USE OF CROWDSOURCED TRAVEL TIME DATA IN TRAFFIC ENGINEERING APPLICATIONS

Sakitha Pathmila Kumarage

(178042H)

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Civil Engineering

Department of Civil Engineering

University of Moratuwa
Sri Lanka

June 2018

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

S.P. Kumarage

Date 20/6/2018

The above candidate has carried out research for the Masters thesis under my supervision.

Dr. G.L.D.I. De Silva

Date 20/6/2018

Dedication

This thesis is dedicated to
Ven. Lankapura Saripuththa Thero,
my parents Lalith Kumarage and Sandhya Kumarage,
my sisters Senuri Kumarage and Osuri Kumarage
and Wanuji Abewickrema.

Acknowledgement

To my supervisor Dr. Dimantha De Silva for the constant encouragement and guidance at all stages of this research without whose invaluable commitment and dedication this research might not have been a reality. I would like to appreciate the valuable suggestions and guidance given by Prof. Saman Bandara, the senior professor at Transport Engineering Group, University of Moratuwa.

Special thank goes to the Senate Research Committee of University of Moratuwa for providing financial support under a short-term grant SRC/ST/2017/32 to facilitate a part of the research work. I like to thank Colombo Municipal Council for giving Infrared Traffic Logger (TIRTLE) to collect traffic data and Sri Lanka Police for granting permission to collect traffic flow data. Special thank goes to Google Inc for enabling Google Maps APIs and allowing free usage limits which made this study a success. I would like to acknowledge Director Postgraduate Studies, Dean Faculty of Graduate Studies, Head of Civil Engineering Department, Research Coordinator Department of Civil Engineering and all the academic staff members of the Department of Civil Engineering for providing the opportunity to complete this research.

I would like to extend my heartfelt gratitude for colleagues at Department of Civil engineering for supporting the research work. Special thank goes to Deshan Bandara, Chamith Dimantha, Chathura Vidanapathirana, Sasika Ranawaka, Gimhan Jayasiri, Milinda Amarasinghe for the support extended in probe vehicle data collection and license plate surveying. I would like to appreciate the support given by Ms. Gayani Rajapaksa to me in conducting the traffic flow estimation using machine learning principles. I would like to acknowledge the support given by Ms. Anuja Fernando, Lecturer Department of Transport and Logistic Studies in preparation of the thesis. I would like to mention the support given by non-academic staff at Department of Civil Engineering. Special thank goes to Ms. Melani Jayakody, the technical officer at Traffic Engineering Laboratory.

Finally, a special thank goes to my parents, my sisters and Wanuji for the constant support and motivation which encourage me to make this research a reality.

Abstract

Transport planning and management are required to provide quality and reliable transport service. Collection of data required for this purpose has been always a challenge and obtaining reliable traffic information will ensure proper planning and management of transport activities efficiently. There are many methods followed in travel time data collection by incorporating both fixed detectors such as traffic sensors and moving detectors such as probe vehicles. Collection of travel time data under both methods requires significantly high investment and technical expertise.

With the development of intelligent transport systems economical ways of traffic data collection based on advanced detection principles were introduced. communication and detection methodologies have faced up with the advancement of crowdsourced data mining allowing more readily extractable information on transport and mobility. this research focuses on development of an economical method for obtaining crowdsourced travel time data. Scalability to larger networks, consistent data collection and data collection at multiple locations simultaneously and ensuring the reliability are issues which are addressed.

Travel time data obtained from Google Distance Matrix API which is a processed information released based on crowdsourced mobile phone data, is used in this study to identify use of crowdsource travel time data and transport planning activities. A cloud-based data acquisition platform was prepared for the data collection by accessing the Google Distance Matrix API. The travel time observed from Google Distance Matrix API was verified with the travel time information collected by using GPS enabled probe vehicles. The results indicate that there is a significant agreement between the travel time given by Google Distance Matrix API and actually observe data for both short distance and long-distance trips.

Several applications are illustrated to understand use of travel time information obtained by the Google Distance Matrix API. A traffic flow estimation model based on machine learning principles is proposed for urban roads, A bottleneck identification method based on spatio temporal analysis of travel time and space mean speed variation is illustrated to analyse corridor traffic. Further evaluating the traffic impact of implementation of bus priority lanes and evaluating the traffic impact of implementation of reversible lanes were discussed with respect to Colombo Metropolitan Area.

With the successful implementation of this research it was identified that use of travel time information given by Google Distance Matrix API is a reliable consistent and economical method of collecting travel time information and it is recommended that the public authorities and organisations responsible in managing city traffic use this tool to improve traffic management plans and transport policies

Keywords: Transport Planning, Crowdsourced data, Google Distance Matrix API, Traffic Analysis, Traffic Flow Estimation, Bottleneck Identification

Table of Content

List of Figures

List of Tables

List of Abbreviations

Acronym	Definition
ANN	Artificial Neural Network
AoA	Angle of Arrival
API	Application Program Interface
AVI	Automatic Vehicle Identification
CMA	Colombo Metropolitan Area
CSV	Comma Separated Values
DGPS	Differential Global Positioning System
ETA	Estimated Time Of Arrival
FHWA	Federal Highway Administration
GPRS	General Packet Radio Service (GSM - UMTS)
GPS	Global Positioning System
GSM	Global System For Mobile Telecommunications
HCM2010	Highway Capacity Manual 2010
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IEEE	Institute Of Electrical And Electronic Engineers
IOS	iOS operating systems
IP	Internet Protocol
ITS	Intelligent Transport Systems
JSON	JavaScript Object Notation
KNN	K- Nearest Neighbour
LOS	Level Of Service
MAC	Media Access Control (IEEE 802)
MAE	Maximum Absolute Error
OS	Operating System
PCs	Personal Computers
PHP	Hypertext Preprocessor
RAM	Random Access Memory

RFID	Radio-Frequency Identification
RMSE	Root Mean Squared Error
RSS	Residual Sum Of Squares
RTMS	Remote Traffic Microwave Sensor
SSID	Service Set Identifier
STM	Spatio-Temporal Marix
SVR	Support Vector Regression
TDoA	Time Delay of Arrival
ToA	Time of arrival
TTFF	Time To First Fix
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

List of Appendices

1 Introduction

1.1 Context

The movement of humans, animals and goods from location to location is known as transportation. The history of transportation goes back to the beginning of the civilisation. Today with the development of human civilisation transportation has become a significant need for human beings to survive. With the development of transportation, people started to move longer distances within a short period of time. On this process different modes of transportation were introduced. In which the transportation on road has become the most popular as it was the most accessible and convenient method of movement for human beings. There is a significant development and expenditure in road infrastructure starting from the past century all over the world. It is expected the worldwide transport expenditure will count for 5.8 trillion US dollars between 2016- 2020(1).

There are many problems and issues arising with the development of transportation infrastructure and continuous demand for transportation. Cities, towns and major urban centres start to improve many problems of mobility year by year. Transport planning and management is a critical contributor which can provide solutions for major problems related to transportation. The transport planning ranges to a highly diversified area in which many types of research were conducted, and studies were published(2).

Transport planning and management are required to provide quality and reliable transport service. Congestion is a major problem faced by many transport systems which need to be managed and maintained at optimum levels to ensure an excellent service for customers.

Congestion occurs due to the demand exceeding the available capacity. Traffic congestion in urban areas is often present in cities with vital economic development. The economic development outcomes in more employment, more consumption, and more exchanges in goods and services, which result the requirement of mobility of people, goods and services more frequent and faster(3).

The management of traffic congestion, the introduction of policies, crowd handling are day to day activities faced by planners and managers in this transport systems. The availability of information for Transport planning is a major requirement to provide any solution. Therefore, many developments were initiated over the past several decades to obtain data required for transport planning. There are many parameters to identify in controlling traffic. Traffic flow, speed, capacity, safety, compatibility, accessibility are some parameters required in transport planning to ensure efficient planning(4).

Obtaining reliable traffic information has been a challenge. Installation of traffic sensors, surveillance systems and other detecting devices is a primary way of obtaining traffic parameters such as traffic flow and speed. In most of the developed countries, these systems are being installed, and data gathering platforms are developed. These detector systems has enabled to manage city traffic and use data in planning activities. The situation is different in developing countries. The investment of traffic detector systems is an expensive alternative for developing economy is to prioritise as there are many other significant requirements to be addressed(3).

The development of intelligent transport systems allows economical ways of traffic data collection-based on advanced detection principles. Communication and detection methodologies have phased up with the advancement of crowdsourced data mining, allowing more readily extractable information on transport and mobility. This development of intelligent transport systems has given a ray of hope for many developing countries to obtain traffic related data with a lesser investment cost(5).

The traffic data service conducted in Sri Lanka includes traffic flow data collection by manual accounting methods, spot speed surveys conducted using doppler speed detectors, license plate matching surveys, traffic flow data collection by traffic loggers. These surveys are conducted after understanding the requirement and conducted only on several days to fulfil the requirement of the intended study. There is no continuous data collection method currently being practised in Sri Lanka to collect continuous traffic information(6).

The study focuses on using travel time information provided by Google Distance Matrix API in transport planning activities. The information provided by the API is-based on advanced detection methods such as crowdsource data mining and big data handling.

Hence the study provides a convenient methodology to collect traffic information and proposes several applications in which data could be used for traffic management and transport planning activities. The methodology proposed by this study is an approach to use processed data in transport planning which would be an economical alternative to implement. The methods proposed in this study would be beneficial to transport-related public authorities which are operating under the objective of reducing traffic congestion in city limits.

1.2 Research Problem

Collection of data required for Transport planning and management has always been a challenge(4). The research looks at using crowdsourced travel time data obtained from Google Distance Matrix API in transport planning activities. The existing methods of obtaining travel time information and speed data are not consistent and reliable(7). Most of the methods used to obtain data involve manual surveys and could not be conducted continuously. The reliability of manual data collection methods are always questionable due to the involvement of human error and low sample rates(8).

The travel time data collection and space mean speed data collection are-based on either license plate recognition surveys or travel time surveys which use probe vehicle to collect travel time. The licence plate machine surveys are conducted for a shorter period using manual methods of transcription and matching(8). Therefore, the error is significantly high. The travel time surveys conducted using probe vehicles are only capable to collect travel time information on a single segment during the trip time. To study the travel time variation, it is required to operate many probe vehicles to collect travel time data(8). Therefore, the current methods of travel time data collection available in Sri Lanka cannot monitor the travel time variation of a road segment for a given time period.

Further, the travel time information collected by probe vehicles are not representative of the vehicle fleet. If a representative value for travel time of selected road segments is required, several probe vehicles have to be dispatched along the segment for several times of the day, and different vehicle types have to be used. This type of data collection is costly and has many practical issues in implementation(9).

In developed countries, it is possible to install traffic sensors and surveillance systems which can identify traffic flow data, travel time and many other parameters. The initial investment cost of such systems are very high, and it is very rare that the developing economy will implement such solutions(10).

Therefore there is a need for the development of a methodology of obtaining reliable and consistent travel time data simultaneously for a larger Road network. The methodology should be economical for developing countries to implement. By developing a such system, it is possible to conduct a continuous evaluation of traffic management and transport planning activities in a city.

1.3 Research Objectives

To address the research problems found in the above section following research objectives were developed.

- Understand the use of crowdsourced data and hybrid positioning systems in transportation engineering.
- Development of a crowdsourced data mining platform and data collection portal to collect travel time information.
- Verification of travel time information obtained from crowdsourced data with actual travel times observed with other methods of travel time data collection.
- Application of crowdsourced travel time information in transport engineering

1.4 Organization of the thesis

This thesis is compiled with 7 chapters and 5 appendices. Each chapter is focused on reaching the objectives of the research and each chapter is structured as given below;

Chapter 1: Introduction to the research topic. The objectives of the research were identified. Current methods have no capacity in covering travel times of large networks in real-time continuously. The available methods are significantly expensive to implement for a developing country. Therefore, there is a need to develop a method and verify to obtain reliable and consistent travel time data simultaneously for a large road network.

Chapter 2: This chapter includes the preliminary knowledge and literature which directs to the research topic. The chapter discusses the uses of travel time data, current methods of collecting travel time data such as manual recording, license plate surveys and video analysis-based methodologies. Further, this chapter illustrates on modern methods of collecting travel time data. Next, it discusses the use of probe vehicle techniques in travel time. Finally, this chapter discusses how crowdsourced data obtained from mobile phones could be utilized in estimating travel time, which leads to being the basic concept of travel time collection in this research.

Chapter 3 : This chapter discusses the use of crowdsourced data and Global Positioning Systems (GPS) in transport planning. The chapter initially provides an introduction to crowdsourced data. It discusses about how the location could be identified using GPS signals and GSM signals which could be traced buy GPS sensors and GSM modules which are embedded in smartphones. Next it discusses about involvement of Google location-based data sharing. On this regard it discusses how Google collect user location and the provision of Google privacy policy on anonymous data sharing. Finally, the chapter discloses the usability of Google travel time data obtained from Google Distance Matrix API in transport planning activities.

Chapter 4 : This chapter discusses developing a travel time data mining platform-based on the crowdsourced travel time data released by Google Distance Matrix API. Initially, it summarises the past literature which directs to use crowdsourced data as the next step in travel time data collection. Next, it discusses the web application and cloud server which were developed to collect travel time data by accessing Google Distance Matrix API. Further, it elaborates the procedure on data collection accessing the relevant APIs, collecting travel time data and storing. Moreover, the chapter discusses the provisions allowed by Google privacy policy on usage and sharing travel time data. In summary, this chapter develops the methodology on collecting Tower time data from Google Distance Matrix API.

Chapter 5 : This chapter works on the verification of Google travel time data. Initially, it discusses the cellular network infrastructure of Sri Lanka and its feasibility to use in verification. On this regard, the mobile usage, subscriptions and signal availability of Sri Lanka are evaluated. Next this chapter focuses on the verification methods. It presents a verification method-based on probe vehicle techniques. It illustrates the results obtained from the probe vehicle data collection and how it confirms with the travel time provided by Google. The evaluation was extended to short distance trips in peak-hour traffic, short distance trips in off-peak traffic and long-distance trips. The next verification method evaluates how Google travel time varies with different vehicle types. On this regard, a license paid survey was conducted, and results of the analysis are represented. Finally, it summarises and concludes the success of verification processes and conformity of Google travel time as an accurate estimate.

Chapter 6 : In this chapter, the applications of Google travel time data are discussed. As the first application, it presents a methodology for traffic flow estimation of urban roads-based on Google travel time data and machine learning principles. This study focused on using machine learning principles to estimate the traffic flow from speed data. A traffic flow dataset with Google travel time was collected and trained using the K-nearest neighbour clustering-based regression method.

The second application refers to Identification of road bottlenecks along corridors using Google travel time data. In this study segmental speed variation along an arterial road was studied. The variation of speed with time and space was illustrated in a spatiotemporal matrix and formation of bottlenecks (low-speed links) were identified with location and time of occurrence. Then the impact of the bottleneck on the traffic movement along the arterial road was evaluated.

The third application refers to using Google travel time data for evaluation of transport projects. In this study, the variation of travel time and average space mean speed was evaluated before and after implementation of the transport projects. Implementation of bus priority lanes in Colombo Metropolitan Area and implementation of reversible lanes in Colombo Metropolitan Area were two projects which were evaluated under this study.

Chapter 7: This chapter concludes the research work and suggests the future work which could be developed in the research area. Initially, it evaluates the methodology of data collection, the methodology of data verification and the applications of the travel time data obtained from Google API. Further, it presents best practices to follow in data collection and analysis with the experience gained on novel data collection method. Finally, it summarizes the conclusions and recommendations resulted from the study

.

2 Preliminaries

2.1 Definitions

In this study following definitions were considered throughout.

Travel time

The time necessary to traverse a route between any two points of interest. The travel time between two points were considered by including moving time in which the vehicle is moving at considerable speed (>5km/h), stop time at stop locations, delay time on traffic stream in which the vehicle is moving slowly (<5km/h) and waiting time for transverse if the mode is public transport.

Time-mean speed

The arithmetic average speed of all vehicles for a specified period of time.

$$Time\ Mean\ Speed = V_{TMS} = \frac{Sum\ of\ Speed\ of\ vehicles}{Number\ of\ vehicles} = \frac{\sum_1^n v_i}{n}$$

Space-mean speed

The average speed of vehicles travelling a given segment of the roadway during a specified period of time

$$Space\ Mean\ Speed = V_{SMS} = \frac{Distance\ Travelled}{Average\ Travel\ time} = \frac{L}{\frac{\sum_1^n t_i}{n}}$$

2.2 Uses of travel time data

Travel time is the most fundamental source of information about travelling between an origin and a destination. Hence travel time of transit and delay studies are being considered by researchers since early times of transport research. Travel time is understandable parameter when compared to other parameters used to evaluate traffic on roads such as traffic flow speed etc. Hence due to the simple understandability of travel time parameter has enabled it is being used widely in transportation engineering(8).

Travel time is a key indicator of efficiency in planning and designing of transport systems. Many government organisations use travel time for developing transportation policies and programs. A classic example would be travel time based policy developed by the Boston Region Metropolitan Planning Organization(11). In the identification of performance need studies and assessments, travel time is widely used. The travel time index is being used on several occasions to rank and prioritize transportation improvement projects for funding(11). Travel time is used in calibration of demand forecasting models, and network travel time is considered as an impedance factor when defining a generalised cost for transport models. Further, in economic analysis travel time is used to identify the utility gain by travelling on the road(8). In defining accessibility of a road network, travel time is a key indicator which identifies the accessibility between nodes of the transport network. Moreover, travel time could be used in estimating parameters such as emissions fuel consumption and particulate matter concentration of roads environments(2).

In operational activities, travel time could be used in many ways. Developing a historical travel time database of a considered network will enable to evaluate the daily performance of the road network(8). Further knowing travel time along road network will enable road users to optimise their travel and select the best path to reach a destination. More precisely travel time could act as a traveller information parameter(8). Mobile applications such as Google maps use this technique to suggest the best path between an origin and destination by selecting the fastest route in real-time. Thus, travel time is helpful in daily commuting activities and navigation. Travel time could be used in incident detection. It could be concluded that the road segment is underperforming due to an incident if the observed travel time is higher than the average travel time.(8) Real-time acquisition of travel time data could open up many research areas which enable real-time freeway and arterial street traffic control using real-time travel time data. Thus, obtaining travel time has many operational benefits in controlling road traffic. In Operation of public transport schedules, the travel time of trips on public transport route is a mandatory parameter to control(8).

On the aspects of evaluation, travel time could be used as a performance measuring parameter. Travel time along the road network could be used in evaluating congestion management systems and in identifying congestion trends(12). By studying the variation of travel time along the arterial road, congested locations along the arterial road such as bottlenecks could be identified. In an analysis of network robustness, many researchers have used travel time as an indicator(13).

2.2.1 Current methods of collecting travel time data

Many methods of collecting travel time data were developed in research due to the diverse applicability and usage of travel time data. There are many guidelines developed to collet travel time data to keep the proper practice and quality of data collection. By referring to literature, four major methods of collecting travel time data could be identified as follows;

1. Test vehicle techniques
2. License plate matching techniques
3. ITS-based techniques
4. Probe vehicle techniques.

Following paragraphs will describe each technique in briefly.

2.2.1.1 Test vehicle techniques

Test vehicle technique is a prevalent method of collecting travel time data. It is also referred to as floating car data collection method. In this method, a vehicle is travelled between origin and destination for the sole purpose of calculating the travel time. In this method, the driver has to take specific care in driving to control the speed of the vehicle and act as an average moving vehicle in a traffic stream. A second person records the time while the vehicle is being moved. Person who records the travel time record the time to reach several checkpoints which were marked before the travel time survey. As an improvement to the manual method of collecting data automated methods could be introduced to increase the accuracy of data collection. Although it requires low initial cost and low skill level to collect travel time using manual test vehicle techniques, there are many disadvantages. The driver has to take proper care in his/her driving pattern so that the test vehicle represents the whole vehicle fleet moving in the traffic. If it is required to take large samples, then the cost is very high as several vehicles has to be implemented. Therefore,in this method, a limited number of samples are always collected. Further, this method could only be used in collecting travel time of a freeway or an expressway. Simultaneous observation of travel time of an urban network is not possible as it is not possible to collect data simultaneously. As an improvement to test vehicle technique, it is possible to introduce electronic distance measuring instruments or GPS technology which could increase the accuracy of data collection. However, it cannot give solutions to increase the number of samples collected simultaneously unless otherwise the data collection is carried out using many vehicles which is not practical in most of the time(8).

2.2.1.2 License plate matching techniques

In this method license plate of vehicles were recorded at adjacent checkpoints of a given Road segment. Then the license plates were matched in between each checkpoint. The arrival times at the checkpoints in which license plate was recorded are compared to derive at the travel time between the checkpoints. This method could be implemented in many ways such as manual recording, computer recording or video analysis of a live video stream. In the manual method of data collection, a person is located at each checkpoint to record the license plates and arrival times. Although this method requires low skill level and low cost, the accuracy and consistency are always questioned. Further, the manual methods have a very high processing time as the matching also to be done by humans. As an improvement to this manual method, a software solution for recording number plates and travel time were introduced which increased the accuracy. Video record-based manual transcription methods were introduced afterwards in which a video of the traffic stream is recorded, and license plates were transcripted manually. This item has increased the sample collection rate. Moreover, computer-aided video analysis solutions were introduced in later. By using video analysis methods, the number plate was read by image recognition techniques and time was recorded with the recognition of license plate. Although the sample collection rate has increased the accuracy was a challenge, due to the efficiency of image recognition algorithms. High initial cost is expected in the implementation of such systems. Although license plate matching technique could be utilized along with a freeway or expressway, it is a cumbersome effort to collect travel time of an urban network using this technique as it requires many video analysis cameras located at several locations of the network and conduct video analysis continuously which requires high computer processing(8).

2.2.1.3 ITS-based techniques

With the advancement of sensing technology and computational processing power, many intelligent transport systems(ITS) were introduced to collect travel time data. In most of the urban road networks, many countries utilise sensors such as inductance loop, Bluetooth sensors motion sensors and pressure sensors. It is possible to identify different vehicles with wheel axle loads with this sensing technology. By implementing sensors at several locations of a road segment, it is possible to identify the travel time on the road segment by considering the time mean speed and wheel axle loading.

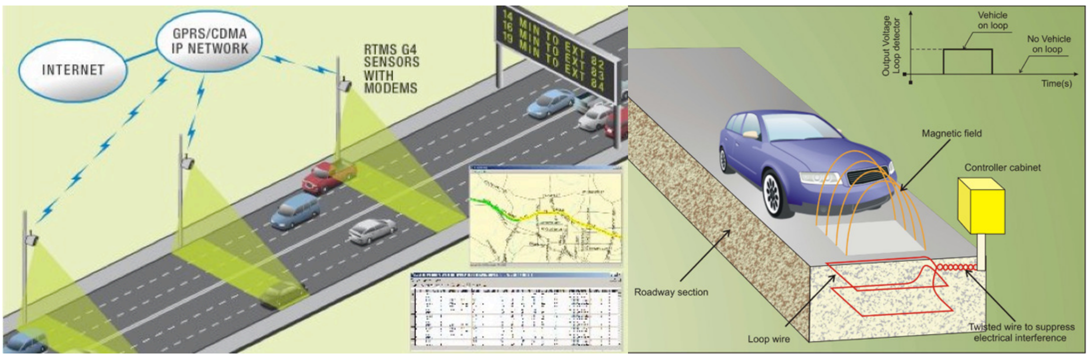


Figure 1 : Left: Microwave vehicle detectors, Right: Inductor loop vehicle detectors
Ref : US Depatment of Transport

http://www.itscosts.its.dot.gov/ITS/benecost.nsf/DisplayRUCByUnitCostElementUnadjusted?ReadForm&UnitCostElement=Remote+Traffi c+Microwave+Sensor+on+Corridor+&Subsystem=Roadside+Detection+(RS-D)

Figure 1(a) shows how traffic sensors equipped with microwave traffic detectors could be used in traffic monitoring. The traffic sensors are connected with an intranet which can identify the progressing traffic condition along the road. Figure 1(b) shows the mechanism of detecting vehicles from loop detectors. The detectors can identify an approaching vehicle or a passing vehicle. The detectors induct a current through the embedded coils which create a magnetic field on the ground. Vehicles are detected by the eddy currents formed when the magnetic field is interfered by a moving vehicle.

Further, video-based surveillance systems were installed in many road networks to control traffic. The surveillance cameras could be utilized in obtaining travel time data. Video analysis using image processing algorithms should be utilized in this process. The advantage of these ITS-based techniques is they could be implemented to a larger network and obtain travel time simultaneously. Further, these methods could provide a representative sample on travel time data collection as it monitors all the vehicles which are moving at its vicinity(8).

Table 1 : Cost comparison of different sensor systems

Sensor Type	Production	Year	Unit Cost	Lifetime Years
Microwave Traffic Detector	United States	2008	$9,638	10
Remote Traffic Microwave Sensor on Corridor	United States	2004	$20,900	15
Sensors (RTMS) on Freeways	United States	2003	$11,500	10
Inductive Loop Surveillance on Corridor	United States	2006	$17,400	15

Table 1 provides a detailed costing on the available sensors used in traffic monitoring. The cost is exemplary high as there is a high component of installation and calibration works before implementing such a sensor network. Installing these sensors in a city which can afford, will enable to collect quality and highly accurate data on the traffic condition in real-time. It is always a question whether high investment on such systems a feasible alternative for developing economies(8).

Probe vehicle techniques

In this method, the vehicle is instrumented with the remote sensing device and driven in the traffic stream. Travel time is collected while the vehicle being driven. Probe vehicles can be personal vehicles public transit vehicles or commercial vehicles. Probe vehicles need not be given for the exact purpose of collecting travel time. The vehicle is fixed with an electronic transponder or receiver to communicate with a tracker constantly. In the tracking purpose, there are many navigation techniques are being used(8).

Signpost-Based Automatic Vehicle Location (AVL) – In this method, a transponder in the probe vehicle communicate with transmitters mounted on signpost structures along the roadside(8).

Automatic Vehicle Identification (AVI) – In this method instead of transponders, electronic tags are fixed to probe vehicles. These tags communicate with roadside transceivers and identify unique vehicles via a vehicle identification code and collect travel times between transceivers(8).

Ground-Based Radio Navigation – In this method, Data is collected by communication between probe vehicles and a radio tower infrastructure. Using radio communication signals. This method is similar to military transmission systems used in small-scale communication(8).

Cellular Geo-location – In this method, GSM/GPRS data transmission is tracked to identify activity patterns. This technology can collect travel time data by tracking cellular telephone call transmissions. In this method, a telephone could be located by triangulation with closer signal posts. However, it is arguable on the accuracy of this method(8).

Global Positioning System (GPS) – In this method, the probe vehicles are equipped with GPS transponders. The GPS devices communicate two-way and receive signals from earth-orbiting satellites. The positional information determined from the GPS signals is transmitted to a control centre to display real-time position of the probe vehicles. Public bus fleets, taxi fleets and commercial trucks which are utilised with GPS trackers are used in many research works to identify travel time of urban networks(8).

In all these navigation systems, the location of the probe vehicle is shared with the time. Thus, by analysing the vehicle location trajectory with time, the travel time of a road segment could be obtained. Although probe vehicle techniques provide higher accuracy and consistent method of travel time collection which could be extended to a large urban network simultaneously, it is highly questionable whether authorities or government agencies could priorities such systems due to the high initial cost and operational cost(8).

The Federal Highway Administration US has conducted a comparison of the existing probe vehicle techniques and Table 2 illustrates the comparison. According to the analysis, Ground-Based Radio Navigation and Global Position System(GPS) were identified as low-cost alternatives in implementation. The GPS could be identified as the most optimum solution to implement. Although it is sufficient to implement GPS systems in collecting traffic data, to get a representative sample of the vehicle fleet, the majority of the vehicle should be implemented with GPS devices. This may arise many privacy issues in tracking vehicles and cause safety problems to the general public. Many types of research have been conducted using GPS fleet data obtained from public transport agencies, taxi services and similar transport-oriented companies which operate vehicle fleets. It is questionable whether travel time data or any other traffic data obtained from these GPS fleets could give reliable estimations on traffic condition as vehicles with GPS devices could not represent the whole vehicle fleet moving along the road(8).

Table 2 : Comparison of different probe vehicle techniques ref : The Federal Highway Administration US

Technique	**Costs**				**Data accuracy**	**Constraints**	**Driver requirements**
	Capital	**Installation**	**data collection**	**Data reduction**			
Signpost-based Automatic Vehicle Location (AVL)	High	High	Low	High	Low	No. Of signpost sites, transit routes, and probes	None – uses transit vehicles
Automatic Vehicle Identification (AVI)	high	high	low	low	High	No. Of antenna sites and tag distribution	Required – but can use toll patrons
Ground-based radio navigation	Low	Low	Low	Low	Moderate	No. Of probe and size of the service area	Required
Cellular Geolocation	High	High	Low	Moderate	Low	No. Of sell users and cell towers	None – uses current cellular users
Global Positioning System (GPS)	Low	Low	Low	Moderate	High	No. Of probs	Required – but can also use currently instrumented vehicles

3 Use of crowdsourced hybrid positioning in transport planning

3.1 Introduction to crowdsourced data

Crowdsourcing is a distributed problem-solving model in which undefined size of people engaged in achieving a common goal(5). It involves exertion of combined intelligence, knowledge or experience of a group of people to answer a question, solve a problem or manage a process. Although crowdsourcing is a new term in the technical arena, crowdsourcing is being practised since ancient times. In building large structures or fighting war many people work for a common objective. The same concept is applied nowadays on technical platforms to collect data, perform tasks and design projects. With the introduction of internet and connected devices and masses, the crowdsourcing potential has increased exponentially. The handheld mobile devices such as smartphones, tablet PCs, smart wearables have increased the involvement of people in generating a huge amount of data(4).

When looked at the generation of crowdsourced data, it could be categorized as active crowdsourcing and passive crowdsourcing. In active crowdsourcing, people who get involved in crowdsourcing are aware of the objectives, and they work towards the project and involve actively as agents. In situations such as surveys, polls, contests, data-entry and protests people involve directly and actively towards achieving the objective of the project. In passive crowdsourcing, the activity of people is being traced as data and people are not actively taking part in the project(14). Analyzing Twitter data, using web traffic information, using public blog comments made by people and using monetary transaction data are situations in which passive crowdsourcing is prominent. The user is not directly focused or involved in crowdsourcing, but their activity provides information and data for passive crowdsourcing. In the proper practice of crowdsourcing, people should be informed that their data will be used as crowdsourced data for analysis and quality improvement. It is a mandatory fact that crowdsourcing agent adheres to laws of protecting the user privacy(5).

3.2 Use of crowdsourced data in transport engineering

The crowdsourced data is an optimum source of information gathering for transport networks. Many people use transport networks for daily commuting purposes. The utilisation of transport networks by users is a highly complex scenario to model concerning time and space(2). For example, the traffic flow in a large network may vary due to different factors such as speed, capacity, human error, weather, special events etc. In such diversified cases controlling traffic and optimising the traffic network in a short period of time is not easy without gathering data required for the analysis(15). Crowdsourced data could be an optimum approach in such a complex situation to gather information from the undefined amount of people who are actively using the traffic network in real-time. With crowdsourced data gathering, it enables to gather information instantaneously on both temporal and spatial parameters(5).

Social media networks are very attractive mode of collecting crowdsourced data. There were many researches which were conducted to use Twitter and Foursquare® data in transport planning and land use mapping. According to Vanessa and Enrique, Individuals generate a vast amount of geolocated content by using mobile social media and search applications. They identify Twitter as a mobility sensor to get the information for urban planning applications. According to authors, twitter activity data could be used for analysing crowd behaviour, land use detection and traffic identification successfully. The methodology was verified in Manhattan(US) London(UK) and Madrid(Spain)(16). Jinghui et al. Suggest road traffic prediction method-based on Twitter data. The authors compare simultaneously gathered data from 943 loop detectors and tweets generated within a box area which covers the 943 loop detectors. The authors observed a very high correlation between traffic intensity and regeneration intensity which suggest that people tend to post traffic-related content in social media and that could be used as information for traffic identification(17). McHugh Suggests a similar approach of using real-time traffic to eat to identify adverse traffic incidents. The author identifies the correlation between traffic-related tweets and adverse traffic conditions and uses twitter data and weather records to predict expected travel time. The author concludes that the methodology is successful and it was verified to Dublin City(18).

Spyratos et al. Suggest using Foursquare® data for estimating nonresidential blocks and land use mapping purposes. The authors used Foursquare®®places application programming interface(API) to gather places details of Amsterdam City. 37,482 Foursquare® places used to identify the land use map of the city. The study was successful in developing a land use map to the area which could be easily updated, and it could be considered as a very low-cost alternative compared to the conventional methods of developing land use maps(19). Sun and Li Suggest a method of identifying travel and activity patterns of users-based on location-based social network data. In this study, the authors use Foursquare® check-ins to investigate how gender influence in travel and activity patterns. The authors use consecutive check-ins made by the same user to produce daily trajectories. They were able to produce 13,162 trajectories of 1835 users over a period of one month in New York City(20).

In addition to using social media networks, there are many commercial platforms developed to gather crowdsourced data for analysis in transport engineering. One of the viral traffic data collecting platform is Waze®. It was developed under the concept of crowdsourcing traffic data. Currently, the Waze® app operates on major mobile platforms and has a user base of over 100 million. The mobile application act as a social network of drivers moving closer to a user neighbourhood. In operation, users within the neighbourhood share traffic information such as road closures, traffic jams, accidents, the location of speed cameras, speed and police traps and other traffic hazards. Further, Waze® app collects the travel time and speed of users anonymously when they use Waze® app. All these data are transmitted to the Waze® servers. Based on the crowdsourced information collected from its users the app provides routing and real-time traffic updates on the app(21).

Currently, only 13 countries have fully implemented Waze® app, and many current users are associated with those countries. The limitation of Waze® app is, it needs an initial user base to create maps and communicate traffic information to make the app an useful for many. Although the app could be used anywhere in the world and also available to download, it is not an efficient form of information as the maps are incomplete and do not exist a strong user base in each country. Concerning Sri Lanka, the situation is same where there is no significant amount of Waze® users(21). If the

Waze® app is used in the Sinhala language, there is a possibility of app becoming an attractive traffic information portal for Sri Lanka.

Similar to the concept of Waze® on crowdsourcing traffic information INRIX® is a service which provides real-time traffic information to users. The speciality of this platform is it does not limit only for mobile phones; It collects information from mobile phones probe vehicles such as connected cars, trucks delivery vans and other forms of vehicles which are fixed with GPS sensors. With all these vehicles INRIX® collects real-time anonymous data on roadway speeds and vehicle counts. With the user base of over 300 million, it is possible to give reliable information about routing and congestion to its uses. Currently, the platform is enabled in 65 countries. Due to its large-scale operation and user base, it could be identified as the largest company specialised in traffic data. INRIX® introduced Global traffic scorecard in 2016 by analysing 1064 cities in 38 countries(22).

HERE® is another search service which provides real-time traffic data. This is in operation since 1985, targeting to provide map services and navigation services. Currently, It uses probe vehicle data and embedded city traffic sensors to provide traffic information. The service provides traffic flow information to over 65 countries and incident traffic information to 31 countries. The advanced algorithms developed by HERE® company on GPS-based Technology has enabled to report traffic information for multiple lanes on the road and congested junctions(23).

TrafficSense® by Cellint® is another Similar application which uses crowdsourced data. It collects data from active phones in a mobile network anonymously. This is a passive crowdsourcing method in which TrafficSense® has to partner up with mobile service providers. Instead of cellular tower triangulation in finding the location, the TrafficSense® use cellular signalling patterns of routes in finding the location of mobile phones. Initially, an instrumented vehicle is driven along the roads in which traffic data is required. Then the signalling pattern along the route is identified, and a GPS coordinate is matched with cellular signal data. When a mobile phone emits a signal, the location is identified by comparing to the reference signal pattern created along the route. Hence the location of the mobile phone is identified by reference to the cellular signalling patterns of the route. This method increases the accuracy of cellular triangulation method and gives reliable information in real-time. With this technology, TrafficSense® can identify vehicles which are moving on a particular road and lanes separately(24).

Telenav® Is a solution which connects car sensors and provides location-based services. It provides intelligent navigation and advanced driver assistance solutions by connecting car sensors moving in a large network. The car producers have to embed the sensors and connect them with Telenav®. RoadSense is a product introduced by TeleNav® which is an early warning system for congestion situations-based on the information collected from other car sensors. SmartHorizon is another such product to provide the best route by using car sensors in real-time. For this solution to become an optimum operating condition autonomous vehicle has to be introduced to the vehicle fleet. Thus, the solution is much futuristic(14).

TrafficCarma is a mobile application developed by TrafficCast® International Inc. The speciality of this application is; it provides traffic information for daily commuters by referring to crowdsource traffic data and other advanced information such as traffic sensor data, speed data, road construction information and camera imaging. The use of static and dynamic data fusion of different sources to derive at optimum results is an advanced feature of this solution compared to previous solutions(14).

In addition to general traffic information gathering by crowdsourcing information, it is possible to gather public transport information by crowdsourcing. Carnegie Mellon University has developed Tiramisu Transit app to crowdsource location information of public transport vehicles. The users can transmit the location of the bus when the user is travelling on public transport. Further bus fullness and other rider experience information could be shared using the application. The mobile app was successfully implemented in Pittsburgh enabling to save investing money on implementing automatic vehicle location systems(25).

Further enhancing the crowdsource information gathering the Boston City Council has implemented a dedicated mobile app to collect information about potholes along roads. The application uses an accelerometer, gyroscope and GPS sensors to track bump events when a user is travelling in a vehicle. Algorithms were developed to identify likely potholes by analyzing the sum of bump events. The user can submit the bumps recorded during the trip for authority to consider as maintenance data(14).

Minnesota Department of traffic has implemented a crowdsourced bicycle network map for users to share bicycle tracks in the city. The users can follow bicycle routes which are submitted by other users or add new bicycle routes. Further users can report issues related to bicycle tracks to the Department of Traffic via this application. The project has successfully attracted many users as the city of Minneapolis, and Saint Paul has many bicycle users(14).

As identified in above cases, the use of crowdsourced data is an optimum solution to gather many spatiotemporal data required in transport engineering. This study is concerned on using crowdsourced data in transportation planning activities. On this aspect, all above solutions could be used. This study especially focuses on using Google travel time data in transport planning. Many limitations found in earlier cases could be addressed when Google travel time data is used. Google APIs could be identified as a convenient and reliable source of information when evaluated the available crowdsourcing platforms gather in data for transport planning. Google does not require embedded traffic sensors unlike HERE® or does not need dedicated probe vehicles such as INRIX®. The user base of Google is smartphone users who use Google services. Thus, the scalability of Google platform is comparatively very high. Further, Unlike Waze® application, the Google crowdsourcing does not need active user involvement in generating data. The Google uses passive modes of collecting user location. Hence it enables to scale up Google to any environment without having an initial user base. Google uses a hybrid positioning system by incorporating all the communication devices and communication infrastructures such as GPS signals, wi-fi and cellular GSM. Therefore, the location accuracy is higher when compared with services such as Traffic Sense which need the involvement of local mobile service providers. By comparing to all the factors which can cause limitations to the study use of Google Traffic data can eliminate many limitations and enhance the usability of the service in developing countries.

