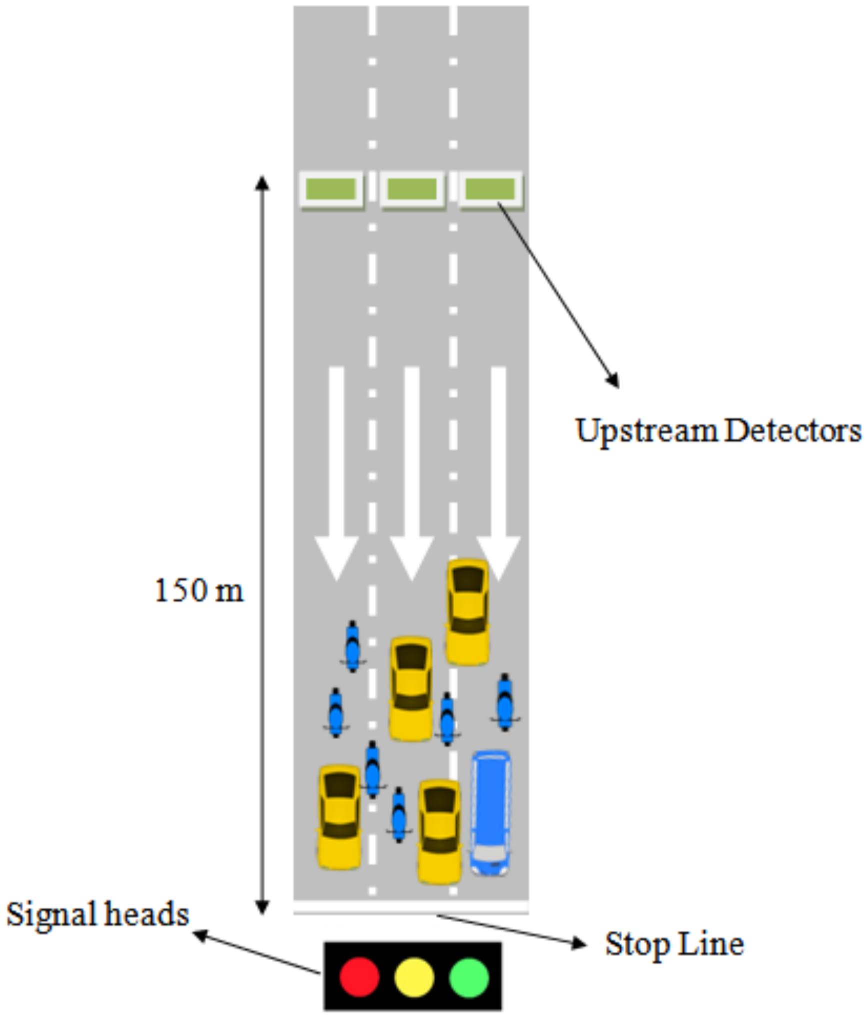


RF Predictor Results

- MAPE = 29%
- More accurate than kNN on Actual data (56%) and kNN on smoothed data (34%)

