Trans-Portal
Context sensitive traffic modeling from smart-phone data

Samanway Ghatak

Trans-Portal
Context sensitive traffic modeling from smart-phone data

A thesis in partial fulfilment for the degree of

Master of Technology

in

Computer Science and Engineering

Submitted By

Samanway Ghatak

16/CS/4115

under the supervision of

Dr. Subrata Nandi

Associate Professor

NIT Durgapur

and

Dr. Sujoy Saha

Assistant Professor

NIT Durgapur

Department of Computer Science and Engineering

National Institute of Technology, Durgapur

May 2018

Dedicated to all you who mentored me to a better person.

Declaration

I certify that

1. the work contained in this thesis is original and has been done by me under the guidance of my supervisor.

2. the work has not been submitted to any other Institute for any degree or diploma.

3. I have followed the guidelines provided by the Institute in preparing the thesis.

4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

5. whenever I have used materials (data, theoretical analysis, figures and text) from other sources, I have given due credit to them by citing them it the text of the thesis and giving their details in the references.

Samanway Ghatak

Certificate Of Recommendation

This is to certify that the thesis entitled "**Trans-Portal, Context sensitive traffic modeling from smart-phone data**", submitted by **Samanway Ghatak** for the partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Engineering with specialization in Computer Science and Engineering**, is a bonafide research work under the guidance of **Prof. Subrata Nandi** and **Prof. Sujoy Saha**. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma. In our opinion, this thesis is of the standard required for the partial fulfillment of the requirements for the award of the degree of **Master of Technology**.

(Counter Signed by)
Prof. Subrata Nandi
Assoc. Professor, Dept. of CSE
National Institute of Technology
Durgapur-713209, INDIA

(Counter Signed by)
Prof. Sujoy Saha
Asst. Professor, Dept. of CSE
National Institute of Technology
Durgapur-713209, INDIA

(Counter Signed by)
Prof. Goutam Sanyal
Prof. & Head, Dept. of CSE
National Institute of Technology
Durgapur-713209, INDIA

Certificate of Approval

The foregoing thesis is hereby approved as a creditable study of technological subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite with degree for which it has been submitted. It is to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

BOARD OF THESIS EXAMINERS

1. _____ 2. _____

3. _____ 4. _____

5. _____ 6. _____

Acknowledgements

First and foremost, I would like to express my deep gratitude towards my research supervisors **Dr. Subrata Nandi** and **Dr. Sujoy Saha**. Without their assistance and dedicated involvement in every step, this thesis would have never been accomplished. I would like to thank them from my heart for the constant support, patience, motivation and understandings over the tenure of this research.

I wish to acknowledge **Dr. Goutam Sanyal**, HOD of CSE, for providing such a supporting environment where I was offered such opportunities to learn and apply my knowledge in practical fields. I would also like to take the opportunity to show gratitude and respect to other faculty members of our department for their encouragement.

I thank my PhD Scholar Mrs. Ratna Mandal for her continued support. I would love to express my gratitude towards all other members of the research group - Partha Sarathi Paul, Munsi Yusuf Alam, Prithviraj Pramanik, Kingshuk De, Bishakh Ghosh, Naman Mehta who were available for discussion, help and suggestion when needed. Thanking would be too small an acknowledgement for my friends Gangotry, Abhay, Om, Prashant, Ankit, Ravikant and all those who were always by my side whether I needed them for academic purpose and as a companion.

Last but not the least, I would like to thank my family and my friends for their unconditional support.

Samanway Ghatak

Trans-Portal
Context sensitive traffic modeling from smart-phone data

Contents

Abstract

In the age of smart-phones and smarter applications planning of trips has been the motivation behind many innovative research projects as well as numerous successful business ideas. Nowadays there are applications available in our cellphones that make life easy by calling nearby cabs, ordering food from restaurants, keeping track of deliveries, traffic monitoring for law enforcement authorities etcetera. There are many state of the art system and services that provide real time as well as historical data that helps users to plan their trips more intelligently. In situations where the real-time data is not available, like planning a trip across the city, planning business strategies based for courier and food delivery websites; the historical data based predictions become useful. Despite the fact that APIs like Google maps provide nice tools for planning and visualizing the traffic condition in a very detailed level, there are conditions when these systems are observed to be not sufficient. In suburban and rural areas of developing nations like India, where the limited use of cellphones and technology limits the scope of such state of the art systems. Apart from that, in sudden exceptional scenarios like bad weather, festivals, political or cultural events the data from these systems are not always adequate to planning trips ahead of time. In this thesis we suggest an approach to come up with a traffic modeling strategy that can model road segment behavior with low volume data collected from mobile devices, which can be useful in extending the coverage of the map services as well as can be useful in modeling special scenarios only using low volume data. GPS trails and a log of surrounding WiFi access points are collected through

crowdsourcing using an Android application. The WiFi signals, used as a feature that indicates human activity around an area, helps to model areas with similar behavior, enabling data from different geographical locations to be modeled together helping to model the different types of areas and their typical traffic behavior. Here we built a system that captures raw data using mobile devices, accumulates the data in cloud storage, from the collected data identifies different routes in case of public buses, identifies landmarks using different landmark identification algorithms, extract features out of the GPS and WiFi sensor data and finally build an unsupervised model based on the vehicle movement in a segment and availability of WiFi access points around the road segment. This built model can then be used to mark segments as Busy, Medium Busy and Normal. The accuracy of this approach is then tested using annotated data collected by volunteers from Durgapur, a suburban city of Eastern India. While testing the model on over 800 Kilometers of collected traffic data, It was observed to estimate the segments with 72% accuracy when compared against annotation provided by human volunteers. These informations can be used further for vehicular communication, helping vehicles approaching certain road segments to avoid those based on information on their recent status from other vehicles. Using these model, segments that stay busy throughout the day, or gets busy in specific times can be identified helping in better planning of trips. Comparison of results from multi-mode transport can also reveal areas where public transports behaves faulty due to various reasons like profit maximization; helping the authorities to plan enforcement activities better.

List of Figures

List of Tables

Chapter 1

Introduction

Necessity is the mother of invention. — Plato.

In the modern day of technology, due to the availability of so many resources around us, we hardly do anything totally unplanned. In such planned lifestyle, travel planning is a vary important part of our daily life. An average person spends significant time on the road everyday for their daily commute. With the help of the available technologies, our travel planning has become remarkably easy nowadays. Specially in case of intra-city travels or short distance trip, we frequently make use pf various map services available to us like Google Maps, OpenStreetMaps, WikiMapia etc. When it comes to real time trip planning, these services, specially Google Map has become extremely popular due to its crowd sourced nature and high availability, precision and accuracy. Using Google Map api one can easily get the traffic condition, expected travel time in different modes both in real time and general scenario. These services use both the real time data from different vehicles, and historical data available to visualize the traffic movement patterns. These services have become so useful nowadays that many technical organizations have successfully developed full businesses out of the traffic prediction. Companies like Ola Cabs, Uber, Swiggy, FoodPanda use such map services to provide quality cab service, food delivery etcetera. On the other hand data from the vehicles that are used by these companies feed live data back to the map services that is used in real time analysis of traffic.

1.1 Motivation

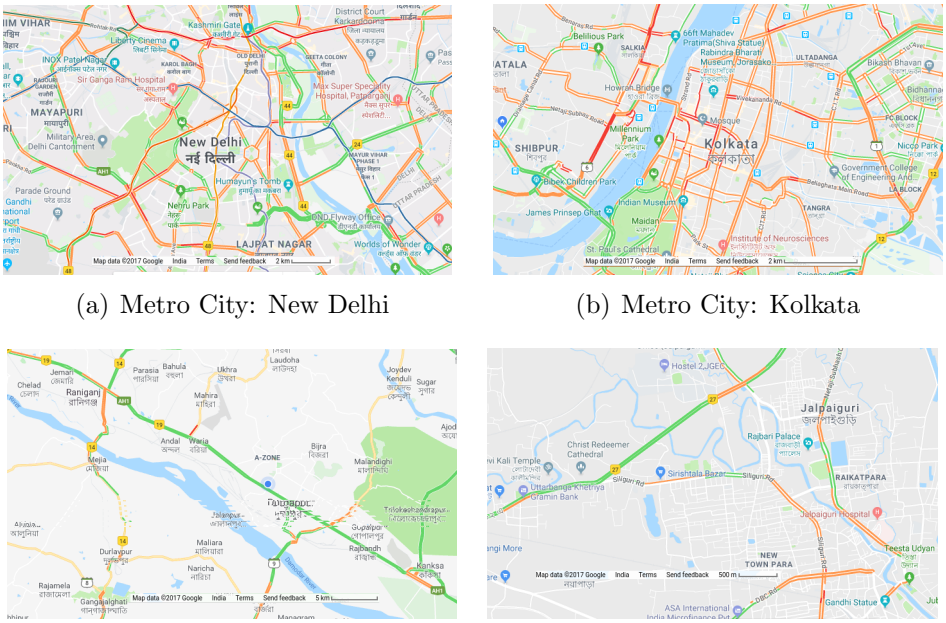
In this section we discuss the motivations behind this thesis. Enrichment of map services is required in different aspects and applications in modern days. The main motivation and possibilities behind this research can be described in two parts. Firstly, increasing the coverage of map services with traffic data and second, making better estimations for some special scenarios. Both these topics are discussed in following subsections.

1.1.1 Coverage

Though these map services can be seen being highly useful and productive, there are some limitations that can be observed. In Figure 1.1 the typical traffic condition prediction of Google Map is displayed here. The four cities out of which New Delhi (1.1(a)) and Kolkata (1.1(b)) being metro cities and the other two, Durgapur(1.1(c)) and Jalpaiguri(1.1(d)) being two suburban cities of India. It can be easily observed that the coverage of google map is significantly less in suburban area due to the less amount of data available there. This scarcity of data makes prior planning of trips difficult. As a result, the services offered that were discussed earlier are limited in these cities compared to the other two suburban cities. So presence of a traffic modeling strategy that depends on less amount of data to make the predictions can be very useful in such areas both for the local people and organizations to start operating in new location and provide service without degradation in performance.

1.1.2 Special Scenarios

Apart from the fact that the limited coverage of some areas can be improved, there are other situations that can be addressed here. While conducting a survey on the college students, it was observed that almost all the people use map services prior to their planned trip to be sure exactly how the traffic can be expected. The Google Map services provides a very detailed 'Typical traffic' section that depicts the change of traffic conditions with 5 minutes



(a) Metro City: New Delhi

(b) Metro City: Kolkata

(c) Suburban City: Durgapur

(d) Suburban City: Jalpaiguri

Figure 1.1: Google Map traffic prediction for different cities of India

interval differently for each day of the week. So special weekly behaviors like weekly holidays in schools, colleges, offices as well as temporal behaviors are considered in the model to predict precise outcomes. As a result until the availability of sufficient amount of data, the model does not display any data on the map. Whereas in some cases merging of data from different day and time can be useful in coming up with a system that is more available trading off the precision.

Even after these precise measures taken, many a time these prior predictions fail in real time due to the presence of some unaccounted factors. After carrying out a detailed study of road conditions predicted typically by Google and outcomes of the same in real time, some external factors were revealed from the knowledge of actual incidents. These factors make significant difference in real time. The real time behavior deviate drastically in several regions in case of sudden rainfall, storm, holidays, different religious festivals and political activities, due to occurrence of sports events like cricket and

soccer matches, cultural events like Book Fair, Food Festivals, Film Festivals etc. It is also observed that frequent and daily commuters, cab drivers, bus drivers can efficiently suggest routes to take or avoid in such scenarios based on their knowledge. Though it is not possible for the map services themselves to predict such incidents, but with prior knowledge of such events or possibilities of natural phenomena, based on the historical data, it should be possible to develop a system that model such cases.

So, development of a system that uses only low volume data to make useful prediction can also be used in such unlikely cases to model special behaviors that can be used alongside map services for more useful predictions. From the fact that the local people, commuters and drivers are able to predict traffic conditions in special cases from their experience also gives an idea that once a system is developed that can make traffic prediction for certain types of scenarios can also be used to predict traffic conditions for new scenarios(traffic model of a cricket match-day may convey some relevant traffic information of a soccer match-day).

These are the major factors that motivates to build a system that can predict traffic conditions only using low volume data that can be used both for improving the coverage of map services and also modeling the traffic conditions for special cases that are not separately modeled by map services like Google Maps.