3.3 Positioning systems

3.3.1 GPS-based location identification

The position of objects was identified with reference location. Defining a location of an object by reference to other objects has been practised since ancient time. Following the same concept, the Global Positioning System(GPS) was introduced in the 1970s in which a coordinate system is used to define a location of the object. GPS receiver can find the location of anywhere in the world when it has a direct signal from four or more GPS satellites. The location of the GPS receiver is determined based on the time difference to receive signals from three GPS satellites which are precisely located. A fourth satellite is used to calibrate the internal clock of the GPS receiver which enables to determine time delays of received signals accurately. Figure 2 shows how location is identified using GPS satellites(26).

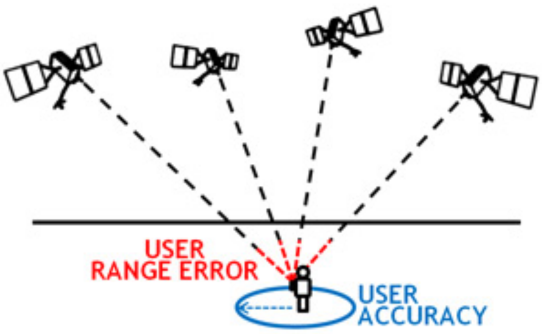


Figure 2 : Location identification from GPS

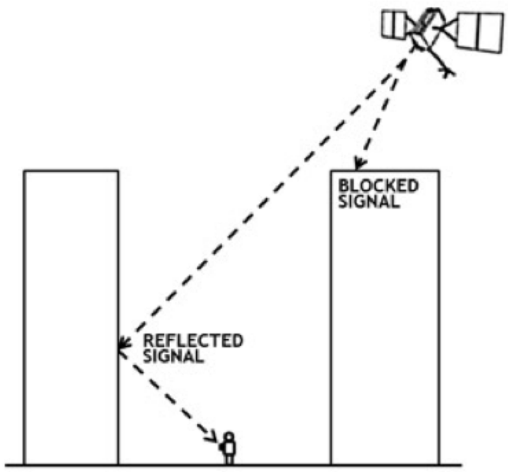


Figure 3 :Reflection of GPS signals

The signal strength is an important, paramount factor in location accuracy. The accuracy of GPS location varies between 10 to 20 meters in most cases. It is tough to find the GPS location at areas with weak GPS signals. The signal weakening happens due to mountains or due to interference from buildings. As in Figure 3, when there is a reflection in the GPS signal due to interference such as building a mountain, the accuracy of the identified location is very low. Therefore, finding the location with weak GPS signal has been a challenge. Hence, they were many types of research being carried out to obtain an accurate GPS location in an area with weak GPS signal.

The initial research of Airforce was able to find that using only GPS signals in location identification can cause specific errors due to the availability of signal(26). Augmentation methods to increase the accuracy were developed. The Differential Global Positioning System(DGPS) is an attempt taken to increase the accuracy. DGPS Uses a fixed network of ground-based reference points to broadcast the location via GPS. Location of the GPS receiver is corrected by referring to the signal sent by the ground-based reference point. By using this method, the accuracy of the location identification was increased up to 1.5 m. Wide Area Augmentation System(WAAS), European Geostationary Navigation Overlay Service(EGNOS) are examples for DGPS systems(26).

Assisted GPS is an improved version of Differential GPS in which mobile network transferring stations were used as ground-based reference points. This could be considered as the initial development of hybrid positioning systems in which cellular network tower locations were used instead of ground fixed reference points(27).

3.3.2 Cellular-tower trilateration

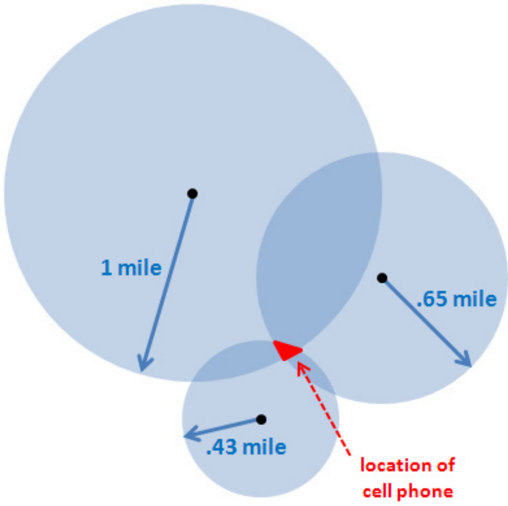


Figure 4 : Cellular tower trilateration

In addition to GPS-based location identification, there are other methods of location identification. Cellular trilateration is such a common technique used in location identification. In this method, the precise location of cell towers is being used. The location is determined in relative to three fixed points in space (28).

In cellular tower trilateration the distance to the device from each cell tower is determined. When the distance from each tower is known it is possible to identify a circle with radius equal to the distance. It is trivial that the location of the device should fall on the circumference. By using three such circles as shown in Figure 4, it identifies the relative area of which the device is possible to be. If the cell towers are closely located, and the circles intersect each other and form a common intersection area, then the device could be in any location within the intersection area as shown in Figure 4. Therefore, the accuracy of cellular tower trilateration is always challenged. To increase the accuracy of trilateration, it is required to determine the distance between the station and the device precisely. There are several methods used in measuring distance.

Time of arrival(ToA) - In this method, a signal is sent from station to the mobile device and back again. The speed of the signal is known and determines the time difference between sending and receiving the signal at the cell tower. Then The distance to the device could be identified (29).

Received signal strength(RSS) - In this method, the loss of signal in propagation is used to compute the distance the loss of signal power could be used to identify the distance by using a free space path loss equation (29).

The angle of Arrival(AoA)- In this method, the angle of the arriving signal least used to identify the geometry and the position of the device. To carry out a proper location identification by Angle of Arrival there should be at least two stations for which mobile device is connected and a third station is required for verification (29).

3.3.3 WIFI Positioning System

The use of Wi-Fi technology in position identification is an economical method with the abundant availability of Wi-Fi access points. When a city is considered, Wi-Fi access points could be found in almost every building. Therefore, using Wi-Fi access point to locate a mobile device is an optimum method to utilise existing infrastructure. It is possible to locate the position of any mobile device which has Wi-Fi enabled using this technology. Further, position identification using Wi-Fi technology could be performed both indoors and outdoors. Thus, the barrier of location identification within buildings using GPS signals is not a problem when Wi-Fi positioning methods are used(28).

Wi-Fi uses electromagnetic waves to transmit data over the space. The signal strength and signal delay are the major parameters used in Wi-Fi positioning systems to identify the location. Similar to Cellular tower trilateration Wi-Fi systems use Received Signal Strength(RSS), the Time Delay of Arrival(TDoA) and Angle of Arrival(AoA) in position identification. Fingerprint Positioning is another method of location identification especially used in Wi-Fi positioning systems. It identifies the position by matching the location patterns in reference to a previously created database of signal patterns. The accuracy of Wi-Fi Positioning is significantly high in the indoor environment while it is low in outdoor environments(28).

3.3.4 Hybrid positioning systems

Hybrid positioning is diffusion of different methods of geolocation together and optimised positioning methods by increasing the accuracy and reduce the delay time of determining the location. In literature, there are many hybrid positioning methods by incorporating GPS signals, cellular signals, Wi-Fi signals, television signals, Radio-frequency identification(RFID) signals. The concept is to increase the accuracy by using two or more methods which can give better results than using an individual positioning system(27)–(30).

This study is concerned on using a hybrid positioning system-based on GPS signals Wi-Fi signals and cellular signals. There are many methods of hybrid positioning techniques by using GPS Wi-Fi and GSM or cellular signals. The hybrid positioning system created by Skyhook combines GPS cell tower triangulation and Wi-Fi Positioning. In this method, an initial position is identified using known Wi-Fi access points while cellular triangulation and GPS signals are used to increase the accuracy. Skyhawk has a database of Wi-Fi access points with its location, and the changing locations of Wi-Fi access points are being tracked with reference to GPS signals. With this method, it was possible to reduce The Time to First Fix(TTFF) up to one to two seconds. It takes approximately 30 seconds to find the location using only GPS signals(30).

A similar solution is provided by Navizon which use Wi-Fi and cell tower signals to identify the position with no delay. In this concept the exact location of Wi-Fi access points and cell Towers format by using the GPS enabled devices. Navizon used GPS enabled devices to map the nearby Wi-Fi access point and cell Towers. The users who have GPS enabled phones or handheld GPS devices were used in the identification of cell tower locations and Wi-Fi access points. When a significant number of cell towers and Wi-Fi access points have been mapped, the users could obtain the exact location even without using GPS. This study could be identified as the starting point to generate crowdsourced hybrid positioning systems(27).

Developing on the above concept, Zirari et al. Proposed hybrid positioning system by using different combinations of GPS satellites, Wi-Fi access points and cell towers(27). In all these positioning systems it is required to find a signal from four visible signal emitters. The signal emitters could be either satellite Wi-Fi access points or cell Towers. Since the accurate location of the Wi-Fi access point and cell towers are known they act as ground-based reference locations on identifying the position. The results of the research show that accurate location could be obtained only having a signal from single GPS satellite and three other Wi-Fi access points or Cell Towers.

3.4 Using crowdsourced data in hybrid positioning systems

The concept of hybrid positioning systems can increase the accuracy of position identification in space and reduce the delay time in position identification. Use of many known locations as reference points in location identification is the principle behind hybrid positioning system. On this regard, many signal emitters such as Wi-Fi access points, cell Towers and other signal generation methods such as TV signals, RFID signals and different bands of radio signals are used. By using all these methods, the target is to multiply the number of a location known reference points(31).

Crowdsourcing is another way of gathering data on signal emitters. With the enhanced usage of Wi-Fi access points and mobile devices, the number of signal emitters in a given space has increased. For an example when a city is considered, there are many Wi-Fi access points and mobile phone users at any given time. Therefore, if the location of mobile phones and Wi-Fi access points could be identified, they could be used as ground-based reference signal emitters in location identification. This could increase the accuracy of hybrid positioning and reduce the delay between turning on the service and determining the location. To achieve this target, it is required to implement a crowdsourcing service and get the location of mobile devices and Wi-Fi access points. When such a hybrid positioning system is enabled in a city, the location of mobile phone users which connect to GSM service, Wi-Fi access point or GPS signals could be identified. As it has a rapid TTFF value compared to a conventional GPS system, the movement and speed of mobile phone users in vehicles and other modes of travel could be identified. Thus, by combining several search users in a city, the traffic condition could be understood(31).

3.5 How Google collects user location

Google Maps Mobile is a mobile mapping service which was started in 2005 to provide location-based services such as interactive maps, satellite imagery, find local businesses, get driving directions and live traffic updates. Currently, Google uses crowdsourced hybrid positioning system in order to identify the location of its users(32). The crowdsourced hybrid geolocation system of Google Maps is being supported by many smartphone users and mobile phone users all around the world. Google Maps gather user location as a passive crowdsourced information for their platforms from mobile phone users(33).

In November 2007, Google Maps Mobile launched an app feature called “My Location” which could be considered as the major transition in gathering crowdsource information. By 2007 GPS sensors were not much popular in smartphones. Less than 15% of the mobile phones had GPS technology at that time(32), (34). My Location technology uses cell tower ID information to provide users with their approximate location and help them to find places around them. Initially, the cell tower ID information was given by cellular providers. However, Google transmitted the cell tower ID information to their servers, and the location database of cell tower IDs was created. Hence Google could provide location information without relying on the cellular service provider(33).

According to Ji and Jain at Google mobile team, Google used mobile phones which are enabled with GPS technology to find the location of cell tower IDs. When a mobile phone makes or receives a call, it has to communicate to a nearby cell tower. Then the mobile phone knows the ID of the cell tower which it communicates. If the GPS is enabled at the time when the phone call was made Google Maps application on the phone could identify the GPS coordinates along with the cell tower IDs.(34) This information was sent to Google location database in which the location of cell tower could be identified with reference to the GPS location of the mobile phone and Cellular signal parameters such as Time of Arrival(ToA), Angle of Arrival(AoA) and Received Signal Strength(RSS). Since Google services are used by millions of people around the world over millions of cellular tower location updates were collected from multiple mobile phones, different careers and at different times. This clustering method enabled phones without GPS technology also to find its location and access Google location-based services(34).

According to Raphael Leiteteritz the product manager of Google, explains how the company provide location-based data by using a variety of signal emitters such as GPS, cell towers and Wi-Fi access points. The Wi-Fi access points all around the world were collected under the Google Street View project(33). An omnidirectional radio antenna was fixed on the vehicle which travelled along regular roads in capturing street view photographs. The antenna was able to receive publicly broadcast Wi-Fi radio signals within the range of the vehicle. This equipment was able to receive Service Set Identifier(SSID) and Media Access Control(MAC) addresses of available Wi-Fi access points. Service Set Identifier(SSID) is the name of the Wi-Fi access point which is comprised of 32-digit alphanumeric characters. Media Access Control(MAC) addresses is a unique 48-digit identifier assigned to each Wi-Fi access point. The location of each identical Wi-Fi access point within the signal range could be identified by using SSID and MAC address of Wi-Fi access points and incorporating the GPS location sensor fixed at street photography vehicle(33).

Google could provide many location-based services when the Google location servers were developed by incorporating above stated methods. Any mobile phone which has GPS or which does not have GPS could use Google location-based services with the help of Google location servers. If a mobile device needs to identify its location, then it send a request to Google location server with a list of MAC addresses of Wi-Fi access points and cell towers, which are currently visible to the device. Then the location server compares MAC addresses sent by the user mobile with the geocoded MAC addresses in within the server and identifies an approximate location of the user by matching the MAC addresses. Then the approximate location is geocoded and sent back to the mobile device. The accuracy of the location identification is increased, if GPS services are available on the mobile device(31).

With this hybrid positioning system adopted by Google, it is possible to predict the movement of traffic on roads. If a person use Google services in his/her mobile phone, it sends anonymous bits of data about the user location to Google. According to Brandt's Google assigned patent when the mobile device sends its location frequently, the Google location servers could estimate the speed of the user and determine the mode of travel(35). When there are many thousands of other mobile devices in a city share information on speed and location at any given time, the Google location servers could estimate the live traffic condition of roads in real-time. This has made the use of crowdsourced hybrid positioning systems in transport planning work(35).

3.6 Google privacy policy on anonymous data sharing

When a user approved to use Google services such as Gmail, Google Plus, Google Maps etc., the user has to agree what the privacy policy followed by Google. The privacy policy of Google requests the consent of users to allow the company to use location and related information. The location information closed under the information Google collects heading states that "*When you use Google services, we may collect and process information about your actual location. We use various technologies to determine location, including IP address, GPS, and other sensors that may, for example, provide Google with information on nearby devices, Wi-Fi access points and cell towers*" Therefore If a user is using any Google service, then the user allows to collect location information while the service is being used either actively or passively(36).

Hence Google has the ability to collect location information from any user who either uses Google services or has a mobile device operating on the Android operating system. With this provision of the privacy policy, a large number of users could be identified with their location information. Moreover, in the privacy and security section of Google privacy policy, Google requests the consent of users to combine personal information from one service with other Google services. Under this clause, Google has the permission to use location information collected from users while they use different Google services(36).

Further, the Google request the consent from users to share non-personally identifiable information publicly and with its partners under the information we share section of Google privacy policy(36). This allows Google to use location information of users in identifying traffic data along roads and popularity of places. Under the privacy and security-related material section Google request the consent of users to allow it to use information gathered from sensors such as accelerometer, gyroscope etc. This clause allows Google to use information sent by sensors embedded in smartphones and similar devices to determine the speed of users and direction of travel. Hence this clause allows Google to identify travel patterns and travel modes of users(36).

In conclusion, when a user agrees to the privacy policy of Google, it allows the company to use many information of users such as GPS location Wi-Fi or cell tower MAC addresses, information provided by sensors such as accelerometer and gyroscope. By combining all these data shared by many thousands of people moving in a city Google could give better estimates and information regarding real-time traffic condition of roads.

4 Methodology of developing a travel time data mining platform

4.1 Use of mobile probes in travel time data collection

Travel time data collection has been a very interesting research topic over time. There were many methods used by different researchers on collecting travel time data. With the improvement of Intelligent Transportation Systems, communication and detection methodologies have phased up allowing more readily extractable information on transport and mobility(15).

Broadly, the methods of travel time collection could be divided into fixed-point technologies and probe vehicle technologies. The fixed-point technologies could be identified as embedded traffic sensors automatic vehicle identifiers, video and image-based vehicle identification methods and similar alternatives in which data acquisition equipment is fixed at a location.

Probe vehicle technologies are-based on moving vehicles on traffic. The probe vehicle has a method of collecting travel time data, and the data acquisition equipment is not stationary. Most commonly used method of data collection in probe vehicle is that the vehicle is equipped with the device which can send its position with the timestamp. Many methods were developed to estimate travel time using this data. Vehicles equipped with GPS sensors is a commonly used probe vehicle mode in many parts of research work. Taxi vehicles which are equipped with GPS sensors are used by many researchers in deriving travel time estimates(37). It could be justified that GPS trajectories of taxi vehicles can cover many areas of the city. Dewulf et al. Use GPS trajectories of 400000 taxi vehicles to identify commuting patterns of drivers in cities(38). Zhao et al. Suggested identifying traffic condition using probe data collected by commercial global positioning system fleet management devices mounted on trucks(39).

There are several methods in estimation of travel time from probe vehicle data. Average speed method, analysing vehicle trajectories, evaluation of iterative travel time and piecewise linear analysis of vehicle trajectories are some of the methods used in research(40). The travel pattern of taxi vehicles in a city is not similar to the movement of other vehicles. Therefore it is questionable whether using taxi vehicles to estimate travel time on roads will be accurate as it cannot represent the whole vehicle feet moving on the road. Moreover, the sparseness in obtaining GPS location along a trajectory will cause errors in travel time estimates as it cannot account for delays. Therefore using only the GPS location shared by taxi vehicles or any other fleet of vehicles which are used for several purposes such as public transport, delivery of goods, fleets of trucks could create errors as they cannot represent the whole vehicle fleet moving on the road.

4.2 Crowdsourcing methods of travel time data collection

Crowdsourced data is an optimum mode of information gathering for traffic planning activities. Therefore, crowdsourcing information from smart mobile phones and embedded deices were discussed in many research works. The concept of crowdsourcing emerged as an alternative for installing many sensors to get a picture of congestion along roads. In the initial research work, the GPS location of vehicles was used to identify the congestion or moving traffic along the road. Jimenez and valacia suggest a methodology to identify traffic congestion-based on GPS data from taxis vehicles in a city limit (1). The GPS location reported by the GPS devices via GPRS (General Packet Radio Service) with a frequency of 10s this information was used to identify traffic patterns along roads(1). This could be considered as a low-cost option in identifying critical points in traffic networks in developing cities without expensive traffic-monitoring systems. Limitation in implementing this type of a methodology is the data collected is only available for taxi vehicles in the city which is an inadequate sample of vehicle movement (1). Moreover, for countries with less number of taxies, this method cannot provide a representative sample in urban traffic (15).

To address the issue of obtaining a representative sample use of Bluetooth data from a mobile phone could be identified as an improvement. Gudishala et al. propose Bluetooth-based travel time estimation system as a cost-effective method. In this method, travel time information is obtained from the Bluetooth enabled devices of which the passengers carry with them in vehicles. The GPS location is transmitted via Bluetooth to the vehicles nearby. With this method, it is possible to get a representative sample of data as there are many vehicle types. The limitation of this study is Bluetooth communication can only cover a small range less than a hundred meter. Therefore, it is not possible to get an idea about the whole city traffic by using this method and only possible to implement for shorter corridors(7).

The methodology suggested by Janecek et al. For estimating vehicular travel times based on the mobile cellular network could be identified as a way to expand for a larger area (16). The research focuses on how vehicle travel times and road congestion can be inferred from anonymised signalling data collected from a mobile cellular network (16). To obtain higher accuracy in estimation of travel times and timely detection of congestions by both active users engaged in voice calls and inactive users were used (16).

In the methodology of Janecek et al., spatially coarse mobility data from all users (both active and inactive) is gathered to capture speed deviations in long road sections and detect congestion events (16). Then finer grained mobility data produced by active users is used to refine the location accuracy and classify the type of congestion event (16). Compared to earlier methodologies stated above this approach presents a higher accuracy and an economical approach which does not require investments in new infrastructure while utilising mobile cellular network as a large-scale mobility sensor (16).

Improving on using mobile network data on congestion identification D'Andrea and Marcelloni suggest a methodology to detect traffic congestion using GPS data provided by smartphone users moving in a road network (14). In their study, traces collected from vehicles moving in the city are analysed in real-time using an expert system, without the need for a learning process or historical data. The outcome of the study presented a system for detecting traffic congestion, traffic state-based on the speeds of vehicles and incidents in real-time (14). With the study, it is possible to send a notification to users on traffic alerts indicating the affected area and traffic state whether traffic is flowing, slowed, very slowed, and blocked (14). This study is much more similar to the information provided by Google traffic layer (17). A significant difficulty in this research was to obtain GPS data from users moving in the city due to privacy issues (14). Next, the scalability of the system with limited resources was a challenge. As future work, D'Andrea and Marcelloni suggested incorporating historical data and real-time information for travel time prediction and traffic state prediction (14). Hence Google Distance Matrix API will be a solution for the challenges faced by authors and future work suggested by them.

4.3 Collecting data from Google traffic layers

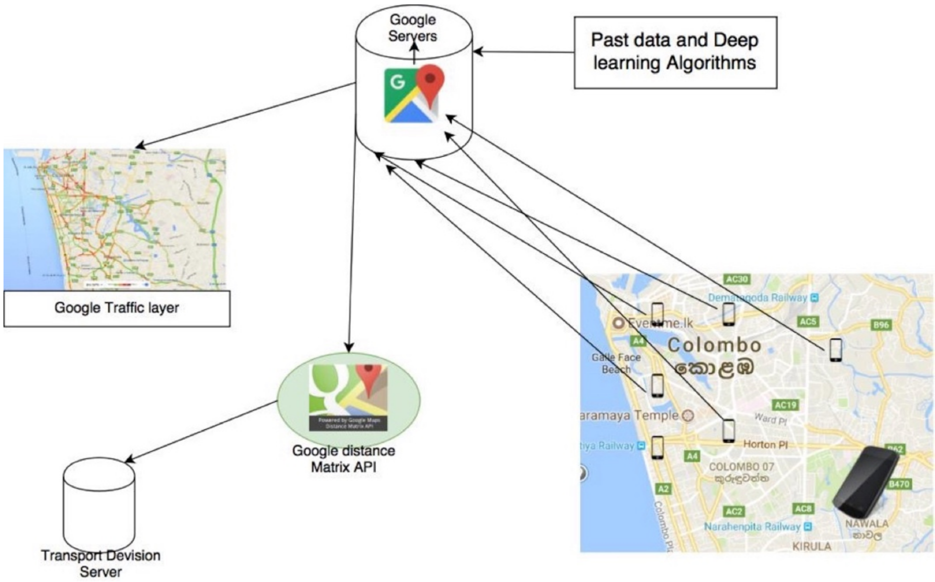


Figure 5: Collecting traffic data from Google Maps

It was identified in the literature review use of crowdsourced data is an optimum approach in mining data for traffic analysis on roads. Use of Google traffic data is an economical solution as it does not need any investment in equipment or calibration. It was identified that Google could give better coverage in Sri Lanka out of all available traffic information providers. Google is the only service available in Sri Lanka of this sort, and in many developing countries it is the same. This study utilises data mining from Google Application Programming Interfaces(API) and obtained data to analyse traffic.

Figure 5 shows the process of collecting traffic information from Google application programming interface. Initially, the uses of Google Services share location information with Google when they use services such as Google browsing, Google

Maps, Gmail, YouTube and similar services provided under the flag of Google. The user location shared by thousands of people in a city is connected to Google location servers. By using the real-time information shared by users and incorporating the past data Google servers derive at traffic information such as traffic along roads, estimated travel times between origins and destinations, the popularity of places and identification of extreme traffic conditions etc. The algorithm or methodology of estimating traffic parameters are not revealed by Google and kept black boxed to the public. However, the traffic information is given to public via Google Maps and Google Maps APIs.

4.4 Collecting User Location from Smartphone users.


```
{
    "timestampMs" : "1507463450322",
    "latitudeE7" : 355221018,
    "longitudeE7" : 1395028185,
    "accuracy" : 20,
    "velocity" : 0,
    "altitude" : 77,
    "activity" : [ {
      "timestampMs" : "1507463430796",
      "activity" : [ {
        "type" : "UNKNOWN",
        "confidence" : 35
      }, {
        "type" : "STILL",
        "confidence" : 33
      }, {
        "type" : "IN_VEHICLE",
        "confidence" : 27
      }, {
        "type" : "ON_BICYCLE",
        "confidence" : 4
      }, {
        "type" : "ON_FOOT",
        "confidence" : 2
      }, {
        "type" : "WALKING",
        "confidence" : 2
      } ]
    } ]
  },
```


Figure 6 : Location information collected from users by Google

Google has a mechanism for collecting location information from its users. The collection of location information is given as a clause in Google privacy policy. In third chapter, it was revealed how the privacy policy requests the consent of users to use their location. Any person who wants to use Google services has to agree to the privacy policy of Google. Hence when users agree to the privacy policy Google receives the consent of using location information. Google services are provided to users in many platforms basically it could be divided into mobile-based platforms and web-based platforms. In mobile-based platforms, Google services are provided for mobile phones which operates on Android operating systems, Windows operating systems and iOS operating systems. Location information of users is being shared with Google when Google services are being used in any of these operating systems either the Google service being actively used or being processed in the background. Figure 6 shows location information sent by the mobile user to Google Service which was retrieved from the user's account. The location information collected from users are stored with the user identity. If a user required to download data which was collected from Google it could be done via the Google takeout platform. It allows users to download location history in JSON format. In Figure 6, the user's location was recorded in GPS coordinates, and timestamp for that gives the approximate moving velocity direction, and the activity type. The activity type could be either moving in a vehicle walking or cycling. Google collect the location information of many users in this format.

Figure 7: Map of Daily activity of a mobile phone user who use Google services

Figure 7 shows daily activity location of a mobile phone user in which the location was share with Google. The location information was downloaded from Google takeout platform, and the information was mapped to illustrate person activity. The map shows over 150 locations which were recorded within 3 days of activity. The person has used a mobile phone made by Apple Inc, and the product name is iPhone X

4.4.1 Location collect from Android phones

Android is an open source mobile operating system(OS) which is developed by Google. The operating system is Primarily designed for mobile devices with touchscreens such as smartphones, tablet PCs And it was extended for televisions (Android TV), vehicles (Android Auto) and wearables (Wear OS). Any Android OS-based mobile phone communicates regularly with Google servers and information such as device type, carrier name, crash reports, location history, app lists, search queries, call logs, activity patterns, routing information, and types of calls etc. This information is used by Google services to improve the user experience. Features like driving directions on the daily commute, personalized advertising, personalized search experience. All these options are focused on increasing the customer satisfaction while increasing the usability of Google services.

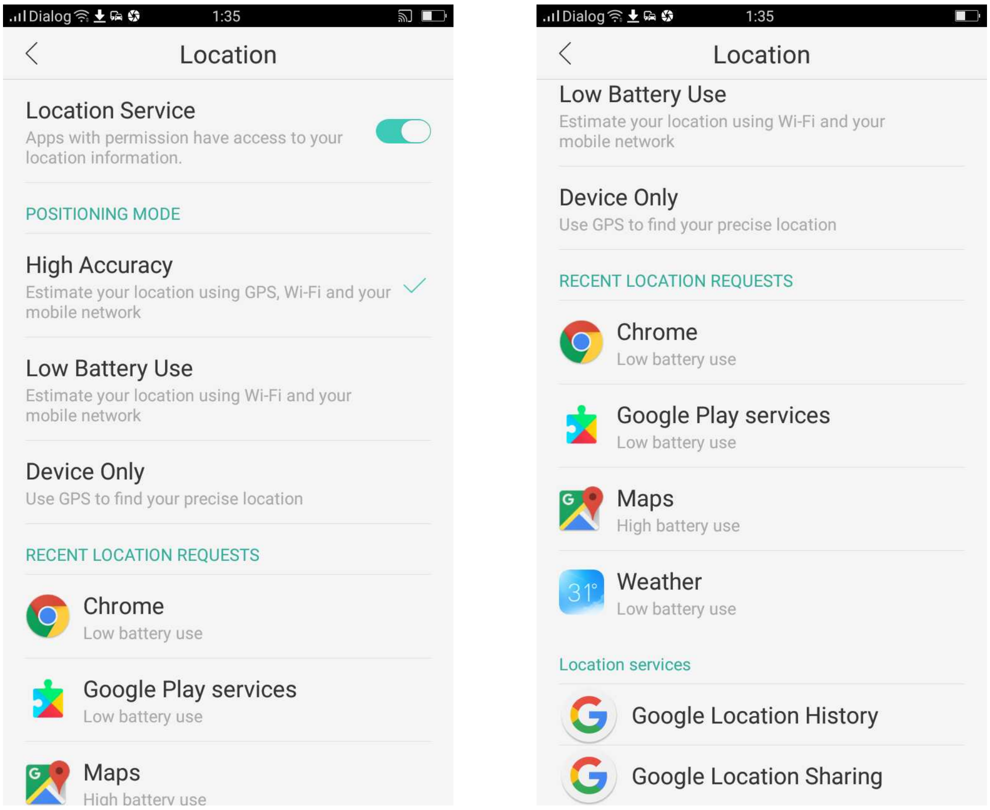


Figure 8 : Location enabling options in Android phones

In the Android OS, the location services could be enabled from location settings as shown in Figure 8. The location services are enabled by default and when you try to use any Google service the user is requested to enable location services to increase service quality. The users can disable location services and use Google services, but users cannot disable sharing information regarding cellular data and MAC addresses of Wi-Fi access points. In addition to the services and applications provided by Google smartphone users tends to install different applications which use location data such as Facebook, Messenger, WhatsApp, Viber etc. Further, many Google applications run on Android OS in the background. When an application runs on background, it runs constantly and refreshes the app frequently. Mobile apps like Gmail, Google Calendar run in the background to notify new emails or events. These background applications have access to cellular data or location information. Therefore, location information, IP addresses, MAC addresses of Wi-Fi Access points are continuously shared with Google. Hence the objective of using mobile phones as probe devices to determine location become a success.

4.4.2 Location information from iOS mobile devices

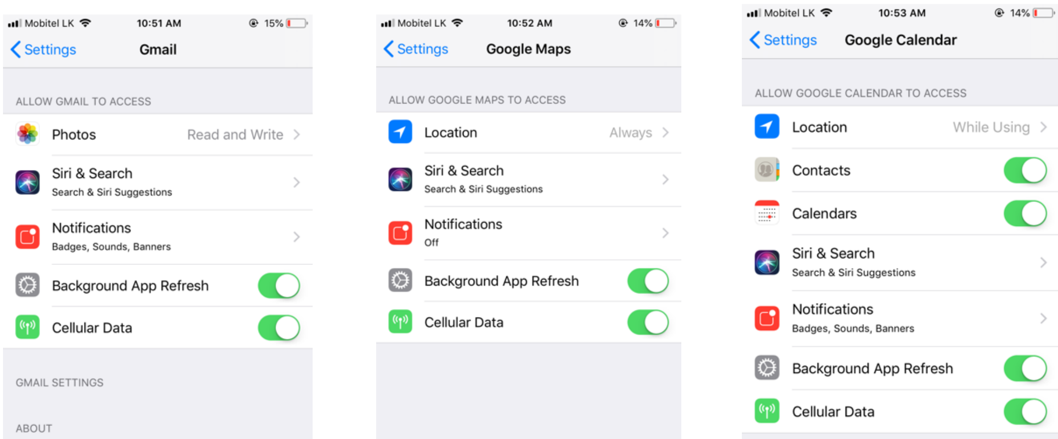


Figure 9: iOS settings to change accessibility of Google Services

The mobile phone market shares majorly owned by the Android operating system which account for 87% globally(41). IOS is the next used mobile operating system which accounts for 12% globally(41). Google services are available in iOS operating system too. Hence location information, cell tower information Wi-Fi access point information and other user details of iOS users could be obtained If Google services are used by them in iOS. The operating system has more flexibility for the user to decide sharing location information, cell tower information and Wi-Fi access points.

Although the users have the flexibility in restricting the use of GPS location, it is not possible to use many of the services which Google offer without cellular data. Figure 9 shows accessibility levels given to several Google services such as Gmail Google calendar Google Maps and YouTube. All these applications have cellular data access ability and enabled background app refresh. Certain applications are given location accessibility. Hence Google has the opportunity of collecting location information from iOS users.

4.4.3 Google Service Architecture of Location-Based Services.

Google services use location information of its users to increase the usability of their services and increase the involvement. By sharing location information, users can get many benefits, and the convenience is very high. Location-based services such as Google Maps gives traffic information, place details, routing options, public transport schedules, nearby places, sharing location with friends and many more. By using location data, Google services personalize applications to suit the choice of the user. Location services and search queries of users are used to view personalize advertisements in which users are interested. Therefore, customers or users tend to use Google products as the services are user-friendly and increase the convenience. This feature of Google has made over one billion of daily active users and more than 2 billion devices globally(41).

Further, Google provides services such as activity monitoring for its users. Figure 7 shows the places visited by a user along the timeline. This feature has enabled Google to monitor activity locations of people and improve their services accordingly. There are several arguments on this regard whether Google or any commercial entity could monitor human activity by providing services. Hence the privacy of service is always argued matter, and many legal and policies are being constituted by governments to protect user privacy while ensuring quality service. Google in their privacy policy states that they take proper measures to protect the user identity and ensure the safety of people when they share location information and daily activity with Google(36).

4.5 Using the Google Distance Matrix API

Google Distance Matrix API is The service provided by Google Inc under Google Maps platform. By accessing this API, it is possible to collect trip information such as distance and travel time about a route between given origin and destination. By using Google Maps, a user can find the distance and estimated time to arrival(ETA). The user has to give the origin and destination to Google map if he/she wants to find travel information on travelling between two locations. Then Google provides information such as distance, directions, travel time-based on different modes of travelling, and more interestingly navigation to the required location.

Similar information could be obtained by accessing the Distance Matrix API when information of different routes, different origins and destinations are required simultaneously. The Distance Matrix API is requested via secured hypertext transfer protocol(https) or hypertext transfer protocol(HTTP). Figure 10 shows the basic method of calling the API. The output format and the parameters have to be defined appropriately to suit the user requirement.

```
https://maps.googleapis.com/maps/api/distancematrix/outputFormat?parameters
```

Figure 10 : Basic method of calling the Distance Matrix API
Ref : Google Maps API

4.5.1 Input Parameters to API

The input parameters are the elements which user request from the API and results are released according to the requested input parameters. In defining input parameters, they are mandatory required parameters of which the user has to define them definitely. And there are optional parameters of which the user can define as per the requirement.

4.5.1.1 Required parameters

Following parameters are mandatory to Define, and the API cannot be called without defining them. If there are any errors in this parameter, the results will be either erroneous or failed.

- **Origins**

This is the parameter for which origin of the route to be passed. It is possible to call a single origin or a multiple number of origins at single API call. Either the address or latitude and longitude coordinates or a place ID could be given as origins. In this study, the latitude and longitude coordinates of origins and destinations were used to reduce the error. Using addresses to collect travel time will make many errors as addresses are not properly defined in Sri Lanka. Moreover, using place IDs or addresses will result in travel time between those locations. This might cause an error in data collection as the study is interested on collecting travel time on roads and not in between places.

- **Destinations**

Similar to origins, this parameter accounts for the destination of the travel from the selected origin. The destination value also passed as latitude and longitude coordinates to reduce errors and loss of data. When calling the API always a single pair of origin and destination was used. If multiple Origins and multiple destinations were used the Distance Matrix API gives information for all the trip starting from each origin and going to each destination. The study was concerned on collecting travel information along roads segments. Therefore, passing multiple origins and destinations will result in complications and errors.

- **API Key**

This is the identification method of the API. For each API call, an authorized API key should be passed and the Google service has to authenticate the key. In order to get API key, the user has to register the data collection as a project on the Google API Console. The user can obtain an API key when he/she creates a project and enable a payment method.

The API calls are not unlimited, and they are being limited according to predefined rules. There is a free limit set out by Google-based on a daily basis or monthly basis. Users have to call the API as a paid service if the free limit is exceeded. The paid services are provided either-based on pay as you go basis or as a premium plan under a contract between Google Inc and the user.

Table 3 : Cost structure of the Distance Matrix API

Standard limit	Pay-as-you-go	Premium
2500 API calls / Day	$0.50 /1000 extra calls up to 100000 API calls/day	After contact with Google

Table 3 shows the limitations and cost involved in data collection. Under the standard usage limits, 2500 free API calls per day are allowed. That is traffic information of road segments could be obtained at most 2500 times every day. After exceeding the daily limit, the pay as you go scheme could be enabled in which it allows to call 100000 API calls per day for which it is billed that $0.50 USD / 1000 additional API calls after exceeding the free limit. The cost structure is decided by Google. Hence it could be changed.

4.5.1.2 Optional parameters

In addition to the mandatory parameters, the API could be called with several other optional parameters. The following paragraphs will explain only the optional parameters which are relevant to the study.

- **Mode**

With this option, it is possible to enable the travel mode. The travel mode could be driving walking cycling or public transport. In the driving mode, the travel time and distance are calculated using the road network. In this study, the driving mode is used, as it is the most optimum to use in traffic studies.

- **Language**

This is the language of which the results could be obtained. There are more than 50 languages available to obtain data. The list of languages possible to obtain data is attached in the Appendix A. In this study, English language was used throughout the data collection process.

- **Units**

This parameter specifies the unit system to be used in data collection. This parameter is very important if the distance and travel time values are expressed in text format.

The Metric unit system and imperial unit system are available, and the metric system was used in data collection for the study.

- **Departure time**

This is the desired time of departure. It is possible to collect travel time information at current time or time in future. This parameter is used to calculate travel time, and it is considered as the starting time of the journey. In the data collection, the departure time is set to "now" which indicate the travel time related to the time which the API was called is collected. When the departure time is set to "now," the real-time traffic information is considered in calculating travel time. If the departure time is set to a time in future, the travel time prediction could be obtained. It is not possible to get the travel time in the past via this API.

- **Arrival time**

Similar to departure time it is possible to collect travel time information by defining the arrival time to a destination. It should always be a time in future and cannot be in the past or present.