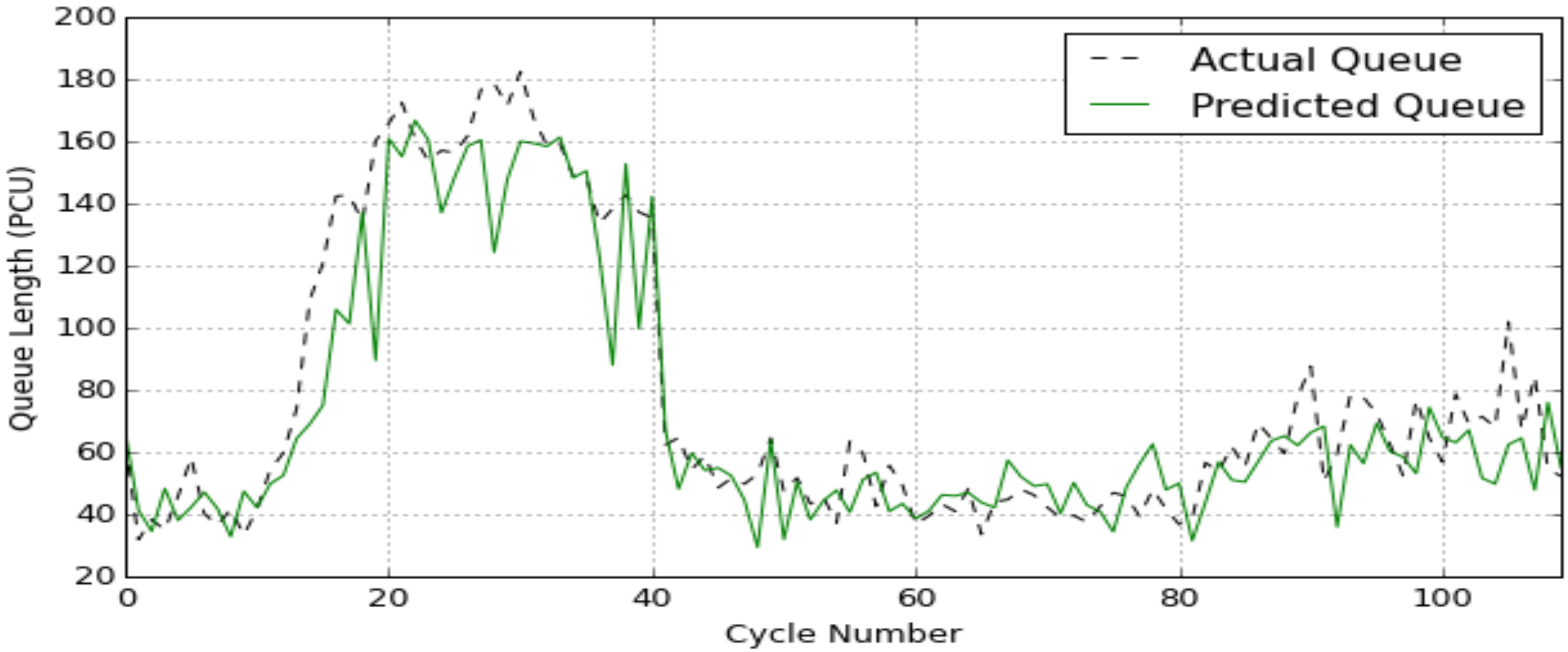


Prediction using location based data



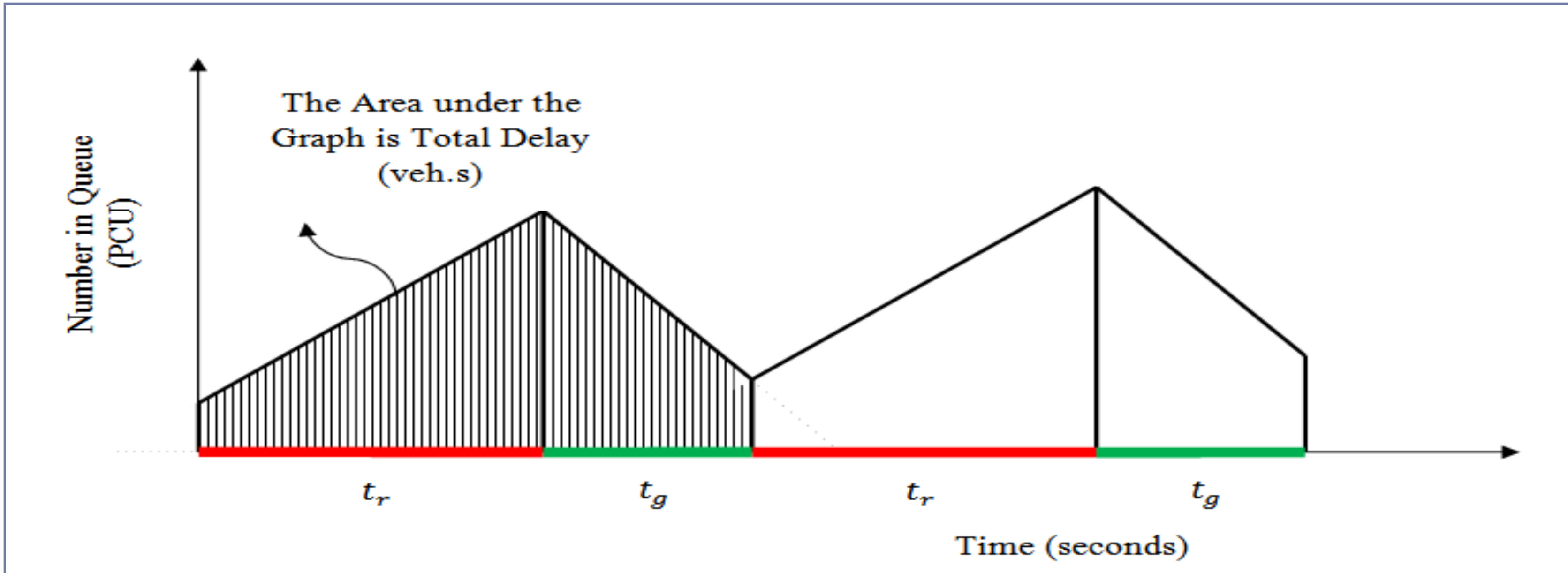
- Data from Simulation in VISSIM (Indian Conditions)
- Thiruvanmiyur Intersection
- kNN trained with upstream detectors occupancy to predict the queue length as the target variable.