1.2 Our Contribution

For next generation of smarter apps and digital lifestyle, there is a scope of improvement in terms of coverage and alternative event based models. In this thesis we present a detailed study of the collected GPS and WiFi data, define features that are useful in traffic modeling and suggest a novel approach to the first step of building a traffic prediction system i.e. identifying traffic conditions of road segments while considering their demography while using only low volume crowd-sourced data. An end to end system is developed

that captures data from volunteers traveling in public buses and motorcycles in form of GPS sensor logs and WiFi signal logs, accumulates the data in a cloud storage, collects the data from cloud for processing, identify different landmarks from the trails using different landmark sensing algorithms depending on travel mode, extract features for different segments, create a model using a novel unsupervised approach that uses the conventional clustering algorithms in combination. The algorithm finally classifies the data as one of the 3 classes - *Chaotic, Med-Chaotic* and *Non-Chaotic* based on low volume data used to build the model.
We developed a data capturing and modeling framework containing an application called **Trans-Portal** that collects data from the volunteers and saves it in Google Firebase Cloud. Later in regular time interval those data are retrieved from the cloud service and processed with python scripts using scikit-learn tools. We tested the outcomes of the algorithm with the help of volunteers and found out that in 72% of cases the estimations turned out to be accurate using nominal amount of data to build the model. The data used for testing the model are annotated while capturing by volunteers for this work. This model can be easily extended further to predict travel times and this extension can be further used by the users and other application that are to be built on top of this layer.

1.3 Thesis Outline

The organization of the thesis discussed in this section. In Chapter 2, we discuss related works in the field of traffic state estimation and landmark detection. In Chapter 3, we propose a system architecture that can be used to collect and analyze traffic data. In Chapter 4, first we challenges in the research. In subsequent sections we discussed the methodology used in landmark identification, analysis of different features and the algorithms used to make the traffic estimation model. In Chapter 5 we discuss the results and analysis of the performance. Here we also depict some of the visualization of the outcomes of the work. Finally in Chapter 6 we discuss the future scope of this research and conclude the thesis

Chapter 2

Literature Survey

Over the years traffic condition modeling has been one of the popular avenues of research. In recent years, since the smartphones came into play, new data has been made available which created new possibilities of research. The total study of related researches can be discussed in two parts. Study of landmark identification algorithms and study of congestion sensing. In the following sections, we discuss some of the papers and their contributions.

2.1 Landmark Identification

The first research in this thesis was to identify the important landmarks in city from GPS trajectory data. The problem of the landmark sensing from GPS data can be thought of a special case of clustering of 2-Dimensional data where 2 dimensions represent the two of the GPS coordinates. DBSCAN[2] has been one of the most popular density based clustering technique used in the field of data mining. But when we cluster data from GPS trajectories, some of the improved and special purpose versions are observed known to perform well. One of such algorithm is TrajDBSCAN[16] which specializes on sparse and diverse multimodal vehicle trajectory data. It was also observed that BusStopFinder algorithm[12–14] outperforms TrajDBSCAN when public buses are taken as probing vehicle. Further, UrbanEye[17] is another research that addresses enriching the landmark set of a map by adding more

information of road landmarks like turns and speed-breakers.

2.2 Traffic condition analysis

Another interesting and popular research topic is traffic condition identification from GPS data. There are many popular researches that uses a large set of historical and real time data to estimate congestion and predict ETA(estimated time of arrival). One of such papers V-Track[15] uses GPS data and other data like GSM to estimate both a users trajectory and estimated time of arrival along a route by using a HMM(Hidden Markov Model) based map matching scheme and time estimation method. In [8] collects data using 100 vehicles carrying GPS enabled mobile devices in a 10 mile stretch for 8 hours. This article focuses on real time data collection mechanism and broadcasting of the traffic data for real time monitoring and analysis. [20] uses the data collected by [8] and applies Kalman filtering approach to highway traffic estimation. In [9] performs Big Data analysis on a large dataset of human mobility, weather conditions and road network data and decomposes the movements in 3 types of flows - seasonal, trend and residual flows. [6] takes a different approach by deployment of low cost sound sensors that reduces the deployment cost. [11] focuses on video streams collected from CCTV cameras and image processing to identify congestions in real time traffic scenarios. In [1] authors surveys a set of algorithms that performs congestion detection from different large scale data. In [19] authors addresses the problem of GPS data being insufficient to fully estimate traffic condition by using other data sources such as social events, road features, Point of Interest (POI), and weather. In [10] the authors suggest the use of limited number of low quality cameras to use the images in an unsupervised algorithm to count the vehicles followed by a inverse Markov chain to infer the traffic conditions. In another paper, [18] the authors combine extensive Twitter data and large scale GPS trajectory data to estimate traffic congestion by using tensor factorization.

As it is found in the above discussion, analysis and prediction from large

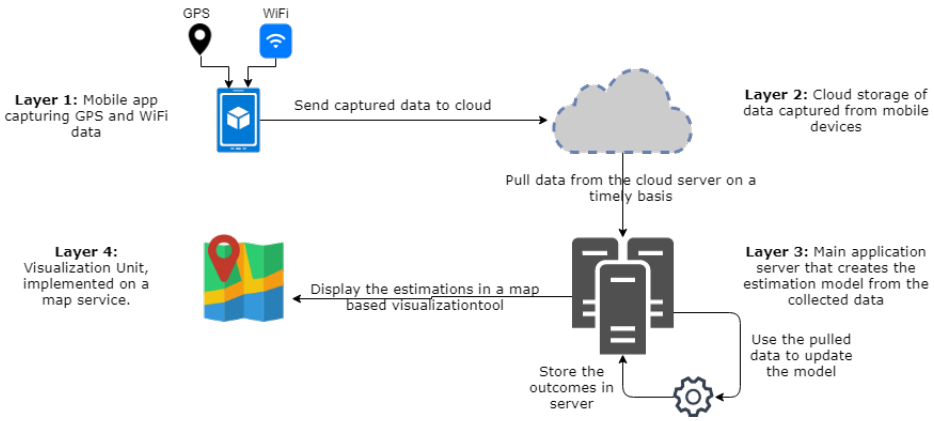
scale different databases has been a popular research over the years. But all these research has one issue is common when we think about traffic modeling in developing suburban regions. These models either depend on installing cameras and sound sensors to collect real time data, or depend on very large volume of traffic and social media data to make necessary estimation. To state a few, VTrack[15] uses a dataset that has over 800 hours of drive data collected from 25 cars. In FCCF[9] 4 different large scale data sets ware used. In [19] a dataset of Chicago with 1257 road segments whose combined length is 700 miles are used to make estimations. In CTCE[18], the data from [19] is combined with 500 million data from public buses to compute expected travel time. But in reality, as seen in the motivation section of the introductory chapter, traffic modeling in suburban area is still largely uncovered due to a lack of data. So development of a context sensitive traffic modeling scheme from low volume data may indeed open new opportunities in increasing the coverage of map services and allow next generation of applications to make the most out of the services.

Chapter 3

System Architecture

3.1 Overview

To build an end to end system that effectively makes traffic prediction based on a model created from historical data, here we propose a 4-layer system model. First layer would consist of devices that can capture data to be used in modeling. Due to the high usage of hand-held devices, this data capturing unit can be implemented as a smart-phone application capturing data from the different sensors it has access to. The second layer would be a data accumulating unit that collect all data from these smart-phone applications and store them permanently for further use. In our case, this layer is implemented using Google Firebase Cloud storage due to its fantastic compatibility with the android platform. The next layer in the model is the one that uses all the collected data to create a model to estimate the traffic behavior. In real time settings this can be implemented in server machines. For this work, this layer was implemented in a personal computer with 2.4GHz Intel i5 processor with 8 Gigabytes of RAM. The final layer is the visualization layer that is implemented using a web-page that displays the information on Google-Map. The basic structure of the whole system is depicted in the diagram 3.1.



(a)

Figure 3.1: System Architecture

3.2 Data acquisition application

The first layer of the system is the data capturing unit, which is a simple android application[4]. The application gets access to the WiFi sensor and the GPS sensor of the android device. A volunteer while starting his/her trip, starts the application after switching on the required sensors. The application then captures data from the sensors in a comma separated file. For the GPS sensor data, Latitude, Longitude, Speed, Altitude and Timestamp is recorded in the file in 1 second interval. For the WiFi data, the sensor continuously searches for available access points around itself. Whenever access points are found in the search, the MACid, ESSID and the RSS is recorded into the trace file along with the Timestamp. Once the trip ends, the volunteer stops the logging and the logged files are sent to the Firebase Storage Unit when the phone receives Internet connection. Apart from the usual logging activity, the volunteers are also given an option to mark the travel experience in a scale below. This information, if available is logged each second and stored in a separate file. For now, it is not used, but the information is kept as a provision to use it in future for evaluating the model.

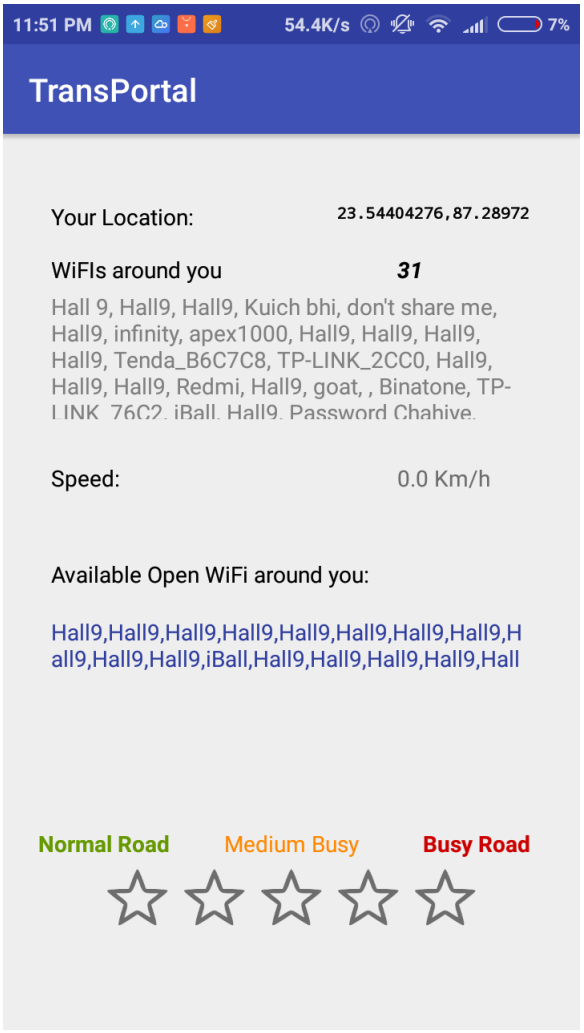
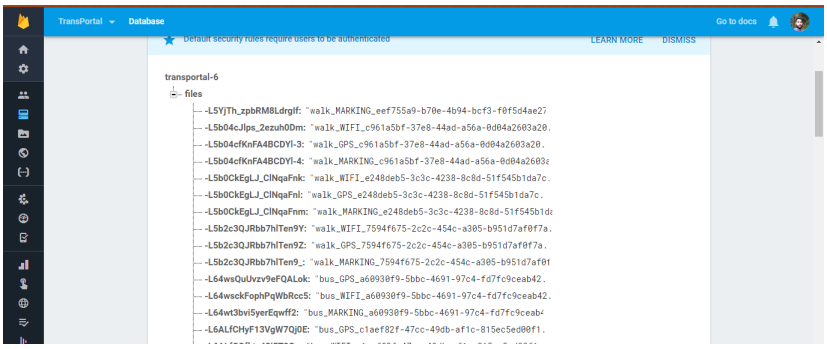


Figure 3.2: Data Acquisition Application

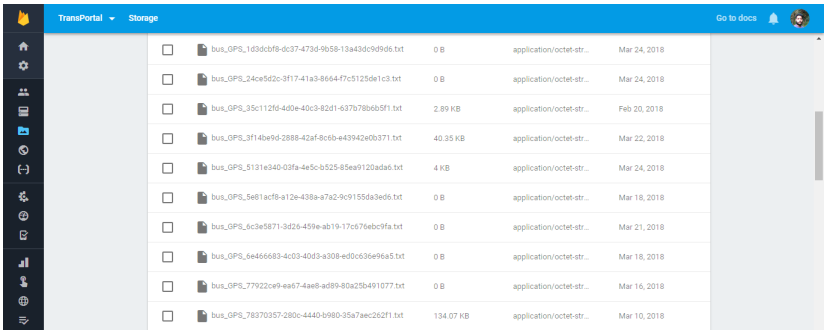
3.3 Cloud based data storage unit

For the data accumulation and storage unit, Google Firebase storage is used. The cloud service is configured to accept raw data and files from authenticated user. For the user authentication purpose, Firebase User Authentication API is used in the application in previous discussion. The authentication adds a layer of security to the whole system, restricting unauthenticated and possibly malicious users from flooding the storage with unwanted data. When the android application completes recording a trip, it makes a call to

the Firebase RealTime Database API with a UUID, this UUID is recorded in the NoSQL database (3.3(a)) of the cloud service. After the id is saved in the database, the files are uploaded to the file store of the cloud (3.3(b)). These files contain the UUID as a part of their names. So, when the files are needed to be downloaded and analyzed, new ones can be easily filtered based on presence of their UUID in the NoSQL database.