- **Traffic model**

This specifies the assumptions to use when calculating time in traffic. The travel time values given by the API includes the time in traffic, delays and running time. The predicted travel time is based on real-time information and historical averages. This parameter is available only when the travel mode is "driving" and the departure time parameter is set to "now". It is possible to give free traffic models when calling in the API. They are "best guess", "pessimistic" and "optimistic". The "best guess" option is the default option, and it considers the duration in traffic as the best estimate of travel time given by historical traffic conditions and live traffic conditions. Pessimistic mode indicates that the duration in traffic should be longer than the actual travel time on most days. This gives a higher value for travel time which accounts for bad traffic conditions which may exist. The optimistic model defines lower duration in traffic, and it is considered that the duration in traffic should be shorter than the actual travel time on most days. This gives a lower value for travel time which considers good traffic conditions which may exist.

In addition to above optional parameters, there are some other parameters which are not much relevant for the considerations of the study.

4.5.2 Results Given by API

When the API is called correctly and authenticated by Google service, the results are given as the output. The output format is defining the format in which the results should be released via the API. The Google Distance Matrix API allows JavaScript Object Notation(JSON) format or Extensible Markup Language(XML) format to release results. The information requested placed in a row array, and JSON decoder or XML decoder can be used to access the output results and store in a file. Following fields are released as output elements via the API

- **Distance**

This is the distance between origin and destination along the selected route. The route is selected by the API to minimize the travel time. The route selection is influenced by the availability of roads on the map, the availability of real-time and historical data. The distance value is given in meters. The distance value has a significant accuracy in which the error is less than 0.44%(42). Road segments with distance more than 10 meters were used to collect data in this study.

- **Duration**

This is the length of time it takes to travel the selected route by the API. In the calculation of the duration Is-based on the posted speed of the road and the distance of the route. This value gives an average travel time based on the information available for the road.

- **Duration in traffic**

This is the length of time it takes to travel the selected route-based on current and historical traffic conditions. It considers the traffic model defined in the API in the calculation of travel time. This value will be only released if the departure time is set to “now”. The values could be obtained in seconds.

This study was concerned on using above parameters in order to carry out analysis-based on our time data. The fare of the trip could be taken as an element if the mode is public transport and search details are being shared by operators with Google.

4.5.3 Example: Calling the Distance Matrix API and obtaining results

The following example illustrates how the Distance Matrix API is called and how results are obtained. The Table 4 shows parameters and the output format used to call the API.

Table 4 : Distance Matrix API call parameters

Parameter	**Value**
Origin	6.934956,79.8537749 (Colombo Central Bus Stand)
Destination	7.2909666,80.6310087 (Kandy Central Bus Stand)
API Key	AIzaSyC01gder$5gxpgxCJVXDuCJkCYWve-gaXO8y0
Travel mode	driving
Traffic model	best_guess
Departure time	now
Language	English (en)

The Table 4 shows how travel time between Colombo Central Bus Station and Kandy Central Bus Station could be obtained by accessing the above parameters. By incorporating the parameters in the Table 4 to the API call, the results could be obtained

After calling the API, it is possible to collect data in the JSON format Figure 11 right shows the JSON output given on the browser. The outputs include distance, duration and duration in traffic. The results could be obtained as values or as text. The distance is given in meters (135462m) and the travel time is given in seconds (12593s). Figure 11 left shows how the same information which could be obtained in a mobile phone by giving the same origin and destination pair.

```
{
   "destination_addresses" :
      [ "Colombo Central Bus Station" ],
   "origin_addresses" :
      [ "Kandy central Bus Station" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "135 km",
                  "value" : 135462
               },
               "duration" : {
                  "text" : "3 hours 35 mins",
                  "value" : 12902
               },
               "duration_in_traffic" : {
                  "text" : "3 hours 30 mins",
                  "value" : 12593
               },
               "status" : "OK"
            }
         ]
      }
```

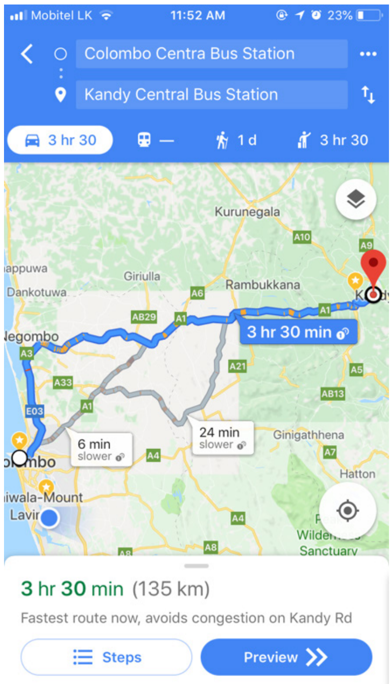



Figure 11 : Results obtained from calling the Distance Matrix API

4.6 Development of data collection server.

This section focuses on developing the data collection server to collect travel time information from Distance Matrix API. In the earlier section, it was understood that by calling the API it is possible to collect trip information. The objective of the data collection service is to collect trip information for a given set of road segments at a predefined frequency.

4.6.1 Development of PHP scripts to collect data

Although travel time information could be accessed via the Distance Matrix API, it is not convenient to call the API manually every time when travel time information is required. Further, it is not possible for a human to call the API simultaneously for several Road segments exactly at a given time. For example, if it is required to collect trip information for 20 Road segments 6 a.m. to 10 p.m. At a frequency of 5-minute intervals, this could not be carried out by a human and it is not practical to do so. Hence there is a requirement of developing travel time data collection server which can run a program to collect required trip information data and store. For this purpose, the development of a web server-based application to collect trip information, which could run at a predefined schedule is required.

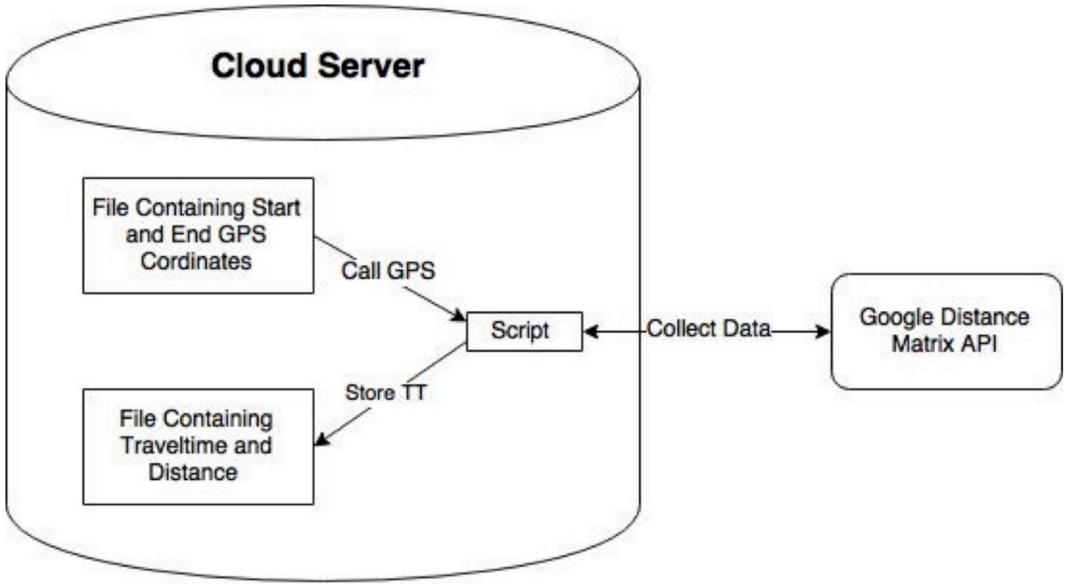


Figure 12 : Data collection framework

Figure 12 illustrates the framework of data collection proposed under this study. In order to established the web server a cloud server platform was purchased from a cloud server service provider. In-house web server is not preferred as there are many complications and cost components involved in establishing an in-house web server. The advantage of the cloud server is, it is ready-made to utilize, and there are no maintenance activities associated. The cost of purchasing a cloud server for the research period is cheaper than establishing an in-house web server. In obtaining cloud server, it is required to ensure that necessary speed and space is available to carry out the data collection process throughout the design period. In this study, the cloud server was consist of 2-gigabytes of space and a random-access memory(RAM) of 512-megabytes.

When the server is established, an application script is required to develop to collect data by accessing the Google Distance Matrix API and store the collected data on the web server. Therefore, a Hypertext Preprocessor(PHP) script was developed for the purpose.

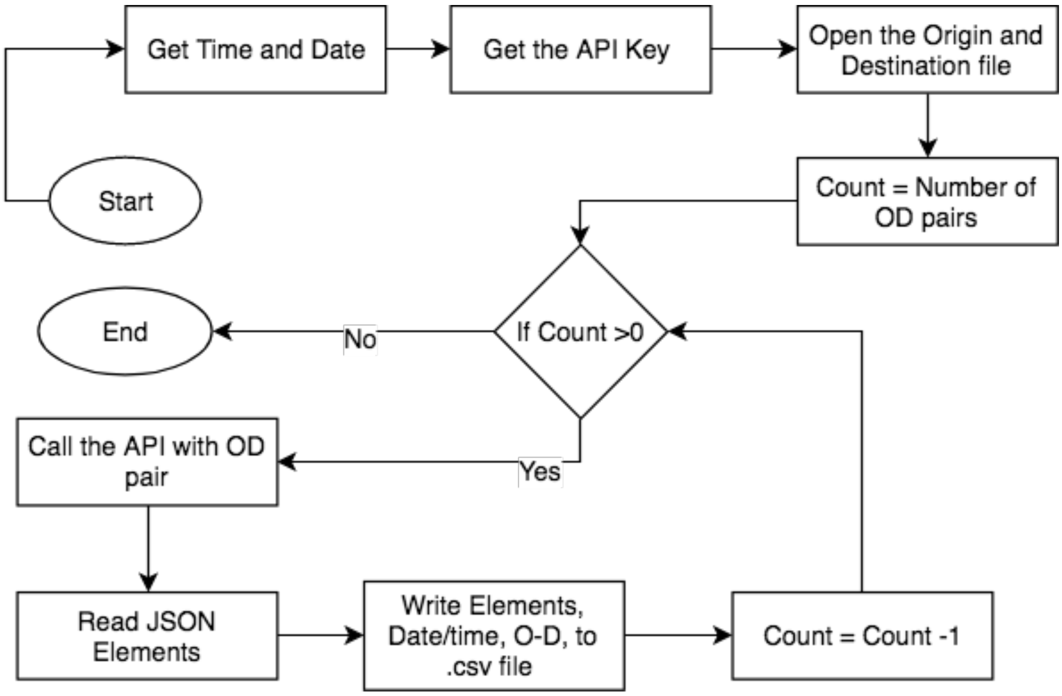
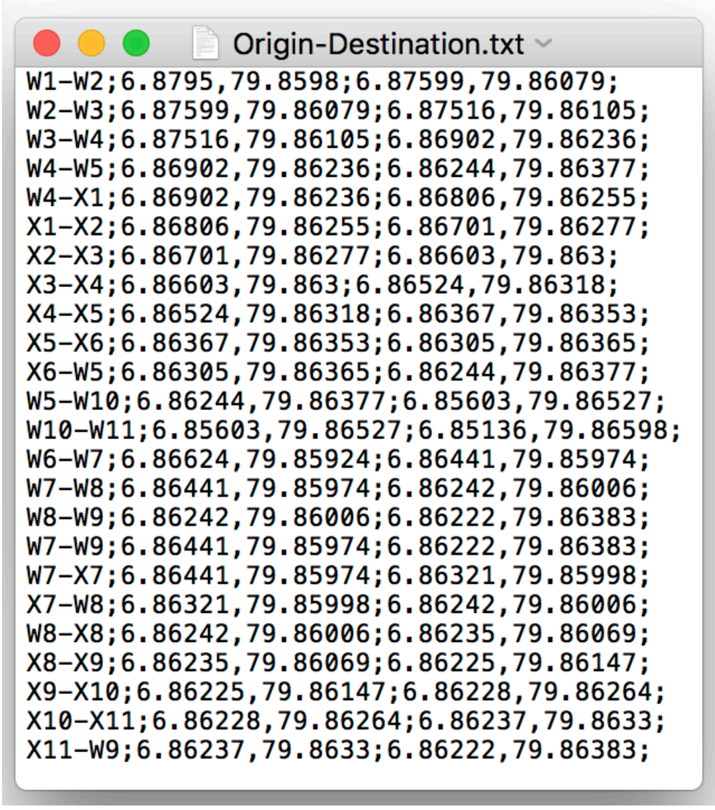


Figure 13 : Flow chart of the PHP script

The flowchart of the PHP script which was developed to collect data is shown in Figure 13. When the PHP script is executed, it collects the date and time of the time zone to which the road segments belong to. The time zone and the conversion factors have to be coded. Then the PHP script collects the API key which has to be given as a parameter. The road segments origin-destination file has to be created in the text format and included in the folder location where the script is located. The script can collect the text file, and it will be the main input. Next, the origin-destination pairs will be put into an array by reading the input file. Then the count of origin-destination pairs in the array will be counted and assigned to another variable. Then the script will call the Distance Matrix API for each origin and destination pair iteratively until the API was called for each and every origin and destination pair in the array.

The origin-destination input file has the latitude and longitude of the origins and destinations formatted according to the way that the PHP script requires. Figure 14 shows a typical example of the origin-destination file. Every row in the file represents an origin-destination pair. The first column has the nominal segment name which is given for identification. The second and third column contains latitude and longitude of origin and destination respectively. The input file is formatted as a semicolon separated text file.




```
W1-W2;6.8795,79.8598;6.87599,79.86079;
W2-W3;6.87599,79.86079;6.87516,79.86105;
W3-W4;6.87516,79.86105;6.86902,79.86236;
W4-W5;6.86902,79.86236;6.86244,79.86377;
W4-X1;6.86902,79.86236;6.86806,79.86255;
X1-X2;6.86806,79.86255;6.86701,79.86277;
X2-X3;6.86701,79.86277;6.86603,79.863;
X3-X4;6.86603,79.863;6.86524,79.86318;
X4-X5;6.86524,79.86318;6.86367,79.86353;
X5-X6;6.86367,79.86353;6.86305,79.86365;
X6-W5;6.86305,79.86365;6.86244,79.86377;
W5-W10;6.86244,79.86377;6.85603,79.86527;
W10-W11;6.85603,79.86527;6.85136,79.86598;
W6-W7;6.86624,79.85924;6.86441,79.85974;
W7-W8;6.86441,79.85974;6.86242,79.86006;
W8-W9;6.86242,79.86006;6.86222,79.86383;
W7-W9;6.86441,79.85974;6.86222,79.86383;
W7-X7;6.86441,79.85974;6.86321,79.85998;
X7-W8;6.86321,79.85998;6.86242,79.86006;
W8-X8;6.86242,79.86006;6.86235,79.86069;
X8-X9;6.86235,79.86069;6.86225,79.86147;
X9-X10;6.86225,79.86147;6.86228,79.86264;
X10-X11;6.86228,79.86264;6.86237,79.8633;
X11-W9;6.86237,79.8633;6.86222,79.86383;
```

Figure 14 : Input Origin-Destination file

When the input file is loaded, and the script calls the API, the script will allocate origin longitude and latitude value to origin parameter and the destination longitude and latitude value to destination parameter in the API. The optional parameters have default values. The travel mode is driving, traffic model is best guess and the departure time is now.

When the API is called for a single origin-destination pair, the response to the API call is finished in JSON format by the Google Distance Matrix API. The script can read the JSON output and find the elements in the JSON output such as the distance in meters and duration in traffic in seconds. Then these elements are assigned to variables. Next, space mean speed is calculated from the obtained distance value and travel time value.

When the JSON file is decoded, and speed is calculated the next task is to write the results to the file. The results could be written to a comma separated text file into a database table. If data is to be written to a comma separated variables(CSV) file, then the script format the variables as a lion and write to the CSV file. Similarly, If the data needs to be input into a database table a database query is created and passed to the database.

Segment	O-Lat	O-Long	D-Lat	D-Long	Distance(m)	Travel time(s)	Speed(km/h)	TimeZone	Date and Time
W1-W2	6.8795	79.8598	6.87599	79.86079	1716	409	15.10	SL/Colombo	2018-05-11 04:00:04 PM
W2-W3	6.87599	79.86079	6.87516	79.86105	1049	349	10.82	SL/Colombo	2018-05-11 04:00:04 PM
W3-W4	6.87516	79.86105	6.86902	79.86236	1525	337	16.29	SL/Colombo	2018-05-11 04:00:04 PM
W4-W5	6.86902	79.86236	6.86244	79.86377	1562	374	15.04	SL/Colombo	2018-05-11 04:00:04 PM
W4-X1	6.86902	79.86236	6.86806	79.86255	977	248	14.18	SL/Colombo	2018-05-11 04:00:04 PM
X1-X2	6.86806	79.86255	6.86701	79.86277	1114	252	15.91	SL/Colombo	2018-05-11 04:00:04 PM
W1-W2	6.8795	79.8598	6.87599	79.86079	1716	408	15.14	SL/Colombo	2018-05-11 04:05:03 PM
W2-W3	6.87599	79.86079	6.87516	79.86105	1049	352	10.73	SL/Colombo	2018-05-11 04:05:03 PM
W3-W4	6.87516	79.86105	6.86902	79.86236	1525	342	16.05	SL/Colombo	2018-05-11 04:05:03 PM
W4-W5	6.86902	79.86236	6.86244	79.86377	1562	383	14.68	SL/Colombo	2018-05-11 04:05:03 PM
W4-X1	6.86902	79.86236	6.86806	79.86255	977	247	14.24	SL/Colombo	2018-05-11 04:05:03 PM
X1-X2	6.86806	79.86255	6.86701	79.86277	1114	252	15.91	SL/Colombo	2018-05-11 04:05:03 PM
W1-W2	6.8795	79.8598	6.87599	79.86079	1716	405	15.25	SL/Colombo	2018-05-11 04:10:04 PM
W2-W3	6.87599	79.86079	6.87516	79.86105	1049	355	10.64	SL/Colombo	2018-05-11 04:10:04 PM
W3-W4	6.87516	79.86105	6.86902	79.86236	1525	358	15.34	SL/Colombo	2018-05-11 04:10:04 PM
W4-W5	6.86902	79.86236	6.86244	79.86377	1562	348	16.16	SL/Colombo	2018-05-11 04:10:04 PM
W4-X1	6.86902	79.86236	6.86806	79.86255	977	249	14.13	SL/Colombo	2018-05-11 04:10:04 PM
X1-X2	6.86806	79.86255	6.86701	79.86277	1114	256	15.67	SL/Colombo	2018-05-11 04:10:04 PM

Figure 15 : Sample of data collected

Figure 15 shows an example of data collected from Google Distance Matrix API via using the PHP script developed. The results file includes items such as segment name, origin longitude and latitude, destination longitude and latitude, distance in meters, travel time in seconds, average speed, time zone, date and time values.

4.6.2 Scheduling the data collection

By developing the PHP script, it is possible to call the Distance Matrix API for many segments simultaneously and collect travel time information. Hence the problem of collecting travel time information of many segments simultaneously was solved. It is required to schedule the execution of the PHP script in order to collect travel information for a period of time with a data collection frequency. To schedule the PHP script execution, a Cron-job scheduler could be used. Cron-job scheduler can execute a task for at predefined times in future.

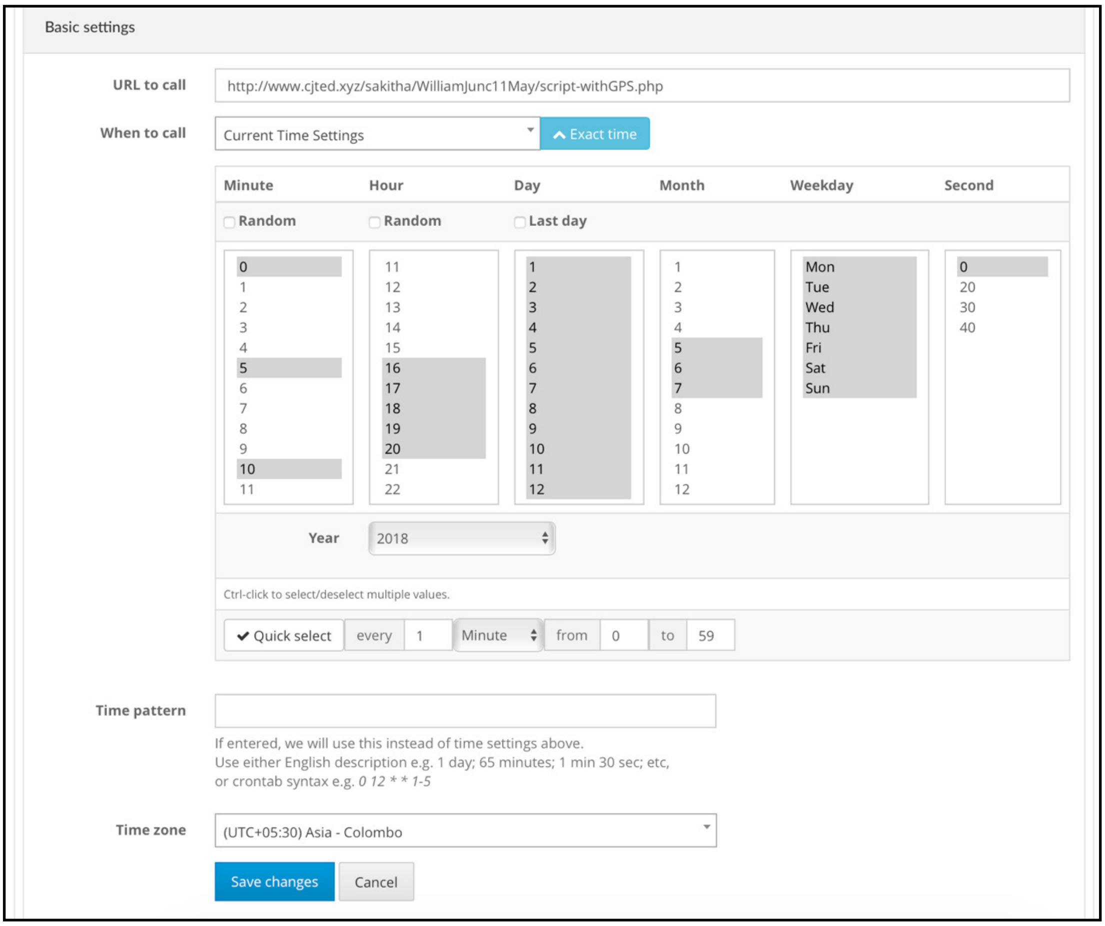


Figure 16 : Cron job scheduler used to execute the PHP script

Figure 16 shows the user interface of a Cron-job scheduler used in the study. The link to the PHP script was called via the Cron-job scheduler. The example in Figure 16 shows how Cron-job was created to collect travel time information in 5-minute intervals from 4 p.m. To 9 p.m. Every day for a period of 3 months. The successful implementation of Cron-job scheduler enables to collect travel time information consistently for a longer period of time. Hence this solves the problem of executing the PHP script for a period of time at a given frequency.

5 Verification of Google travel time data

This chapter elaborates on the verification process carried out to ensure the reliability of travel time information collected via Google Distance Matrix API with the field observed travel time. By accessing the Google Distance Matrix API, it is possible to collect travel time information and average speed information for a given segment on the road. This information could be collected continuously for a period of time and at a given frequency. Thus, this information could provide a base for many analysis related to the performance of roads with traffic. Before traffic information being used in the analysis, it is required to establish that obtained data is correct to the actual situation which could be observed on the road. Hence the verification process was carried out in following steps;

1. Evaluation of the network infrastructure of Sri Lanka.
2. Using probe vehicle techniques in verifying the Google travel time
3. Using public transport vehicle GPS data to verify the Google travel time
4. Verification of Google travel time data for different vehicle types

5.1 Evaluation of the network infrastructure of Sri Lanka.

In order to verify the results obtained from Google Distance Matrix API, it is required to evaluate the mobile network infrastructure facilities available in Sri Lanka. The travel time estimates provided by Google are based on the mobile communication methods such as cellular data Wi-Fi and GPS signals. In order to obtain correct results, it is required to establish that Sri Lanka has a proper network infrastructure which can support the information communication. Hence this section will evaluate the network infrastructure of Sri Lanka.

5.1.1 Mobile usage and subscription in Sri Lanka

The Government of Sri Lanka has identified accessibility and affordability of Information and Communication Technology as a priority area(43). According to statistics, Sri Lanka has 6.71 million internet users, and this is nearly 32% of the total countries population. Having 32% of internet penetration is an average performing value compared to the countries in the South Asian region. Nearly 5.5 million people are active mobile social users in Sri Lanka which account for the 26% of the population in 2018(44).

It was identified that there are 27.38 million mobile connections and that is 131% of the country's population. This indicates that there are many mobile connections in the country. There's increasing growth in mobile connections, and the Figure 17 shows how the mobile connections growth since 1997 to 2017(45).

Year	Subscriptions
1992	2,644
1993	14,687
1994	29,182
1995	51,316
1996	71,029
1997	114,888
1998	174,202
1999	256,655
2000	430,202
2001	667,662
2002	931,403
2003	1,393,403
2004	2,211,158
2005	3,361,775
2006	5,412,496
2007	7,983,489
2008	11,082,454
2009	14,264,442
2010	17,267,407
2011	18,319,447
2012	20,324,070
2013	20,447,508
2014	22,123,000
2015	24,384,544
2016	26,227,631
2017 Jun	28,113,153

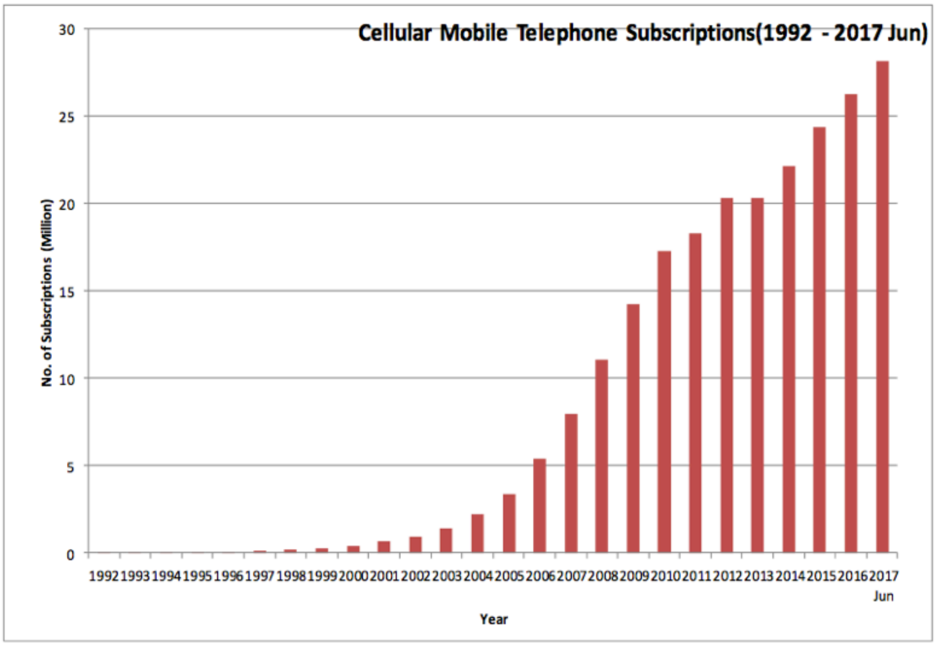
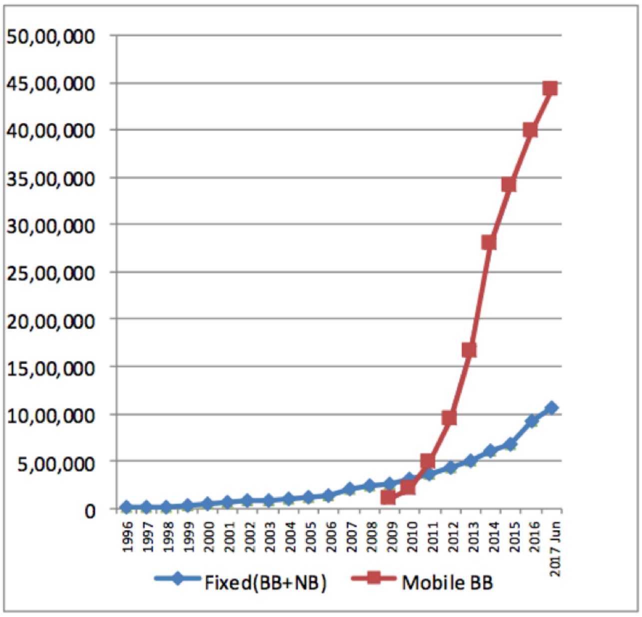


Figure 17: Cellular phone subscriptions 1997-2017 (Ref: Telecommunication Regulation Commission Statistics Division)

The number of mobile broadband users has been increasing since 2009. It was recorded nearly 4.5 million broadband users in 2017. It could be observed that there is a rapid growth in mobile broadband subscriptions since 2009. (See Figure 18)

Year	Fixed(BB+NB)	Mobile BB
1996	2,504	
1997	10,195	
1998	18,984	
1999	25,535	
2000	40,497	
2001	61,532	
2002	73,468	
2003	85,500	
2004	93,444	
2005	115,000	
2006	130,000	
2007	202,348	
2008	234,000	
2009	249,756	91,359
2010	302,000	200,000
2011	359,000	485,000
2012	423,194	942,461
2013	507,845	1,664,003
2014	606,100	2,790,195
2015	682,512	3,408,408
2016	929,089	3,991,465
2017 Jun	1,060,529	4,418,799



Note: BB – Broadband, NB – Narrow Band

Figure 18 : Mobile Broadband subscriptions from 2009-207 (Ref : Telecommunication Regulation Commission Statistics Division)

When the mobile internet usage is considered it is identified that there are nearly 6.15 Million of active mobile internet users. And when considered the share of web traffic by device mobile phones and handheld tablet devices account for 78% of the total web traffic shared by devices(44).

According to the GSMA connectivity index, Sri Lanka has an overall connectivity index of 61.1 in which the index where is from zero to hundred. The GSMA Mobile connectivity index is an analytical tool to measure the performance of mobile network connectivity in various countries. The index for a country is developed by considering the infrastructure affordability consumer readiness and content of mobile networks in that country(46).

GSMA mobile connectivity index is influenced by many indexes which represent infrastructure network connectivity, network performance indexes. Figure 19 shows a comparison of different indexes with reference to several South Asian countries and other high-income countries. All indexes range from 0-100 in which 0 is the worst and 100 is the best.

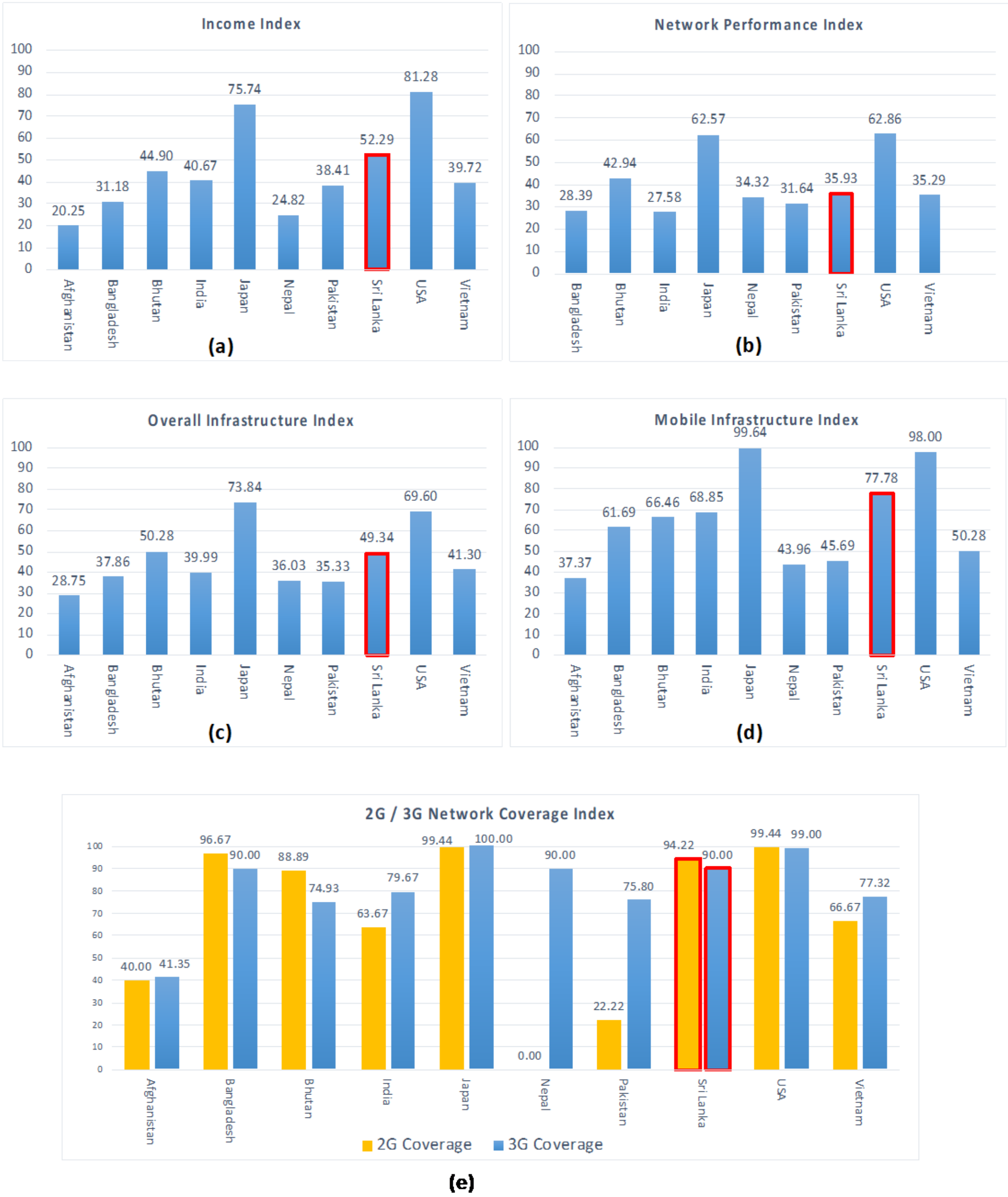


Figure 19 : GSMA Connectivity indexes

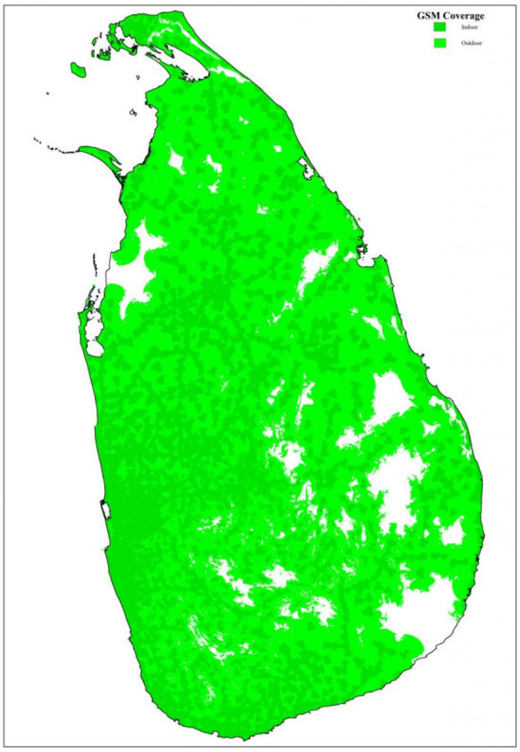
The Figure 19(a) illustrates the income index of the countries which are used in the comparison. Sri Lanka could be identified as a mean income index country with compared to the countries considered for comparison. The Figure 19(b) shows the network performance index of several countries. 25% of mobile download speeds, 25% of mobile upload speeds and 50% of mobile latencies Work considered in calculating this index. Sri Lanka could be identified as a low performing country when compared to other countries with similar income indexes. The major reason behind could be the latency of mobile networks. The Figure 19(c) shows how the overall infrastructure index varies with other countries. Access to electricity, number of secure services in the country, international bandwidth per user are some parameters which were used in the calculation of this index. When comparing the results, it could be identified that Sri Lanka is performing better than Pakistan, Nepal, Afghanistan, Vietnam and India. The Figure 19(d) shows the mobile infrastructure index. This index concerns the network coverage of each country. Sri Lanka is performing better than all other South Asian countries in this index. The Figure 19(e) shows the 2G and 3G network coverage indexes. Sri Lanka is performing very well compared to all other neighbouring South Asian countries in network coverage.

5.1.2 Signal availability of Sri Lanka

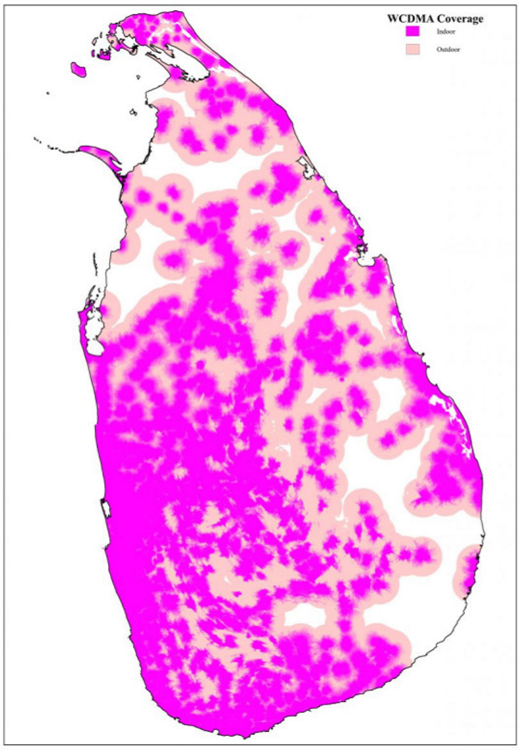
Network signal coverage is a very important parameter for crowdsourcing information. Google use cellular communication channels such as GSM, WCDMA, 4G, Wi-Fi access point and GPS signals to identify location information. Therefore, the signal availability along roads is a paramount requirement. If proper signal coverage is not available the travel time, information and prediction could be erroneous.

The network coverage maps a shared by network operators to ensure that customers are aware of the network availability in their locality. In Sri Lanka, there are five mobile network providers. They are commercially known as Dialog, Mobitel, Etisalat, Hutch and Airtel. The competitive companies have different market shares in Sri Lanka. Hence the coverage of each network provider differs with the customer base. Mobitel and Dialog have a significant share of the market. Thus, they provide higher coverage than other competitors.