Queue Prediction

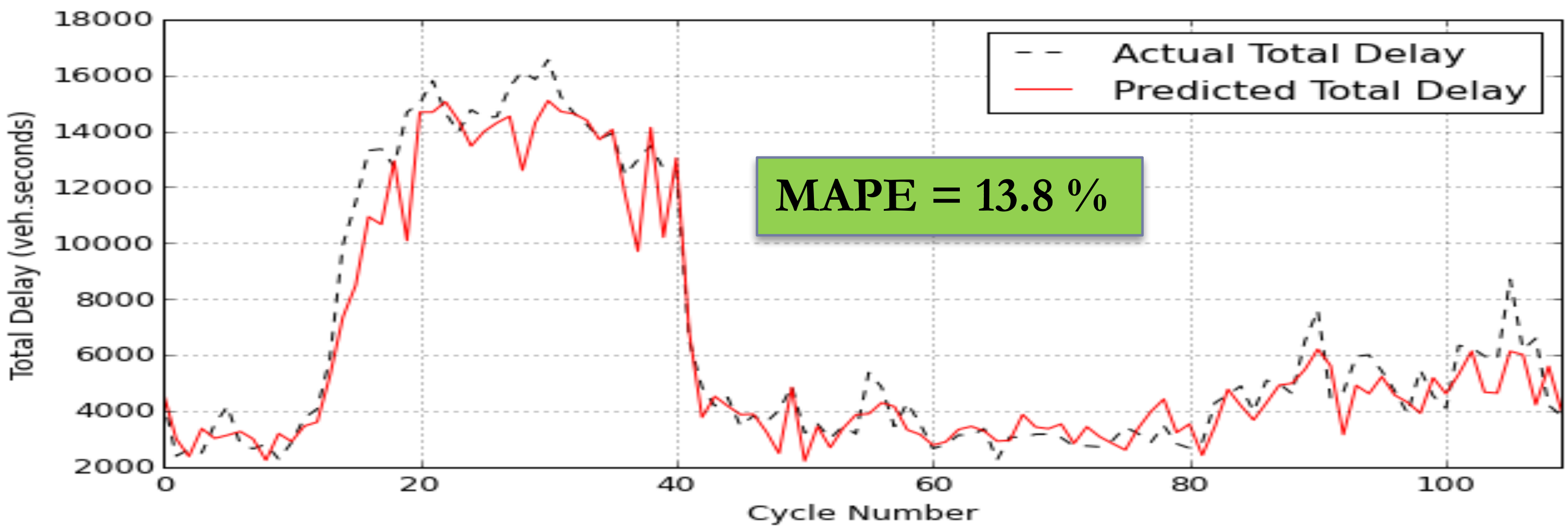


- $MAE = \frac{1}{n}\sum_{i=1}^{n} |q_{i,actual} - q_{i,estimated}|$
- MAE = 13.4 (Reasonable accuracy because the maximum number of vehicles that occupy this stretch during fully congested periods go up to 180 vehicles [PCU].)