(a) Firebase Realtime Database



(b) Firebase Storage

Figure 3.3: Data storage layer

3.4 Data processing unit

The next unit in line, the processing unit is the main application server in the architecture. This unit[3, 5] downloads new files from the cloud storage, filters the trails depending on the mode of transport (Bus, MotorCycle, Car), for bus trails checks whether the trail is from a new route or an existing one.

Once the trails are segregated, landmark detection algorithms are applied on the trails to find the set of landmarks along the route. Detected landmarks are stored accordingly. Identification and analysis of the segments are performed in this layer. The built model is updated as needed as new behaviors, landmarks and segments, bus routes can be identified dynamically with availability of new trails.

3.5 Visualization

The final layer in the system is the data visualization unit which is a simple web page that uses the Google Map api where collected trails are displayed to the users after estimating traffic conditions and displaying it in a color scheme similar to the one used in Google maps.

Chapter 4

Methodology

4.1 Challenges

While designing a traffic behavior estimation system, first step was to go through the use case of the system and defining what are the things needed to build the application. The problems and scope of improvement are discussed in the motivation section(1.1). First challenge that is faced while performing such research is the availability of data. The existing popular system like Google-Map uses a huge volume of data collected from mobile devices to make predictions. The same data can be analyzed in detail to make interesting estimations on the traffic behavior in places where there the services are limited. Modeling special cases can also be done using these data. But unfortunately such huge data set are not made available for research purpose by the owners. So this research was limited to the data available on-line in research communities.

The next challenge is to make the data that are available online useful. There are some data sets available that provides GPS traces of different traffic modes, but to build a system that can estimate traffic with sparsely available data need to contain some additional information apart from the obvious GPS traces. In this research, available WiFi signal around a place is used as an indicator of the amount of human activity or business around the

place. No data set were available that contained these data simultaneously. So for this research, data was collected according to requirement with the help of volunteers in the town Durgapur, in Eastern India, where the movement patterns of public buses were analyzed and modeled to make useful estimations about a road segment.

4.2 Need of recreating the transport layer on top of map service layer

While analyzing the behavior of a specific road segment in terms of traffic, the first idea we need to define is segment. What is a road segment? A segment can be defined as the part of a trail that connects two landmarks. Landmarks can be simply considered as some important places in the map, that are connected by road segments. So the question that can come up is that how do we define the segments and landmarks. Popular map services like Google Maps has millions of landmarks stored in their databases. Similarly, there are number of segments in same scale. Google map has road segments as small as 100 meters. To define such high number of road segments, the map service also have landmarks with no or less physical significance. Such granularity is useful when we have relevant amount of data that can use such granular informations. Practically, as per the regular commuters, they are mostly considered about the behavior of the road segments that connect places of human interest like Bus Stops. Comparing it to the map services like Google Maps, the road segment connecting two bus stops can correspond to a set of segments as shown in such map service.

So why do we prefer less granular road segments? First, as the target of the system is to enhance the service of existing map services by increasing the coverage and modeling special situations, it is considerable to trade off the quality of the service for generalized estimation with higher coverage. For the areas that are not covered in popular map services, overall idea of roads being busy can be useful in planning trips even if the system cannot provide

exactly which part of a road segment is slow. Secondly the generalized idea of segments would be user friendly enough to take decisions based on the estimation model. For example, the information that place A to place B is busy at evening can be useful enough to plan trip even if the is information is not as granular as "The first 100 meters and 5th 100 meters between the places A and B stays slow form 5.20 PM to 6.40 PM". Specially considering the system is designed to provide road behavior analysis in areas where no information is present, can be helpful even if the information lacks details.

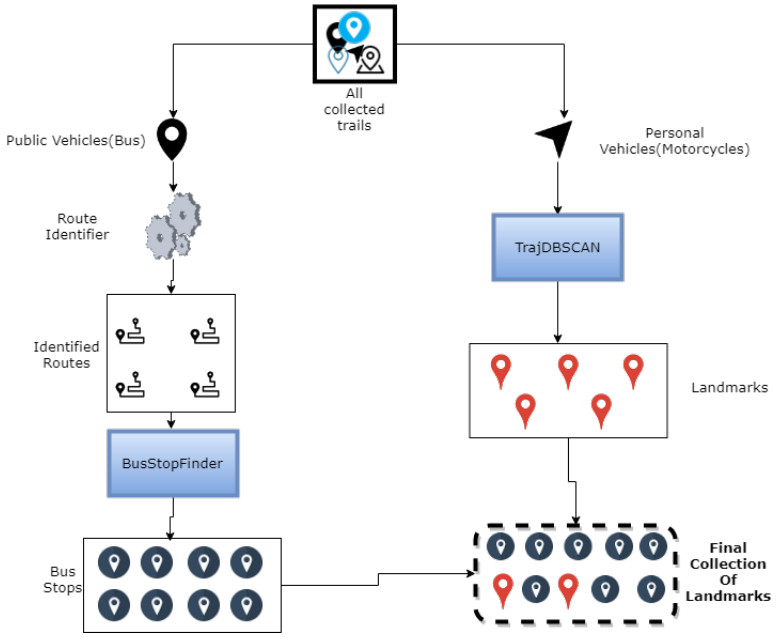
So to achieve this, we first create a simplified road network were the nodes can be considered as the landmarks and the roads connecting them can be considered as the edges in the network. To create this network, we use the GPS trails to identify the points of interest along the trails. From this set of points of intersections, referring to as *landmarks*, we define the road segments as the parts of the trails joining them. Then we extract features of these segments and model them according to their behaviors.

4.3 Landmark detection in different travel modes

Landmark detection from GPS trails is a vast field of study. There are a number of algorithms that identifies landmarks from GPS traces. Popular algorithms like CB-SMOT, TrajDBSCAN are built on the popular clustering algorithm DBSCAN perform well in different cases. BusStopFinder, an algorithm that also follows the same basic principle, performs well when there is a set of trails available in a specific route. But this algorithm has a constraint that it needs data in same route. As we have a collection of trails gathered from public buses and two wheelers, we have taken two different approaches in identifying the landmarks.

4.3.1 Personalized two wheeler vehicles

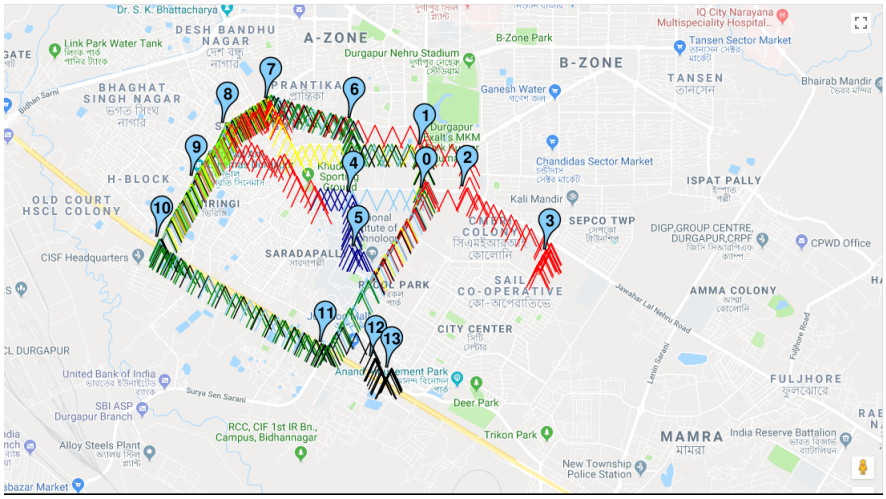
First obvious observation in the data collected from motorcycle is that these vehicles has a very low probability of stopping down at areas due to its small



(a)

Figure 4.1: Workflow of landmark detection from heterogeneous GPS trails.

size and high flexibility. Such vehicles having a compact size, can move past some obstacles that larger vehicles cannot. On the other hand, motorcycles being a personalized vehicle, has fewer chances of stopping at important landmarks, until that landmark is of importance to the rider. Moreover, it was also observed that personalized vehicles, specially motorcycles have a higher tendency of taking routes that are less taken by other vehicles. Though it provides a larger view of the movement pattern in areas other than main bus roads, due to its driving nature, identifying landmarks is a difficult task. We applied DBSCAN and TrajDBSCAN algorithm on the data collected from motorcycles and identified the landmarks according to the algorithm. These found landmarks were saved into a set of landmarks that are used later to identify road segments.



(a)

Figure 4.2: TrajDBSCAN results on Motorcycle trails

4.3.2 Bus/Public transport

For public transport like buses, BusStopFinder is considered as the preferred algorithm as it is known to perform well in the special case of trails collected from buses of same route. To use this algorithm properly, first step was to identify the different bus routes from the collection of trails. Once the routes are identified and trails are arranged properly in different routes, the landmarks can be easily identified using the BusStopFinder algorithm. The whole process of bus stop identification from trails is described below in two different subsections. The first one discusses the route identifications and the second one describes landmark identification in a specific route.

4.3.2.1 Route Identification

Identification of routes are done through a novel algorithm discussed in this section. On identification of a new route, an outline of the route is kept in a specific folder. The outlines are called the skeletons of the route, which are GPS points along the route with a distance of 100 meters. First, when a new trail is processed, the system lists all the different routes that are present in the routes folder. Now the trail is compared against all the known routes.

This process is described in Algorithm 1 Evan though the idea is to follow each point in the route one by one to identify whether the trail matches a route, it is a really slow process as in the best case, only one route would be successfully matched with the trail. So a faster rejection of the other routes are required to make the matching process faster.

Algorithm 1 FindMatchingRoute

```
1:  procedure FindMatchingRoute                                  ▷ Input: trail
2:      trail ← Collection of GPS and Timestamp in the trail.
3:      estimatatedTrail ← A rough estimate version of trail taken in 100 seconds interval.
4:      notMatch ← true
5:      skeletons ← Collect all the available route skeletons
6:      toRemove ← []
7:      for skeleton in skeletons do
8:          closePoints ← FindClosePoints (estimatatedTrail, skeleton)
9:          result ← FindMatch (trail, skeleton, points)
10:         if result == 'match' OR result == 'superroute' then
11:             notMatch ← false
12:         if result == 'superroute' then
13:             toRemove ← toRemove + [skeleton]
14:     if notMatched == false then
15:         Create new route and its skeleton using trail.
16:     for skel in toRemove do
17:         move all trails under skel to the newly created route.
18:         Remove skel.
```

In the FindMatchingRoute 1 algorithm, first a minimal version of the trail is computed with less granular points - in 100 seconds interval. This approximated trail is then compared to the known skeletons in the FindClosePoints algorithm 2 which returns pairs of points, one from the approximate trail and the other from the skeleton that are observed to be within 60 meters of each other. These approximated information is then sent to the FindMatch algorithm 3 where the relations of the trail with respect to the skeleton is returned. These relation can be either of the following values - *different, crossing, opposite_direction, superroute, match.* If the result is *different*, a new skeleton is created taking the current trail as the new route. In case of *superroute*, a trail is found which not only matches the existing trail, but

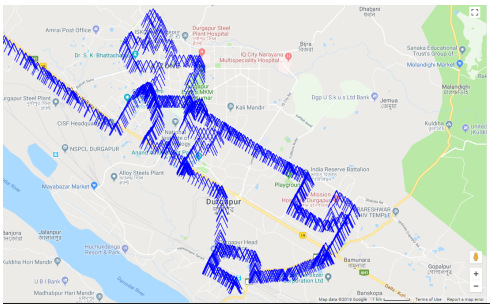
Algorithm 2 FindClosePoints

```
1: procedure FindClosePoints                ▷ Input: skelPoints,trailPoints
2:     skelPoints ← All points in the skeleton
3:     trailPoints ← All points in the trail
4:     closePairs ← {}
5:     for skeletonPoint in skelPoints do
6:         for trailPoint in trailPoints do
7:             if Distance between skeletonPoint and trailPoint less than
   60 meters then
8:                 closePairs ← closePairs + {(skeletonPoint, trailPoint)}
9:     return closePairs
```

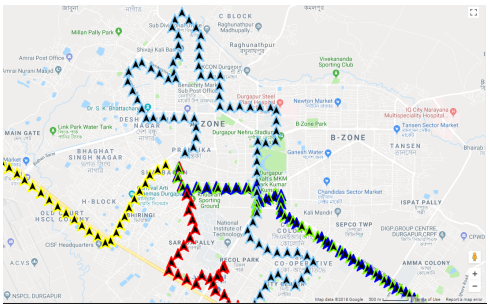
Algorithm 3 FindMatch

```
1: procedure FindMatch                      ▷ Input: trail, skeleton, closePoints
2:     if length(closePoints) < 1 then return 'different'
3:     if length(closePoints) = 1 then return 'crossing'
4:     if The first skelPoint entry in closePoints comes later in the original
   skeleton compared to the last skelPoint entry of closePoints then return
   'opposite_direction'
5:     closeSkels ← skeleton point entries of closePoints
6:     for skelPoint in closeSkels do
7:         if Difference of index of skelPoint and index of next point in clos-
   eSkel is more that 20 then
8:             return 'different'
9:     Now check each point in trail and verify if they follow all points in the skeleton.
10:    if The trail skips a point in skeleton then
11:        return 'different'
12:    if The trail lasts longer than 1000 meters between two consecutive
   points in the skeleton then
13:        return 'different'
14:    return 'match'
```

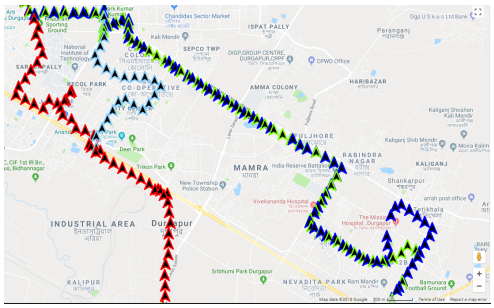
extends it. In such case a new skeleton is created using the current trail and all trails in the route that is found to be the subroute of the current trail are moved to the newly created trail. The older route and its skeleton are deleted as a new route is already found that contains it.