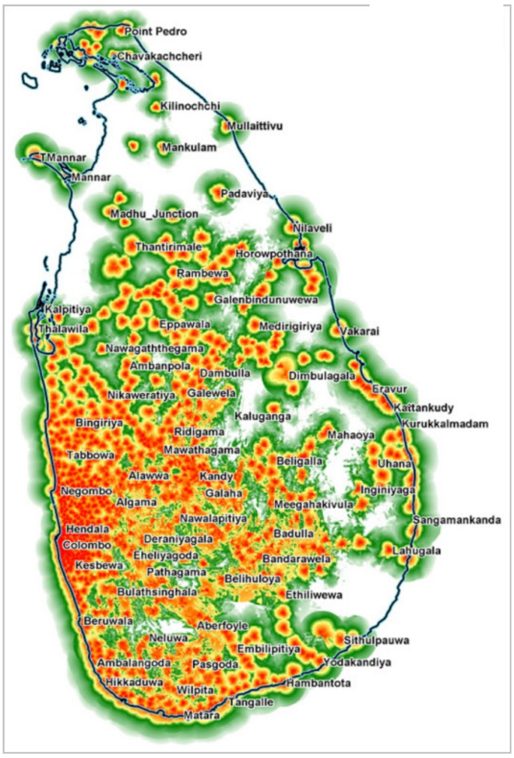
Figure 20 shows network coverage maps shared by different network operators.
The maps give the network coverage of each network provider by 2010. The network providers have not released a coverage map since then.



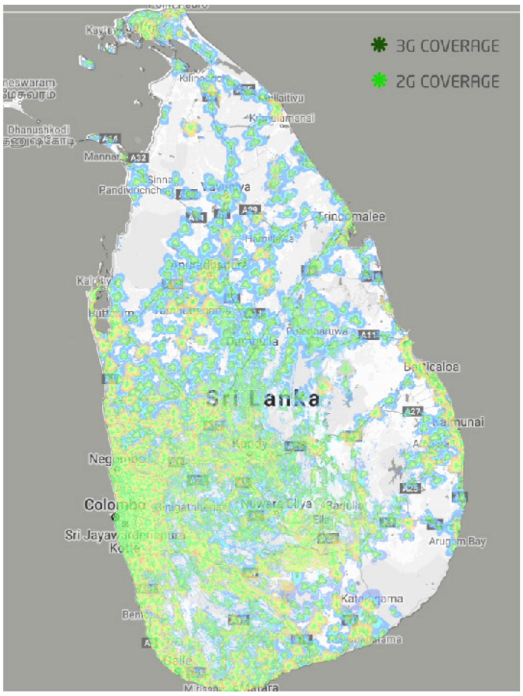
Mobitel – 2G Coverage
Ref : Mobitel(PVT)LTD, www.mobitel.lk



Mobitel – 3G Coverage
Ref : Mobitel(PVT)LTD, www.mobitel.lk

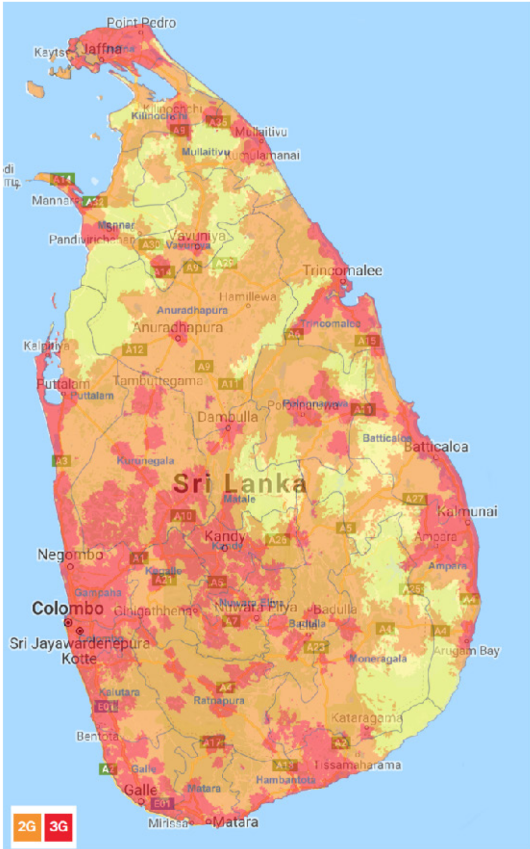


Dialog – 3G Coverage
Ref : Dialog AxiataPLC, www.dialog.lk



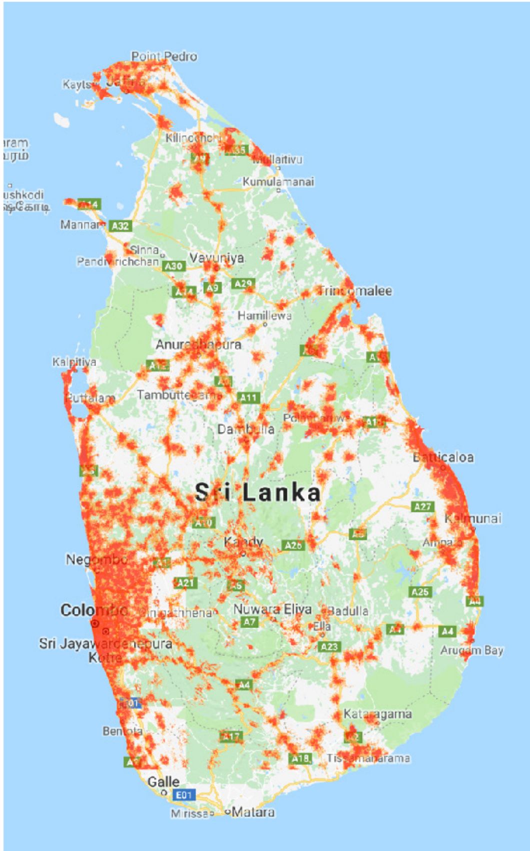
Etisalat – 3G & 2G Coverage
Ref : Etisalat Lanka(PVT)LTD, www.etisalat.lk

Figure 20 : Network coverage of Mobile network providers in Sri Lanka



Airtel –2G/ 3G Coverage

Ref : Bharati Airtel Lanka(PVT)LTD, www.airtel.lk



Hutch – 3G Coverage

Ref : Hutchison Telecommunications Lanka(PVT)LTD, www.hutch.lk

Figure 20 : Network coverage of Mobile network providers in Sri Lanka

In addition to the coverage maps given by operators, there are signal coverage maps developed by third-party organisations. Signal maps developed by OpenSignal Inc is used for the analysis due to the unavailability of recent data released by national network operators. By referring tp the maps shared by OpenSignal company, it could understand that the signal availability along major roads, at major cities and sub towns are adequate enough for mobile users to use services such as Google. OpenSignal is a company that specializes in wireless coverage map. The company collect data by crowdsourcing carrier signal quality from the OpenSignal mobile application users. According to the company statistics OpenSignal mobile applications are used by Millions of users. Hence the network coverage maps are updated every six months to ensure the quality of service. Therefore, it could be concluded that the coverage maps provided by OpenSignal are much reliable than the older maps provided by network providers.

Figure 21 illustrates signal availability map developed by OpenSignal using outsourced information. The signal along arterial roads and city centres could be found in green colour which indicates that there is a good signal availability along roads. This is the most important information for this study when considered the signal availability.

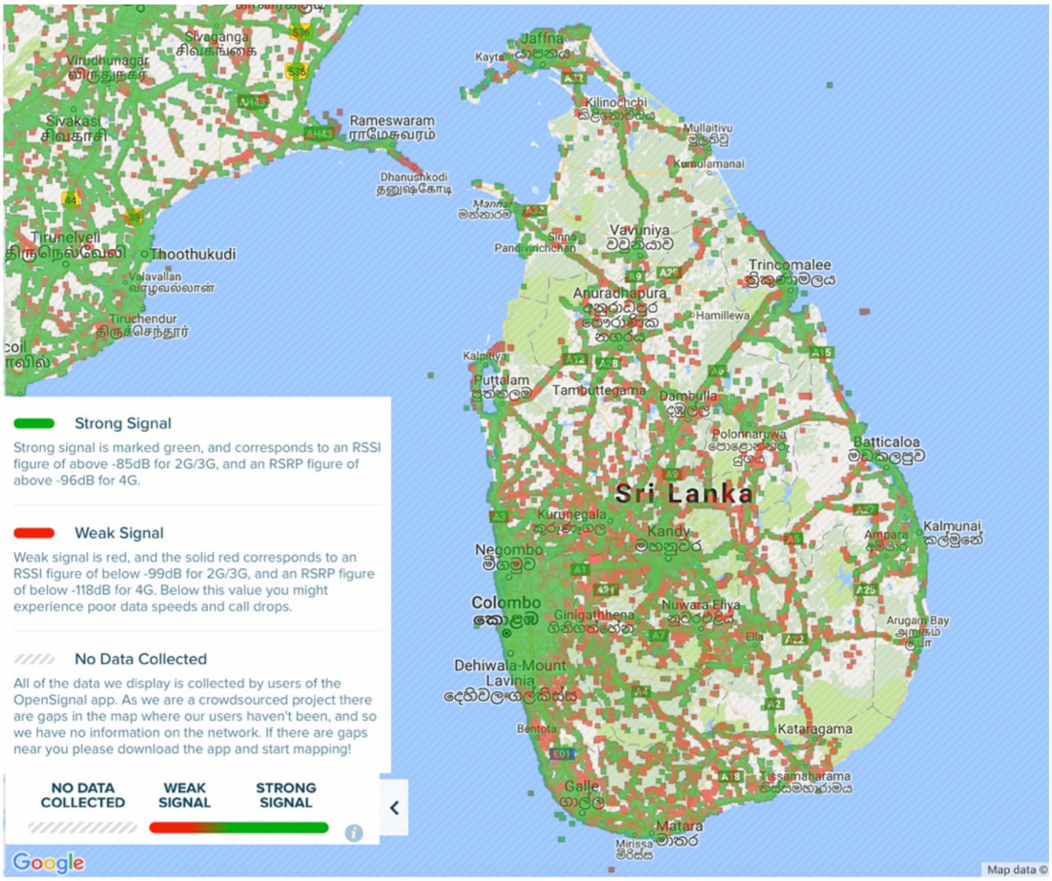


Figure 21: Network signal coverage map developed by OpenSignal Ref: OpenSignal PVT LTD (https://opensignal.com)

Figure 22 shows the signal coverage map developed by OpenSignal which shows the signal availability of different network providers in Colombo district. It could be found that Mobitel and Dialog network providers have a higher coverage in the Colombo district. Moreover, these two network providers have a majority of the market share. The network coverage of other three network providers are clustered to the city centre of Colombo Metropolitan Area. In overall evaluation, the network availability could be concluded as adequate.

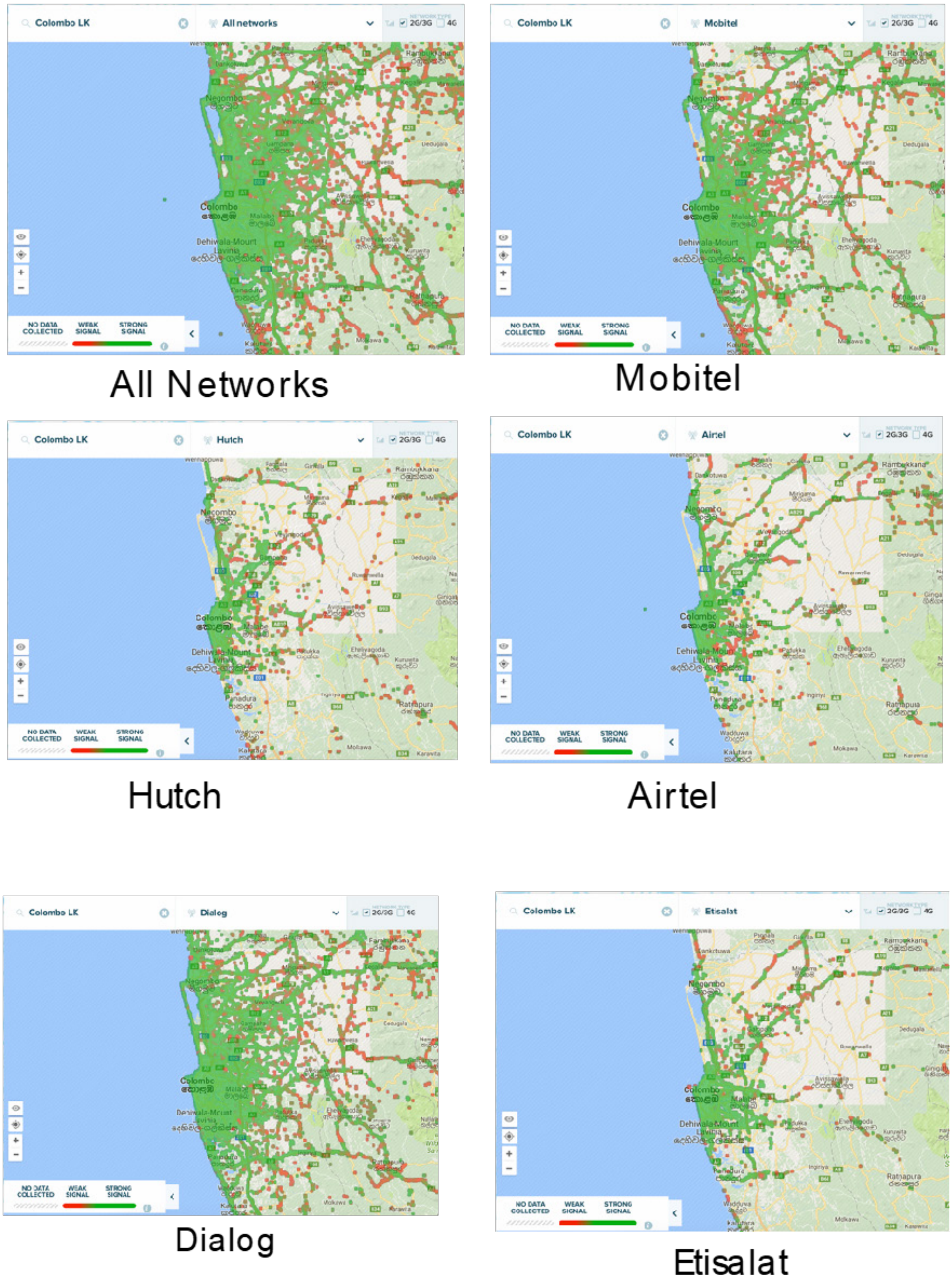


Figure 22:Network Coverage map of different network providers
(Ref: OpenSignal PVT LTD (https://opensignal.com))

5.2 Using probe vehicle techniques in verifying the Google travel time

This section will observe how the travel time information collected by Google Distance Matrix API agrees with the information collected by other modes of travel time collection. It is important to identify the significant correlation of Google travel time data with travel time data obtained from other modes. If the correlation between two travel time observations results in a higher agreement, then it could be concluded that Google travel time data has a significant accuracy in travel time collection with respect to other modes of travel time collection.

Using probe vehicle techniques to verify the Google travel time data is a feasible option in the environment which the study was carried out. Vehicles equipped with GPS sensors are used in this study as probe vehicles. A GPS data collecting device was developed by the author in order to support the data collection activities. Further Android applications which support GPS logging were used. The verification was carried out for both short distance trips and long-distance trips separately. A motor car was used in most of the time for data collection.

5.2.1 Development of GPS device to collect travel time data

A mobile device which can collect GPS location and transfer the location information to a web server was developed to collect data. A single-board microcontroller and a GSM/GPS module were connected and powered by the vehicle battery. Following boards and models where used in assembling the device.

1. Arduino Uno SMD R3 - Single-board microcontroller.
2. sim808- GPS/GSM module
3. Power controlling relay
4. Power switch, Connectors and casing

Figure 23 shows the circuit design of connecting single-board microcontroller with the sim808 module. In assembling the device, support was taken from an electronic engineering specialist. A power controlling relay was used to ensure the safety of the

GPS device from the variability in current input from the vehicle. The device was sentenced in a dust-proof casing to ensure a long-term functionality.

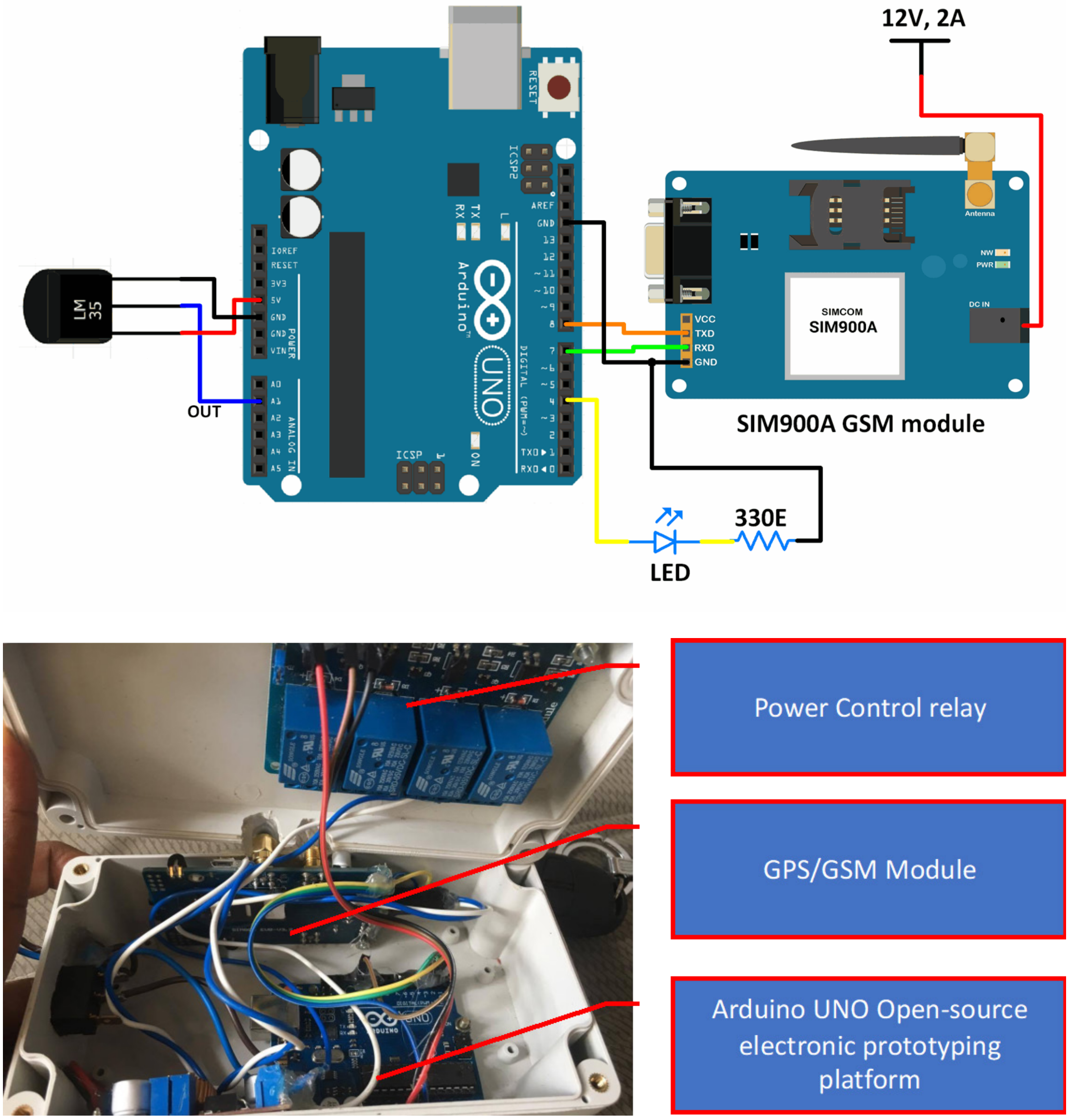


Figure 23 : Design diagram of the Arduino-based GPS Device

The sim808- GPS/GSM module used in data collection has a horizontal pointing accuracy of GPS location less than 2.5 meters. The accuracy is increased with 3G GSM signals. The GPS sensor has a maximum time to first fix the value of 30 seconds in cold start and minimum time to fix the value of 1s in continuous operation. The power consumption at sleep mode is around 1mA and at operation is around 42mA. Therefore, the accuracy of location data collection is adequate enough for the verification of travel time data.

The module was programmed to collect GPS location when it is powered. It was configured to collect GPS location at 30-second intervals and keep at continuous operation throughout the operation period. The collected data was sent to a database in the cloud server via Mobile data (GPRS/GSM). The GPRS/GSM module was kept activated throughout the data collection process to ensure the accuracy. The Arduino code is attached in Appendix B for further reference.

Figure 24: GPS Device mounted in a probe vehicle

The assembled GPS device was mounted inside a motor car, and power supply was given from the vehicle power supply. Figure 24 shows how the device was mounted. The device was mounted in between the driving seat and the passenger seat. It was fixed to the locker in between two seats by a bracket. To operate the SIM808 module mobile data package was purchased with the monthly subscription.

5.2.2 The methodology followed in collecting data

A private vehicle was used in data collection and the driver was aware about the floating car driving guidelines. The data collection trips were carried out throughout

14 months starting from January 2017. The trips were scheduled-based on the other travel purposes of the to minimize the travelling expenses and increase the convenience of data collection. Scheduling of trips was categorized-based on distance time, and type of traffic. Trips which crossed the Colombo city limits and had distance more than 50 km were considered as long distance trips. The short distance trips were limited within the Colombo Metropolitan Area (CMA), and long-distance trips covered several major towns of Sri Lanka.-based on the time of the trip daytime, and nighttime trips were selected. Peak traffic time and off-peak traffic conditions were selected as the type of traffic.

Special care was taken in driving to ensure that the motor vehicle act as a representative vehicle of the moving vehicle fleet. Overtaking manoeuvres, High-speed manoeuvres were voided, and intermediate stopping during long distance trips was limited, and the stopped duration was recorded. Round trips were considered as two independent trips as there was no significant correlation between the inward trip and outward trip.

5.2.3 Analysis to verify data with Google travel time

The data collected by the GPS device and other mobile applications were analysed under three categories. It was identified that verification is required for following categories;:

1. Evaluation of short distance trips - Peak time
2. Evaluation of short distance trips - Off-peak time
3. Evaluation of long-distance trips

A set of origins and destinations identified-based on the author's decision to collect travel time data by GPS logger and Google Distance Matrix API simultaneously. Travel time information to travel between each origin-destination pair was collected from Google Distance Matrix API continuously every day from 6 a.m. To 10 p.m. At 30-minute intervals. 78 origin-destination pairs were selected to allow the data collection within the free limit of Google Distance Matrix API. Data obtained from GPS logger was matched with Google Distance Matrix API data at the end of every week during the research period. The matched data set of travel times were used to analyze short distance trips. (see Appendix D)

Data collection for long distance trips was based on the requirement. Travel time between several towns with a distance more than 50 km were collected along the long-distance trip route. The data collection was carried out throughout the journey period at a 15-minute interval. The data collection was also carried within the free limit of Google Distance Matrix API. Travel time collected from the GPS logger data are matched with travel time of Google Distance Matrix API after the journey and a data set with manual travel time data and API data was created (see Appendix E).

5.2.3.1 Evaluation of short distance trips - Peak time traffic

The evaluation of short distance trips during peak hour traffic conditions will ensure how the travel time estimates during traffic conditions agree with actual travel time data. To evaluate the correlation between travel time, obtain from GPS logger and Google Distance Matrix API a matched data set was prepared for both long distance and short distance. Figure 25 shows a map of GPS locations in which travel time data was collected for the short distance peak traffic condition analysis

The travel time observations conducted in the morning and evening peak hour traffic conditions were filtered from the dataset and taken for analysis. Morning peak traffic conditions could be observed from 6:30 a.m. To 9 a.m. And evening peak traffic conditions could be observed from 4:30 p.m. To 6:30 p.m. Hence travel time observation belonging to these times were considered for peak hour traffic condition analysis. After filtering and omitting outliers, it was possible to collect 44 identical data points. Figure 26 shows the variation of distance for different segments which were used in the analysis of short distance peak traffic condition. There are Road segments ranging from 3 km to 62 km in the dataset which was used to analyze for short distance travel time estimates during peak hour traffic conditions. These 44 identical observations were made from 32 different road segments. The road segments cover Colombo Metropolitan Area (CMA). Segments from the main arterial corridor, segments which goes across the city limits, the expressways, highly congested roads and less congested roads are included. (see Appendix D)

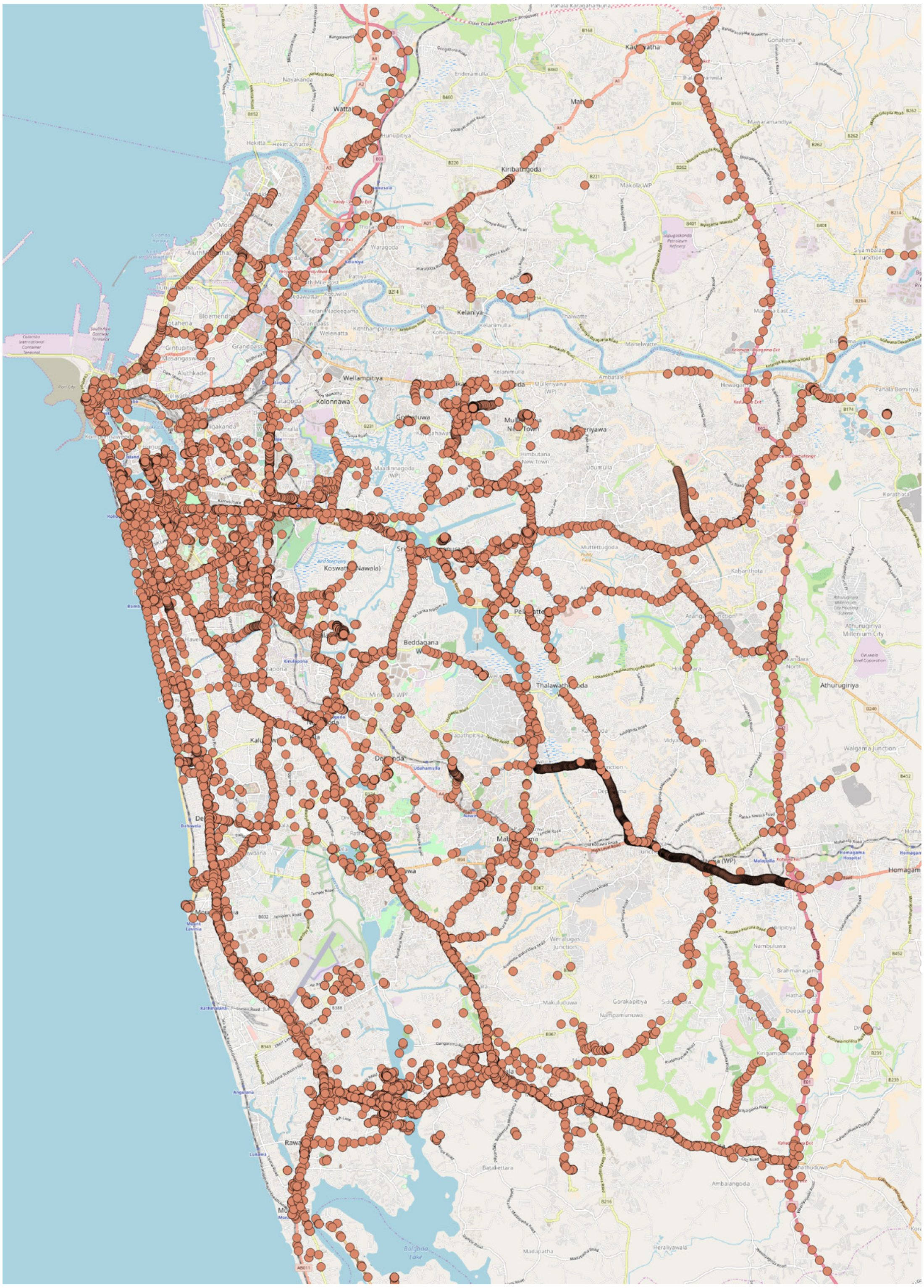
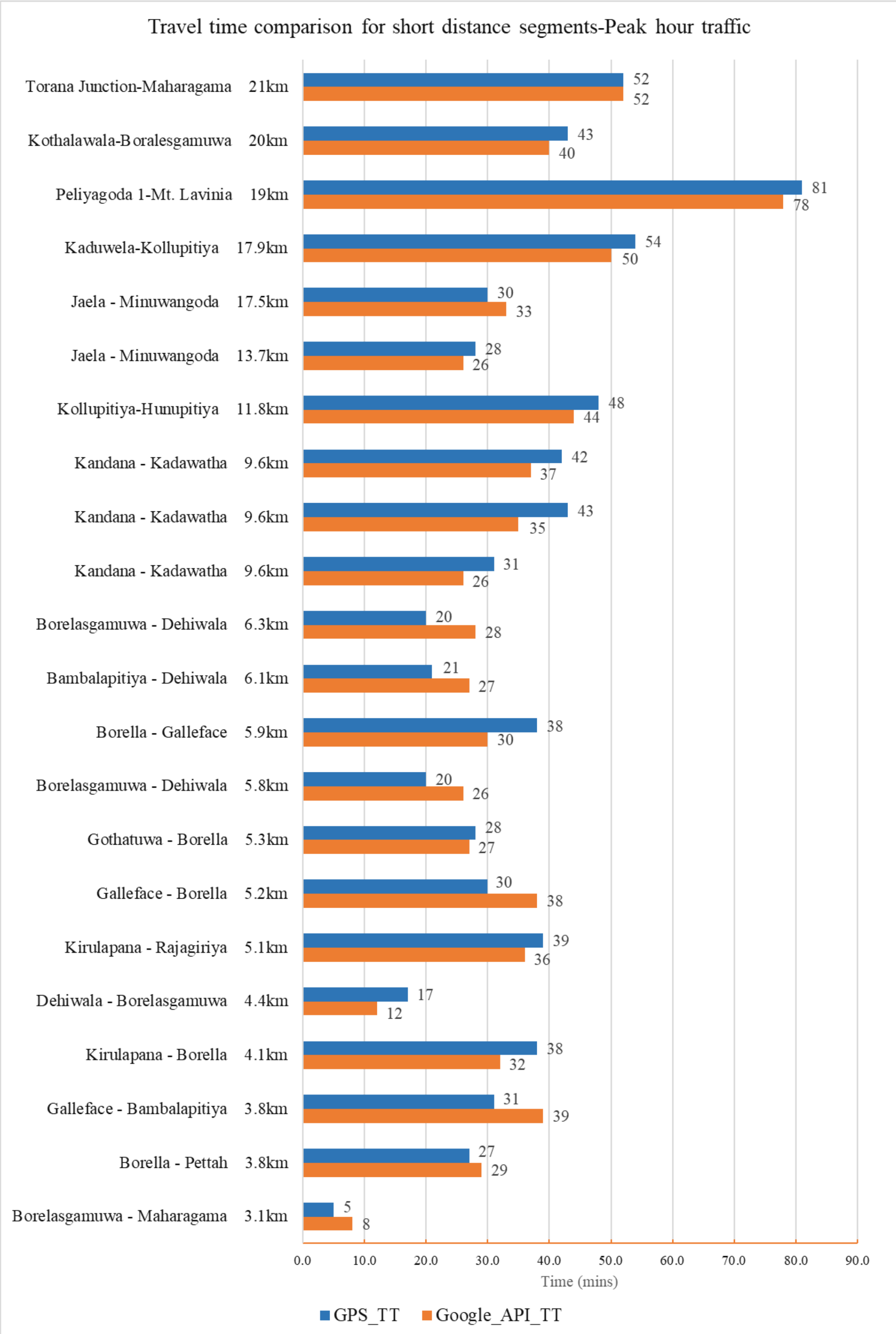


Figure 25 : Map of GPS locations use for travel time evaluation of short distance peak hour traffic condition


Travel time comparison for short distance segments-Peak hour traffic
Torana Junction-Maharagama 21km
Kothalawala-Boralesgamuwa 20km
Peliyagoda 1-Mt. Lavinia 19km
Kaduwela-Kollupitiya 17.9km
Jaela - Minuwangoda 17.5km
Jaela - Minuwangoda 13.7km
Kollupitiya-Hunupitiya 11.8km
Kandana - Kadawatha 9.6km
Kandana - Kadawatha 9.6km
Kandana - Kadawatha 9.6km
Borelasgamuwa - Dehiwala 6.3km
Bambalapitiya - Dehiwala 6.1km
Borella - Galleface 5.9km
Borelasgamuwa - Dehiwala 5.8km
Gothatuwa - Borella 5.3km
Galleface - Borella 5.2km
Kirulapana - Rajagiriya 5.1km
Dehiwala - Borelasgamuwa 4.4km
Kirulapana - Borella 4.1km
Galleface - Bambalapitiya 3.8km
Borella - Pettah 3.8km
Borelasgamuwa - Maharagama 3.1km
52 52
43 40
81 78
54 50
30 33
28 26
48 44
42 37
43 35
31 26
20 28
21 27
38 30
20 26
28 27
30 38
39 36
17 12
38 32
31 39
27 29
5 8
0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0
Time (mins)
GPS_TT
Google_API_TT

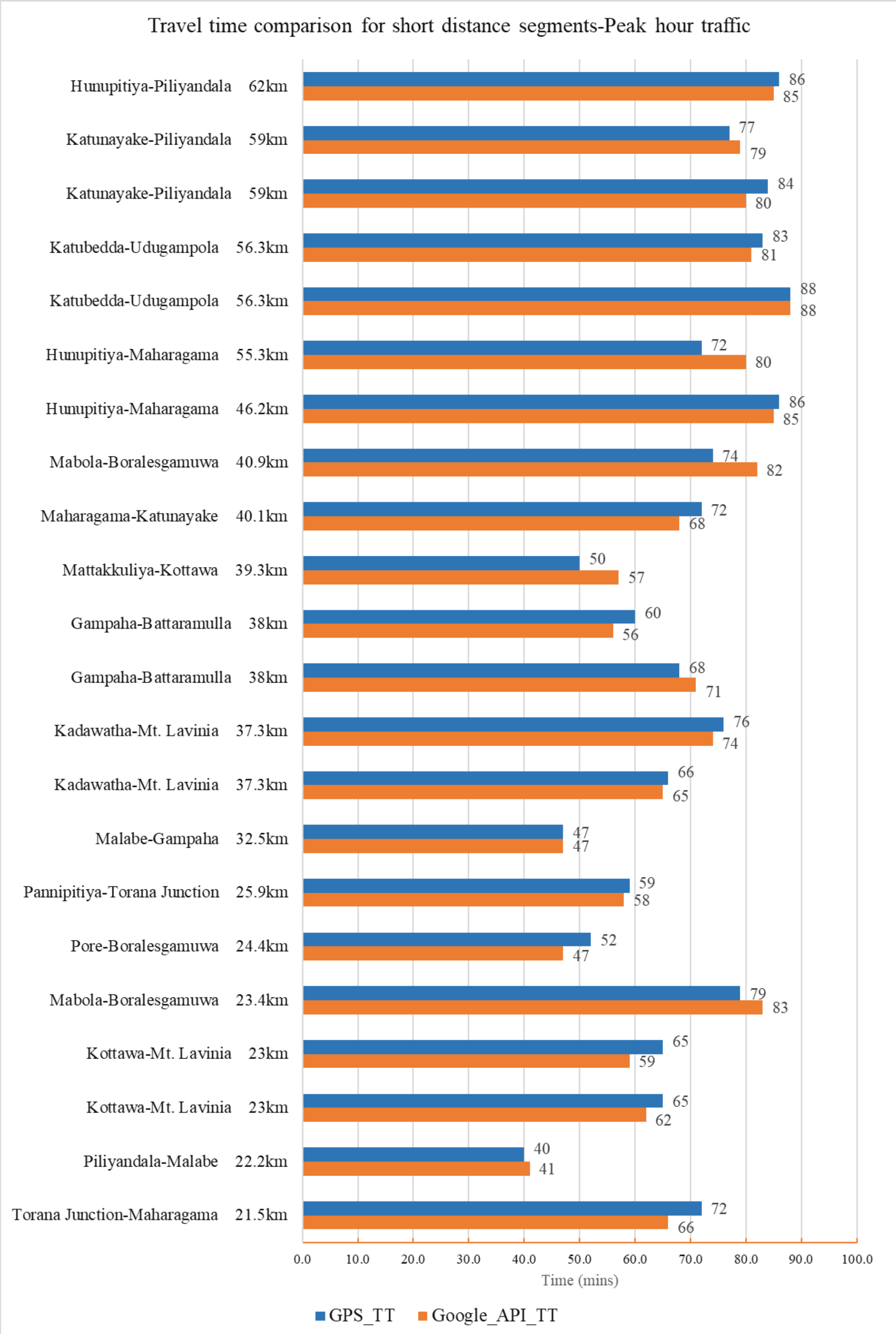


Figure 26 : Travel time Comparison of two methods, compared to segment distance – Peak traffic

Figure 26 shows a comparison of observed travel times from Google Distance Matrix API and probe vehicles. There are repeated segments with different travel time values, as these observations were made on different days and at different times. It could be observed that the travel time has a positive relationship with the segment length usually. But this relationship is being deviated in Peliyagoda -Mt. Lavinia segment due to the heavy traffic observed at the peak traffic conditions. There are two observations for Hunupitiya-Maharagama segment with two different distance values. This has caused because different routes have been selected by Google Distance Matrix API in getting the shortest travel time by considering the real-time traffic conditions.