Total Delay

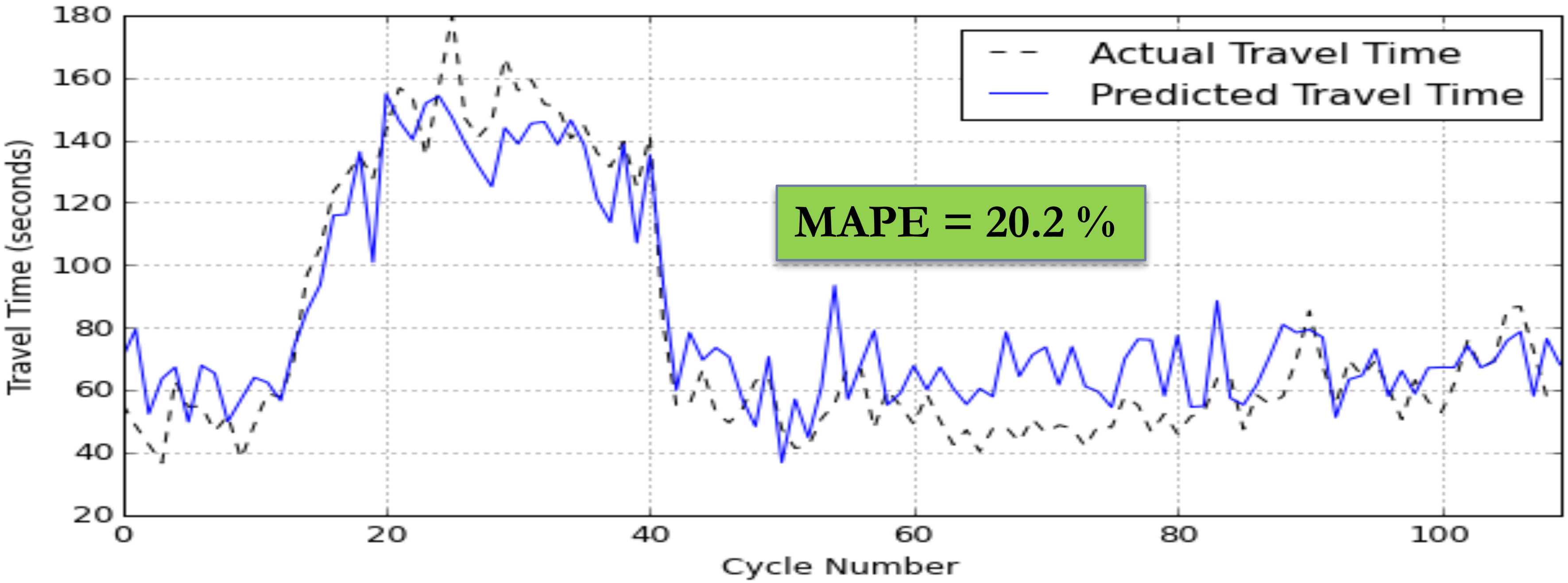


The Area under the Graph is Total Delay (veh.s)
Number in Queue (PCU)
t_r
t_g
t_r
t_g
Time (seconds)



Actual Total Delay
Predicted Total Delay
MAPE = 13.8 %
Total Delay (veh.seconds)
Cycle Number
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
20
40
60
80
100