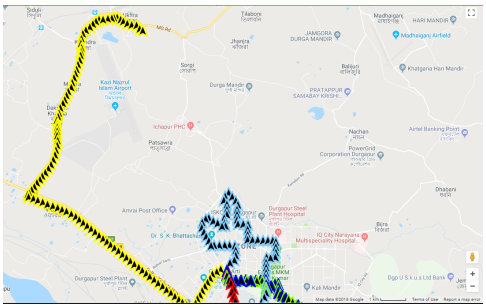
(a) Trails before route detection



(b) Identified routes in different colors:1



(c) Identified routes in different colors:2



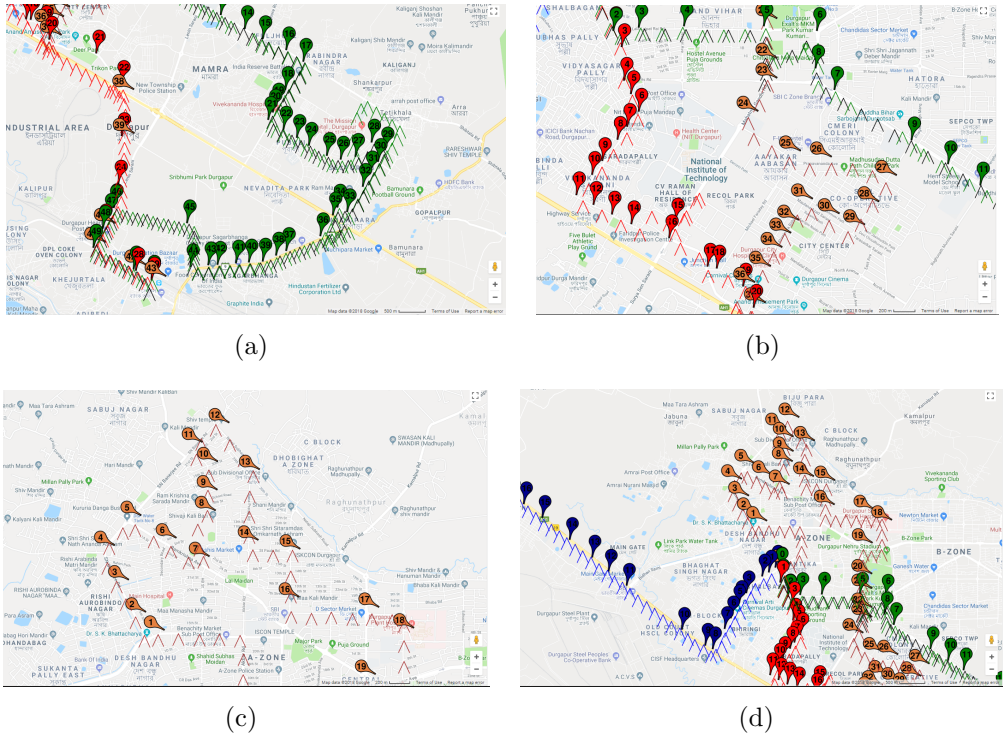
(d) Identified routes in different colors:3

Figure 4.3: Raw trails and identified routes

In the figure 4.3, 4.3(a) is the representation of a collection of raw trails visualized on google map. 4.3(b), 4.3(c) and 4.3(d) are the results of the Route Identification algorithms discussed in algorithm 1. In the figure 4.3(b), the green and the blue outline looks as if they identified the same route twice. But a closer look into the bottom right side of 4.3(c) reveals that due a small area where the buses take different direction, the algorithm identifies 2 different routes out of the trails in the said region.

4.3.2.2 Landmark Identification

Once different routes are identified using the route finder algorithm, we can use the BusStopFinder algorithm to identify the different bus stops in those routes. Once all the bus stops are identified from all the routes, we store them into a common database and later refer to this database to identify the present road segments and store their details. The figure 4.4 contains the details of the bus stops found in the different routes.



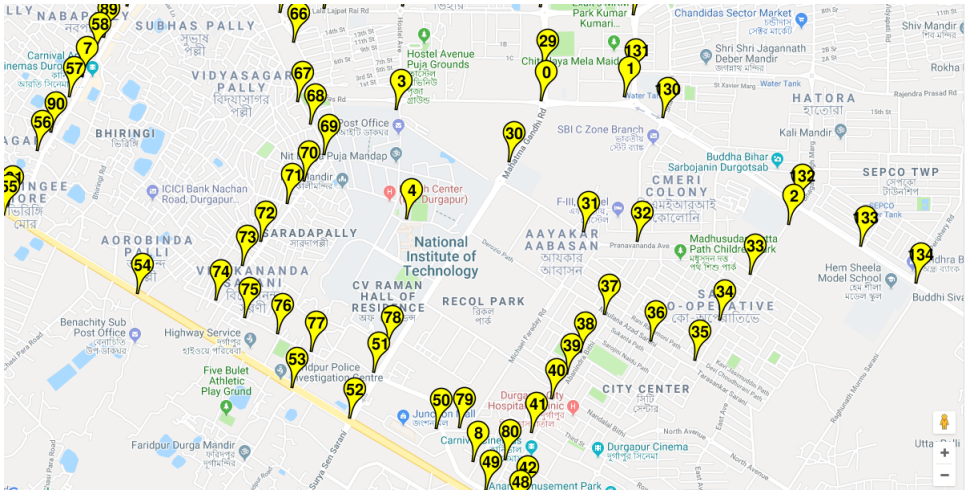
(a) (b)

(c) (d)

Figure 4.4: Bus stops identified by BusStopFinder

In Figure 4.4(a) it can be seen that there are some bus stops that are common multiple routes. Comparing the landmarks found in Figure 4.2 we can also spot some of the landmarks detected by personal vehicles (Motorcycles) appear as bus stops too. A simple clustering algorithm is then used to identify the distinct stops/landmarks. In case multiple stops or landmarks are detected within 30 meters of each other, they are considered as a single landmark and updated in the database accordingly. In figure 4.5 a subset of

the final collection of landmarks are displayed.



(a)

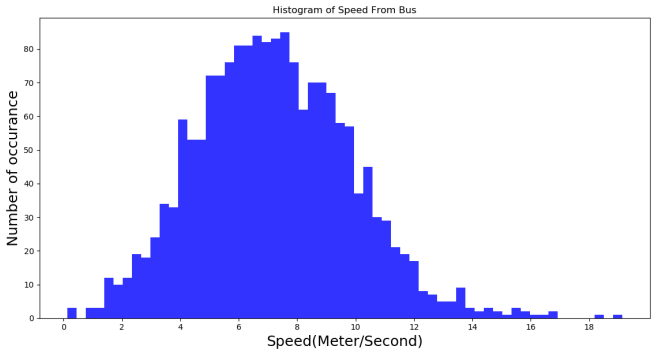
Figure 4.5: Snapshot of all detected landmarks from Bus and Motorcycle

4.4 Features

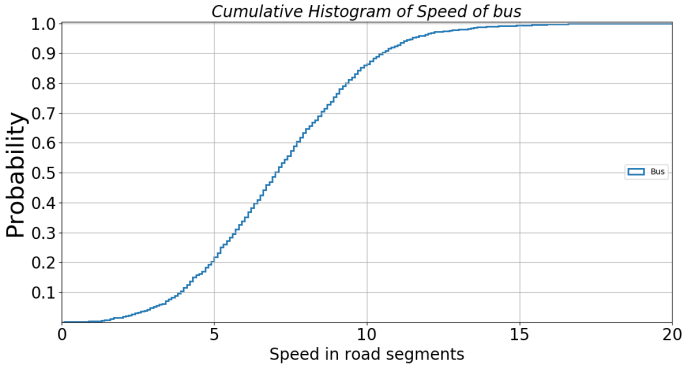
To estimate the traffic condition of a certain road segments, we need to identify some of the aspects of vehicular movement that captures the behavior of the traffic. From the smart-phones used by the volunteers, we are capturing information about GPS location and WiFi access points active around. From road behavioral studies and surveys conducted on daily commuters, we identified some of the features that can be used in modeling the traffic conditions.

4.4.1 Speed

The first and the most obvious aspect of vehicular movement is the speed of the vehicles. Though different features contribute to the sense of congestion or a sense of a road being busy, speed is the most obvious one. Most of the state of the art systems like Google Maps, OpenStreetMaps, Bing Maps use speed and other derived features of speed to create nice visualization



(a) Histogram

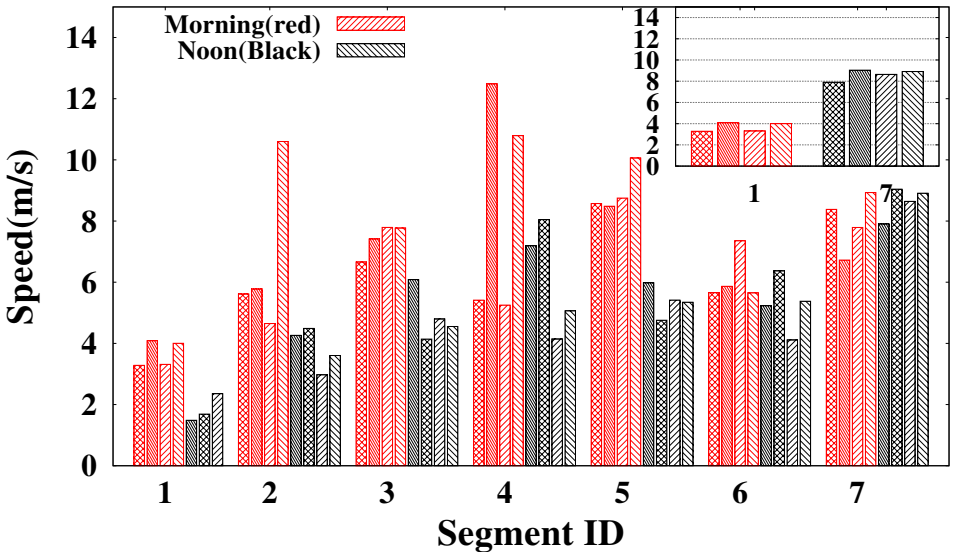


(b) Cumulative Histogram

Figure 4.6: Histogram and Cumulative Histogram of speed collected from public bus,

of the speed profiles of cities. Here, in this thesis, speed is also a very important feature. To estimate the behavior of a segment with respect to a probing vehicle, speed is captured for each of the segments the vehicle move through. From these collected data, we by using a novel approach try to classify the movement as slow or fast. Figure 4.6(a) shows the histogram of speed collected from buses. It can be seen that the histogram is more like a bell shaped curve. The most of the observed speed ranges from 5m/s to 10m/s. The same information can also be visualized in the cumulative histogram 4.6(b). The histogram has the steepest raise from 5m/s to 10m/s. It can be also seen that the histogram reaches the top mark of 100% near the 15m/s mark. Even though after 10m/s there are very few samples that

have a higher speed.



(a)

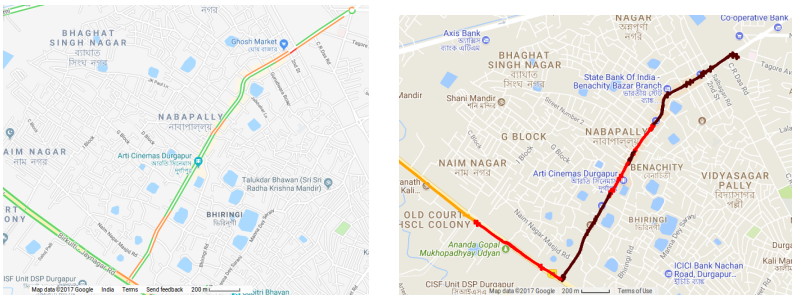
Figure 4.7: Speed in a sequence of road segments in two different times of the day.