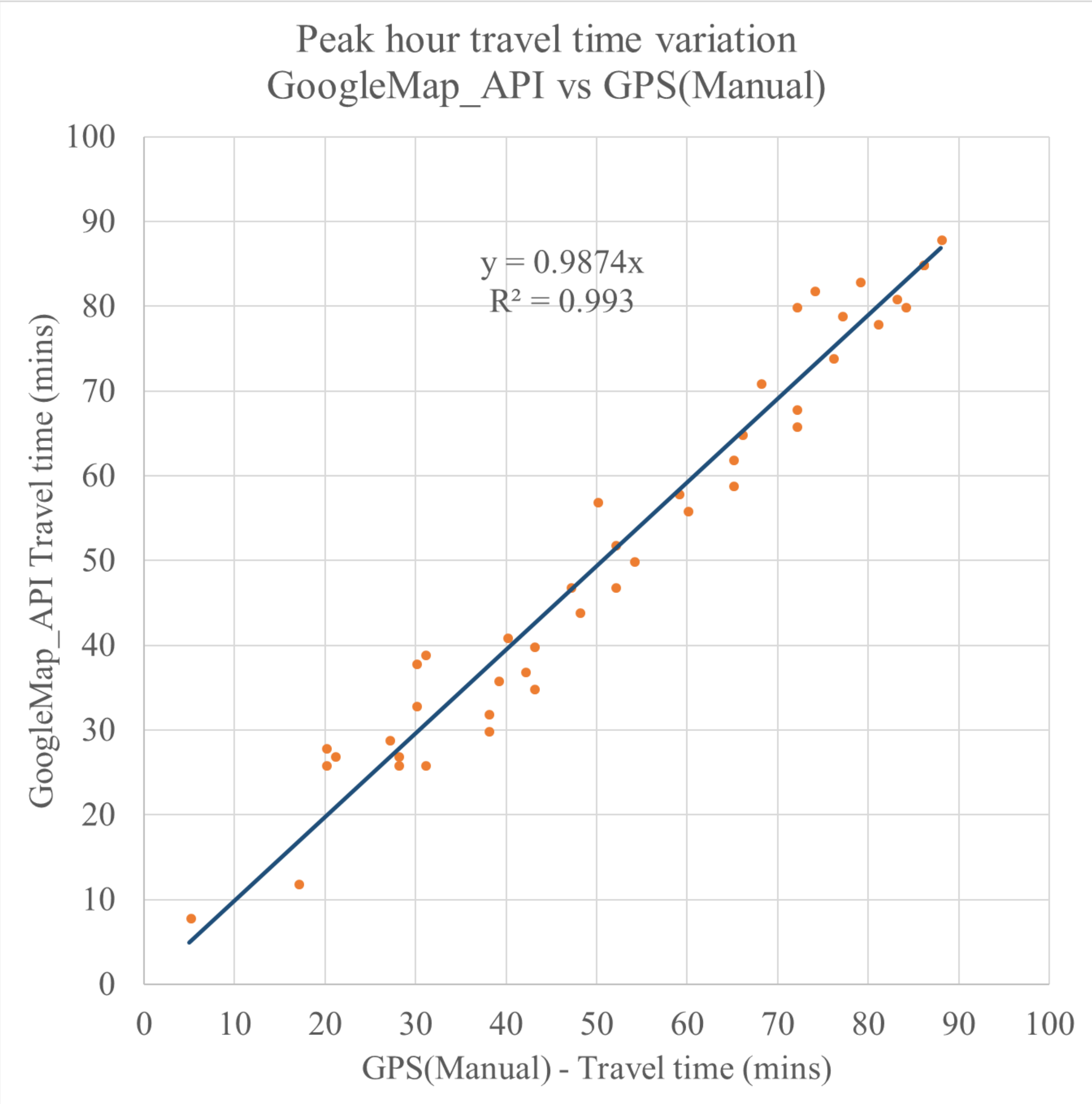


Figure 27: Comparison of Google API travel times with Probe vehicle travel times - Short Distance-Peak traffic Conditions

The Figure 27 shows a linear regression model developed to identify the explanatory power of travel time data of Google Distance Matrix API by the travel time data given by probe vehicles. The Figure 28 shows the statistical analysis conducted for further explanation of travel time comparison obtained from two methods. The analysis indicates that there is a linear unitary relationship between two travel time parameters It could be observed that the linear regression model through origin ($y = \beta x$) has a regressor coefficient (gradient) β closer to 1. Further, the null hypothesis of regressor coefficient β not equal to one is rejected under 95% confidence level which indicate that Google API data significantly agree with GPS traveltime data The R^2 value of the least square estimates having a value of 0.99 shows that there is a very high association between two parameters. The standard error of the estimate is low at 4.67 minutes which suggest the explanatory power is high. (See Appendix F) In simpler terms, the Google traffic has a 99% accuracy.

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Peak	.987	.012	.997	79.291	.000	.962	1.012

a. Dependent Variable: Google_TT_Peak

b. Linear Regression through the Origin

Model Summary[c,d]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.997[a]	.993	.993	4.67444

a. Predictors: GPS_TT_Peak

Figure 28: Statistical analysis of Google travel time and GPS travel time for short distance trips in peak traffic conditions

Therefore, by considering all these factors, it could be concluded that the travel time estimates given by Google Distance Matrix API have a significant agreement between the actual travel time observed on the road. Therefore, it is possible to use Google Distance Matrix API travel time data to analyze road segments with short distance during peak traffic conditions.

5.2.3.2 Evaluation of short distance trips – Off-peak time traffic

The evaluation of short distance trips during off-peak time traffic conditions was conducted similarly to the above analysis. It is required to conduct this analysis to identify how travel time estimates given by Google Distance Matrix API agrees with the actual travel time when vehicles are moving at a considerable speed than traffic condition. The travel time observations which does not fall within the time frame of peak traffic time was filtered from the data set and taken for analysis. 52 identical data points were available after filtering and omitting outliers from the matched dataset. The 52 identical data points were collected from 41 Road segments which lie across the study area. Unlike the road segments used for travel time analysis during traffic conditions, this data set has road segments which run from one end to another end of the western region of Sri Lanka in which the study was conducted. The Figure 28 shows the map of GPS locations used for the analysis. The length of road segments ranges from 0.8 km to 73 km. The road segments cover many arterial roads, expressways and other highways of the study area. The road segments were selected to cover Colombo district and Gampaha district. The data collection was carried out occasionally from January 2017 to April 2018. (See Figure 30)

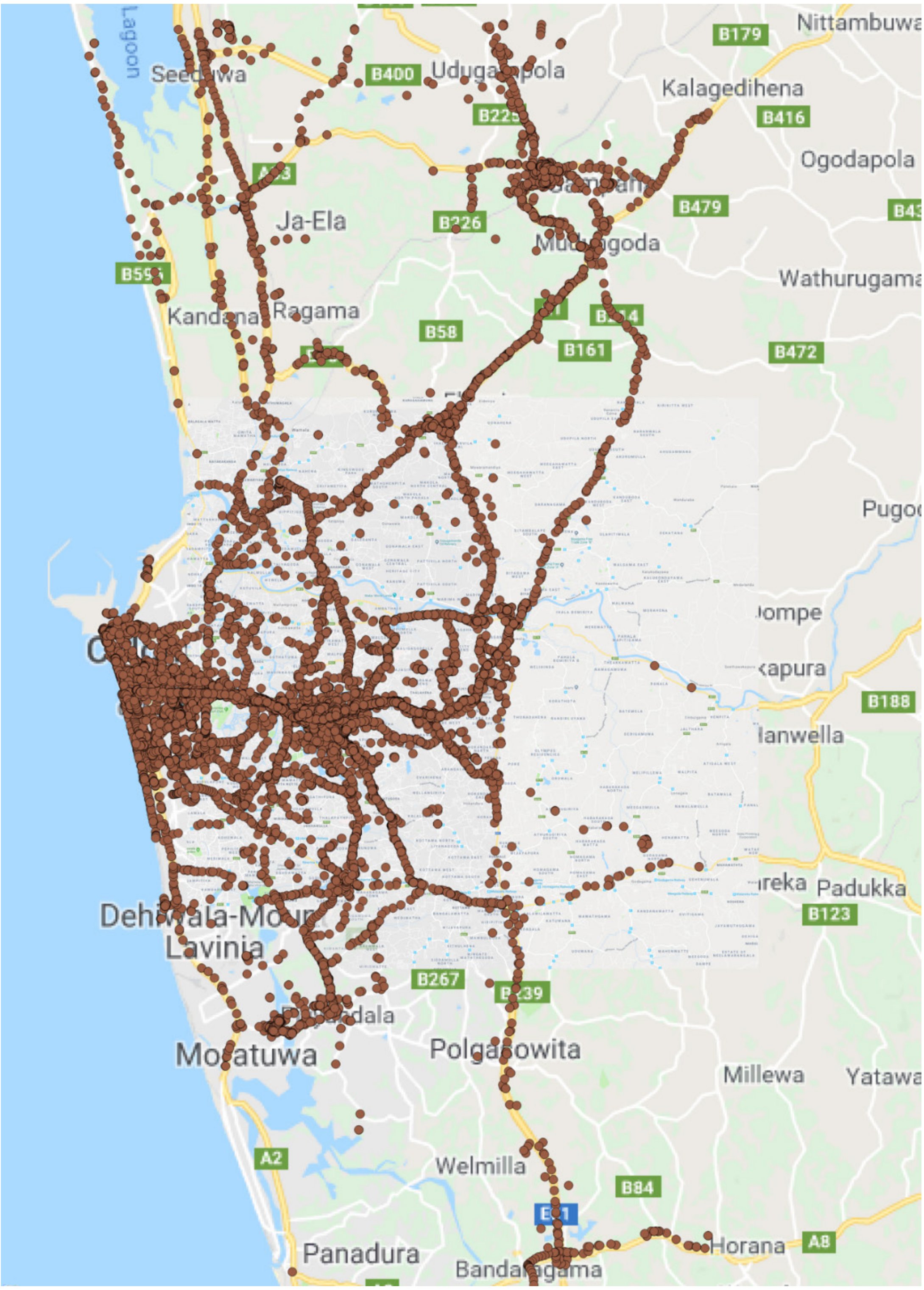
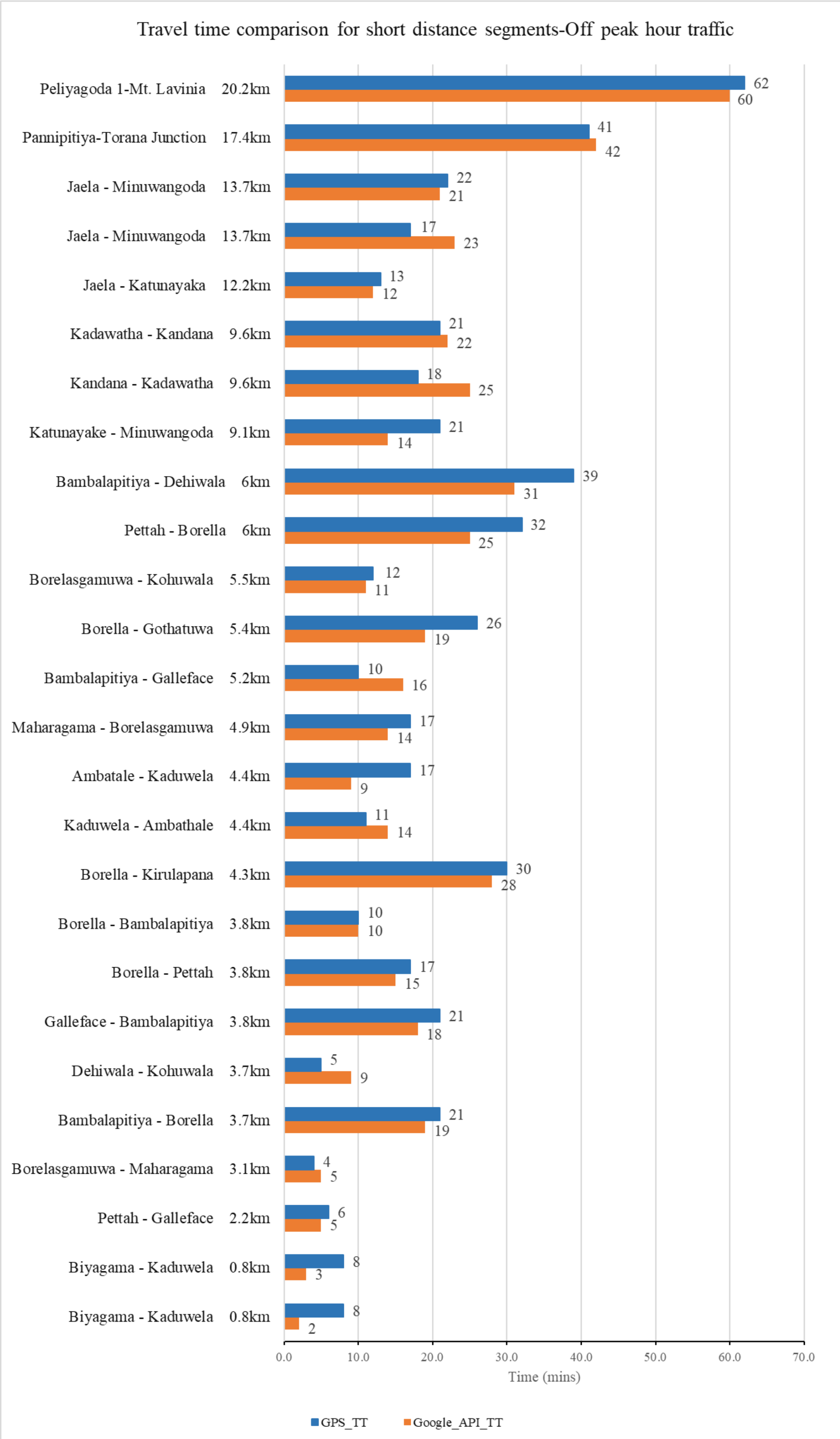


Figure 30: Map of GPS locations use for travel time evaluation of short distance - peak hour traffic condition


Travel time comparison for short distance segments-Off peak hour traffic
Peliyagoda 1-Mt. Lavinia 20.2km
62
60
Pannipitiya-Torana Junction 17.4km
41
42
Jaela - Minuwangoda 13.7km
22
21
Jaela - Minuwangoda 13.7km
17
23
Jaela - Katunayaka 12.2km
13
12
Kadawatha - Kandana 9.6km
21
22
Kandana - Kadawatha 9.6km
18
25
Katunayake - Minuwangoda 9.1km
21
14
Bambalapitiya - Dehiwala 6km
39
31
Pettah - Borella 6km
32
25
Borelasgamuwa - Kohuwala 5.5km
12
11
Borella - Gothatuwa 5.4km
26
19
Bambalapitiya - Galleface 5.2km
10
16
Maharagama - Borelasgamuwa 4.9km
17
14
Ambatale - Kaduwela 4.4km
17
9
Kaduwela - Ambathale 4.4km
11
14
Borella - Kirulapana 4.3km
30
28
Borella - Bambalapitiya 3.8km
10
10
Borella - Pettah 3.8km
17
15
Galleface - Bambalapitiya 3.8km
21
18
Dehiwala - Kohuwala 3.7km
5
9
Bambalapitiya - Borella 3.7km
21
19
Borelasgamuwa - Maharagama 3.1km
4
5
Pettah - Galleface 2.2km
6
5
Biyagama - Kaduwela 0.8km
8
3
Biyagama - Kaduwela 0.8km
8
2
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
Time (mins)
GPS_TT
Google_API_TT

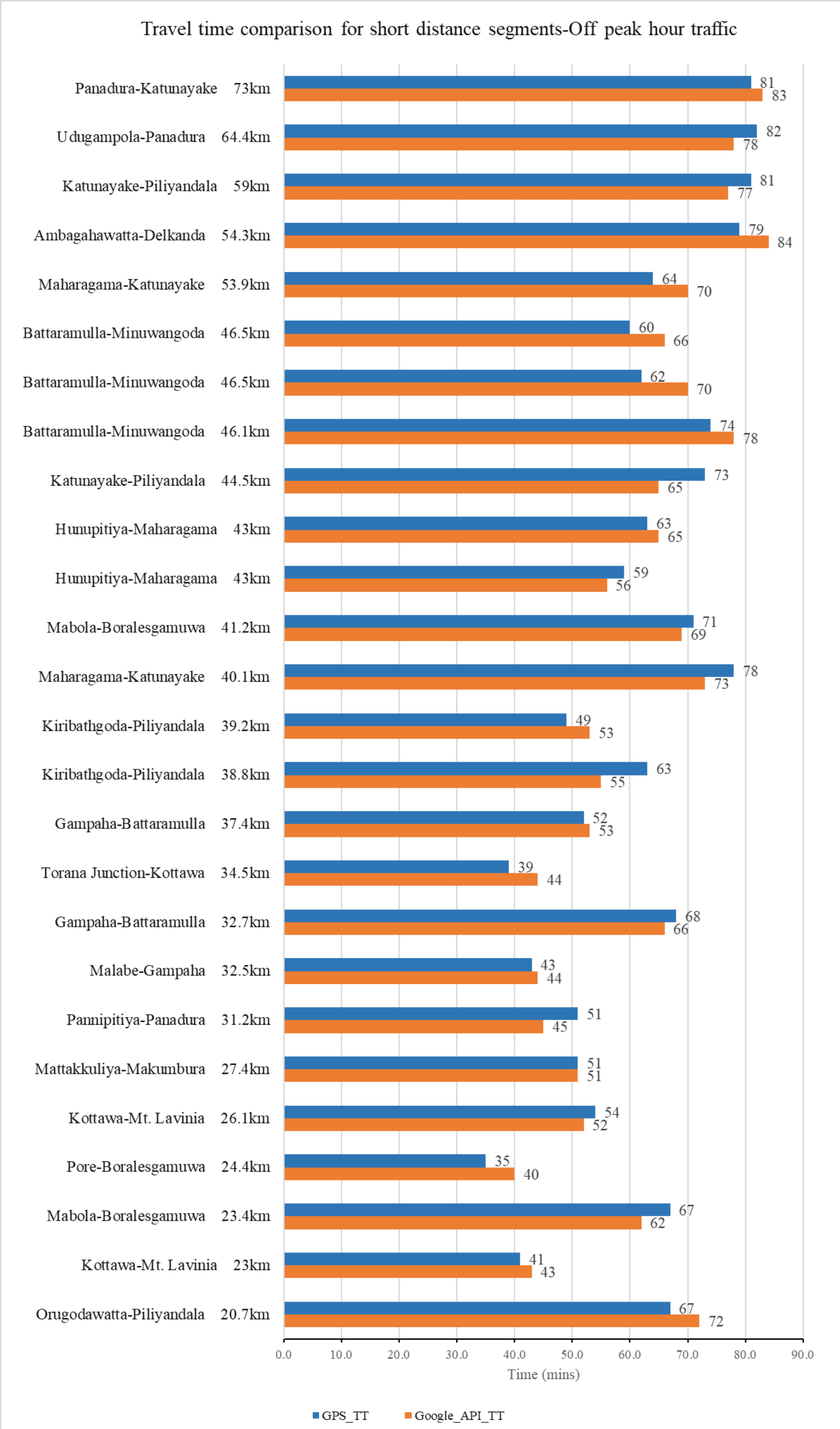


Figure 31: Travel time Comparison of two methods, compared to segment distance – Off - peak traffic

Similar to the analysis conducted for peak traffic condition, the Figure 30 and Figure 31 shows the comparison of travel time values obtained from two different methods with relevant to the distance. Maharagama-Katunayaka, Battaramulla Minuwangoda, Gampaha-Battaramulla, Peliyagoda-Mt.Lavinia, Bambalapitiya-Dehiwala Segments show relatively high travel time compared to the segments with similar distances.

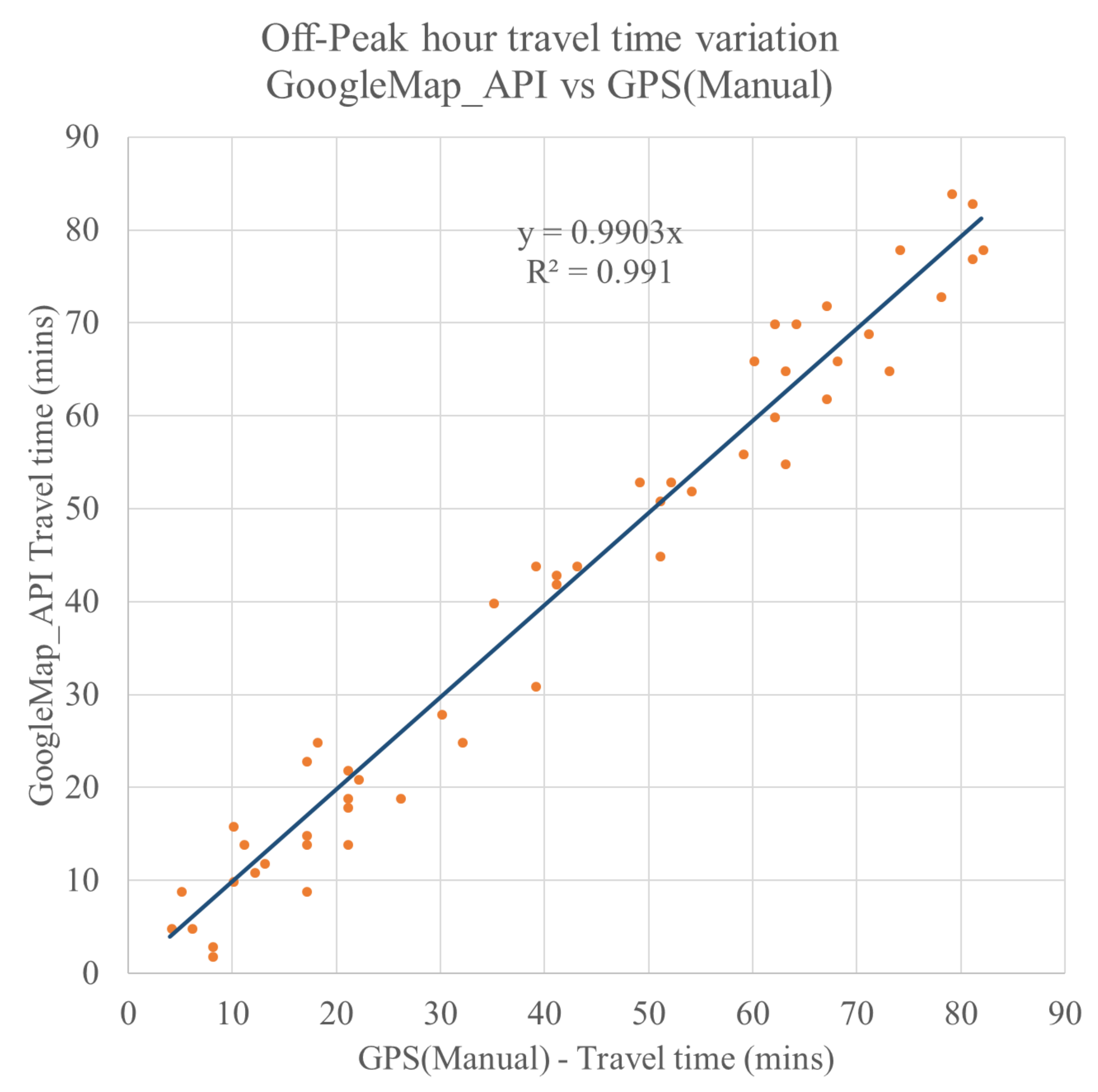


Figure 32: Comparison of Google API travel times with Probe vehicle travel times - Short Distance-Off-peak traffic Conditions

The Figure 32 shows the association of observed travel time from Google Distance Matrix API and travel time from probe vehicles. The Figure 33 shows the statistical analysis conducted by using the linear regression though the origin model in which the dependent variable is traveltime given by Google API and independent variable is traveltime given by GPS device. Similar to the short distance peak time analysis it could be observed that there is a linear unitary relationship between two travel time parameters The linear regression model through origin ($y = \beta x$) has a regressor coefficient (gradient) β closer to 1. Further, the null hypothesis of regressor coefficient β not equal to one is rejected under 95% confidence level which indicate that Google API data significantly agree with GPS traveltime data The R^2 value of the linear regression model having a value of 0.99 shows that there is a very high association between two parameters. The standard error of the estimate is low at 4.56 minutes which suggest the explanatory power is high. (See Appendix F) In simpler terms, the Google traffic has a 99% accuracy.

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Offpeak	.990	.013	.995	74.678	.000	.964	1.017

a. Dependent Variable: Google_TT_Offpeak
b. Linear Regression through the Origin

Model Summary[c,d]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.997[a]	.993	.993	4.67444

a. Predictors: GPS_TT_Peak

Figure 33: Statistical analysis of Google travel time and GPS travel time for short distance trips in off-peak traffic conditions

By considering both analyses conducted for travel time estimates during traffic conditions and off-peak traffic conditions it could be concluded that Google travel time estimate has over 99% accuracy and reliability in providing travel estimates for short distance trips.

5.2.3.3 Evaluation of long-distance trips

The evaluation of travel time estimates given by Google Distance Matrix API for long distance trips is a very important analysis in the verification process due to many discrepancies and challengers involved in obtaining an accurate estimate. Travel time for long distance trips varies due to many factors such as driving behaviour, unexpected incidents, change of weather and many more external parameters. Therefore, an initial estimate for travel time for a long-distance trip is prone to change during the trip. This study focuses on identifying variation between the initial estimate of travel time and the actual observed travel time

Long distance trips were scheduled occasionally throughout the research period. The data collection was conducted as a secondary outcome of long distance trips which were planned for other purposes such as holiday, visiting friends, delivery of goods and chauffer guiding. Therefore, the data collection was carried out on weekends most of the instances. Data collection from Google Distance Matrix API is scheduled when a long-distance trip is planned. The manual data collection was conducted by using the GPS device and mobile apps with log GPS location. Road Segments with near continuous driving was selected as there could be errors when the vehicle is stopped before its destination.

In driving the long distance trips, the driver was instructed to drive without over speeding and like an average vehicle which moves on the road. The data collection was carried out to cover different types of roads in Sri Lanka. Trips with a length more than 50 km and crossed Colombo city limits were considered as long-distance trips.

Most of the trips originated from Colombo district and covered Southern province, Eastern province Central province, North Central province and Sabaragamuwa province. Figure 34 shows a map of GPS locations which was collected during the data collection period.

A data set was prepared by matching the Google Distance Matrix API data and travel time data calculated from GPS records. After matching the data set, it was possible to obtain 32 data points of 32 road segments. The comparison of travel time obtained from two methods, related with distance is illustrated in Figure 33. It could be observed that the percentage difference in travel time increases with the distance. Travel time variation percentage is shown in red colour. It was identified that average difference between GPS logger time and Google API travel time is 20.3% , which indicates that there is a 79.7% average accuracy rate.

The Figure 36 illustrates linear regression model developed to identify the Association between travel time estimates given by Google API and observed travel time. The Figure 37 shows the statistical analysis conducted by using the linear regression through the origin. The R^2 value of 0.96 indicates that there is a good association between two parameters. The standard error of the estimate is at 40.9 minutes which suggest that the estimater faces challenges in explaining the predictors. Unlike in short distance analyses, the null hypothesis of regressor coefficient β not equal to one is not rejected under 95% confidence level which indicate that there is a slight under estimate in Google travel time and the GPS traveltime has experienced several errors. (See Appendix F)

By considering all these factors, it could be concluded that the travel time estimate given by Google Distance Matrix API has an accuracy of 80% over the observed travel time. This is a reduced accuracy compared to short distances. By considering the factors which affect long-distance travel estimate the accuracy given by Google Distance Matrix API could be considered as a satisfactory estimate.

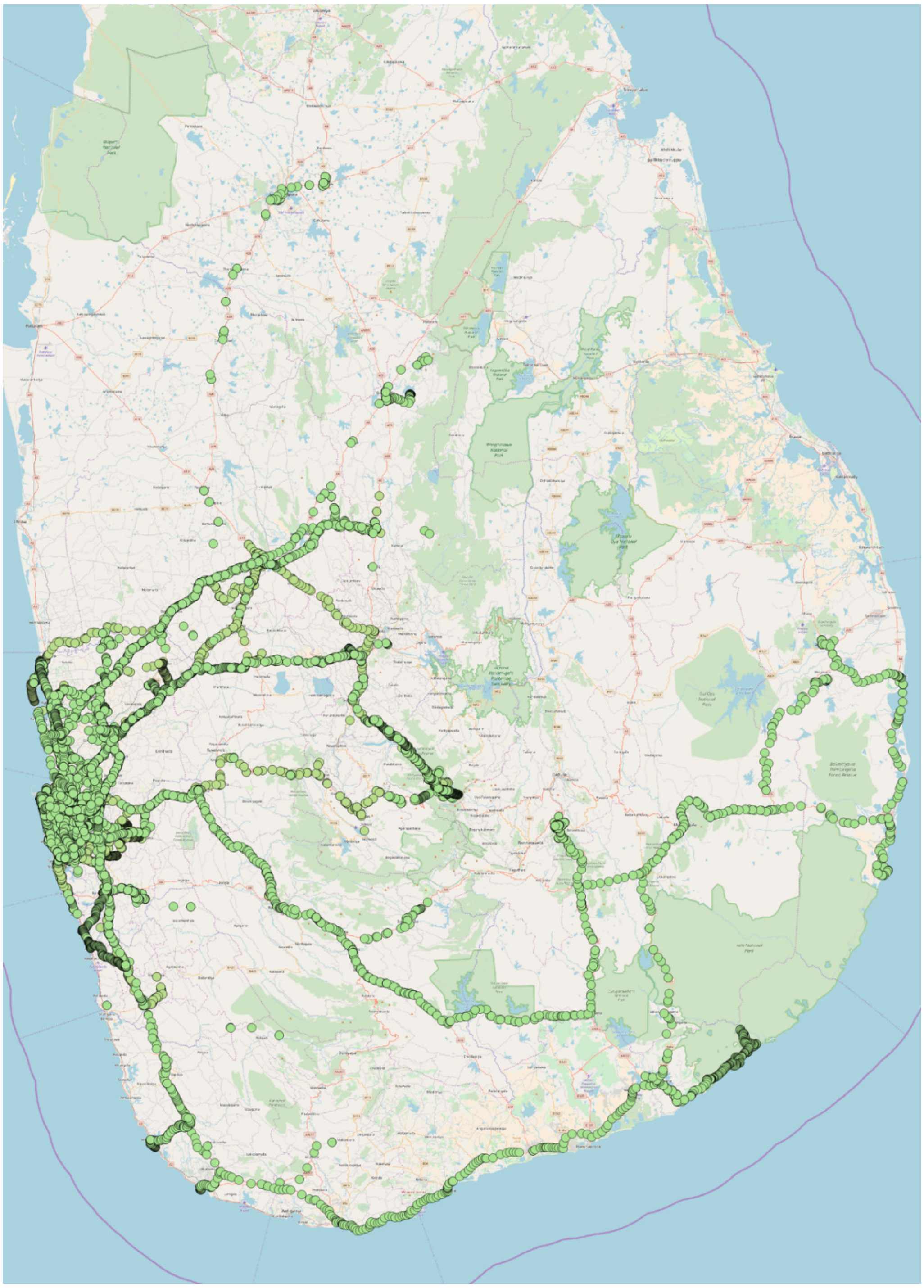


Figure 34 : Map of GPS locations use for travel time evaluation of long-distance trips

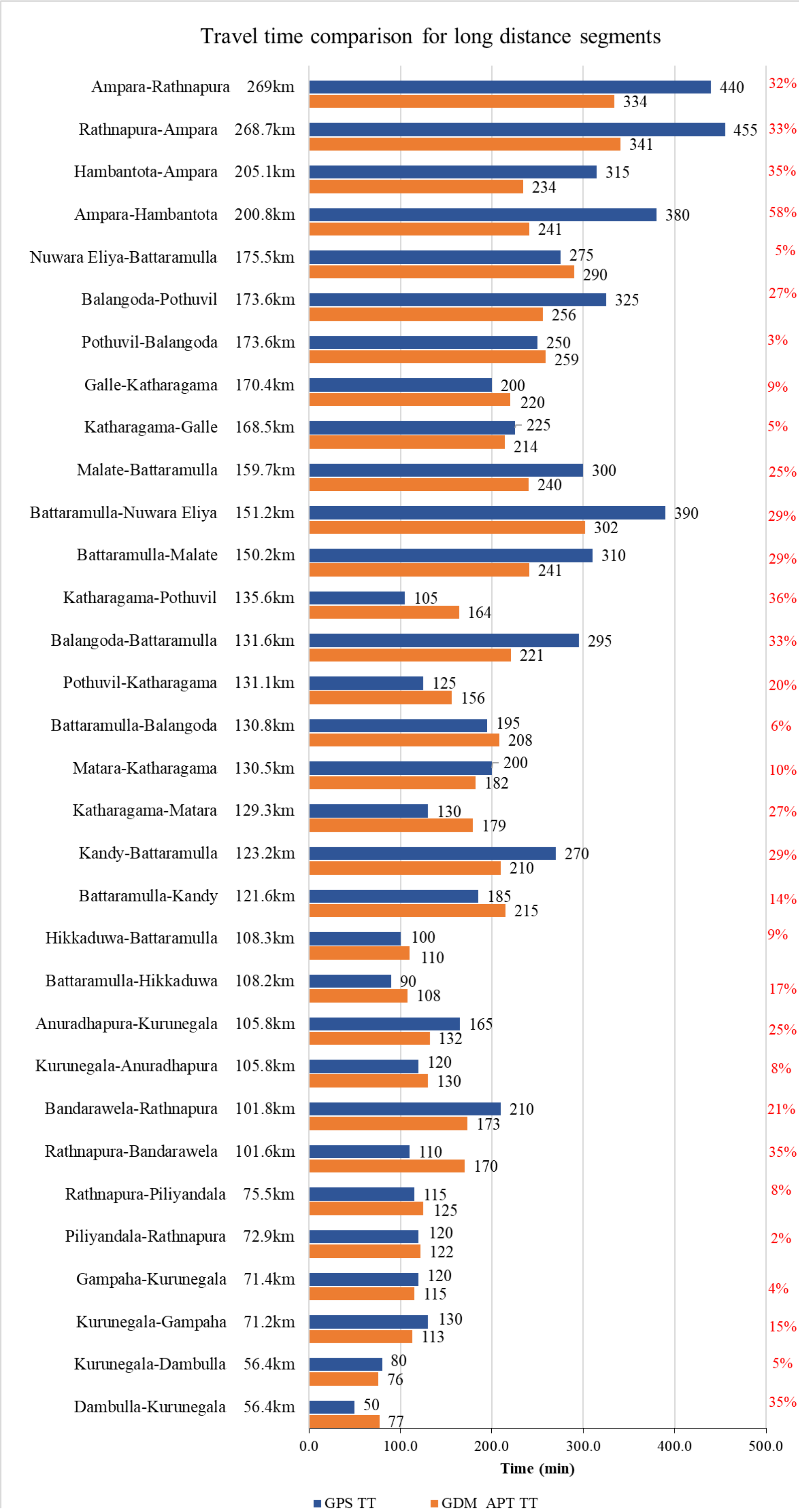


Figure 35:Travel time Comparison of two methods in Long Distance trips

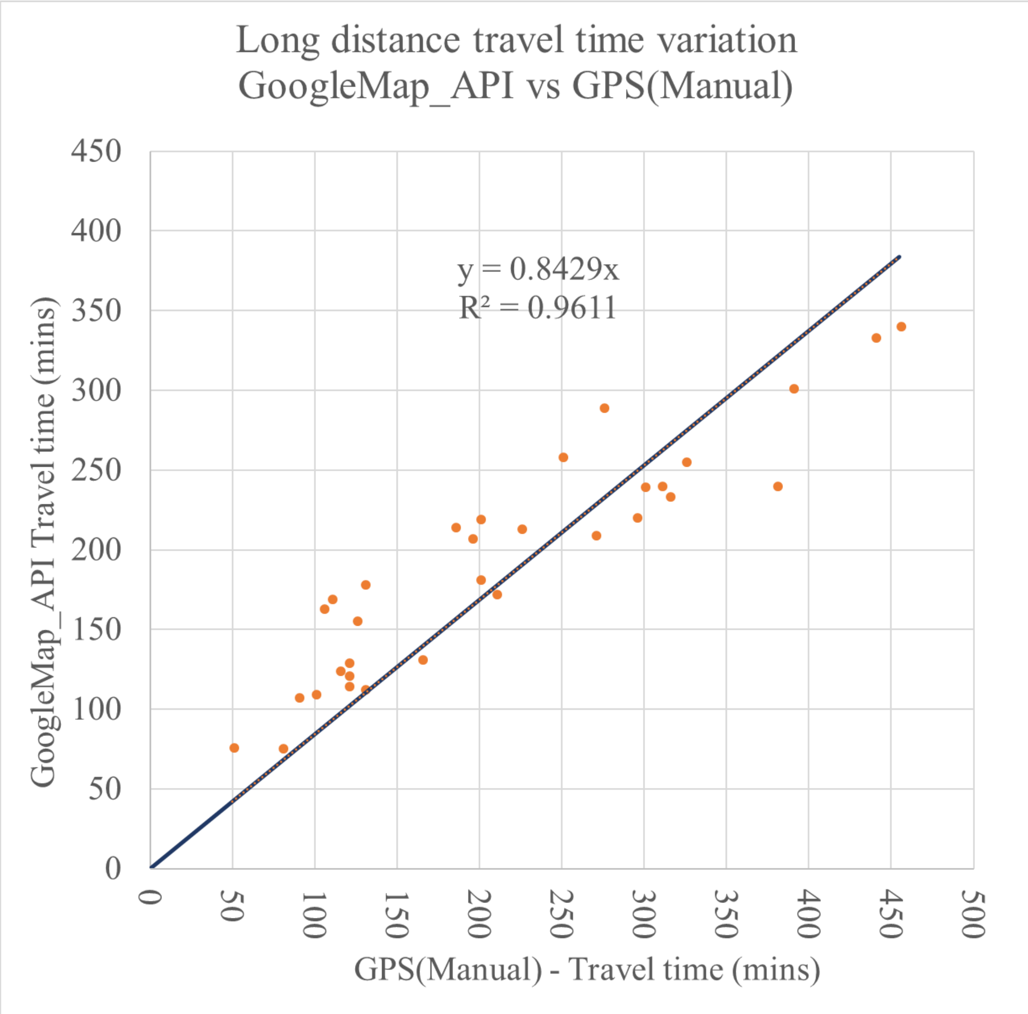


Figure 36: Comparison of Google API travel times with Probe vehicle travel times - Long Distance

Coefficients[a,b]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.	95.0% Confidence Interval for B Lower Bound	95.0% Confidence Interval for B Upper Bound
1	GPS_TT_Long	.843	.030	.980	27.728	.000	.781	.905

a. Dependent Variable: Google_TT_Long

b. Linear Regression through the Origin

Model Summary[c,d]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.980[a]	.961	.960	40.96725

a. Predictors: GPS_TT_Long

Figure 37: Statistical analysis of Google travel time and GPS travel time for long distance trips

5.3 Verification of Google travel time data for different vehicle types

The time taken by different vehicles types to travel the same distance is different from each other. Verification of travel time for short distances and long distances were carried out by using a representative vehicle from the vehicle fleet. The driver has taken necessary care to make sure that the probe vehicle represents the average moving vehicle. In a real scenario, different vehicle types have different methods of movement. For an example Travel time taken by motor three-wheeler to travel 1 km is different from the travel time taken by a motor car to travel the same distance on the same road at the same time. This may be due to many reasons such as driver behaviour, engine capacity, existing traffic, road parameters etc.

Google Distance Matrix API gives travel time estimates-based on the mobile phones moving on the road. It is not possible for Google to know the moving vehicle type as it tracks removing mobile phones and not vehicles. Therefore the travel time estimates given by Google Distance Matrix API is representative travel time for all the moving vehicles on the road. If a user is navigating with Google Maps, then the travel time estimates get adjusted to user's moving speed. But when the travel time is collected using the API duration in traffic parameter is representative value estimated by considering the real-time traffic and historical data.

It is required to identify how the travel time and space mean speed of different vehicle types vary with the travel time values provided by Google Distance Matrix API. In order to carry out this analysis, the average space mean speed of different vehicle types has to be identified. Therefore license plate matching survey was conducted.