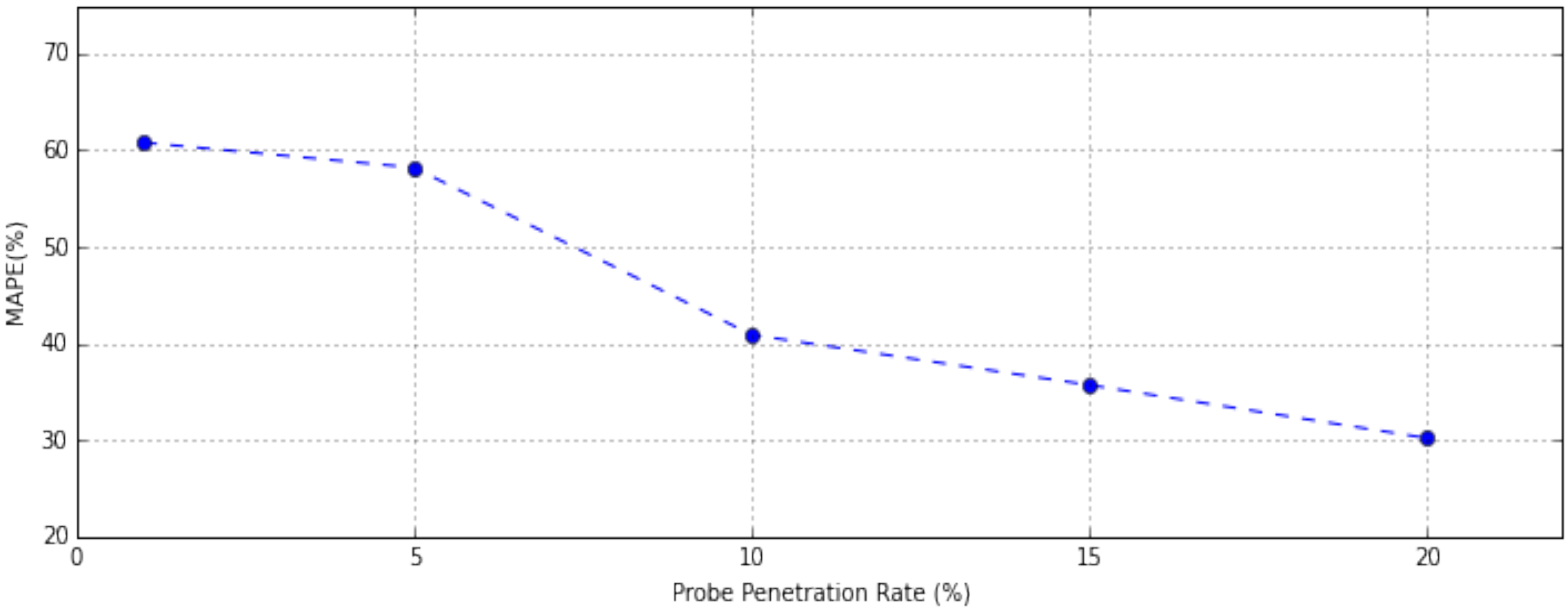
Travel Time



- Predicted Delay + Free Flow TT on the link
- Compared to the actual average travel time for that cycle

Comparison to RF Predictor

- RF predictor tried with different values of probe penetration rate using the same VISSIM simulation
- MAPE reduces with increase in probe penetration rate