In the following Figure 4.7(a), the speed of few segments are displayed. The captured data are from 7 continuous road segments in Durgapur city. The initial 5 segments are through a market area and the last segment, segment 7 is connected to a highway. The segment 6 has a mixed kind of demography, it has few shops around it, but it has 3 speed breakers placed on the road because of the presence of two schools on both sides of the road segment. From the sampled data, some observations can be made. The speed in segments 6 and 7 stay consistent when compared with the first 5 segments. As the first five segments are through market area, the segments have a tendency to become slower in noon compared to early morning, as more shops become active later in the day. Which is not the case for segments 6 and 7, where the speed stay the same. Though segment 6 has lower speed due to the presence of speed breakers in the middle.

But, to identify the busier segments, it is not correct to evaluate the

speeds observed in different segments in same scale. A segment in the market area tends to stay slower than normal city roads due to high activity around them. So if the speeds are considered in an absolute scale, market segments may always appear busy compared to other segments which might not be the case. Figure 4.8 depicts two images, 4.8(a) is the google map traffic analysis of the area discussed in 4.7 and 4.8(b) is the result after clustering the speeds in 4 clusters and marking the segments based on the data collected from those in the morning.

It can be easily inferred that the figures are not at all conveying the same information. So it gives an idea that the expected speed is segments are different from each other. Speed in a segment even if it is slow, cannot be used to infer the segment to be busy until we have an idea about the expected speed of the segment. Again the expected speed in a segment is dependent on the surrounding area and many properties of the road like the road surface condition. So if we can come up with a framework that can measure what the speed in a segment should be, we can effectively estimate how busy or chaotic the segment is.



(a) Google map typical traffic on morning

(b) Visualization of collected speed data from morning

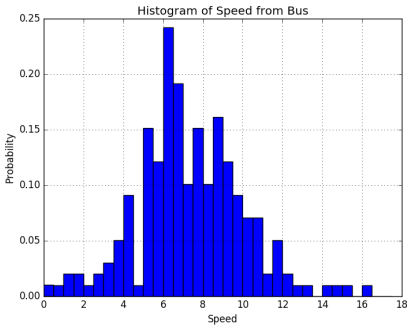
Figure 4.8: Speed based profiling results.

Another interesting study has been carried out regarding the speed patterns in trajectories of heterogeneous vehicles. For this study we the movement patterns of bus and motorcycles in a section of a route that has a part of it through a market, a part through the residential areas and a small part

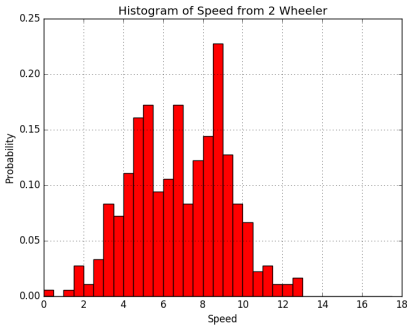
in the highway across the city. The figure 4.9 has two histograms of speeds in the experiment. 4.9(a) is the histogram of bus speeds and 4.9(b) is histogram of speeds of motorcycle. The figure 4.9(c) depicts the cumulative histogram of both types of vehicles. From this image, it can be seen that both the histograms reaches 100% almost at the same point, which indicates the highest speed observed in these two modes of travel are almost same. It can also be observed that the histogram line of bus always lies over the motorcycle line. In other words, if we look into a particular speed, bus has higher number of examples upto that speed - indicating that throughout the experiment motorcycles display an overall higher speed. Again from 4.9(a), we can also see that bus in rare case it displays very high speed, possibly when the road is empty and the bus is able to use its full potential to speed up in the situation. But the overall speed is high in motorcycle because that a significant quantity of the sampled data are taken from market and normal city segments where a bus has higher tendency of slowing down whereas motorcycles can move pass some obstacles due to its compact size that a bus cannot.

4.4.2 Coefficient of Variation in speed

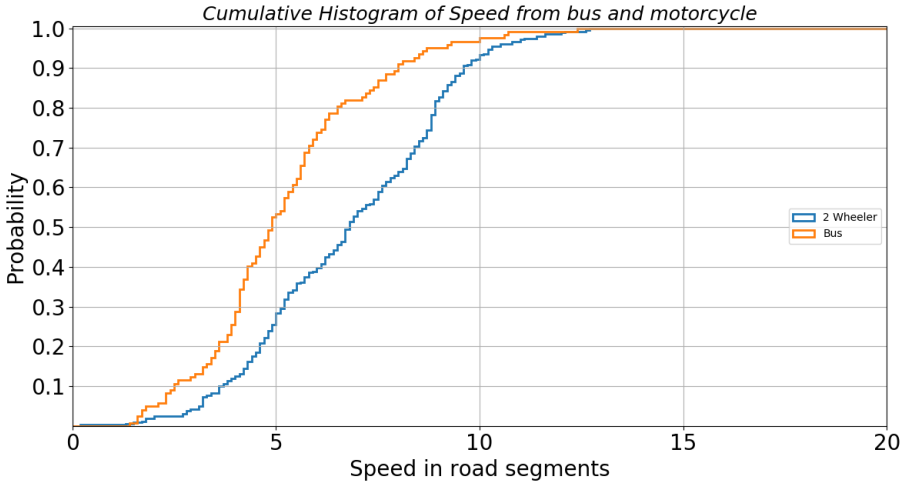
To understand and model the sense of business or chaos in a road segment, standard deviation of speed is a vital feature. It was observed from the experiences of the volunteers that a segment is often considered busy by people when the traffic condition interrupts a steady speed movement of a vehicle. If a vehicle speeds up and slow down to and fro in a segment rather maintaining a steady speed, the segment is considered chaotic or busy by the travelers. To model this perception of the travelers, the standard deviation was considered to be a feature and was observed in detail. The problem of using standard deviation of instantaneous speed as a feature is that its not really meaningful without considering the mean of the distribution.
For an example, let's say two segments A and B has same standard deviation observed, say 3. Let's consider the mean speed in the two segments be 7m/s and 15m/s respectively. Now, it is obvious that when the mean speed in a



(a) Histogram of speed of bus



(b) Histogram of speed of motorcycle



(c) Cumulative histogram of speed from motorcycle and bus

Figure 4.9: Comparison of speeds in different vehicles.

segment is high, minor changes in speed may cause the standard deviation to increase due to the higher magnitude of the sample instantaneous speeds. In other words, the same amount of by only looking at the standard deviation, it is not justified to comment on a segments' situation without taking the mean into account.

If we look into Table 4.1, a collection of labeled samples are displayed in the tabular form containing their Mean Speeds, Standard Deviation of instantaneous speed, Coefficient of variation(CoV) of speed and labels tagged by the volunteers. All the said examples have a standard deviation in range of 3 to

3.1. But the behavior tagged by the volunteers differ for the observations. When we look into the Coefficient of variation column of the table, this data with the labels seems more correlated with the label column. In each case the volunteers labeled an example to be *CHAOTIC*, the CoV display a higher value. In 3 of such cases, the value of CoV is over 0.53, which is more than most of the other entries in the table. On the other hand, looking into the *NONCHAOTIC* entries of the table, we can see the CoV values to be in lower magnitude - all the 5 observations having their values less than 0.44.

Table 4.1: Speed features and resulting labels

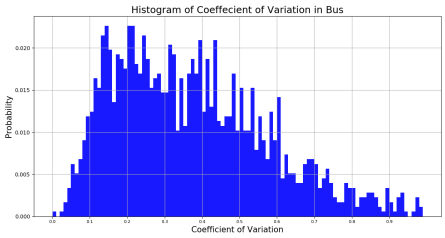
Mean Speed	**Standard Deviation**	**Coefficient of Variation**	**Label**
7.198165804	3.082986515	0.428301681	NONCHAOTIC
9.410659697	3.061029172	0.325272539	NONCHAOTIC
7.099858584	3.061313619	0.43117952	NONCHAOTIC
5.351830515	3.004744678	0.56144242	MED-CHAOTIC
4.180040293	3.053987021	0.730611862	CHAOTIC
10.5567917	3.081804085	0.2919262	NONCHAOTIC
5.895124299	3.037567883	0.515267826	MED-CHAOTIC
10.12673542	3.0756749	0.303718303	NONCHAOTIC
4.952794584	3.083595734	0.622597138	CHAOTIC
5.632153932	3.02791979	0.53761311	CHAOTIC

Relation of speed features and labels tagged by volunteers.

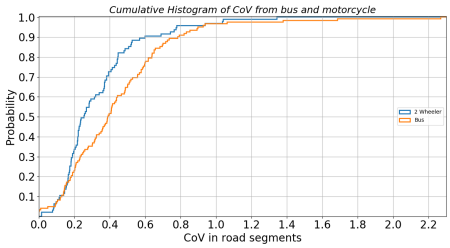
From this above discussion, it can be said that Coefficient of Variation(CoV), as a feature has a correlation with the traffic condition experienced or perceived by the volunteers/travelers. CoV is a feature that is the combination if both the Mean Speed and Standard Deviation. It is computed as:

$$CoV = \mu/\sigma$$

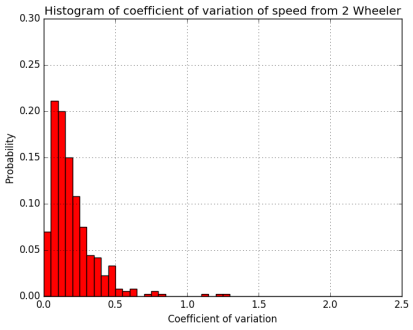
Where μ and σ are Mean and Standard Deviation of the instantaneous speeds respectively. So by taking the Coefficient of Variation as a feature instead of Standard Deviation, we achieve two goals. First we take into account the amount of inconsistency in the data and also we take the data scaled as fraction of the mean of the data, which is seen to make the data more relevant to the human perception of traffic chaos.



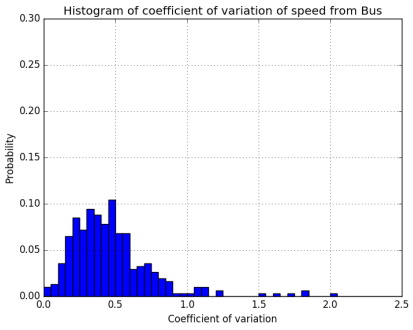
(a) Histogram of CoV in Bus



(b) Cumulative histogram of Bus and Motorcycle in Market



(c) Histogram of CoV in motorcycle in market area



(d) Histogram of CoV in bus in market area

Figure 4.10: Plots on Coefficient of variation

Figure 4.10 contains some of the motivational plots on CoV. The first plot4.10(a) is the histogram of all the collected data. From this graph, it can be seen that the bus has roughly two types of behaviors. There are a large number observations that have a CoV less than 0.35. Similarly, there are also a large number of observation that can be seen having a value higher than 0.4. The figures 4.10(d) and 4.10(c) are histograms of CoV taken in two different modes of transport, motorcycle and bus respectively. These data are taken only from a market section of the roads where significance difference is observed in these two modes of transport. 4.10(b) is the cumulative histogram of the same data as 4.10(d) and 4.10(c). From these figures, it can easily be spotted that the data collected from motorcycles have significantly low values of Coefficient of variations then data collected from the bus. Which indicates bus has a lot high chance of slowing down, stopping in between and speeding up again whereas the motorcycles show a lot steadier movement pattern. This matches with the earlier observations made that for small vehicles like motorcycles can easily bypass some obstacles due to its compact size that a bus cannot. This causes more impact to the speed and movement pattern of a bus which can be seen here.

4.4.3 Segment Length

Apart from the features related to speed, to develop a system that can model traffic behavior taking the condition of the surroundings and road profile into consideration, some features has to be considered that impact the traffic condition. The length of a segment can be considered as one of such features that impact the speed. It is often observed in case of public transport like buses that by nature stops at most of the landmarks, the length of a segment plays a significant role. If the length of a segment is really small, indicating the segment connects two landmarks not far from each other, the speed is often observed to be slow. Such case happens in road segment that passes through high human activity like market regions. It is also observed that the landmarks in market area tend to be situated in close proximity. Thus, when a vehicle moves through the segment, it does not have enough opportunity

to speed up as it has to slow down and stop again in short time interval. As a result in such cases the speed of public buses tend to stay slow throughout. But this information does not necessarily mean that the segment is busy or chaotic, it suggests that the expected speed is low in these segments.