5.3.1 The methodology of license plate survey

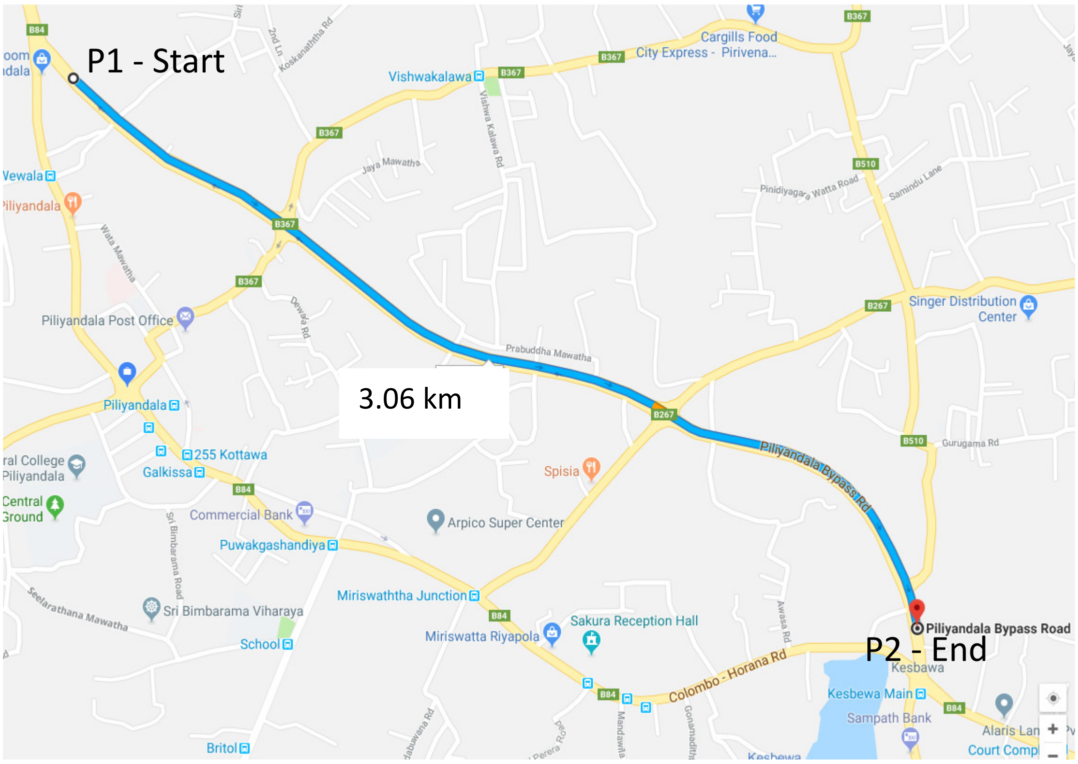


Figure 38: Map of the road used for license plate matching survey
Ref: Google Maps (www.maps.Google.lk)

The license plate survey was conducted on a road segment by collecting license plate numbers at two known locations along the road segment. The data loss is very high if license plate data collection was carried out by manual methods such as manually writing down the license plate number. Therefore, in this study, a digital method of collecting data was utilized by photographing the moving vehicles at 1-second frequency. The data loss in the collection could be reduced by this method.

Figure 38 shows the map of the road which was used to collect license plate numbers. The selected road segment is a speed bypass road to BB084-Colombo Horana road. The road starts before the Piliyandala junction and reconnects to BB084 at Kesbewa. This road has only two connecting roads within the whole distance of 3.06km. Therefore, this road segment is an appropriate road to conduct a license plate matching survey as there is limited access to the road and mobility is higher compared to other roads.

Figure 39: Identification method of license plates

The Figure 39 shows how license plate could be identified from a high-resolution photograph taken at the license plate data collection point. Although photographs are capable enough of reading the license plate, there were no accurate image recognition techniques suitable to Sri Lankan condition which can transcript license plate numbers. Hence manual transcription was used. The license plate read from the photographs were taken into a data sheet with the timestamp of the photograph. Then the data sheets of two data collection point much together by matching the license plate. The time difference between the timestamp of the photograph was considered as the travel time.

In the identification of license plate numbers, only the vehicles within and the demarcated area was considered as shown in Figure 39. The vertical lines indicate the width of the road while slanted lines demarcate the distance from the reference points along the road. License plates which fall in the photographs within 2 meters from the reference line are only selected. In Figure 39, the license plate of the motor car falls within the reference area, and the license plate of the three-wheeler does not fall within the reference area. Hence the license plate of the motor car was only recorded with the timestamp of the photograph. The license plate of the Three-wheel was recorded in the earlier photograph which was taken 1 second before the current photograph. The accuracy of getting the correct timestamp was increased in this way.

5.3.2 Results obtained and analysis

The data collection was carried out for 90 minutes from 3:30 pm to 5:00 pm on a weekday. The capacity of the camera was a limiting factor. After matching the license rates, it was possible to detect 198 vehicles which travelled in between the two data collecting points continuously. The number was adequate enough to carry out the analysis as the sample collected was representative of the vehicles which moved along the road during the data collection period.

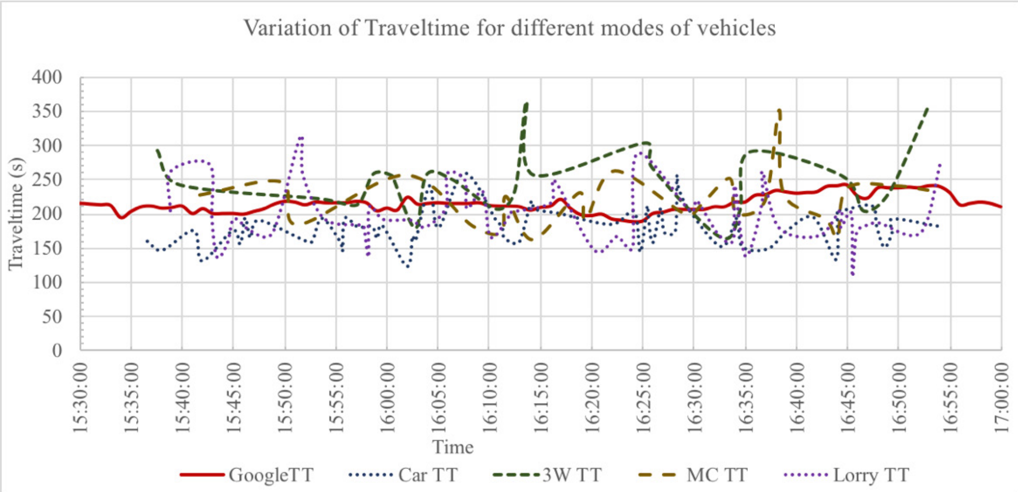


Figure 40: Temporal variation of travel time for different vehicle types

The Figure 40 shows temporal variation of travel time for several types of vehicles undular pattern is due to the different driving behaviours of different drivers in driving the similar type of vehicle. The undular behaviour is comparatively high in motor cars. This made you today availability of motor cars in different engine capacity is, and it was the highest percentage of vehicles in this sample. The variation of Google travel time which is indicated in red colour line is the most important thing to observe. It lies in between undular wavy curves of different vehicle types. This mid-range movement indicates that Google travel time is representative of the vehicles moving on the road.

Table 5: Statistical analysis of travel time and space mean speed for different vehicle

Vehicle			Travel time (s)		Speed (km/h)	
Vehicle Type	**Number**	**Composition**	**Mean**	**ST.Dev**	**Mean**	**ST.Dev**
Motor Car	60	30%	204.8	85.4	59.8	13.5
three Wheelers	23	12%	250.8	49.3	45.6	8.8
Motor Bicycle	23	12%	221.2	40.8	51.4	8.7
Motor Jeep	13	7%	181.3	22.5	61.7	6.7
Motor Van	26	13%	186.2	36.8	61.7	12.8
Lorry /HV	53	27%	214.6	90.7	56.4	14.3
Total	198	100%	209.8	54.3	56.1	10.8
Google Data			**213.5**	**12.3**	**51.9**	**2.9**

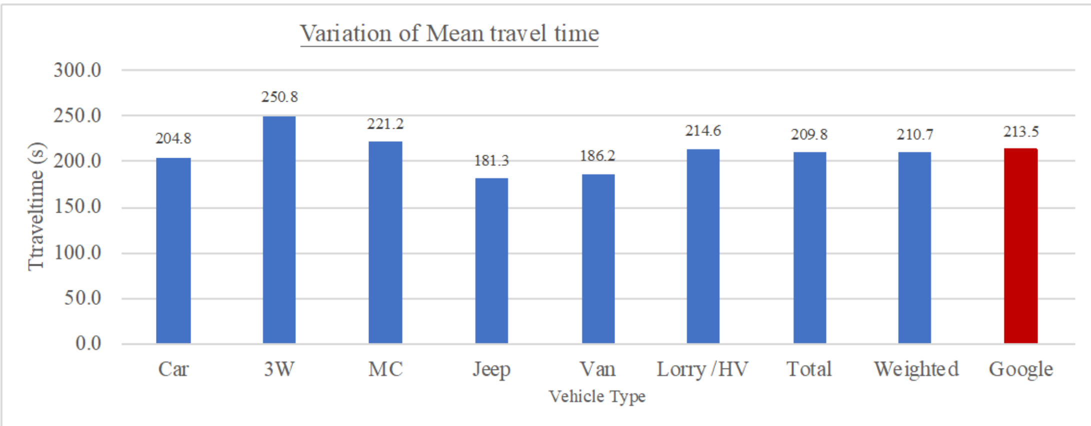


Figure 41 : Variation of travel time for different vehicle types

The Table 5 shows the mean value and standard deviation of travel time and space mean speed for different vehicle types. In the vehicle sample the highest percentage was represented by motor cars and heavy vehicles. The lowest available in the sample was motor jeeps. The three-wheelers and motorcycles were present in equal numbers. It could be observed that the highest mean travel time was recorded for three wheelers while lowest travel time was recorded for Motor Jeeps. Heavy vehicles which has a higher composition in the sample shows a closer travel time of 214.6 seconds to that of Google travel time data. Heavy vehicles and motor cars shows higher variation in travel time having high standard deviation values. This could be due to the availability of vehicles with different engine capacities and each vehicle type. The average travel time calculated by considering all vehicle types deviates by 3 seconds from travel time given by Google Distance Matrix API. The variation of travel time for different vehicle types as shown in Figure 41. By going deep into the analysis, the Figure 42 shows percentage difference in travel time of different vehicle types from Google travel time. It could be observed that three-wheelers have the highest variation from the Google travel time which account for 17%. The average travel time variation from Google travel time is only 2%. These values are based on the sample collected. The travel time given by Google is based on the movement of smartphones along the road in real-time. Therefore, the accuracy is expected to be high and be representative of the vehicle composition. By analyzing all these facts, it could be concluded that Google travel time estimates are representative value of the vehicle fleet which is moving on the road.

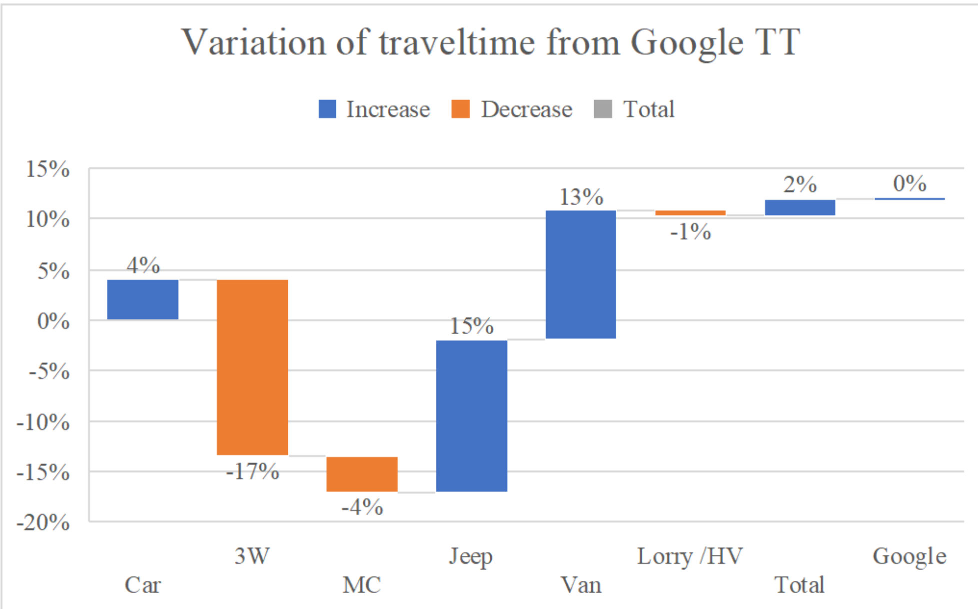


Figure 42: Percent change of travel time from API travel time for different types of vehicles

6 Applications of travel time data obtained from Google Distance Matrix API

The congestion in urban road networks is a common problem across all urban centres. Understanding the traffic flow across the road segments are necessary to provide viable solutions, but a very expensive task especially for developing countries. This section proposes some economical approaches to understand traffic flow parameters by using the Google Distance Matrix API. In earlier chapters, the methodology was proposed and verified. Thus, this chapter will focus on the applications. Following applications are discussed:

1. Traffic flow estimation urban roads-based on Google travel time data and machine learning principles.
2. Identification of road bottlenecks along corridors using Google travel time
3. Using Google travel time data for evaluation of transport projects

6.1 Traffic flow estimation urban roads-based on Google travel time data and machine learning principles.

6.1.1 The requirement of the study

Traffic flow is a key contributor to traffic management and control systems. Availability of traffic flow data for urban road networks will enable planning and decision making efficient for transport planners and policymakers. However, the collection of consistent traffic flow data for an urban road network in developing countries has been challenged due to funding gaps. In most of the developed countries, the flow data could be obtained from traffic sensors and surveillance systems embedded in road networks, which become an expensive alternative for developing countries to prioritize.

With the improvement of intelligent transportation systems, traffic flow prediction has become a major consideration [3]. Communication and detection methodologies have phased up with the advancement of crowdsourced data mining, allowing more readily extractable information on transport and mobility.

It was understood in earlier chapters that the use of crowdsourced data has become more consistent and reliable as it enables to collect a large number of data samples continuously. The investment in infrastructure is minimized, resulting crowdsourced data as a low-cost alternative since the crowdsourced data collection techniques are-based on consumer services such as telephone calls, GPS navigation, Geotagged data transfer etc . The chapter four describes how Google Distance Matrix API provides travel time for a given origin and destination-based on the above concept.

The objective of this study is to develop a non-parametric non-linear traffic flow estimation model-based on K- nearest neighbour regression, which uses travel time and speed data obtained from Google Distance Matrix API and road geometry data. The study includes a detailed literature survey, methodology identification, data validation and results of the analysis model.

6.1.2 Literature review

Traffic Flow Estimation

Traffic Flow Rate (q) is defined as a number of vehicles passing a point on space during a unit period of time, which is given by equation 1 (47).

$$q = \frac{n}{T} = \frac{n}{\sum_{i=1}^{n} h_i} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} h_i} = \frac{1}{\bar{h}} \quad (1)$$

q = flow/volume;
n = Number of vehicles;
T = time duration;
h = time head way;
$\bar{h}$= mean time headway

In 1936, Greenshield assumed a parabolic flow-density relation corresponding to a linear speed-density relation (47). Improving from that, transport specialists developed different mathematical models for uninterrupted traffic flow considering macroscopic and microscopic traffic flow characteristics (47). Microscopic traffic simulators [e.g., MITSIMLab, AIMSUN, VISSIM] involve detailed models of driver behaviour, comprising car-following, gap-acceptance, lane-changing, and other disaggregate behavioural models (47). In developing countries, due to the existing heterogeneous traffic nature and complexity in obtaining data, it is difficult to use microscopic simulations(47). Vehicle distribution of developing countries shows a higher percentage of motorized two-wheelers and three-wheelers compared to four-wheelers (48). Hence traffic flow characteristics are much different when compared with developed countries which have more than 80% motor cars in traffic flow [7]. Further, it could be observed that two-lane roads are abundant in developing countries (48). In two-lane roads lane changing and passing manoeuvres typically performed when sight distance and gaps being available in the opposing traffic stream (48). Hence the directional flow is influenced by opposite flow which has to be taken into consideration when characterizing directional flow(48).Chandra has found that road width, shoulder width, and directional split are significantly affected by the free flow speed and capacity of two-lane highways (48). Moreover, classical methods of traffic flow analysis omit the temporal variation of traffic flow (49). Although incidental analyses are possible, analysing long-term behaviour is challenging with classical method. Hence researchers looked at time series analysis and machine learning principles to incorporate spatiotemporal variables in traffic flow prediction (49).

6.1.2.1 Use of Machine Learning in Flow Prediction

Use of machine learning principals has become an optimistic approach to traffic flow prediction due to its stochastic and non-linear behaviour. In literature, three broader categories of models that are used for traffic flow prediction could be identified as, linear parametric models, non-linear parametric models, and non-linear non-parametric models [1, 13–15]. On the evaluation of linear parametric models, historical average prediction models, time series prediction models, exponential filtering model and Kalman filter model [as cited in 13] are very popular. The researchers who consider the non-linear parametric behaviour of traffic flow followed wavelet analysis-based models [16], cellular automata model [17], fuzzy regression model [18] and the catastrophe theory-based models [16]. However, limitations in above methods caused poor performance in traffic flow prediction due to linearity and parametric approach. Hence, the researchers have focused on non-linear non-parametric approaches-based on machine learning principles. On this aspect, use of k-nearest neighbour's regression model [8, 19], random forest regression model (49), support vector regression model [20], Gaussian process regression model [21] and artificial neural network-based models (51) could be observed.

6.1.3 Methodology

To predict travel time, the methodology was structured in three stages as data collection, training model and analysis. Google Distance Matrix API and the Infra-Red Traffic Logger were used to collect travel time data and traffic flow data. Data collected from both sources were merged together by matching spatiotemporal parameters. The analysis was carried out using the IBM SPSS Modeler machine learning platform [22].

In data collection, traffic flow and travel time data were collected in Sri Lanka at ten locations in the city of Colombo which covered mid blocks (more than 500m away from intersections) of two-lane, two-way roads. The time interval was set to be 5 minutes, and data collection was conducted continuously in the daytime under dry weather condition for three months.

.

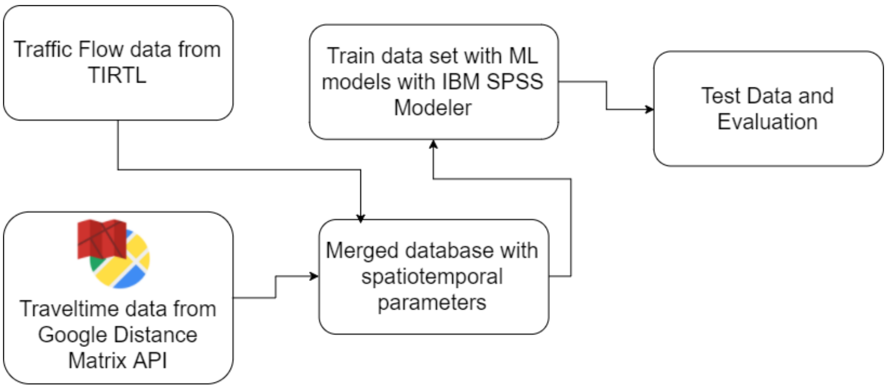


Figure 43:Flowchart of the Methodology of data collection for traffic flow prediction

Traffic flow data was collected through an Infra-Red Traffic Logger (TIRTL) system which is capable of classifying types of vehicles that are defined by axles and vehicle lengths. Video survey was conducted to validate TIRTL data accuracy. A linear regression model was developed to calibrate the instrument [24].

Google Distance Matrix API was called in one-minute intervals to collect travel time data in between a specific origin and destination locations where the infrared traffic logger was placed. The data collection was not within the Free limit most of the time.

6.1.3.1 Evaluation Using Machine Learning

In this research, traffic flow is considered as a non-linear non-parametric function of temporal inputs and spatial inputs (see table 1). A clustering-based regression method is proposed by following the objective of the study. Zhang et al. And Zhong et al. have concluded that a relatively accurate model can be obtained with an averagely sized dataset when K- Nearest Neighbours (KNN) is used [25].

KNN is a non-parametric algorithm which model parameters do not have to be calculated. K is the number of nearest neighbours considered in classification. When K is defined, the prediction is made by identifying the K nearest data points and counting the frequency of each class among the K nearest neighbours. The K value should be defined in order to minimize the absolute error of the model [26].

In this study, the processed dataset was analyzed using the KNN model supported by IBM SPSS Modeler [22]. Eight spatiotemporal attributes were considered in flow prediction as given in Table 1. The equation 2 illustrates the flow prediction concept used in KNN algorithm. When the model is trained, it was verified using the test dataset for prediction accuracy. The equation 3 illustrates the use of distance weights in obtaining higher accuracy [26]

$$Flow_{KNN} = f\left(T(t, s, s_o)_{1,2,\ldots\ldots n}, S(l, w, w_o, h, h_o)_{1,2,\ldots\ldots n}\right) \quad (2)$$

$$Flow_{predict} = \frac{1}{K}\sum_{i=1}^{K} w_i Flow_{KNN} \quad (3)$$

Table 6: KNN Input Attributes

Temporal attributes (T)	**Spatial attributes (S)**
t = Time (hh:mm)	l = Link Length (km)
s = LinkSpeed (km/h)	w = Lane Width (m)
s_o = Opposite Link Speed (km/h)	w_o = Opposite Lane Width (m)
	h = Shoulder Width (m)
	h_o = Opposite Shoulder Width (m)

The main evaluators of the machine learning models are the Root Mean Squared Error (RMSE) and the Maximum Absolute Error (MAE). In order to compare the results obtained the same dataset was trained under support vector regression (SVR) and artificial neural networks (ANN).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x_\iota})^2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - \overline{x_\iota}|}{x_i}$$

Where, x_i is trthe ue value, $\overline{x_\iota}$ is predicted value and N is number of patterns.

6.1.4 Analysis

6.1.4.1 Results and Discussion

A 70% of the dataset was used to train the model while 30% was utilized for testing the trained model. Figure 44 shows the distribution of training dataset and the test dataset. K- Nearest Neighbours (KNN) regression was carried out with K=3 as it shows the minimum sum of squares error when compared with K greater than 3(Figure 45). Further KNN regression with K=3 gives the minimum prediction errors since RMSE and MAE is lower when K=3 than K=4 or K=5. (seeTable 7)

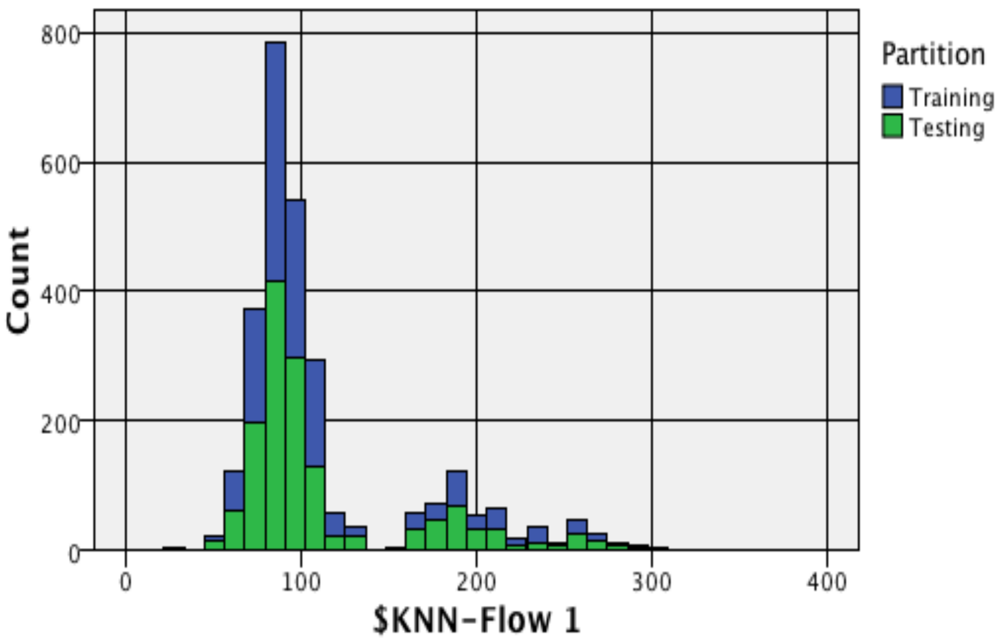


Figure 44: Training and Test Sample Distribution

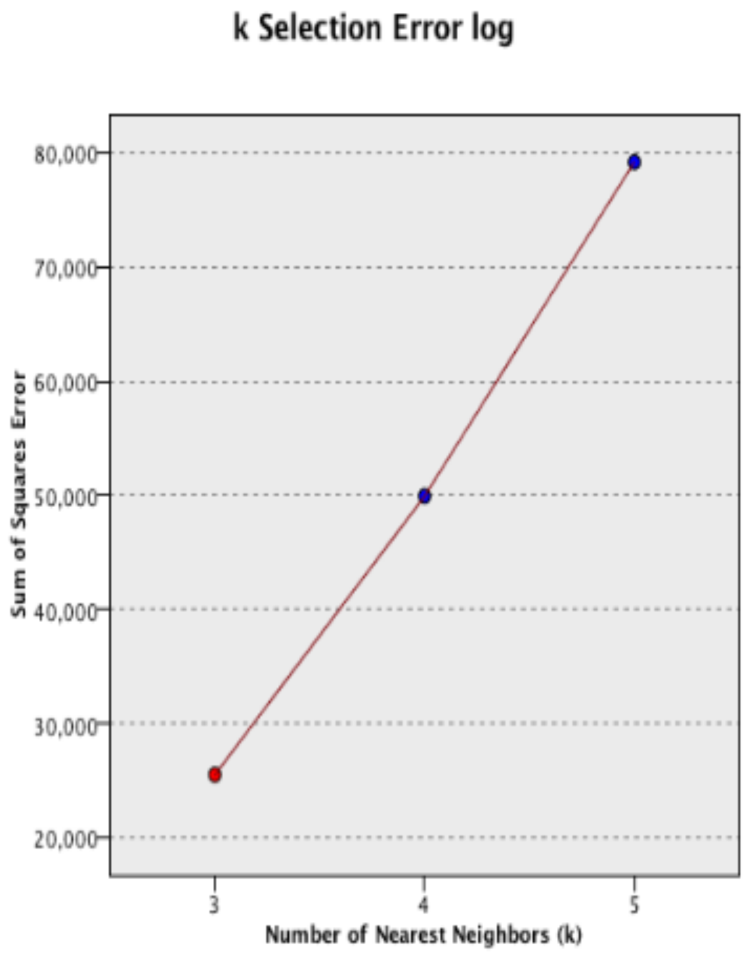


Figure 45:K- Selection Error Graph

Table 7. Evaluation of Regression Models

Parameter	RMSE	MAE	Linear Correlation
KNN k=3	9.479	2.318	0.983
KNN k=4	12.762	3.67	0.962
KNN k=5	15.452	4.237	0.924
SVR	44.691	25.854	0.528
ANN	29.13	18.31	0.710

Compared to K- Nearest Neighbour regression the artificial neural network and support vector regression has given higher prediction errors (see Table 7)

The Figure 46 shows the variation between predicted flow data and actual flow data obtained from neural network regression and Figure 47 shows the agreement of support vector regression. The distribution of predicted data does not match with the distribution of original dataset in both neural network and support vector regression. The reason for prediction error could be illustrated nicely with the two figures.

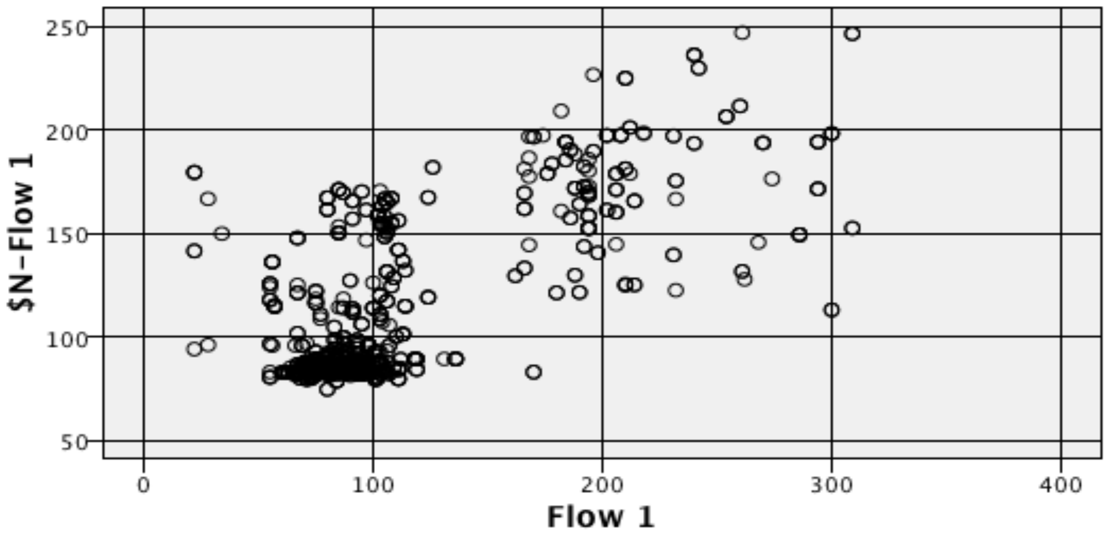


Figure 46:Predicted Vs Observed Traffic Flow for ANN

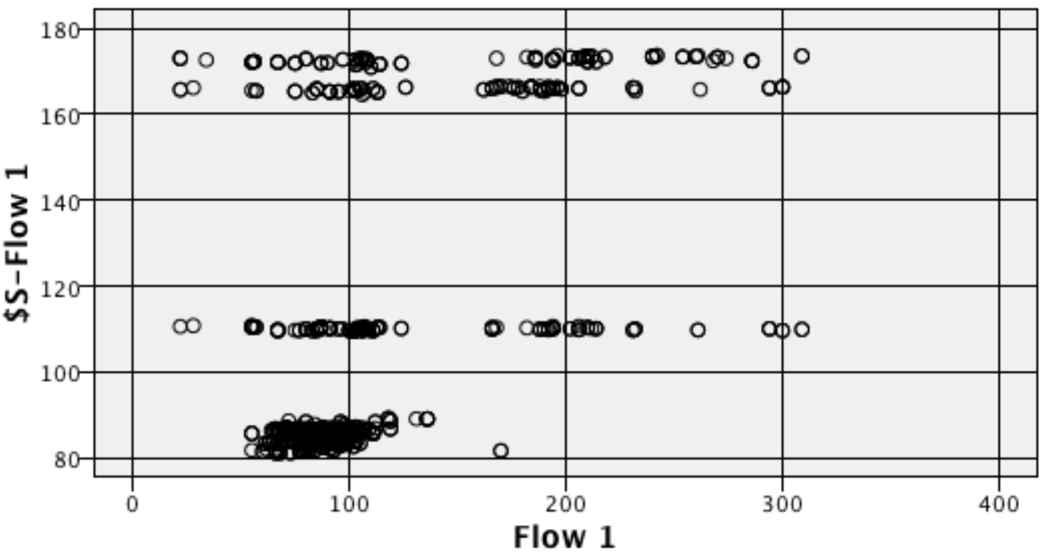


Figure 47: Predicted Vs Observed Traffic Flow for SVR

Figure 48 shows the comparison of predicted values with actual values of the directional flow when plotted against time. The predictions are much accurate with the K- Nearest Neighbour analysis when compared the linear correlation. (see Table 7)

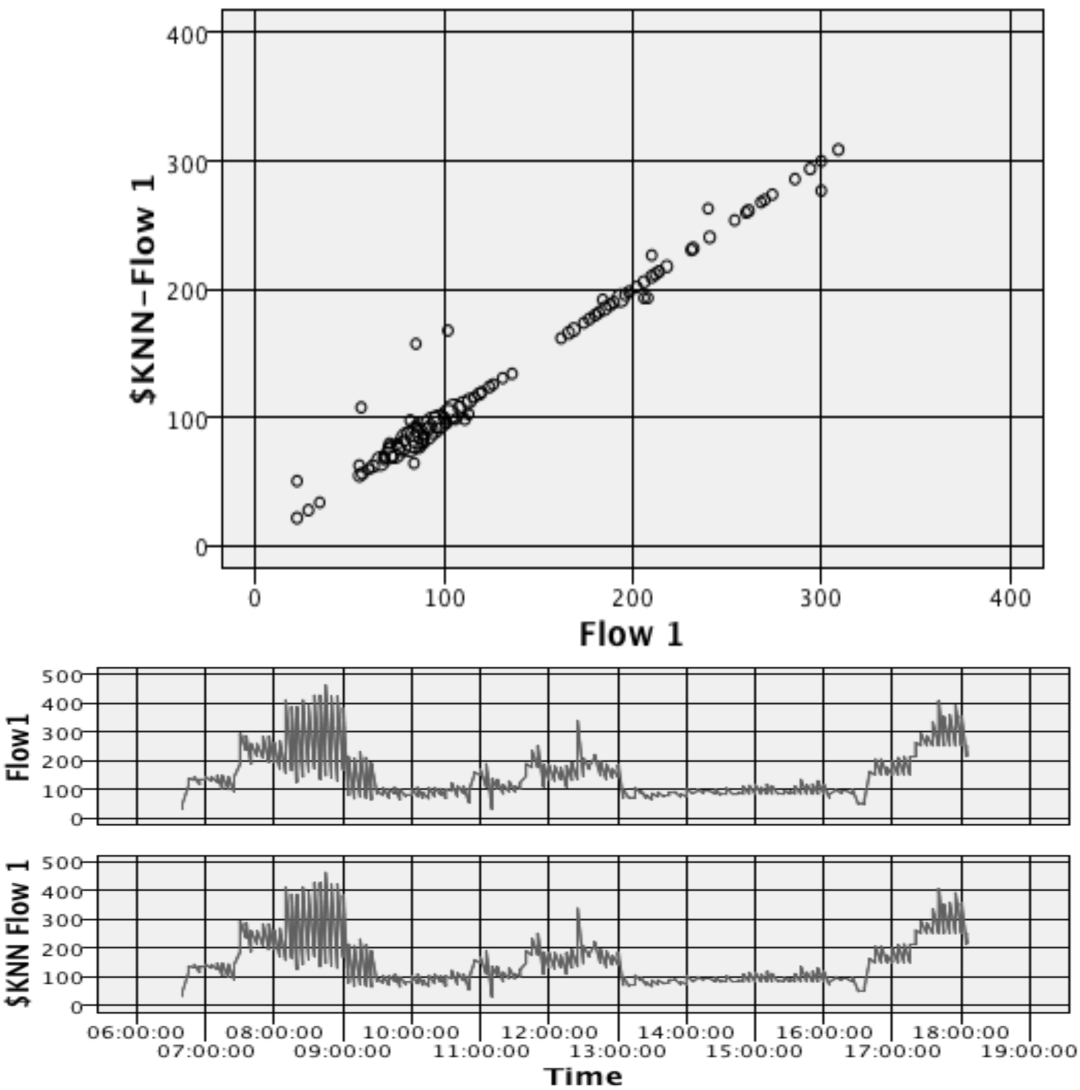


Figure 48:Predicted and Observed Flow Values Under KNN Regression

When considered the input parameters which contribute to flow prediction the directional link speed and opposite direction has the major influence. Figure 49 shows the scatter of directional speeds with predicted Flow values which depict the non-linear behaviour.

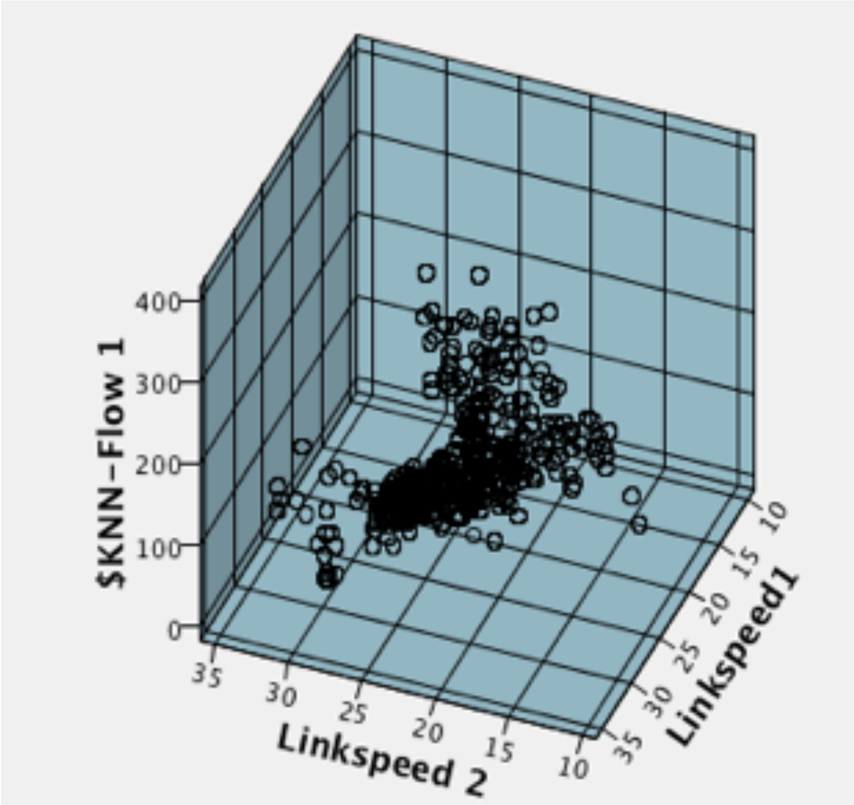


Figure 49: Distribution of Directional and Opposite Link Speeds with Predicted Flow

The verification of the model by cross-validation is conducted using n-fold validation by splitting the dataset into 12 folds. The agreement between predicted KNN-Flow vs actual Flow for each fold was analyzed by taking the RMSE and MAE of training dataset and testing datasets for each fold. Table 8 gives the average RMSE and MAE obtained by averaging each fold RMSE and MAE.

As the percentage difference of RMSE and MAE are at low values, it can be concluded that the model is not overfitting. Figure 50 shows the percentage gain of each training dataset which indicates that the model could be validated for better results.