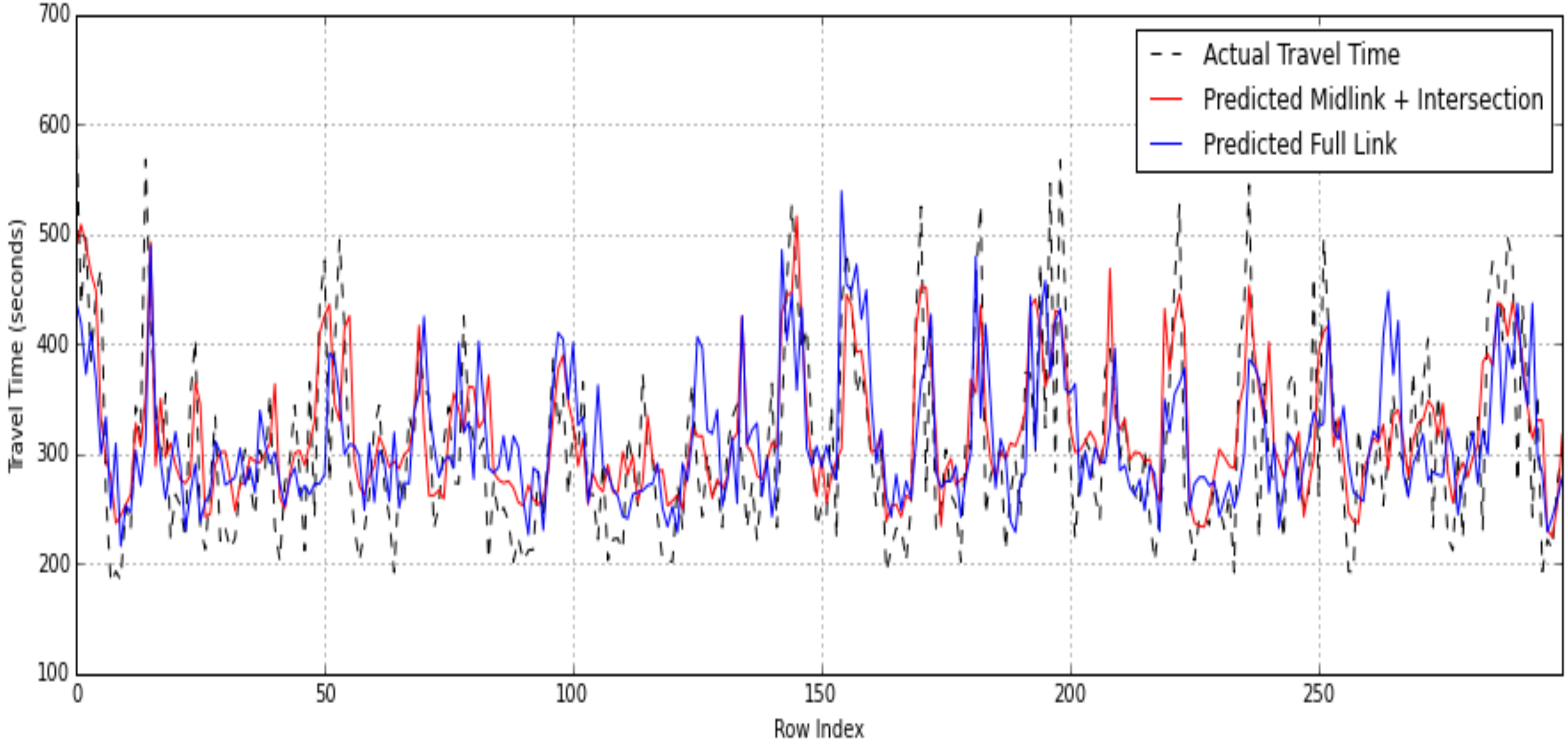
Link Prediction Schemes

- **Scheme 1:** The entire link (one intersection to another) was considered for prediction [kNN alone method]

- **Scheme 2:** The link was split into a mid-link and intersection link and both were analyzed separately. The total link travel time was taken as the sum of the predictions on the mid link (with kNN algorithm) and the intersection link (with the Random Forest predictor) [kNN-RF method]