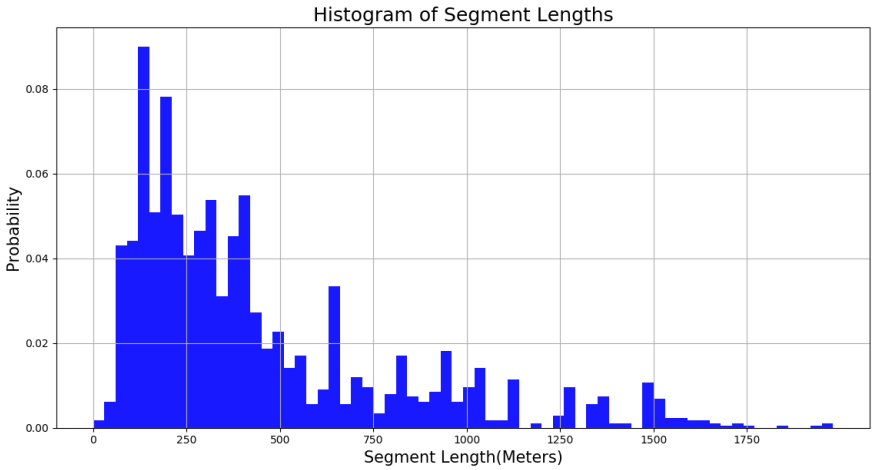


Figure 4.11: Histogram of segment length.

In the histogram 4.11 we can see that there are a high number of segments with length less than 250 meters. Similarly, there are also a number of segments observed having length ranging from 250 to 500 meters. Also, there are segments with even higher length but those are not that dense. From this figure it can be inferred that there are roughly two types of segments having length less than 500 meters, and there are also some segments with even higher lengths. In a naive way we can consider these segments to be from market, normal residential areas and highways respectively as the lengths of these segments correlates with the underlying idea.

4.4.4 WiFi Density

All the features discussed above captures information about different aspects of traffic behavior. Mean and Coefficient of variation of speed contains information about how the probing vehicle behaved subjected to a traffic condition. The length of a segment captures information about the proximity of

the landmarks, giving an idea about the surroundings of an area. But these informations are insufficient in modeling the movement pattern of vehicles when the plan is to build a context sensitive traffic modeling system. A small road section might indicate that the segment is through an area where there are landmarks present close to each other, but it does not necessarily conclude about the speed that cab be expected out of it. Moreover, segment length is a static, time-invariant feature. Even in areas like market, at certain times free flow of vehicles can be expected. In early morning, late night, holidays and weekends certain road segments are observed to be congestion free even though they have a small length. Thus, modeling traffic behavior in cities will be more accurate if we can include a feature that can capture the amount of human activity happening around a road segment. Higher human activity around a segment can be a reason of slow movement of vehicles due to safety reasons. Often in areas of high activities like market, office areas speed limit is imposed in daytimes to avoid accidents. Similarly, at night when the amount of activities around the segment goes down, faster movement can be expected in those areas.

It was observed that this behavior around road segments surprisingly can be modeled using the available WiFi access points around the segment. There are two types of explanation required in this case, first is that the data really have the same physical significance and second, it changes as expected. First, higher WiFi density around a segment indeed signifies higher human activity. WiFi access points are set for people to use Internet facility. In general the wifi signal can be coming from different sources like access points set in stores, offices and mobile phones that people are using, acting as WiFi access points. Either way, higher number of available access points would indicate more number of stores, offices and people, which might in terms cause the overall decrease in speed in some segments of the road. Second the fact that it has a temporal behavior. Unlike the previous feature segment length, WiFi density is not a static feature. WiFi density, as it indicates the active WiFi access points around is a temporal feature as it drops when those access points are switched off or moves away. Thus, it can be stated that with time , when human activity around a segment drops, the WiFi

density also decreases. This can be easily visualized in the following figure 4.12 where WiFi density (in Unique access points/meter) is plotted against segment ids. Data from two different times of the day are displayed in the figure.

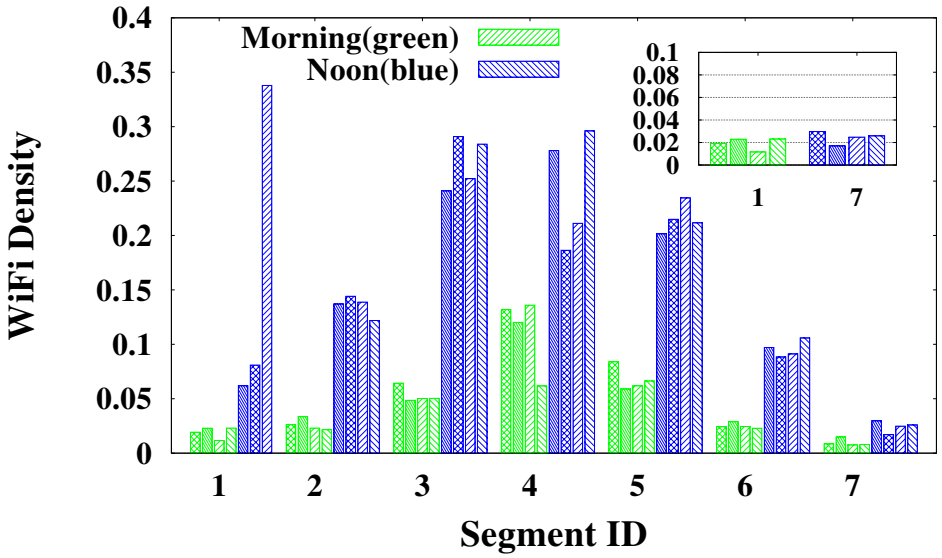


Figure 4.12: Distribution of WiFi density in segments in two times of the day.

In figure 4.12 WiFi density is displayed in 7 different segments of a road. The road segments discussed here are same as discussed in figure 4.7(a). The initial segments the wifi density is observed to be lower in morning hours when compared to the data collected in noon. This shows that the feature captures the information about the change of human activity around the segment. In early morning when very few shops are opened in the market, the wifi density stays low, it raises with time. Whereas in segment 7, which is a segment that meets the highway, there is not much observable change in WiFi density suggesting a consistent speed in that segment. In the inset figure, WiFi of segment 1 in morning and of segment 7 in noon are compared. It can be seen here that they display very similar feature. Indicating the level of human activity around a market road segment in early morning is similar to a normal city road segment in busier hours. Thus, similar speed behavior can be expected in these cases.

In the figure WiFi density is plotted as a histogram. We can see that there

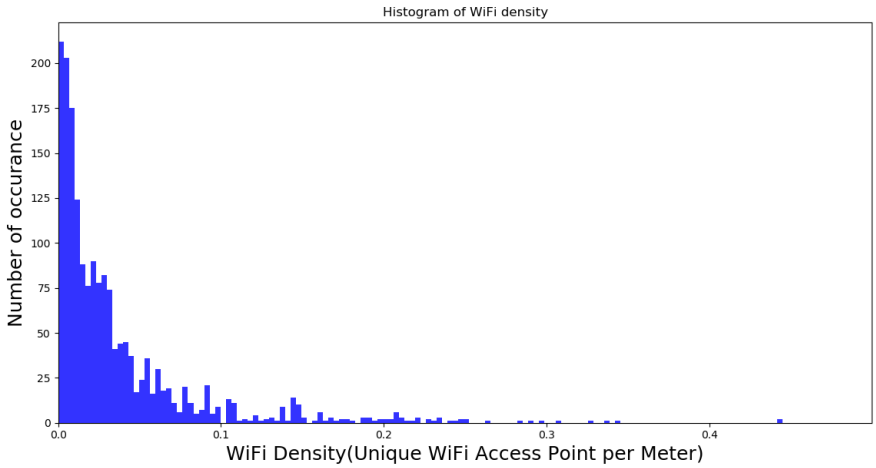


Figure 4.13: Histogram of WiFi Density

are a very high number of observations having very low WiFi density and the number of observation drops exponentially.

But in reality when trying to understand the difference of the amount of activity occurring around a segment, simple difference of WiFi density is not sufficient. It depends on the areas we are comparing too. Let's discuss this with an example, say a segment A have 1 WiFi access point per Kilometer, segment B has 11, segment C has 250 and segment D has 260. So the wifi densities in the four segments will be 0.001, 0.011, 0.25 and 0.26 respectively. The difference in WiFi densities between A-B and C-D are 0.1 in both cases, yet A and B can be considered as two entirely different areas, one being lonely and the other having somewhat of activities around segment. The same difference hardly means any difference between C and D as they both are already having a very high amount of activities happening around them. If we can modify the WiFi density such that when the values are low, their difference would be more significant compared to the same difference when the values are high. To achieve that we would need use a function that maps the values so that the differences are more useful in our case. Luckily this conversion can be done very easily by computing a constant fractional power

of the values. For simplicity, having all the values in the example raised to the power 0.5 would make the values like: 0.032, 0.104, 0.5, 0.509. Now if we evaluate the difference between the samples A-B and C-D, the difference between A and B is 0.072 and the difference between C and D is 0.009. This information correctly captures the information that there is more difference between the segments A and B compared to the difference between segments C and D.

So the next question is what should be the value of the constant power of used and how the performance of this conversation can be evaluated. To find a good value, we created a labeled set of WiFi densities, where WiFi densities of segment are stored with labels in a scale of 1 to 3 where 1 means less activity around a segment and 3 being very high activity around a segment. Now we tried a number of values as constant power and clustered the result in 3 groups. The performance of the clustering with respect to the labels would give an idea how good the data was able to clustered. Figure 4.14 is the diagram that depicts the performance of the approach for different values taken as power.

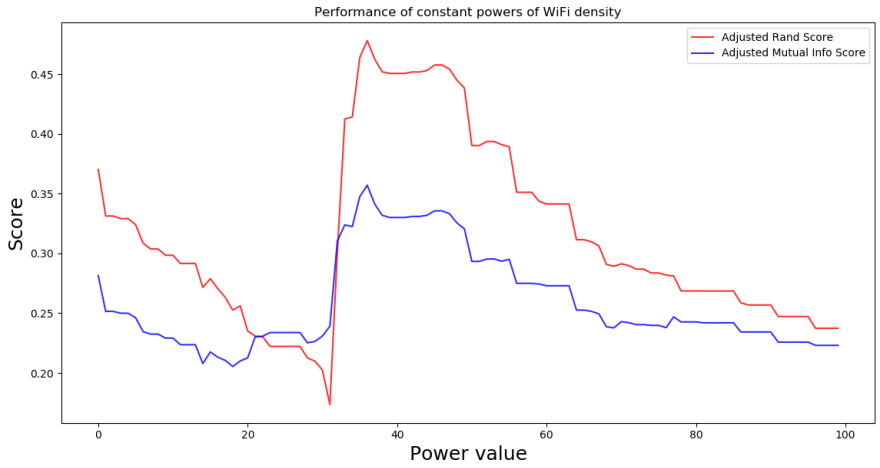


Figure 4.14: Clustering performance of WiFi density raised to a constant power

The performance is evaluated using two clustering performance metrics - Adjusted Rand Score and Adjusted Mutual Info Score. In both the cases, it was seen in 4.14 that the best performance is achieved in clustering when the wifi densities are raised to a power of 0.37. In later computations, we have used this value as power to rescale the WiFi density values.

Taking features such like WiFi also imposes some restrictions. Though it provides useful data in many sub-urban and rural areas. But as here the conditions of developing nations are being considered, there are areas present where WiFi is not used as much. This feature becomes less significant in such cases, but still in major cases, WiFi successfully captures the feature discussed above.

4.5 Feature extraction

In this section we discuss the feature extraction process in detail. In previous sections we discussed how we accumulate the set of landmarks in a city. Now from the information extracted about the stops, we define the segments to be the trails joining any two of the landmarks. When a trail is given, we look into the trail each point at a time and define two states of that point - *stop*- and *move*. The *stop* state is when the gps point is within 30 meters radius of a landmark and *move* is when it is more than 30 meters away from the nearest landmarks. When a sequence of consecutive points in a trail are in continuous move, we keep track of the instantaneous speed and WiFi access points scanned at that specific time. When a sequence of moves as the trail reaches a stop, we compute the features as mean(μ), standard deviation(σ), coefficient of variation(CoV) of collected instantaneous speed and WiFi density computed as unique WiFi access points found in the last movement, scaled by the length of the last movement. These features are stored along with an id of the trail and the segment the features indicate.
For this part of the thesis we used only speed related features and WiFi density to estimate the traffic state. In the following section we describe the steps of the unsupervised approach taken to model the movements of vehicles.

4.6 Unsupervised Traffic Condition Estimation Algorithm

In this section we discuss the main work-flow of the unsupervised estimation approach. The main algorithm can be divided in two parts - building model and estimating the state of a new data. Building the model can be explained in 4 steps. An overview of the algorithm can be found in 4. A detailed explanation of each step is given afterwards.