Table 8. Evaluation for Model Over fit

Partition	**RMSE**	**MAE**	**Linear Correlation**
Training	11.504	5.91	0.979
Testing	11.92	6.445	0.976
% Difference	**3.49%**	**8.3%**	**-**

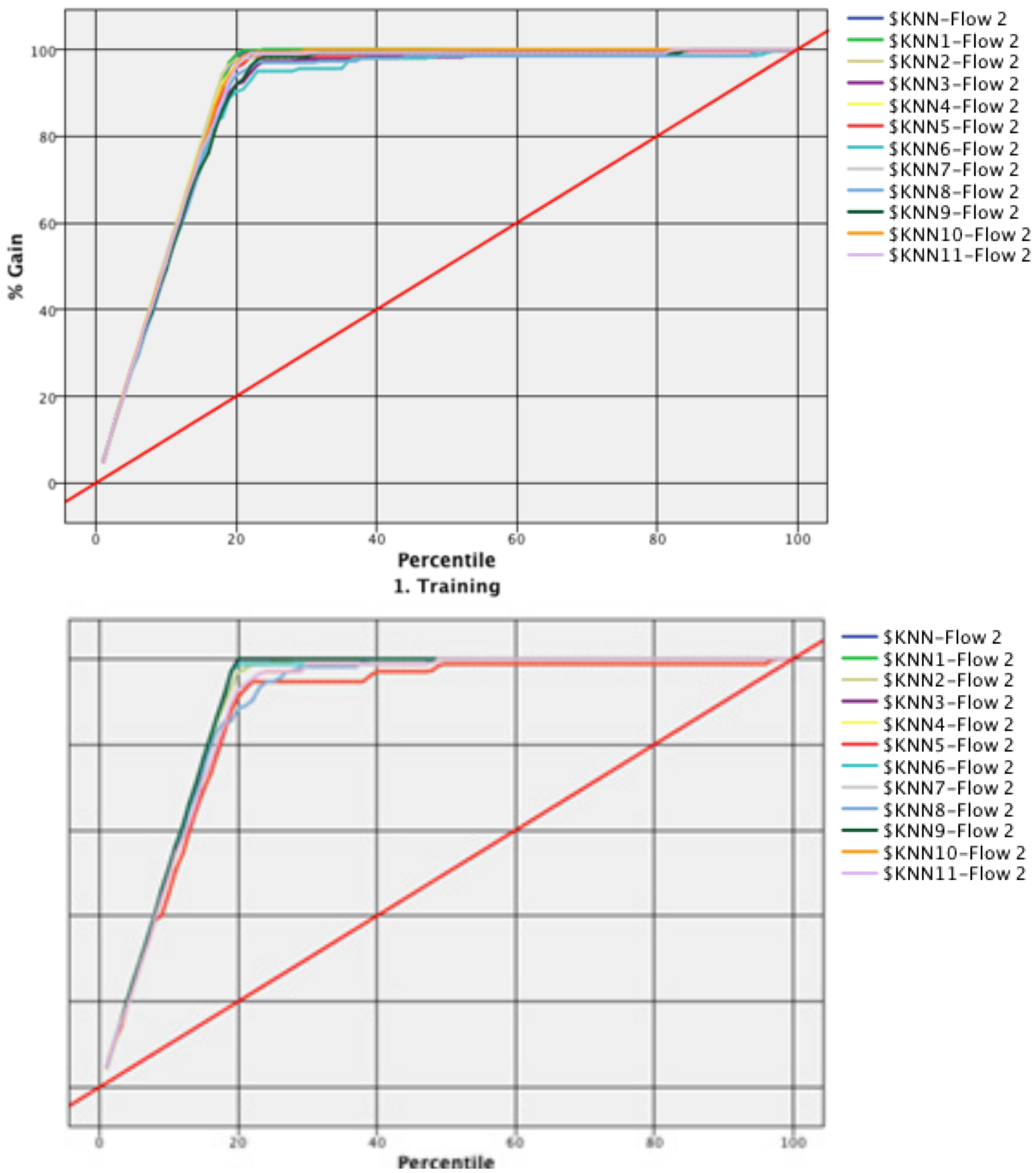


Figure 50:The Percentage Gain of Each Training Dataset

6.1.5 Conclusion and future work in traffic flow prediction

The results show that traffic flow prediction for urban two-lane roads using machine learning principles become a success. Deviating from traditional methods of linear parametric approaches for flow prediction the study evaluates the use of the non-linear non-parametric approach. The study develops a traffic flow estimation model-based on K- Nearest Neighbour regression which uses spatiotemporal inputs. Travel time and speed data obtained from Google Distance Matrix API and road geometry data are used as inputs to the model.

K- Nearest Neighbour regression was able to give a higher prediction accuracy with a linear correlation of 0.97. The prediction error was minimum at K=3 neighbours. The model was cross-validated using the N-Fold cross-validation method and showed the model does not overfit.

In future work, it is suggested to incorporate continuous data feeding methods. Further incorporating data sources such as terrain details, pavement conditions, and land use as categorical variables could enable the estimation model being used in rural areas.
The success of the study enables to use Google Distance Matrix API travel time data in estimating the traffic flow which is an economical alternative for developing countries.

6.2 Identification of road bottlenecks along corridors using Google travel time

Bottlenecks are considered an important research topic in the field of traffic engineering since bottlenecks can be a major source of delays in road traffic. Delays due to traffic congestion can create many socio-economic implications which can toughen day-to-day activities. Identification of regular congestion patterns and bottlenecks along an urban road network can improve the traffic management of that area. Identification of static and dynamic road bottlenecks are carried out in diverse ways. Use of Advanced Transport Information systems in diagnosis process of road bottlenecks is a novel approach to existing traffic management techniques. Most of the bottleneck identification methods in the literature cannot expand for large networks simultaneously or very expensive. Therefore, identifying road bottlenecks for an urban traffic network with existing methods are both a time consuming and costly process. Hence, there is a need for bottleneck identification method which will be economical to implement.

This section presents a method for identifying static road bottlenecks and moving road bottlenecks-based on dynamic traffic data gathered from crowdsourced data and spatiotemporal analysis. Not limited to single day data analysis, the proposed method analyzes dynamic and concurrent traffic data collected for a longer period for multiple links-based on Google Maps Distance Matrix Application Programming Interface. Collected data being fed into spatiotemporal traffic state matrix and bottlenecks are identified by visual evaluation. The proposed solution performed well within an experimental workbench and could be incorporated in urban traffic management.

6.2.1 Objectives

Congestion occurs when the demand exceeds the capacity (64). From the basic definition, congestion is a condition on transport networks that occurs as use increases. It is depicted by slower speeds, longer trip times, and increased vehicular queueing.(64) According to Jimenez and Valenciz there are two broad groups of definitions of congestion: the first is related to the macro-level demand for road use(65) and the second group of definitions is related with micro-level factors which incorporate differences between the roadway system performance that users expect and how the system actually performs (65). Hence road bottlenecks are belonging to the micro level factors (65).

A traffic bottleneck is a localized disruption of vehicular traffic on a road segment in which separates upstream queued traffic and free-flowing downstream traffic (66). When compared to a traffic jam, a bottleneck is a result of a specific physical condition, often caused by merging and diverging traffic, lane drops, grade changes or badly timed traffic lights, intersections (66). Identification of time and duration of congestion in bottlenecks can enable investigations and provide mitigatory actions (66).

The objectives of this application are the identification of road bottlenecks-based on travel time variation and variation of space mean speed below a threshold value. The bottlenecks should be able to identify under spatiotemporal elements and be able to obtain an idea about their occurrence and significance. The third objective is to illustrate the validity of using Google Distance Matrix application programming interface travel times for bottleneck identification.

6.2.2 Literature review

6.2.2.1 Definitions on road bottlenecks

There were several studies which have been carried out to identify road bottlenecks. It was identified that the methods used in identification involve direct and stochastic approaches (67). When looked at the literature for road bottlenecks, many authors have taken different approaches in defining road bottlenecks (65), (67). According to the fundamental definition of Daganzo, an active bottleneck is a restriction that separates upstream queued traffic and free-flowing downstream traffic which is time-dependent (64). Chen et al. Described freeway bottlenecks as certain freeway locations that experience congestion at nearly the same time almost every day (68). According to Bertiney and Mayton(69) who followed Daganzo, a bottleneck is a point upstream of which there is a queue and downstream of which there is freely flowing traffic (69).

Their research was much concerning on using occupancy and flow data for bottleneck identification, and it confirms it is possible to identify freeway bottleneck activations without pre-specifying any arbitrary speed thresholds (69).

Ban et al. defined bottlenecks as sections of the roadway that have either capacities less than or demand greater than other sections (70). Zhao et al. Defined a bottleneck as a poorly performing roadway segment on the basis of speed measurements and statistical predictability (39). Deriving from above concepts for this research a bottleneck is considered as a road location which experience recurrence congestion measured by speed reduction.

6.2.2.2 Bottleneck Identification

Identification of road bottlenecks become the next challenge. In the initial work, according to Jia et al. Spatiotemporal variation should be incorporated in the identification of freeway bottlenecks (71). Cassidy and Bertini developed a method to observe the transition from free flow to queued conditions by visually comparing curves of cumulative vehicle arrival number vs time and cumulative occupancy vs time measured at neighbouring loop detectors (72). This method is effective for detailed analysis of features of a recurring single bottleneck (72). But the method is too time-consuming in identifying and analyzing multiple bottlenecks in a corridor (72).

Following the concept, Chen et al. developed an algorithm for locating active freeway bottlenecks and estimating their delay impact on the basis of loop detector data (68). The basis of identification was on speed differential between a pair of upstream and downstream loop detectors (68). With their method, it was possible to identify the times for which each bottleneck is active, the delay it causes and ranks the bottlenecks in terms of their frequency of recurrence and the magnitude of their delay impact (68). However, the algorithm was limited by data availability and was based on single day data which neglected the day-to-day traffic variations (68). Improving on loop detector data Ban et al. proposed a percentile-speed-based approach by using loop detector data from multiple days to identify and calibrate freeway bottlenecks (70). Bottleneck identification occurred on a speed contour map (SCM) automatically (70). This method converted the speeds on the SCM into either zero or one, depending on whether the speed was higher than a congestion threshold, and identified the areas marked by 1s to obtain the queue length and time duration of the bottleneck (70). A limitation of the method was that it was based on the assumptions of continuous freeway detection and neglected day-to-day traffic variation. Wieczorek et al. Suggest an automated tool for identifying recurrent freeway bottlenecks using historical data archived from freeway sensor data (73). The research was successful in evaluating the Chen et al..(68) method, using a sensitivity analysis approach. The suggested tool compares each pair of longitudinally adjacent detectors at each 5-min time point and declares that there is an active bottleneck between them when the speed at the upstream detector is below the maximum upstream speed threshold, and the difference in the speeds at the upstream and downstream detectors is above the minimum speed differential threshold (73). Further, this method incorporates travel reliability across days, weeks, months, and years which was a weakness in loop detector data (73). The drawback of the method suggested is the scalability (73). There will be practical issues such as funding when this method is considered implement in developing countries (74).

With the implementation of ITS technologies such as vehicle detection by global positioning system (GPS), it enables to obtain a larger amount of information than earlier types of data collection such as sensor data and loop detectors (3). Following the ITS concepts, Zhao et al. Suggest a methodology for identifying and ranking bottlenecks using probe data collected by commercial global positioning system fleet management devices mounted on trucks (39).

With this methodology, it was able to measure the performance of the vehicles directly without being inferred from general measures of traffic performance (39). Unreliable travel conditions over a roadway section indicated by increased travel time were considered as the major parameter in detecting bottlenecks (39). The major drawback in this research was GPS data sample was only representing trucks which are travelling on the road (39). Therefore, identifying road bottlenecks generally applicable to all the vehicles moving on the road is not addressed in this method (39).

Improving on the concept of using GPS data obtained from probe vehicles. Jimenez and valacia (2016) suggest a methodology to identify bottlenecks-based on GPS data from taxis vehicles in a city limit (65). The GPS location reported by the GPS devices via GPRS (General Packet Radio Service) with a frequency of 10s was used to identify road bottlenecks (65).

Finding recurrent low-speed sections, beyond expected delays in the road network was enabled by identifying road network segments that perform poorly in terms of speed, compared with upstream and downstream conditions (65). This could be considered as a low-cost option in identifying critical points in traffic networks in developing cities without expensive traffic-monitoring systems. Limitation in implementing this type of a methodology is the data collected is only available for taxi vehicles in the city which is a poor sample of vehicle movement (65). Moreover, for countries with less number of taxies, this method cannot provide a representative sample in urban traffic (75).

In summary, identification of road bottlenecks methods literature suggests methods-based on loop detectors, traffic sensors and tracking GPS location of probe vehicles, using the cellular mobile network and using GPS location of anonymous moving smartphones as commendable approaches(3), (39), (68), (76). When evaluated among all these methods using mobile phone data as a mobility sensor could be the most appropriate methods as it can generate large amounts of information simultaneously and enables to gather a holistic idea on a large road network with higher accuracy. Therefore, the research will focus on gathering travel time estimates given by Google Distance Matrix API as it uses GPS location and speed of moving mobile phones as the source of information and derive at travel time estimates-based on real-time information and historical data available.

6.2.2.3 Ranking and Reliability of Bottlenecks

When a road bottleneck is identified, the next consideration is to measure the reliability of the bottleneck. Travel time index used by FHWA include the 90th or 95th percentile travel time and the buffer index, which is the extra time needed to allow the traveler to arrive on time (FHWA 2011). This index is computed as the difference between the 95th percentile travel time and the mean travel time, divided by mean travel time. Hale et al. Suggest three evaluation criteria for bottleneck ranking-based on spatiotemporal traffic state matrix (67). Impact Factor for ranking of bottlenecks-based on the duration of congestion and length of congestion over a longer time period was proposed to rank the bottlenecks (67). Annual reliability matrix(ARM) is suggested by authors to understand the annual variability of a bottleneck impact factor (67). Further improving the ARM a bottleneck intensity index is proposed to capture the size and shape of ARM to a single number (67). Emanm and Al-Deek developed a methodology to estimating travel time reliability-based on stochastic approach (10). Their research confirms lognormal distribution is best-fit to compute the the travel time reliability as the probability that a trip between a given origin-destination pair could be made within a specified time interval (10). Zhao et al. Classified the travel reliability of roadway segments into three categories as unreliable, reliably slow, and reliably fast (39). It was based on the hypothesis that roadway reliability is stochastically represented by a unimodal or a bimodal probability density function over a certain time period (39). Zheng and Chang suggested that Weibull and Log logistic statistical distribution best fits to evaluate capacity and congestion duration respectively (77). Hence confirming to estimate the probability of congestion occur and the time it will last using those estimates (77).

6.2.3 Methodology

In obtaining data for bottleneck identification, a spatiotemporal variation of speed in between two adjacent road segments is evaluated. Average travel time and space mean speed are the governing variables which will be considered in this research for evaluation. Both parameters could be obtained using the Google Distance Matrix API. Initially, the API calls are utilized to gather distance value in meters and travel time estimates in seconds to the considered segments. Then travel time value and distance value obtained from the API could be used to calculate the space mean speed of each considered segment. Using the Distance Matrix API travel time between origin and destination could be obtained in seconds and distance in meters. For the experiment, the minimum road length used was 100 meters, and the highest travel time collection frequency was 1 minute. For the experiment, data was collected over a period of 3 months.

6.2.3.1 Bottleneck identification

As identified in the literature survey many works have been done on the identification of road bottlenecks. Out of all the methods used in literature they all used occupancy, space mean speed, travel time or variation in travel time as the governing parameter in identification and ranking of bottlenecks (39), (65), (67), (68). On the evaluation of bottlenecks many analysis methods have followed graphical identification or stochastic estimates (39), (67), (77). When considered the data obtained for analysis the spatiotemporal representation of travel time and speed data will enable to identify bottlenecks (78). Spatiotemporal distribution of travel time and speed data can be the basis for defining reliability for bottlenecks. A similar approach was taken by Elhenawy by suggesting the spatiotemporal traffic state matrix (STM) which could be considered as a successful approach when dealing with large data sets (79). (see Figure 51)

Compared to traditional traffic engineering analyses. The STM facilitates to consider temporal variation over a long period of time. Instead of just peak-hour analyses with the STM weekly, monthly and annual analyses could be obtained by providing a more comprehensive picture of traffic problems. With the STM it is possible to evaluate the duration of bottlenecks, the intensity of bottlenecks, variability over space and time by referring to spatiotemporal variation in travel time and space mean speed.

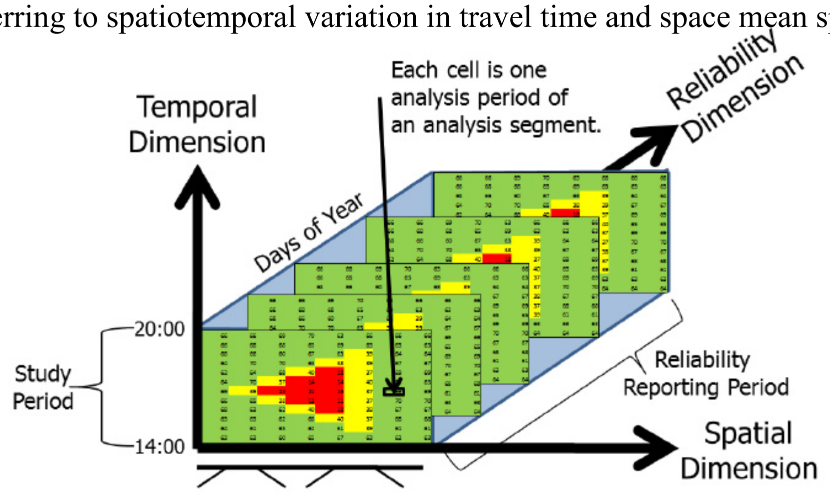


Figure 51: Spatiotemporal traffic state matrix STM; HCM-2010 Transport Research Board(36)

In this paper, a spatiotemporal variation of speed data is recorded in the STM for each corridor or road networks considered. On identification of bottlenecks, the average space mean speed of each segment over a period of a week is evaluated. For identification of bottleneck initially, the segment need to be evaluated for congestion. In that aspect, any segment with space mean speeds less than 21km/h was considered (78). The criteria were to consider average speeds of arterial roads at the level of service (LOS) class E and Class F as congested scenarios (78).

When a road segment is identified as congested the next step is to check for its recurrence. In ranking bottlenecks recurrence is a significant factor to be considered as mitigation of such occurrences is important. In the research the methodology adopted was to prepare STM with the weekly average the speed data and identify congested segments along with spatiotemporal identities. In this way, recurring bottlenecks could be identified. Further on validating the bottleneck visual observing the spatiotemporal traffic state matrix could be used.

6.2.3.2 Ranking of Bottlenecks

When a bottleneck is identified its reliability and influence is the next challenge to be addressed. As congestion is a random event and formation of bottlenecks is followed by such a random event, (77) the attention is significant to the aspect of the influence of the bottleneck and its reliability (77).

Although the literature provides many methods-based on stochastic regression and with many independent parameters (*6*, *14*, *19*), most of such methods given in literature cannot be applied when travel time and space mean speed are only available. Therefore, in order to identify the influence of a bottleneck parameters such as duration of a bottleneck, network average speed at the time of bottleneck, a minimum speed of the bottleneck, the maximum speed of the segment could be incorporated (67). Hence in the research variation of speeds at bottleneck period from normal conditions are observed.

To gather an idea on the speed reduction at the bottleneck percentage variation from existing network average speed was evaluated. Hence that will decompose an idea on how worse the performance of the bottleneck to the overall network traffic.

$$\%Effect = \frac{V_{Network} - V_{Bottleneck}}{V_{Network}} * 100$$

The effect of a bottleneck on the segment performance is another parameter to be evaluated when ranking bottlenecks in this regards. This will be a reliability measure of the segment considered. In literature, there were many approaches-based on either the number of occurrences found the queue size or the duration of the bottlenecks (67). Out of all those available methods, the Travel time Index is an optimum measure as it compares the travel time during congestion to the time required to make the same trip at free-flow speeds. (78). With respect to the bottleneck ranking, Travel time index was taken as the ratio of travel time at bottleneck congestion situation to travel time at free-flow.

$$TTI = \frac{TT_{Bottleneck}}{TT_{Freeflow}}$$

Although a bottleneck gives a high travel time index, it does not become significant for ranking if the bottleneck exists only for a short period of time (67). Further vice versa a bottleneck with lower TTI could be much impactful if it exists for a longer period of time of the day (67). Hence the congestion period should be taken into consideration. In literature, Impact factor was defined using the duration of congestion with the length of congestion (67). Therefore, in order to incorporate the duration of congestion into the evaluation of the impact of a bottleneck the research defines Impact Factor incorporating travel time index and duration of congestion.

$$IF = \frac{TT_{Bottleneck} * D_{congestion}}{TT_{Freeflow} * D_{Observed}}$$

IF = Impact Factor
$D_{congestion}$ =Duration of Congestion
$D_{Observed}$ = Duration Observed

In defining the Impact Factor, the ratio between the duration of congestion to an observed duration which the analysis was taken into consideration, was included as shown in above equation. Hence with this approach, it is possible to distinguish the impact of travel time index and congestion duration ratio as explained earlier.

6.2.4 Analysis and evaluation

The above-explained methodology of data collection for travel time and space mean speeds for urban road segments using Google Distance Matrix API and analysis for identification and ranking of road bottlenecks was carried out at three road networks in Colombo metropolitan area, and one analysis on a major arterial road is presented. The details and map of the analysis on location is given in Figure 52

As per the collected data, the next procedure of the methodology is to illustrate the travel times and calculated space mean speeds on spatiotemporal traffic state matrix. The illustrated spatiotemporal graph shows the road segments in the horizontal axis and time on the vertical axis. The average speeds over a period of three months were shown in the matrix after excluding weekends speed values and extreme outliers.

The matrix was colour coded to identify the formation of bottlenecks. The threshold speed value for detection of bottlenecks was 21km/h which was suggested by the level of service criteria of HCM2010 (78). The speeds which are lower than 21km/h are coloured as red, and red colour gets darker when speed reduces from a threshold level. When the speeds are greater than 21km/h the indicated colour becomes green, and when speed levels increase further from threshold level the green colour gets darker.

In every STM the first column indicates the average speed value between the endpoints of the network which was considered as the network average speed. A bottleneck location is identified when adjacent segments of the selected segment show higher speeds than the threshold level (i.e. Either side of the red coloured segment is shown as green.

After identifying the formation of bottlenecks at each road segment and duration of such formation the impact of the event was analyzed from the theoretical approaches mentioned in the methodology. If bottlenecks were not enounced in a segment throughout the analysis period, then such segments are not considered and considered as well performing segments. Figure 52 shows the map of the segments used in the analysis

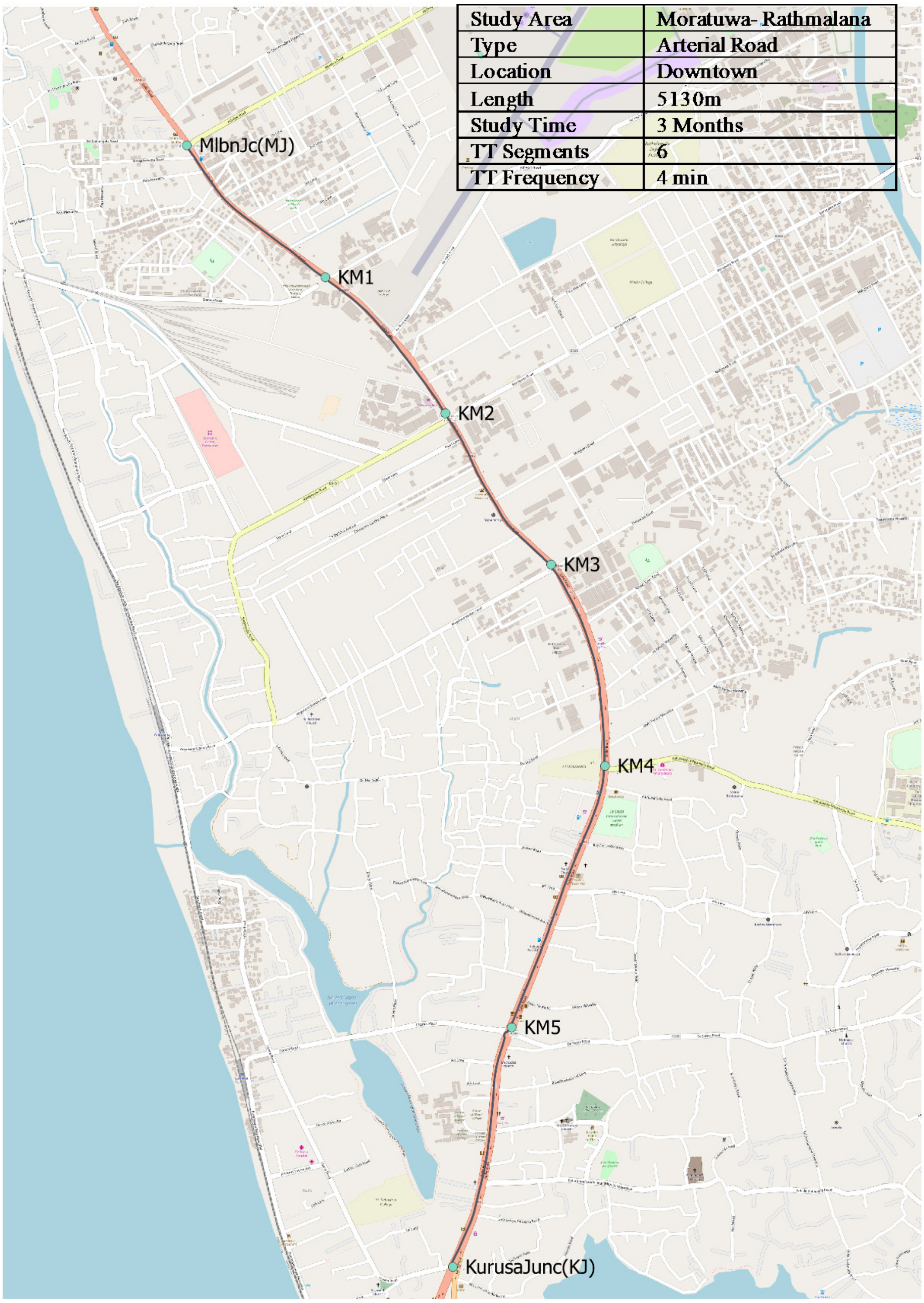


Figure 52: Map of the segments used for bottleneck analysis

KJ KM5 KM4 KM3 KM2 KM1 MJ

Hour	Min	TOTAL	S1	S2	S3	S4	S5	S6
5 AM	0	43.51	43.99	44.37	46.76	35.26	48.02	47.52
	8	42.61	41.74	44.53	46.94	34.56	46.87	46.93
	16	42.71	41.57	44.74	47.40	34.89	46.85	46.83
	24	41.66	38.25	44.09	46.52	34.71	46.21	46.05
	32	42.53	39.60	44.30	47.87	34.88	47.70	45.21
	40	42.25	40.25	45.41	47.71	33.66	46.94	45.31
	48	41.32	39.78	44.81	45.68	32.41	46.26	45.22
	56	39.62	38.64	42.79	43.95	31.25	43.63	43.77
6 AM	0	39.18	38.60	42.23	42.69	30.48	42.98	43.74
	8	35.63	37.11	40.10	39.11	25.74	36.65	40.01
	16	33.54	35.42	39.48	36.81	24.10	34.23	37.59
	24	30.03	30.10	35.97	32.35	22.32	30.18	34.11
	32	28.19	29.78	34.08	29.67	19.35	28.15	32.00
	40	26.70	28.40	33.85	27.79	18.62	24.91	30.50
	48	25.01	26.47	32.43	26.28	16.59	21.84	29.23
	56	23.24	23.03	31.31	24.13	16.62	20.84	28.24
7 AM	0	22.66	23.15	31.42	24.36	16.39	19.12	29.15
	8	21.93	20.45	30.79	23.51	15.88	18.28	28.58
	16	20.77	17.95	29.77	21.86	15.81	18.65	28.14
	24	20.97	17.10	29.93	23.29	15.67	20.37	28.83
	32	19.31	11.97	28.66	22.85	16.85	21.02	29.39
	40	19.73	14.63	27.25	22.33	16.80	20.63	29.39
	48	21.22	20.91	27.64	20.94	15.84	20.60	28.88
	56	20.70	23.31	28.33	19.16	14.53	19.50	29.30
8 AM	0	21.13	22.59	27.98	20.30	14.67	19.08	29.38
	8	20.75	23.06	27.48	19.44	14.89	18.94	29.20
	16	20.47	24.21	27.61	19.06	14.33	19.13	29.48
	24	19.52	23.43	27.74	16.34	13.57	19.00	28.66
	32	20.16	24.47	26.69	17.01	15.15	19.14	28.04
	40	20.14	23.87	26.69	18.47	14.85	18.74	28.73
	48	21.63	25.30	27.67	20.18	15.21	18.99	28.19
	56	21.72	24.22	27.35	22.32	16.03	19.21	27.38
9 AM	0	22.88	24.28	27.88	22.47	16.40	21.94	25.62
	8	24.67	25.67	27.89	26.44	18.08	22.18	27.10
	16	25.25	25.50	27.38	29.11	19.46	24.91	26.95
	24	25.66	25.68	28.20	29.00	19.13	25.70	27.46
	32	26.09	24.93	28.85	30.00	20.82	26.23	27.57
	40	26.18	25.15	29.26	30.20	20.65	26.70	27.64
	48	26.41	26.01	28.75	29.68	20.87	26.98	27.83
	56	26.52	25.60	28.54	30.64	21.06	27.88	28.43
10 AM	0	26.48	25.85	28.76	29.83	20.64	27.99	28.38
	8	26.38	25.19	28.55	30.26	20.99	28.00	28.30
	16	26.40	25.37	28.79	29.89	20.41	28.11	28.13
	24	26.50	26.08	29.12	31.29	20.76	27.97	26.25
	32	26.19	25.10	28.75	30.50	20.96	28.19	26.49
	40	26.03	25.03	28.42	30.58	20.26	27.99	27.10
	48	26.23	25.54	28.66	31.10	20.38	26.83	26.43
	56	25.62	25.52	28.43	30.07	18.97	26.86	27.31
11 AM	0	26.15	25.60	29.55	30.80	19.89	26.68	27.01
	8	25.67	24.19	29.41	31.49	19.25	25.87	25.76
	16	25.18	22.68	29.20	30.57	19.59	25.26	26.72
	24	25.62	23.58	29.01	31.97	20.13	26.26	26.05
	32	25.62	23.89	28.45	31.14	19.70	26.89	26.61
	40	26.18	23.74	28.50	32.03	20.71	28.43	26.74
	48	26.22	24.02	29.95	31.01	20.02	27.95	27.77
	56	25.99	22.79	29.37	32.46	20.45	27.49	27.76

Figure 53: Spatio-temporal graph of bottleneck formation

When the location was analyzed, it was able to observe 4 bottleneck incidents. Further, there were two segments which did not indicate any bottlenecks. The observed bottleneck incidents are recurring bottlenecks as the weekly average speeds were considered. Hence the STM method was able to capture the effect of day to day variations.

Table 9: Performance measure of bottleneck events

Segment		Start	End	% Effect	TTI	IF
Total	KJ-Ml	7:16 AM	8:40 AM	5.7%	1.95	0.395
S1	Kj-KM5	7:08 AM	7:48 AM	29.2%	2.02	0.194
S3	KM4-KM3	7:48 AM	8:48 AM	20.8%	2.23	0.322
S4	KM3-KM2	6:40 AM	9:00 AM	37.0%	1.78	0.599
S5	KM2-KM1	6:56 AM	8:56 AM	12.7%	2.27	0.654

Table 9 shows the performance of bottleneck locations under different performance measure. The effect of each incident was identified under three performance measures. When percentage variation from existing network traffic is considered segment 4(S4) has a higher change. When travel time index is considered segment 3(S3) has the highest drop. Segment 5 (S5) is critical under influence factor considerations, which consider both travel time index and duration of congestion.

The segment 1 shows a bottleneck incident in the morning this is mainly due to a high school located in that segment and vehicles which arrive at school create the traffic by parking at roadsides. When observed the segment 4 long-term congestion could be observed. The reason behind this incident is a long-distance bus stop and traffic signal lights not been updated.

The above analysis reveals STM illustration is a successful tool to identify congestion patterns and road bottlenecks. The advantage of this method is it can incorporate spatiotemporal variations and long-term impacts. Use of this method in planning and management is an effective approach. When a segment is identified as highly congested a finer experiment by analyzing smaller road segments of the congested segment will enable to identify exact locations which cause bottlenecks. Any field visit too can be helpful in this regard. A regular traveller might have an idea about the congestion in a road, but for a planner, this method could be an effective tool to identify bottlenecks and congestion patterns.

The objective of this research is the identification of road bottlenecks-based on travel time variation and variation of space mean speed below a threshold value the objective was archived after validating the use of Google Distance Matrix API travel time estimates. In future work, the experiment could be extended to a larger network and observe congestion patterns and bottlenecks. The visual identification method could be incorporated into a pattern identification algorithm and conduct the manual work much faster.

6.2.5 Conclusion and future work in bottleneck identification

This section focused on identification of road bottlenecks-based on travel time variation and variation of space mean speed below a threshold value. The methodology proposed was to gather data from Google Distance Matrix Application Programming Interface (API). The travel time given by the API is used to calculate the space mean speed of the road segment. The methodology is justified by a stepwise literature review on suitability. Travel time samples obtained from API were validated through floating car data and aerial photography.

Identification of road bottlenecks with a spatiotemporal variation of speed data was graphically visualized on traffic state matrix and impact over time is evaluated. The reliability and significance of road bottlenecks were identified using three evaluator measures, travel time index, bottleneck influence factor and speed variation form overall network average speed. The proposed methodology was illustrated using an arterial road in Colombo Metropolitan Area which showed successful results. Therefore, it could be concluded that the methodology is very advantageous in planning and forecasting traffic management activates which require higher accuracy and low cost of implementation. Hence this is a promising and economical method to identify congestion patterns in large urban networks in developing countries.

6.3 Using Google travel time data for evaluation of transport projects

The travel time information provided by Google Distance Matrix API could be used in the evaluation of transport projects. Especially in traffic Management related activities, this could be a tool to evaluate before and after impact of an executed project. Hence travel time information and space mean speed values could be analysed in a situations such as the implementation of new traffic plans, evaluation of peak hour traffic conditions, managing corridor traffic, implementation of new transport solutions. This section presents two such case studies which were conducted in Colombo Sri Lanka.

1. Implementation of bus priority lanes in Colombo Metropolitan Area
2. Implementation of reversible lanes in Colombo Metropolitan Area

In both these examples, the temporal variation of segmental traffic and spatiotemporal variation of corridor speeds were evaluated.

6.3.1 Implementation of bus priority lanes in CMA

6.3.1.1 Objectives

The bus priority lane is a technique used to reduce delay for mass transit vehicles by allocating a separate priority lane. These lanes are implemented with the objective of giving priority for high occupancy vehicles on roads. Therefore only permitted vehicles such as buses, school vans and similar vehicles with high occupancy could use these lanes. This could be considered as an optimum strategy to increase the public transport ridership by providing higher mobility than other vehicles.

Figure 54 illustrates the commuter composition of western region in Sri Lanka in 2012. According to the composition, nearly 50% of the passengers use public transport in their home to work trips and home to school trips.

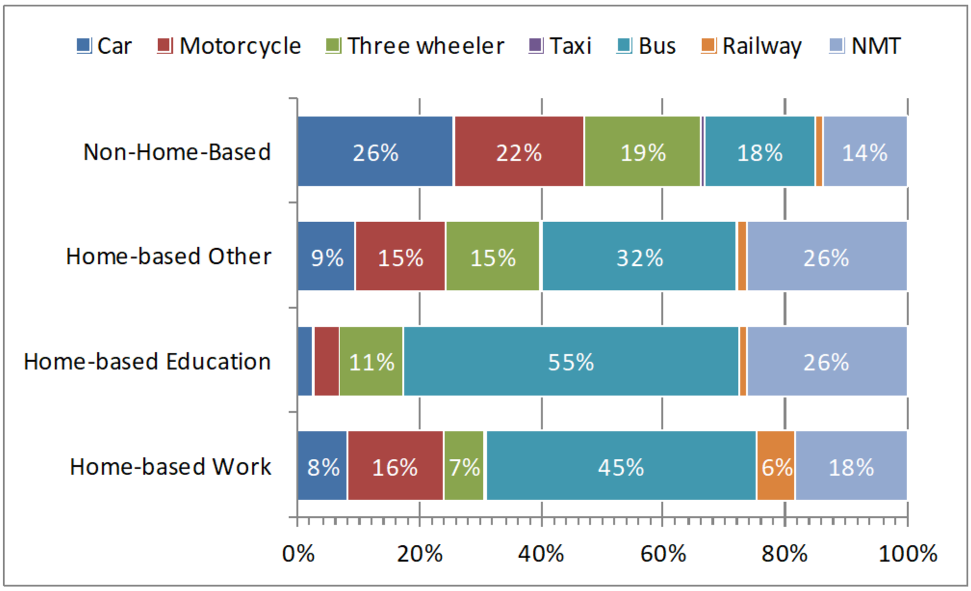


Figure 54: Mode share by trip purpose - Ref: Urban Transport System Development Project for Colombo Metropolitan Region and Suburbs; Technical Report 3: Characteristics of Present Transport Demand

The public authorities implemented bus priority lanes in several roads of Colombo district with the objective of reducing delay for mass transit vehicles on daily traffic conditions. The intention of the project was to Increase the public transport ridership by reducing the travel time for buses and other high occupancy vehicles. The lanes were operated during the peak time traffic of morning from 6.00 am to 9.00 am and evening 4.00 pm to 6.00 pm. During the off-peak traffic conditions, the lanes were operated as normal.

In the implementation of bus priority lanes, several arterial roads in Colombo Metropolitan Area were used. The Table 10 gives details about the segments which were used to implement bus priority lanes. Most of the roads in which the bus priority lanes were implemented consist of either 4 lanes or 6 lanes. Therefore in operation, one lane was converted to a bus priority lane. A lane towards Colombo in the morning and a lane outwards from Colombo in the evening were converted to bus priority lanes.

The analysis presented in the study was conducted to evaluate the pilot project which was carried out before implementation of Bus Priority Lanes.

6.3.1.2 Methodology

The travel time information collected from Google Distance Matrix API was used to evaluate the success of implementing bus priority lanes in Colombo. The data was collected for two segments in which the bus priority lanes implemented in the morning. The table shows the segments in which the analysis was conducted.

Table 10: Segments in which Bus priority lane analysis was carried out

Segment	Length	Start Date	End Date
Rajaririya – Ayurveda Junction	1 km	09.03.2017	17.03.2017
Rajaririya – Senanayake Junction	5 km	28.08.2017	04.10.2017

The data collection was carried out one week before starting the bus priority lanes to understand the existing traffic situation, and it was continuously carried out throughout the operation of bus priority lanes. Figure 54 shows the map of segments which were used to collect travel time data. Data was collected from 28th August 2017 to 4th October 2017 and for the pilot project from 9th March to 17th March 2017.

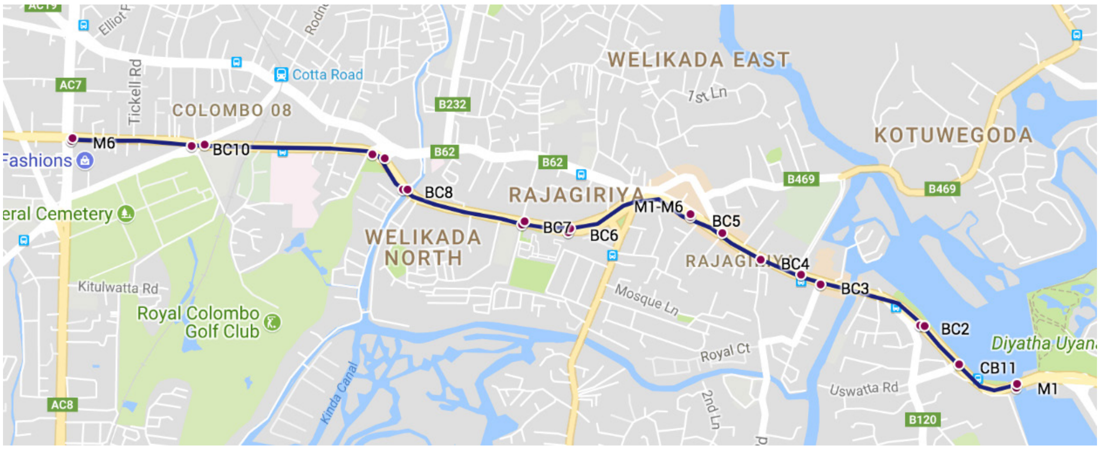


Figure 55 : Bus Priority lane implementation Rajagiriya; Diyatha Uyana – Senanayake junction

6.3.1.3 Analysis and evaluation

Data collected from Google Distance Matrix API was used to evaluate the performance of the bus priority lanes. Spatio-temporal graphs of speed and temporal variation of travel time graphs prepared to evaluate before and after situation. The daily change in travel time was evaluated while the weekly change in speeds for each segment was analyzed. Following sections will present the analysis for each segment separately indicating the performance of bus priority lanes.