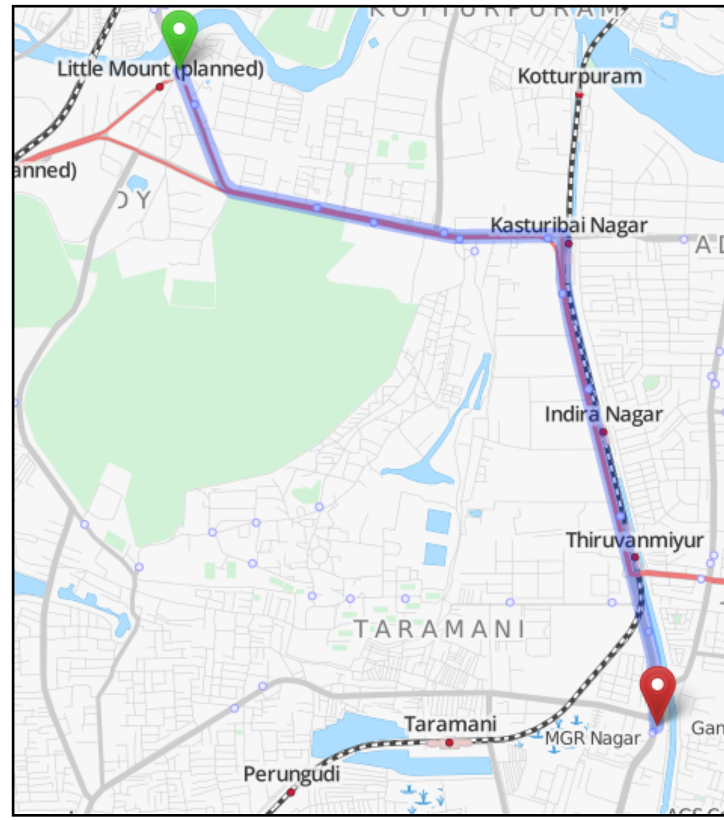
Link Prediction Schemes

- MAPE for Scheme 1: **21%**
- MAPE for Scheme 2: **17%**



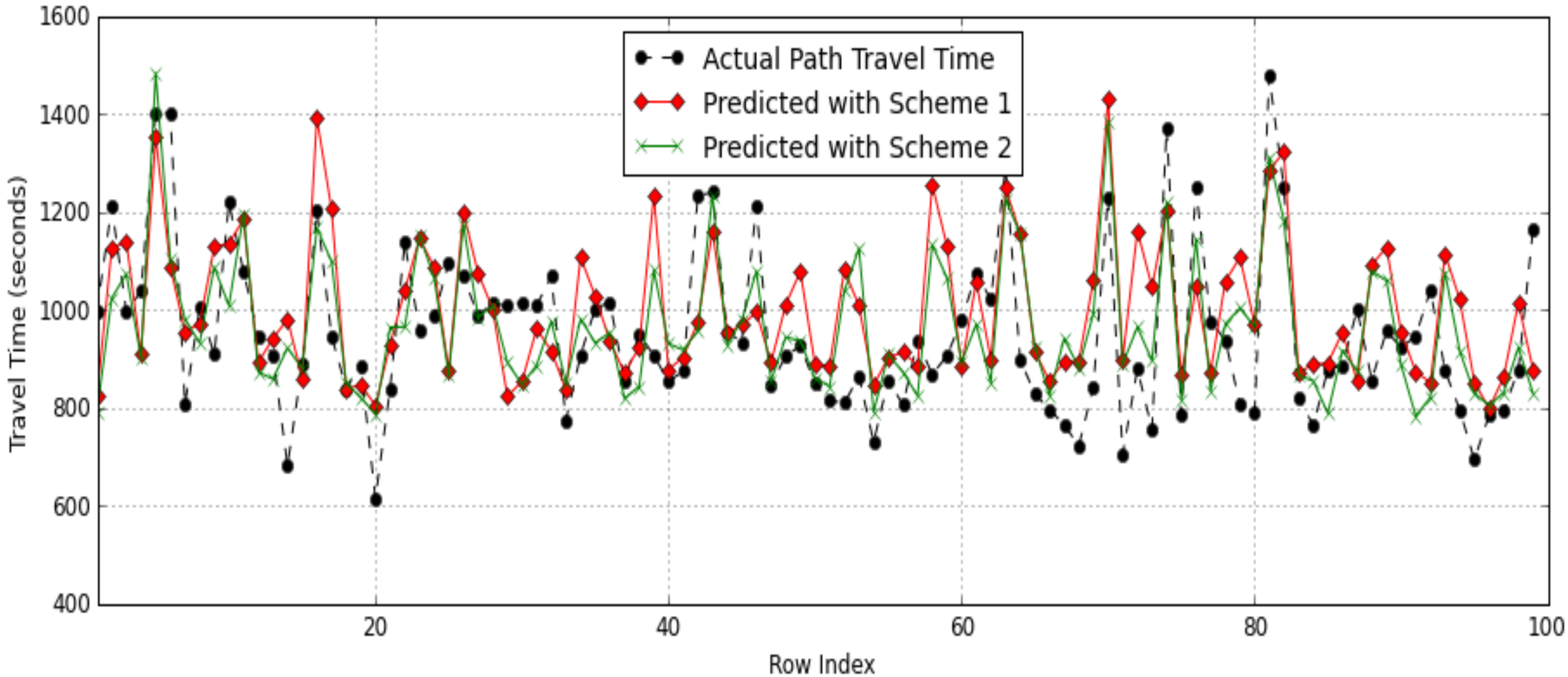
Path Level Prediction

- From Little Mount to SRP Tools
- 6 km long and made up of the following 5 links: L1-L3-L5-L9-L11
- Average travel time is 17.5 minutes
- **Scheme 1:** All links considered as full links
- **Scheme 2:** Links L3 and L9 were split into mid-link and intersection, the remaining links (L1, L5 and L11) were considered as full links.



Path Level Prediction

- Instantaneous Prediction
- MAE for Scheme 1: 141 seconds
- MAE for Scheme 2: 111 seconds



Conclusions and Summary

- TTP for arterial roads in an urban network using GPS data.
- kNN algorithm for prediction at link level
- RF predictor for intersection link TT prediction using spatial data (Smoothing the data using smoothspline reduced error significantly)
- Alternate method using location based sensors -> queue length -> delay -> travel time
- Model Validation was done at Path level – better accuracy when mid link and intersection were considered and analyzed separately

Scope for Further Research

- Buses have been used as probe vehicles -> Need to identify and correct for bus stop delays
- Work with GPS data other than from buses to be able to cover even minor roads in the network
- A higher frequency GPS data would always be better while inferring the exact path of the vehicle and extracting the travel times on individual links
- Links were assumed to be straight lines, better accuracy in Map-matching could be achieved using actual shape files of links
- Field testing with real-world loop detector data for intersection delay prediction

References

- Hoffmann, G., & Janko, J. (1990). Travel times as a basic part of the LISB guidance strategy. In *Road Traffic Control, 1990., Third International Conference,* (pp. 6-10). IET.