Algorithm 4 CreateTrafficEstimationModel

1: **procedure** CreateTrafficEstimationModel
2: ▷ Input: totalDataSet, numberOfDistinctAreaTypes, wifiScalingFactor
3: $wifis \leftarrow$ All the WiFi density data from *totalDataSet*
4: $scaledWiFi \leftarrow wifis^{wifiScalingFactor}$
5: assignment, centers $\leftarrow kmenas\,(scaledWiFi, numberOfDistinceAreaTypes)$ ▷ // Clustering scaledWiFi into numberOfDistinctAreaTypes clusters.
6: Create numberOfDistinctAreaTypes lists of speeds by combining assignments and totalDataSet.
7: For each of the lists of speed compute the average speed.
8: $averageSpeeds \leftarrow$ Average speeds of each list of speeds found in last step
9: $covs \leftarrow$ All the CoVs from totalDataSet
10: $covAssignments, covCenters \leftarrow kmeans\,(covs, 2)$ ▷ // Create two clusters of CoVs, one indicating low CoV and the other indicating high. Low and High can be easily identified by looking at the covCenters
11: **return** $(centers, averageSpeeds, covCenters)$

The first step of algorithm 4 is to extract the information about WiFi. So in line 3, we take out the collection of WiFi densities from the whole data set. In line 4, we raise the wifi to the power of *wifiScalingFactor* which can be an input to the algorithm, in our case which is a constant value of 0.37 as discussed on section 4.4.4 in the discussion of the figure 4.14. Now from the scaled WiFi values, kmeans clustering is performed on the data in line 5 of the algorithm. The number of clusters for kmeans algorithm is the number of different types of area observed in the testing area which is an input to the model creation algorithm. In our case it was 3 as it was seen in field survey that volunteers informed that there are 3 types of distinct behaviors in the

city. One is highly active like market area, hospitals and school areas; one being residential areas and other one being lonely and similar to highways.

The second step is to accumulate the data from similar zones of the city. From the output of the kmeans cluster assignment we can figure out which of our examples are from which area of the city. So by segregating the data as per the cluster assignment of the step mentioned in line 5, we can accumulate the data set of similar surrounding areas. So when we only look at the speed from the total data set, we can create 3 collections(as in our case numberOfDistinctAreaTypes = 3) collection of speeds where each collection is speeds from the examples clustered in a single cluster. In line 6 of the algorithm, speeds from the same clusters are accumulated in a collection.

The third step, done in line 7 and 8 of 4, average speed of each cluster is computed and stored in a list named *averageSpeeds*. So, in subsequent steps, if we get the WiFi density of a certain segment, we can follow the same steps, i.e. raising the value to the power of *wifiScalingFactor*, fit the result to the result of kmeans clustering by computing its distance from the centers found in line 5, and then when we know the cluster it belongs, we can know the average speed of that cluster by indexing the same position of the list *averageSpeeds*.

The final step is to figure out the use of coefficient of variation(CoV). In this case, we do not depend on the individual cluster. As CoV directly corresponds to a stop-and-go movement pattern and that being one of the fundamental reason behind the perception of travelers of a segment being CHAOTIC or BUSY, we consider the CoVs as a whole. We cluster the whole set of CoV values in two clusters so that one would correspond to lower values, indicating smoother flow; and other corresponding to the higher values indicating the busier or more irregular flow. In lines 9 and 10 of the algorithm 4 first all the CoVs are extracted, and then they are clustered into 2 clusters using kmeans method. The results of the clustering are stored in variables *covAssignments* and *covCenters*.

Finally, the model is returned as the tuple of *centers*,*averageSpeeds* and *covCenters*. This model can be later used to estimate the traffic conditions from a given trace of gps. The algorithm of estimating the traffic state is a simple decision table based approach. The said algorithm is described in 5.

In the algorithm, first step is to understand the surroundings of the sample to be estimated. So in line 3, the wifi density associated with the example is scaled by raising it to the power of the scaling factor(0.37). Then in line 4, the result is compared with the cluster centers in the model to understand which cluster of the wifi densities is most similar with the sample. In line 5, the observed average speed in the cluster with which the sample is most similar to is extracted. Then the CoV of the sample is checked in line 6. The center of the cluster having higher CoV values are extracted in line 7 to make later computations easier. Finally, the speed of the sample is compared with the average speed found in line 5 and final decision about the state of the segment is taken from the decision table shown in Table 4.2.

Table 4.2: Decision Table for Traffic State Estimation

	Slower than the average speed	**Faster than the average speed**
High CoV	CHAOTIC	MED-CHAOTIC
Low CoV	MED-CHAOTIC	NONCHAOTIC

Relation of speed features and labels tagged by volunteers.

As described in 4.2, if a vehicle moves faster than the observed average speed in of the cluster corresponding to its surroundings, the state can be either MED-CHAOTIC or NONCHAOTIC. In such condition the CoV is checked. If the vehicle had a low CoV, then it is estimated to move through a NONCHAOTIC segment as the speed is not less than the average speed and the movement is also smooth. The exact opposite case is considered when

Algorithm 5 EstimateTrafficCondition

```
1: procedure EstimateTrafficCondition
2: Input: estimationModel, sampleToEstimate
   estimationModel: wifiCenters, wifiScalingFactor, averageSpeeds, covCenters, speedThreshold
   sampleToEstimate: meanSpeed, CoV, WiFiDensity
3:     scaledWiFi ← sampleToEstimate.WiFiDensity^estimationModel.wifiScalingFactor
4:     wifiZoneIndex ← fit(scaledWiFi, estimationModel.wifiCenters)
5:     observedAverageSpeed ← estimationModel.averageSpeeds[wifiZoneIndex]
6:     cov ← fit(sampleToEstimate.CoV, estimationModel.covCenters)
7:     highCoV ← Index of the higher value in estimationModel.covCenters
8:     if sampleToEstimate.meanSpeed < estimationModel.speedThreshold
   then
9:     return 'CHAOTIC'
10:    else if sampleToEstimate.meanSpeed < observedAverageSpeed then
11:        slow ← true
12:    else
13:        slow ← false
14:    if estimationModel.covCenters[cov] =max(estimationModel.covCenters)
   then
15:        highCov ← true
16:    else
17:        highCov ← false
18:    if slow then
19:        if highCov then
20:            return 'CHAOTIC'
21:        else
22:            return 'MED-CHAOTIC'
23:    else
24:        if highCov then
25:            return 'MED-CHAOTIC'
26:        else
27:            return 'NONCHAOTIC'
```

the speed is lower than the average speed of the cluster. Then the state can either be CHAOTIC or MED-CHAOTIC. If the vehicle has a steady speed - thus a low CoV, the state is estimated to be MED-CHAOTIC. This scenario is rare, and happens when a vehicle moves in a steady low speed. When the CoV is higher, the segment is estimated to be CHAOTIC as the vehicle is both moving in a slow speed and also having a variation in speed.

In the next chapter, the accuracy of this estimation scheme is tested against human tagged data and the performance is compared with the conventional machine learning algorithm.

Chapter 5

Results

In this chapter, we discuss the results of this research. First, we discuss the result and correctness of the first step of the unsupervised estimation strategy. Then, we talk about the outcomes of the methodology, the correctness evaluation strategy of the estimation model is discussed in the following section titled 'Evaluation Strategy'. The subsequent sections display the performance of the approach and a comparison between the outcomes of the method suggested in the thesis and other conventional machine learning techniques.

5.1 Collected Data

In this research, we collected over 800 Kilometers of trails from public buses in Durgapur, a suburban city in Eastern India. The trails were collected from 4 distinct routes of the city where the routes are having different characteristics. The data was processed and from that, we extracted the features as described in the previous chapter. From the data, over 1750 sample segments were found. The annotation of the volunteers were used to label the collected data samples. Among the 1768 samples 945 samples were found to be annotated properly. After discarding partial and incomplete annotations given by the users, we had a total of 370 Kilometers of labeled data that were used to evaluate the model.

5.2 Results of Clustering by WiFi

At the second step of the proposed methodology, we create clusters of segments in the city based on the WiFi density. The intuition behind this step is to accumulate segments that have similar amount of business or human activity happening around them. Now, WiFi density being a temporal feature, it helps to take the temporal behavior of road segments into consideration as well. The results of this clustering approach is displayed in the following table. Table 5.1 displays the percentage of highway segments, city road segments and market road segments falls into each cluster.

Table 5.1: Results of Clustering by WiFi Density

Cluster	**% of all highway segments lying in this cluster**	**% of all locality/township segments lying in this cluster**	**% of all market segments lying in this cluster**
Cluster 1	0	0	55.44
Cluster 2	90.87	22.78	6.9
Cluster 3	9.13	77.22	37.66

As seen in the table, all the highway segments are clustered in cluster 2 and 3. The major portion of the highway segments residing in cluster 2 indicates that cluster contains segments that had very less human activity happening around at the time of taking the data. It can also be seen that a fraction of city segments and a very small fraction of market segments belong to this cluster. The data from these two are either taken in early morning when less activity happens around a segment or in certain areas that have very few WiFi access points available around. Similarly, cluster 1 only consists of the major 55% of the market segments. Indicating that the market segments have such a unique behavior that is never observed in other types of segments. The cluster 3 contains the majority of city trails and also contains a small fraction of highway segments mainly which are junction to important

city roads. A large amount of market segments also fall in this cluster as in early morning and late night the market indeed behaves like city roads because less amount of activity happening around them.

5.3 Evaluation Strategy

In the methodology chapter we described the unsupervised approach to estimate the state of the traffic by building a model using low volume of traffic data. First we apply clustering mechanism on the total data set and try to evaluate the correctness of the clustering with respect to the labels provided by the volunteers. After that, we tested the same data set with the proposed approach and evaluated the performance of the model. After that the same data was classified using conventional machine learning algorithms. So first we discuss the improvement in the proposed unsupervised learning approach over the conventional clustering approaches. Next, by looking at the performance of classification algorithms on the same data set, we address the trade-off of quality of the estimation of traffic data and the overhead of collecting annotated data.

5.4 Performance of Unsupervised Estimation

First, all the labeled data was clustered using conventional KMeans clustering with 3 clusters. This is taken as the baseline to the proposed unsupervised learning algorithm. In the following table 5.2 we display the performances of unsupervised algorithms.

Table 5.2: Results of Unsupervised Traffic Estimations

	10 fold cross validation	**10% training (data used to build the model)**	**30% training (data used to build the model)**	**50% training (data used to build the model)**	**Using a training set uniformly sampled with all segments (9% training)**	**Baseline KMeans Clustering**
F1 Score (micro)	0.6508	0.6433	0.677	0.6813	**0.7195**	0.6391
F1 Score (CHAOTIC segments)	0.6194	0.6299	0.649	0.6381	**0.68**	0.6295
F1 Score (NON-CHAOTIC Segments)	0.7797	0.7678	0.7967	0.8049	**0.8389**	0.732
F1 Score (MED-CHAOTIC Segments)	0.4149	0.4081	0.4254	0.4762	0.4429	**0.6559**

Here it can be seen that the micro F1 score and F1 scores of CHAOTIC and NONCHAOTIC states of the total data set is better in all the cases of training. But in case of MED-CHAOTIC states the baseline algorithm performs better. But when the performance is better in the proposed approach when we consider the micro F1 score.

5.5 Comparison of results with Different Supervised Learning Technique

To evaluate the trade-off between collecting labeled data to build the model and drop of accuracy in the unsupervised model, we evaluated the accuracy of a supervised model developed from the labeled data set. Different supervised learning algorithms were tested upon the data using the WEKA[7] tool. The performances of the supervised learning algorithms can be found in the table 5.3.

Table 5.3: Results of supervised learning on labeled data

	10 fold cross validation	**10% training**	**30% training**	**50% training**	**Evenly Sampled Data (9%)**
F1 Score (Unsupervised)	0.6508	0.6433	0.677	0.6813	**0.7195**
F1 Score (MLP)	**0.742**	0.721	0.703	0.743	0.684
F1 Score (Logistic Regression)	0.737	**0.738**	**0.734**	**0.746**	0.672
F1 Score (J48 Decision tree)	0.730	0.691	0.684	0.691	0.685

From the table it can be observed that the supervised learning algorithms outperforms the proposed algorithm in most of the training strategies. It can also be observed that for the training data set in which the proposed method performed best as seen earlier, it outperforms the supervised learning methods as well.

5.6 Analysis of the results

As seen in the results, when the proposed method is compared with the baseline unsupervised clustering approach, in the best case an improvement of 12.6% can be observed in micro F1 score. When the proposed methodology is performed after random sampling of the collected data, with 10% of training, no significant improvement in micro F1 score was observed. But with 30% and 50% of the data used to build the model, the F1 score was observed to improve by 6% and 6.7% respectively.

When we look into the individual F1 scores of each state of traffic, it can be observed that F1 scores for CHAOTIC states stay the same in both the approaches except when the proposed methodology is tested in an evenly sampled dataset. In that case the F1 score is improved by 8%. In case of NON-CHAOTIC state, an improvement can be observed in all the testing plans, the best case being an improvement of 14.6%. In case of MED-CHAOTIC state, the performance was observed to drop significantly. However, due to the low number of samples observed in this state, the overall performance improved as discussed above.

When the results of the proposed method are compared with some of the conventional machine learning classifiers performing supervised learning, it was observed using Multi Layered Perceptron network and Logistic Regression classifier, the classification task is performed best.

The observations that could be made from the results are described as follows:

1. The decision boundary of the different traffic states are not really clear. As a result, the supervised learning algorithms also found it difficult classifying the data set. This case of having soft decision boundary are caused by mixing up annotations of different users as the annotated data are collected by volunteers. In our case, 7 different persons were collecting the data. As a result, the annotations were made according

to their own perception of business or chaos in road segments. In same segments, in similar movement pattern, two users can annotate the data differently following their perceptions.

2. When the segments that were mostly estimated wrongly was observed in detail, it was observed that in many of the cases, the incorrect estimation was due to the incorrect driving behavior of the bus. To maximize profit, the drivers tend to move slow in certain segments resulting the features to be similar to a busier segment. But the annotations does not contain these informations on the driving pattern, as a result these segments seemed to be incorrectly estimated.

3. The road condition play an important role in the movement patterns of vehicles. When the incorrectly estimated segments were investigated further, it was observed that certain road segments that are incorrectly estimated frequently, are due to the fact that the road condition is poor in those segments. Due to poor road conditions, the vehicles move in a slow speed in these segments throughout and appears similar to cases where bad driver behavior is observed.

4. Another irregularity was observed as some of the segments displayed high CoV but the volunteers marked the segment as NONCHAOTIC behavior. It was later observed that those segments has speed breakers in the middle and that caused the slowing down of the vehicles. This situation makes the CoV high and as a result these segments are estimated more chaotic than they actually are. Some relevant diagrams are displayed in the figure 5.1. In the diagram, 3 pairs of observations are displayed where in each case the black observation was labeled busier by the volunteers. In-set we can see the number of speed-breakers present in each segment in which CoV are displayed. In each case we can see the green observations are having more speed-breakers in them. Thus, in the case of black observations, the increase of CoV is not due to the speed-breakers, rather that is the actual busy road condition.

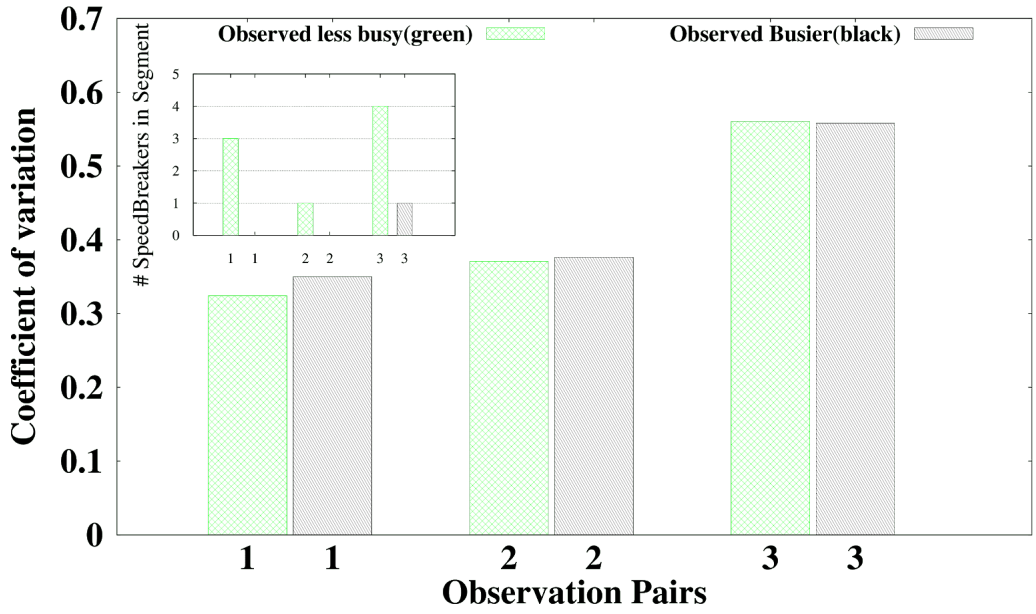


Figure 5.1: Impact of Speed Breakers on CoV

5.7 Visualization of Results from collected trails

To visualize the data collected and analyzed, we developed a web based tool that displays the retrieved information on the google-map api. In this section some of the screenshots are displayed and described.

In the map 5.2(a) a set of segments through a market area is displayed. The bus starts from the terminal point located at the top right section of the image. The state of the segments are displayed as: green color for NON-CHAOTIC, orange for MED-CHAOTIC and red for CHAOTIC. When we look into the next 5.2(b) image, the situation looks similar as a large distance after the initiation point stays CHAOTIC and MED-CHAOTIC. But in reality, these segments were found to be incorrectly estimated by the algorithm as we later discovered that at the initial phase at the area marked in orange and red, the bus moves slowly to get more passengers. In the next phase, the orange segment between the red and green segments, the bus moves normally but due to the bad road conditions, the speed in that segment always appears to be less than the expected speed. This results in incorrect estimation of the movement. In the last image 5.2(c), the segment marked in orange

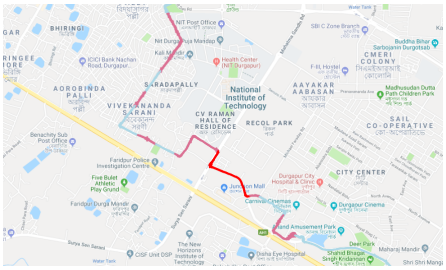
(a) (b)

(c)

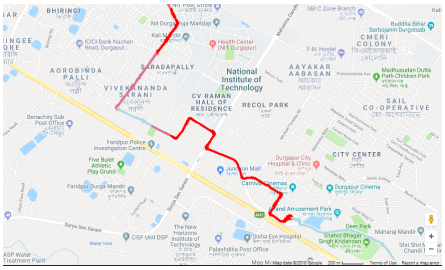
Figure 5.2: Visualization of Traffic conditions in different parts of the city

appears to be MED-CHAOTIC to the algorithm, but the volunteers always observed this segment to be NONCHAOTIC. The reason is that because of the presence of 4 speed-breakers in the segment. The sudden change of speed due to the speed breakers placed in the segment makes the movement pattern resemble a CHAOTIC movement.

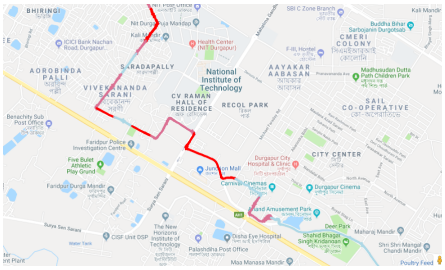
In figure 5.3 we visualize the WiFi density in segments. The light blue color in the images indicates low wifi density, purple means higher and red indicates very high wifi density. This gives an overall idea about the amount of human activity happening around the segments. In image 5.3(b), we can see market areas, areas beside college, shopping mall marked in red. The residential areas are also marked in purple and empty areas like the one that can be seen in the bottom right side of the figure is colored in blue. The other two images depicts the change of the WiFi density in different times of the day.



(a) WiFi Signature in Early Morning



(b) WiFi Signature in Late Day



(c) WiFi Signature in Night

Figure 5.3: Visualization of WiFi Density in different times of day

Chapter 6

Future Scope and Conclusion

6.1 Future scope

In this section we discuss the future scope of this research. From the analysis of the results, we saw that there are few parts on which the algorithm can be improved. The scope of research in this section is discussed as follows:

- It was seen that the proposed approach makes incorrect estimation of the traffic condition when the road condition is not good. So if we separately take the road condition into account, better estimation is possible. More available information on the road condition would result in enriched map services. There are some ongoing research on road profiling using smart-phone sensors. Though this researches focuses on identification of certain road anomalies, they can also be extended to perform an overall road profile and score a road segments according to its surface condition.

- It was also seen that the proposed approach make incorrect estimation due to the bad vehicle behavior for reasons like profit maximization. So finding a way to separate the actual chaotic situation from the fake one like slow movement to get more passengers can improve the estimation. The amount of honking in a segment can be looked into as a feature to separate these two incidents. In an actual busy traffic situation, more

honking can be expected than the situation where the vehicle moves slowly in spite of having a smooth road.

- The outcome of the approach can be used in inter vehicle communication to propagate traffic information without the need of a centralized server approach. Where each vehicle can follow the estimation algorithm and decide on the traffic condition and pass the message to other vehicles. Other vehicles on receipt of such messages can decide wisely which route to take.

- The existing map services can be enriched by extending the coverage and adding new informations to the map layers like the level of activity around with the information of WiFi density color coded into the map.

6.2 Conclusion

In this thesis we have completed a detailed analysis of mobile sensor data and mining traffic information from them. We defined some features that are derived from GPS and WiFi sensor data that can be available via crowdsourcing and used them in an unsupervised traffic estimation approach to estimate the traffic state. We validated the estimations against volunteer provided labels and evaluated the accuracy of the approach. In the best case we achieved 72% accuracy in estimating the traffic state, which improves the baseline clustering technique by 12.2%. We also compared the performance of the approach with some of the most popular supervised learning approach to confirm that we can achieve a graceful degradation in accuracy even in absence of labeled data. We also discussed the reason of low accuracy in estimation due to various facts like soft decision boundary, varying road surface conditions, incorrect driver behavior for profit optimization etcetera. In the final section we discussed the future scopes where the model can be used and how we can further improve the performance of the approach. We also suggested and end-to-end work-flow and a system architecture that can be used as a full system to record, accumulate analyze and visualize information. One android application was also developed to capture the

data we needed for this research. A database from the collected data has been developed using the android application with the help of volunteers that capture the movement pattern of public buses in 4 different routes in Durgapur city. Though this thesis does not address the common issues of crowd-sourcing like incentive strategies for data providers, it rather focuses on making interesting estimations of traffic state from the crowd-sourced data when low volume of data is available to model that can enrich the existing map services.

Bibliography

[1] Pooja P Dubey and Prashant Borkar. Review on techniques for traffic jam detection and congestion avoidance. In *Electronics and Communication Systems (ICECS), 2015 2nd International Conference on*, pages 434–440. IEEE, 2015.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[3] Samanway Ghatak. Roadnaturedetection codebase. `https://github.com/devil1993/RoadNatureDetection`, 2017.

[4] Samanway Ghatak. Trans-portal-android codebase. `https://github.com/devil1993/Trans-Portal-Android`, 2018.

[5] Samanway Ghatak. Trans-portal codebase. `https://github.com/devil1993/Trans-Portal`, 2018.

[6] Dimitar Goshev. *Road Traffic Monitoring with Location-aware Sound Sensors*. PhD thesis, 2012.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[8] Juan C Herrera, Daniel B Work, Ryan Herring, Xuegang Jeff Ban, Quinn Jacobson, and Alexandre M Bayen. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment.

Transportation Research Part C: Emerging Technologies, 18(4):568–583, 2010.

[9] Minh X Hoang, Yu Zheng, and Ambuj K Singh. Fccf: forecasting city-wide crowd flows based on big data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 6. ACM, 2016.

[10] Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, and Robert Morris. City-wide traffic flow estimation from a limited number of low-quality cameras. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):950–959, 2017.

[11] Vipin Jain, Ashlesh Sharma, and Lakshminarayanan Subramanian. Road traffic congestion in the developing world. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, page 11. ACM, 2012.

[12] Ratna Mandal, Nitin Agarwal, Projan Das, Shreyasi Pathak, Himansu Rathi, Subrata Nandi, and Sujoy Saha. A system for stoppage pattern extraction from public bus gps traces in developing regions. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 72–75. ACM, 2014.

[13] Ratna Mandal, Nitin Agarwal, Subrata Nandi, Projan Das, Aniket Anvit, Sunandini Sanyal, and Sujoy Saha. Stoppage pattern analysis of public bus gps traces in developing regions. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 276–279. IEEE, 2015.

[14] Ratna Mandal, Nitin Agarwal, Subrata Nandi, and Sujoy Saha. Personalised route-map generation using crowd sourced gps traces. In *Business and Information Management (ICBIM), 2014 2nd International Conference on*, pages 154–158. IEEE, 2014.

[15] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack:

accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pages 85–98. ACM, 2009.

[16] Le Hung Tran, Quoc Viet Hung Nguyen, Ngoc Hoan Do, and Zhixian Yan. Robust and hierarchical stop discovery in sparse and diverse trajectories. Technical report, 2011.

[17] Rohit Verma, Aviral Shrivastava, Bivas Mitra, Sujoy Saha, Niloy Ganguly, Subrata Nandi, and Sandip Chakraborty. Urbaneye: An outdoor localization system for public transport. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2016.

[18] Senzhang Wang, Lifang He, Leon Stenneth, Philip S Yu, and Zhoujun Li. Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 34. ACM, 2015.

[19] Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S Yu, Zhoujun Li, and Zhiqiu Huang. Computing urban traffic congestions by incorporating sparse gps probe data and social media data. *ACM Transactions on Information Systems (TOIS)*, 35(4):40, 2017.

[20] Daniel B Work, Olli-Pekka Tossavainen, Sébastien Blandin, Alexandre M Bayen, Tochukwu Iwuchukwu, and Kenneth Tracton. An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 5062–5068. IEEE, 2008.