- Nikovski, D., Nishiuma, N., Goto, Y., & Kumazawa, H. (2005). Univariate short-term prediction of road travel times. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE* (pp. 1074-1079). IEEE.
- Kwon, J., Coifman, B., & Bickel, P. (2000). Day-to-day travel-time trends and travel-time prediction from loop-detector data. *Transportation Research Record: Journal of the Transportation Research Board*, *1717*(1), 120-129.
- Khoei, A. M., Bhaskar, A., & Chung, E. (2013). Travel time prediction on signalised urban arterials by applying SARIMA modelling on Bluetooth data. *36th Australasian Transport Research Forum (ATRF) 2013*.
- Guin, A. (2006). Travel time prediction using a seasonal autoregressive integrated moving average time series model. *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE* (pp. 493-498). IEEE.

References

- Chien, S. and Kuchipudi, C. (2003). Dynamic Travel Time Prediction with Real-Time and Historic Data. *Journal of Transportation Engineering*, 129(6), 608–616.
- Padmanaban, R. P. S., Vanajakshi, L., & Subramanian, S. C. (2009). Estimation of bus travel time incorporating dwell time for APTS applications. *Intelligent Vehicles Symposium, 2009 IEEE* (pp. 955-959). IEEE.
- Lee, Y. (2009, July). Freeway travel time forecast using artificial neural networks with cluster method. *Information Fusion, 2009. FUSION'09. 12th International Conference,* (pp. 1331-1338). IEEE.
- Wu, C. H., Ho, J. M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *Intelligent Transportation Systems, IEEE Transactions,* 5(4), 276-281.
- Gather, U., & Fried, R. (2004). Methods and algorithms for robust filtering. *COMPSTAT 2004—Proceedings in Computational Statistics* (pp. 159-170). Physica-Verlag HD.

THANK YOU