Analysis 1 - Rajaririya – Ayurveda Junction – Pilot Project

The first analysis was conducted to evaluate the pilot project which was carried out before implementing the project. The Pilot project was carried out for a period of 1 week in the month of March. Figure 56 shows travel time variation along the segment Diyatha Uyana to Senanayaka junction on usual days without implementation of bus priority lanes. The graph shows high peaks at 7.00 am and 8. 15 am. Which are due to home to school trips and home to work trips respectively. The average travel time is about 12 minutes during off-peak period. Figure 57 illustrates the travel time variation of all five days in which the pilot project was conducted. It could be observed that the travel time during most of the days has increased with respect to the usual travel time observed before the pilot project implementation.

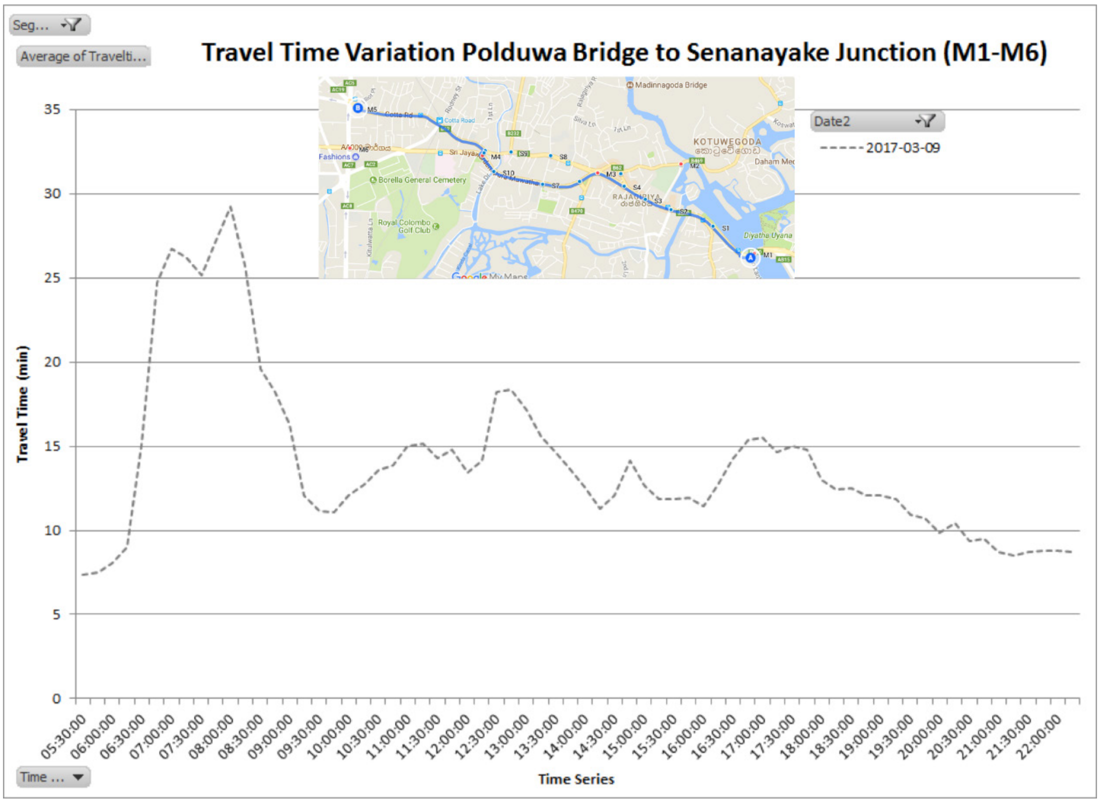


Figure 56: Travel time variation before implementation of bus priority lanes

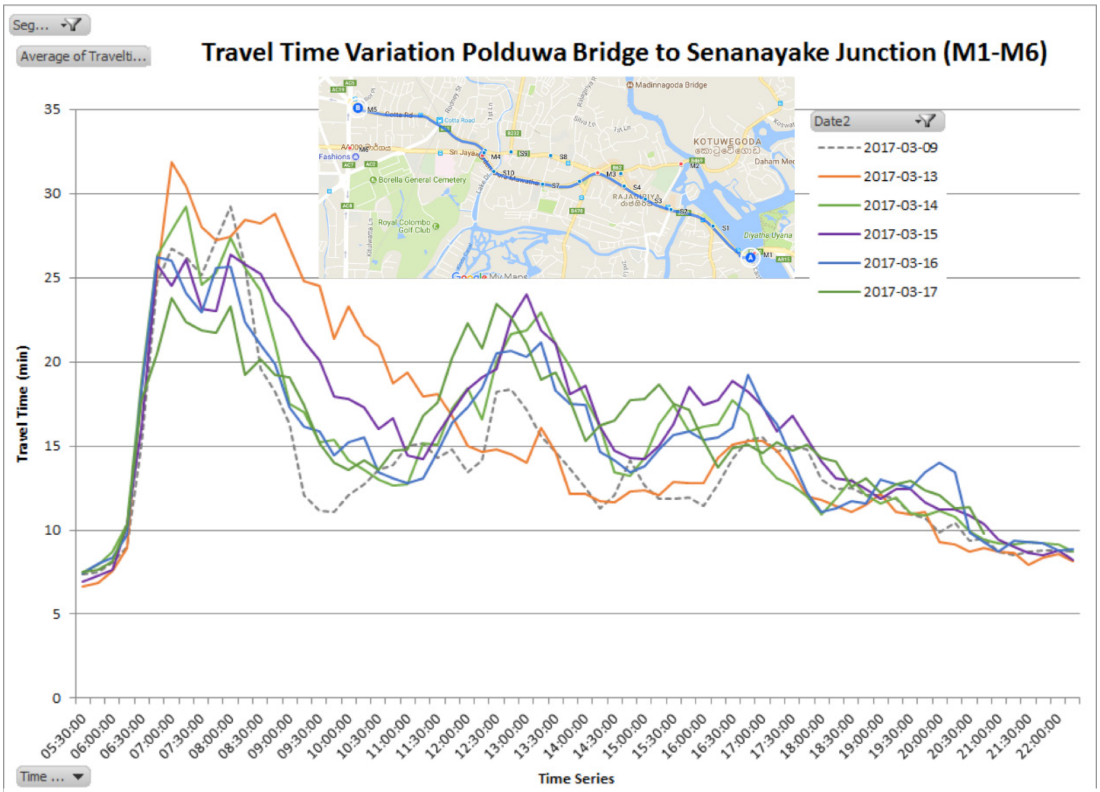


Figure 57: Travel time variation after implementing bus priority lanes

There is a significant increase in travel time from 12.30pm to 2.30 pm which indicates that the school to home trips are being affected due to the implementation of bus priority lanes.

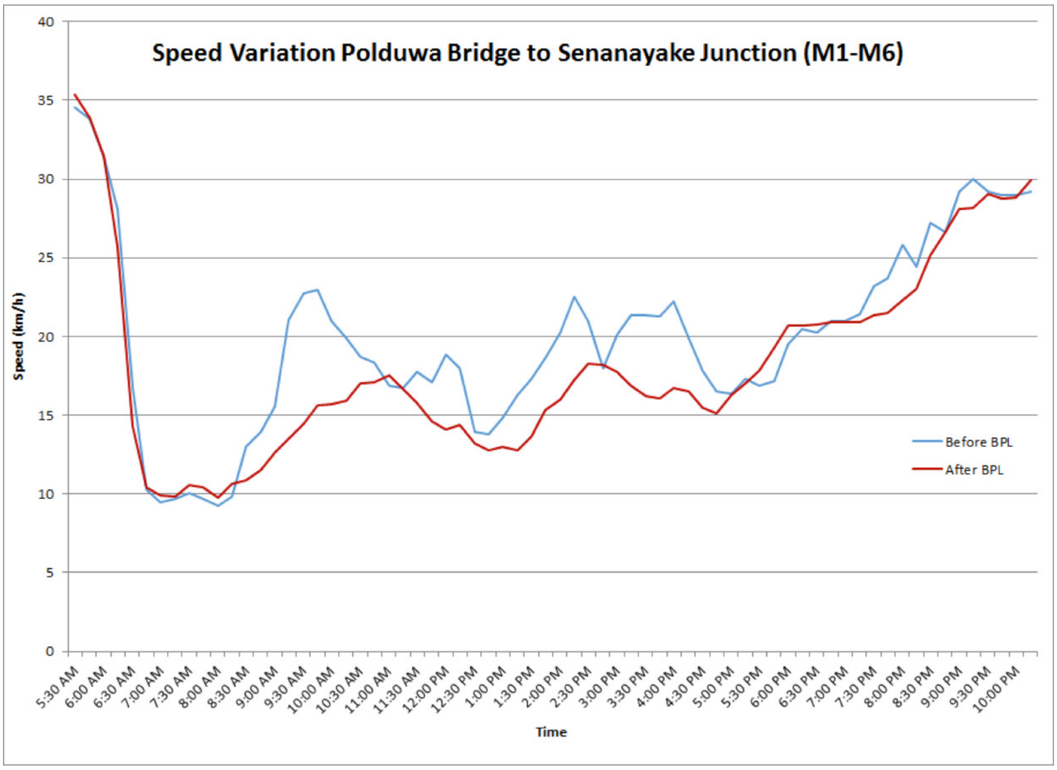


Figure 58: Speed variation before and after implementation of bus priority lanes

The Figure 58 indicates the speed variation along the road from Diyatha Uyana to Senanayake junction. It could be observed that the average space mean speed on the segment has reduced after implementation of bus priority lanes. Before the implementation of bus priority lanes, the off-peak space mean speed was between 15 to 20 km per hour. After implementation of bus priority Lane it has reduced to 12 km per hour during off-peak time. There is no significant change in average space mean speed during peak hours in the morning while there's a slight reduction in speed from 5:30 p.m. to 6:30 p.m.

Analysis 2 – Diyatha Uyana – Senanayake Junction

After implementation of the pilot project, the bus priority lanes were implemented on roads from 6 September 2017. The lanes operated during the morning period and it was not operated in the evening. After conducting a similar data collection, the analysis was developed. Figure 59 shows the temporal variation of space mean speed from Diyatha Uyana to Senanayake junction. There is a special case which could be observed in the analysis. The school holidays were observed during the period in which the bus priority lanes were operated. Hence the analysis was extended to identify variation of space mean speed during school holidays and after school holidays. It could be observed that there is a significant increase in speed during school holidays during the morning peak traffic time. The series with dotted lines show the travel time variation on normal days with school holidays and without school holidays. The series with orange line shows the speed variation after implementation of bus priority lanes during school holidays. The series with green line shows the variation of speed after school holidays. After evaluating all the scenarios, it could be observed that the speed has reduced due to the implementation of bus priority lanes in both school holidays and normal days.

The Appendix C keeps the spatiotemporal graphs for average weekly segmental speed variation from Diyatha Uyana to Senanayake Junction. The total segment was divided into 7 minor segments to collect speed information. The average speed was illustrated on the graph for each week starting from 28 August 2017. By evaluating the spatiotemporal graphs it could be identified that the speed has reduced and it is possible to identify critical links in which the speed has dropped significantly. For example the speed reduction in between Diyatha Uyana Junction to Kotte Road is significantly high compared to the speed reduction in other segments. In other words, this segment operates as a bottleneck during morning peak hours

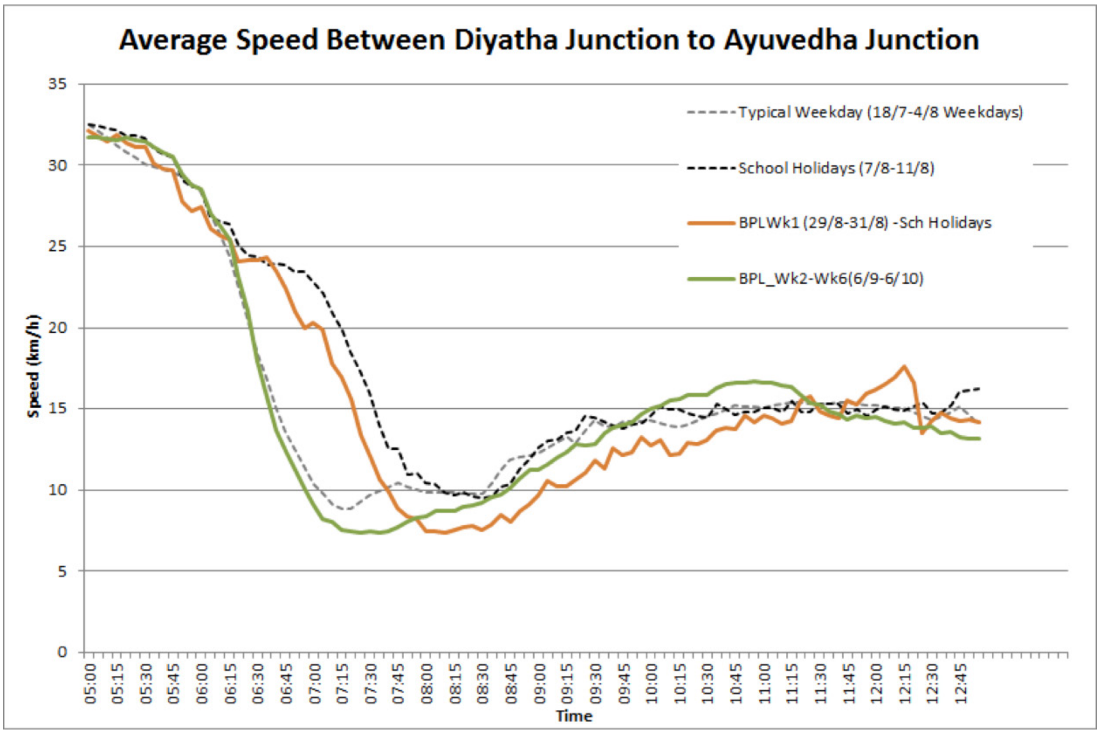


Figure 59:Average speed variation with the implementation of Bus Priority Lanes

.

6.3.2 Implementation of reversible lanes in CMA

Reversible lanes are implemented in a city when there is an asymmetric traffic flow in either direction. Asymmetric traffic flow could be observed during peak hour traffic conditions in which there is a high demand for a single direction compared to the traffic to opposite direction. By implementing reversible lanes capacity required during peak time traffic could be accommodated by shifting the direction of a lane in the opposite direction. This could be identified as an economical alternative in situations where the peak time traffic demand is significantly higher than the normal(80).

An implementation of reversible density is required to ensure that a significant success rate could be achieved within a short period of time. And this record proper communication and public participation is a crucial factor to ensure the success of the strategy. Local authorities should identify the best location for implementation of reversible lanes which would reduce the travel time and would not cause a significant effect in opposite lane due to a reduction in the facility. When reversible lanes are implemented on roads which have non-motorized passenger lanes proper safety measures should be taken to ensure pedestrian safety and reduce accidents(80).

The evaluation of the success should be based on parameters such as an increase in average speed reduction in travel time and increase in traffic flow. Hence evaluation of reversible lanes using traffic data has a significant importance on achieving success.

6.3.2.1 Methodology

The public authorities decided to implement reversible lanes in Rajagiriya area during the peak time traffic conditions of the morning by converting opposite direction lane towards Colombo. Rajagiriya is situated between Colombo and Sri Jayawardenapura which are commercial city and the administrative city of Sri Lanka respectively. Therefore, extreme traffic situations could be observed during morning peak time traffic conditions towards Colombo and towards Battaramulla due to home to work tips and home to school trips. It was observed that traffic flow towards Colombo is significantly higher than the flow towards Baramulla. Therefore, it was proposed to implement reversible lanes from 6 a.m. to 8:30 a.m. On weekdays.

The objective of this project was to reduce the travel time towards Colombo and accommodate passenger mobility. Reversible lanes operated for a length of 3 km in which significant traffic congestion is observed. In order to evaluate the results of project implementation, before and after analysis was conducted using travel time information given by Google Distance Matrix API

The segment in which the reversible lane was implemented (Diyatha Uyana - Ayurweda Junction) was divided into 11 minor segments as shown in the map.(see Figure 60) travel time between these minor segments was collected and analyzed. The data connection was started one week prior to the implementation of the project, and it was carried out throughout the project period.

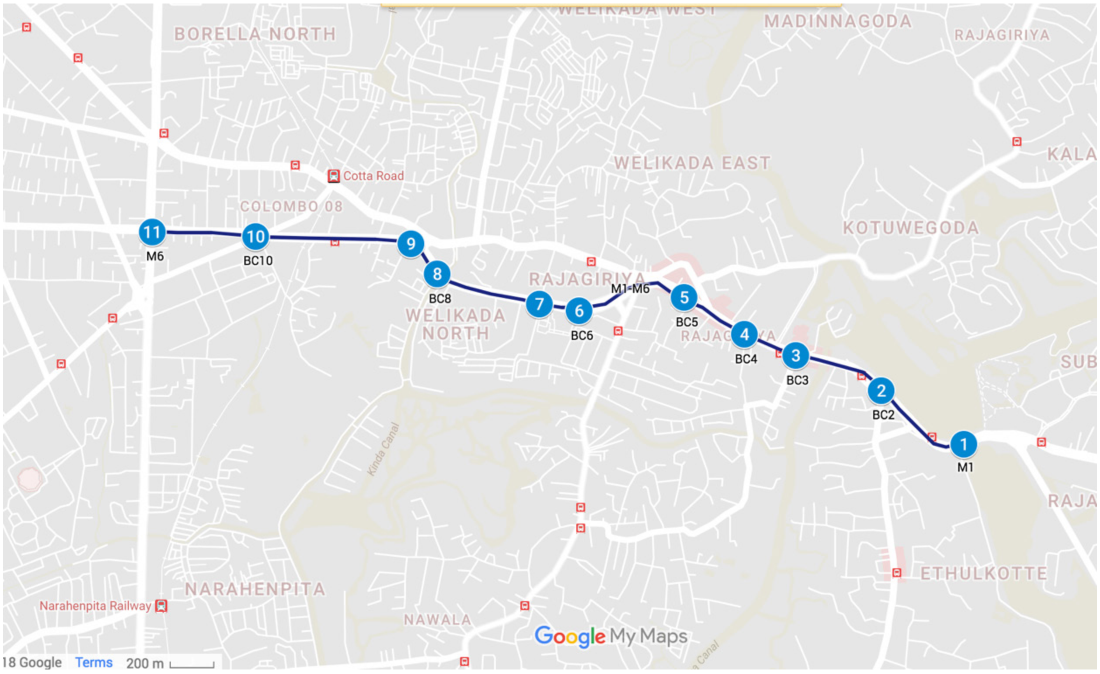


Figure 60: Map od the road segment in which reversible lanes were operated

Data collection was started on 5th March 2017 and conducted on 28 March 2017. Segmental travel time data was collected from 6 a.m. To 10 a.m. At 5-minute intervals.

6.3.2.2 Analysis and evaluation

After collecting data, it was analyzed to identify the impact of implementing reversible lanes on travel time and average speeds. Due to the implementation of reversible lanes, the traffic towards Colombo and the traffic towards Battaramulla were affected. During the operation of reversible lanes, the turning moments from either direction were limited along the road and allowed only at major junctions. Hence the through movement was not affected.

The Figure 61 shows travel time variation on both directions before implementation of reversible lanes. The travel time towards Colombo during peak time traffic conditions is about 25 to 30 minutes while the travel time towards Battaramulla during that time is about 8 minutes. The B240 road in which the lane revision was planned, is a 4 lane road having two lanes each towards Colombo and towards Battaramulla. Hence it could be identified that by adding another land towards Colombo could reduce the travel time during peak time traffic conditions. By implementing the reversible lanes, there's a possibility of increasing the travel time towards Battaramulla from the existing value as only a single Lane be available for the vehicle movement towards Battaramulla.

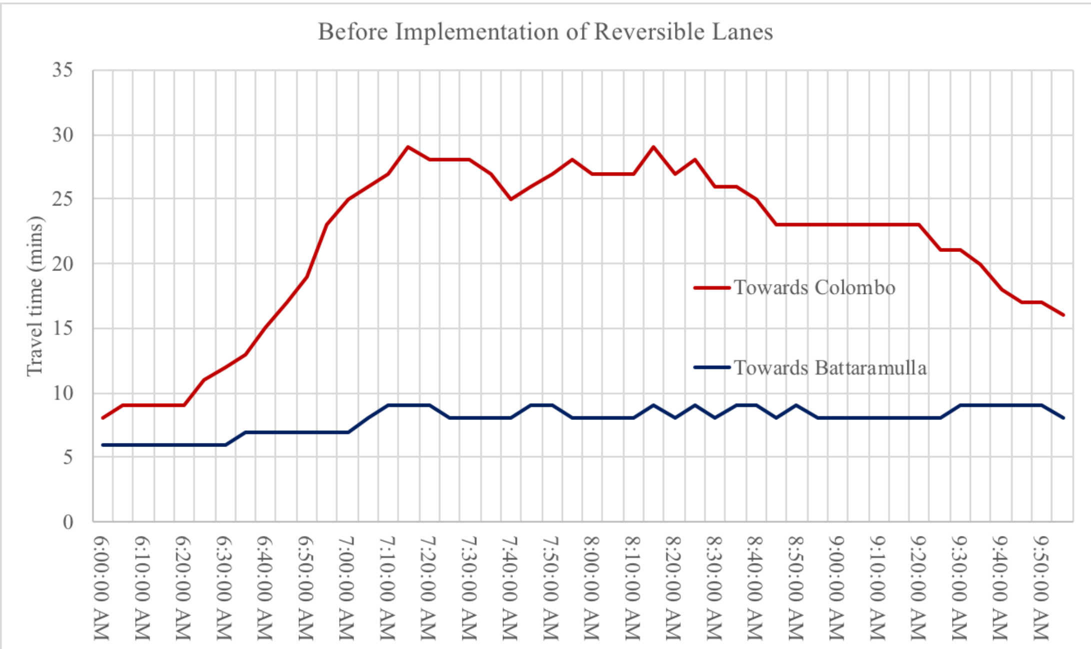


Figure 61 : Travel time variation before implementation of reversible lanes

The Figure 62 and Figure 63 show travel time variation after implementation of reversible lanes. The variation of travel time for 3 days are shown in the graph. It could be observed that the travel time towards Colombo has reduced by about 8 to 10 minutes while the travel time towards Battaranulla has increased by about 6 minutes. Due to the different methods of traffic controlling and allocation of police officers to direct the traffic the subttle variation in travel time during the three days could be observed.

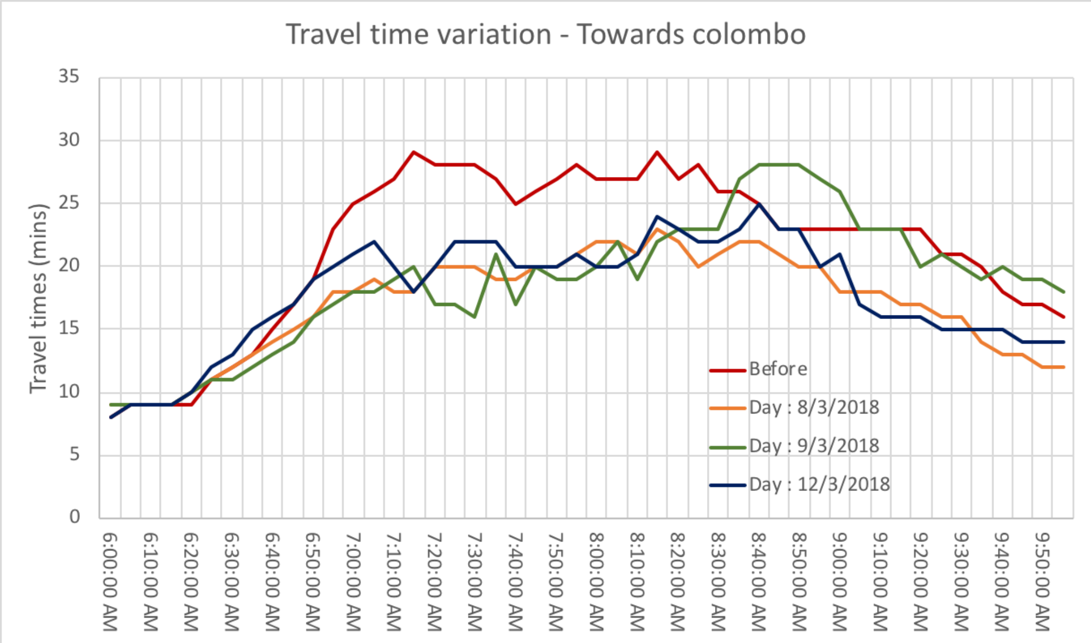


Figure 62 :Travel time variation towards Colombo after implementation of reversible lanes

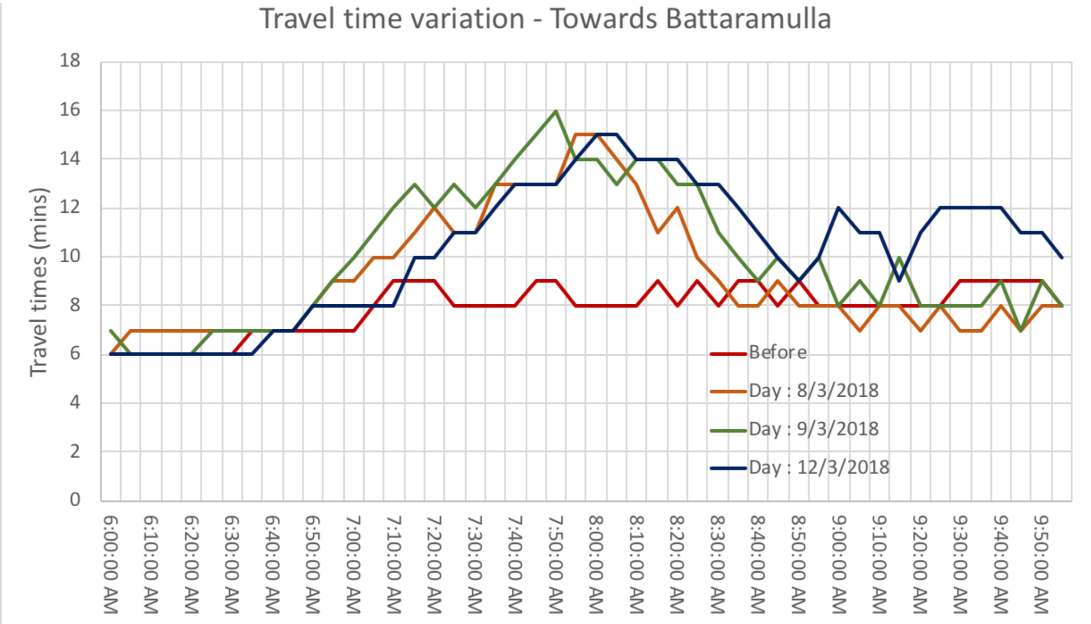


Figure 63:Travel time variation towards Battaramulla after implementation of reversible lanes

The reversible lane was operated until 8:20 a.m. on all three days. But it could be observed that there is a reduced travel time towards Colombo even after ending reversible lanes at 8:20 a.m. This indicates that the cumulative waiting time on traffic has reduced with the implementation of the reversible lanes. The travel time towards Battaramulla has come to its normal condition after 8:40 a.m. oboth 8th and 9th of March. On 12th March there was a protest near parliament which increased the travel time.

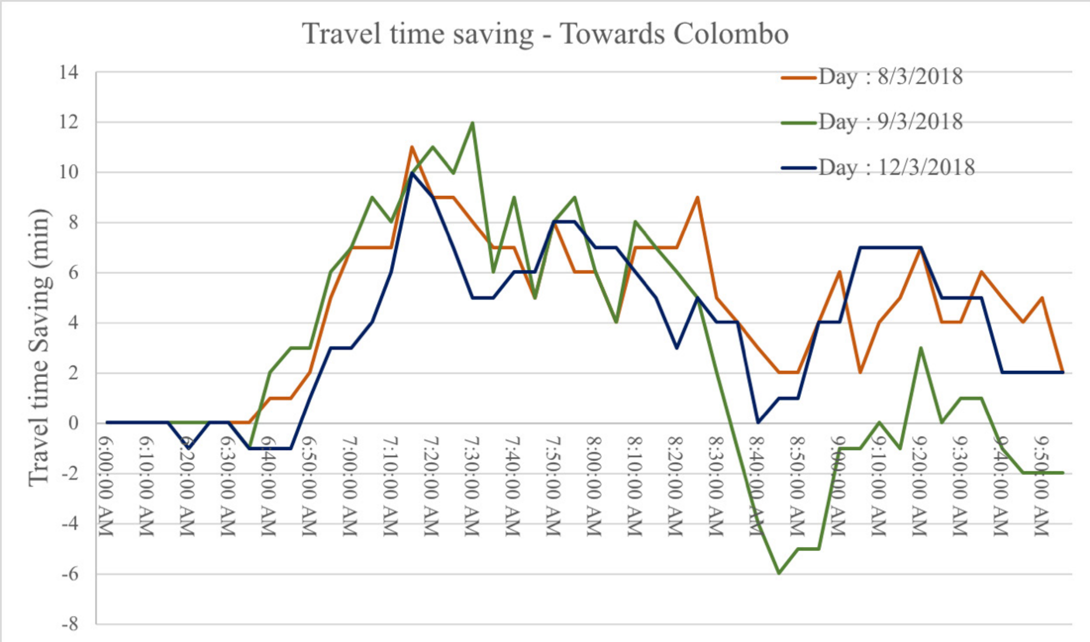


Figure 64 : Travel time saving towards Colombo after implementation of reversible lanes

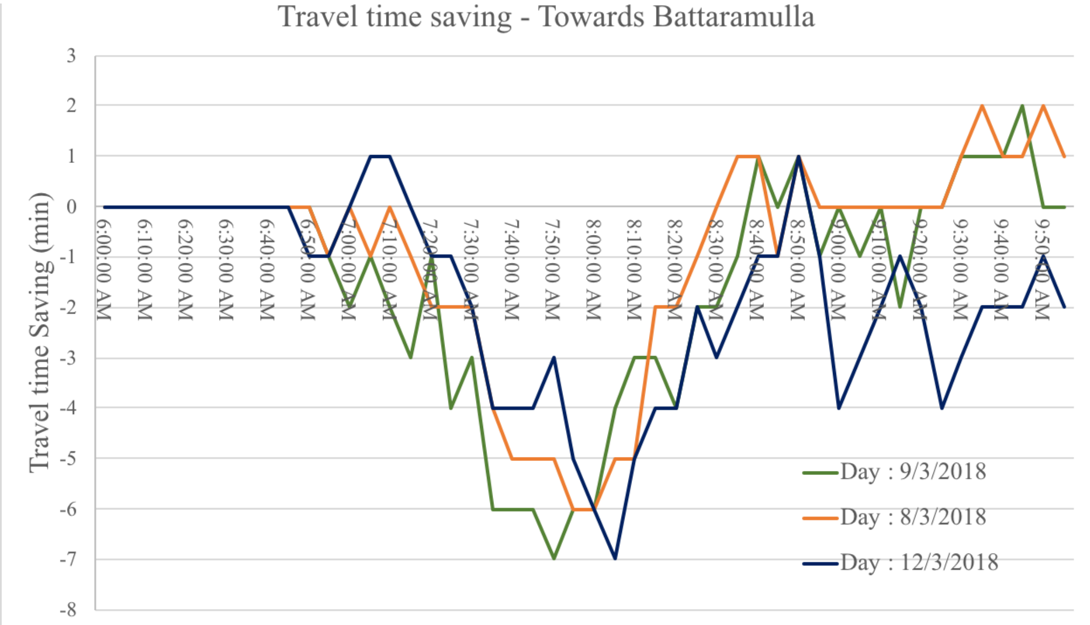


Figure 65:Travel time saving towards Battaramulla after implementation of reversible lanes

The temporal variation of travel time saving could be understood by observing Figure 64 and Figure 65. The two graphs illustrate travel time saving towards Colombo and travel time saving towards Battaramulla respectively. The maximum travel time saving of 12 minutes and an average travel time saving of 6 minutes could be observed towards Colombo. Maximum travel time delay of 7 minutes and an average delay of 3 minutes could be observed towards Battaramulla.

When considered the traffic flow travel time saving of 6-10 minutes during the peak traffic condition towards Colombo, this could make a significant movement in traffic as Rajagiriya is a major bottleneck on this corridor.

7 Discussion and Conclusion

7.1 Summary of main findings

It was possible to realize all the objective after a comprehensive analysis which is celebrated in earlier chapters. The first objective was to understand the use of crowdsourced data and hybrid positioning systems in transportation engineering. This objective was realized by conducting a detailed literature review which illustrated how active methods and passive methods of crowdsourcing data could be used in transportation-related activities. It was identified that crowdsourced information is an optimum source of data collection in real-time traffic management and related activities. Using mobile phones as a crowd sensor will enable to obtain reliable and consistent traffic information. It was identified that the travel time obtained from Google Distance Matrix API is processed information which is based on crowdsourced mobile phone data. Hence travel time information given by Google Distance Matrix API is found to be a good source of gaining travel time information to evaluate city traffic.

The second objective of developing a crowdsourced data mining platform and data collection portal to collect travel time information was realized by developing web application and a cloud server to collect data. This data collection portal has enabled collection of travel time data for multiple road segments simultaneously for any given period of time at any given frequency. This feature has outrun all the restrictions of existing travel time data collection methods. The ability to collect travel time data simultaneously for multiple links and getting a representative sample of the whole vehicle fleet was not possible earlier. Possibility to collect data continuously for a long period of time at any given frequency will enable to collect a large set of travel time information which could be incorporated in data analysis to identify existing problems with high accuracy.

The third objective of verification of travel time information obtained from crowdsourced data with actual travel times observed with other methods of travel time data collection was achieved by comparing the data obtained from the API with the travel time given by probe vehicle data. The data analysis was possible to identify that there is over 90% accuracy of travel time information given by Google Distance Matrix API for short distances and a 70% accuracy for long distances. The verification was extended to identify the accuracy of travel time estimates given for very short segments, which resolve that that there is more than 75% accuracy of the travel time estimates given for very short segments. The license place matching survey conducted to identify the travel time variation of different vehicle types with the Google travel time estimates. The results show that there is a significant correlation in which the difference between Google travel time estimates and weighted mean travel time obtained from the survey is below 1%.

The fourth objective of application of crowdsourced travel time information in transport engineering was successfully achieved by applying the collected data in several applications. A traffic flow estimation model using traffic flow data and Google travel time estimates by using machine learning principles was developed successfully for urban single lane roads. A road bottleneck identification method-based on Google travel time estimates was developed. The use of Google travel time data to evaluate transport projects was illustrated by considering two transport projects which were implemented in Colombo Metropolitan Area.

7.2 Evaluation of the Data Collection Method

The data collection method is based on accessing the Google Distance Matrix API and collecting travel time information for the required segments. The travel time information given by Google Distance Matrix API is based on the crowdsourced information provided by mobile phone users who use Google services. The methodology illustrates how mobile phone users share the location information with Google location servers. In the methodology chapter, it could be identified that there is a vast amount of people who use Google services and Google has the comparative advantage of collecting user location information by providing services such as Google Maps, navigation, traffic information and other location-based services.

The location identification of mobile phone users is based on a hybrid positioning methodology in which Google uses GPS signals, cellular Signals and Wi-Fi access points data to identify user location efficiently. The hybrid positioning system could be identified as an optimum approach in situations where GPS signals are not readily available, or mobile users do not use GPS facilities due to high power consumption. This type of scenario was available in the environment in which the study was conducted.

The data collection method is based on a web application and a cloud server. The methodology used in this study ensures the consistency of data collection using the Google Distance Matrix API. This cloud-based data collection method could be identified as a commendable approach in data collection when compared to the existing methods of traffic data collection practised in Sri Lanka. Most of the research work in this study was conducted within the free data limit of the Google Distance Matrix API.

The data collection was only possible for driving mode. Therefore the travel time of a selected segment could be obtained only for driving that segment. Although there is provisions to collect travel time information of walking, cycling and transit, it was not an interested in this study due to unavailability of data for the environment in which the study was conducted. Although the Distance Matrix API supports many languages, the study was conducted only in English.

In order to use this methodology, the user has to have a good background in computer programming and handling web applications. The usability of the methodology for users without programming knowledge was not addressed in this study. The amount of data collected depends on the free limit allowed by Google. Hence the user has to be aware of the Google policy and payment structure of accessing the Google Distance Matrix API before collecting travel time information.

The study was not able to extend data collection for large sets of segments in which the input file has more than 500 segments. There could be compatibility issues in the proposed methodology when the user required to collect a very large set of data which has more than 500 segments in its input file.

7.3 Evaluation of the Verification method

After developing the research methodology and the data collection platform, the study was extended to verify travel time information obtained from Google Distance Matrix API. The verification was based on manual travel time information collected via different modes of travel time collection. The study develops GPS device which could create GPS log files of a moving vehicle and transfer information to a web server in order to collect travel time information manually. The use of GPS device could be identified as a commendable approach to collect travel time information manually.

The study elaborates on the telecommunication infrastructure of Sri Lanka in order to give a better understanding about the usability of Google services in Sri Lanka. The study evaluates the cellular signal connectivity by using several data sets and Identify the signal availability of Sri Lanka with respect to the signal coverage information given by network providers and open data sets.

The verification of Google travel time was conducted to both short distance trips and long-distance trips. Short distance trips were evaluated in both peak traffic time conditions and off-peak traffic conditions. The analysis of the verification process identifies that there is a significant accuracy of Google travel time data when compared with the manual data collected from the GPS location information. It was established that there is over 98% accuracy of Google travel time estimates with actual travel time for short distance trips. The accuracy of travel time estimates for long distance trips was identified as 79%. Hence the verification method establishes that Google travel time data could be used to evaluate short distance trips and long-distance trips with significant accuracy.

The verification was extended to identify how the Google travel time estimates vary with different vehicle types which are moving on the road. It was understood that the Google could not monitor a vehicle while it can only identify mobile phones moving in vehicles. License plate survey was conducted to understand the association of Google travel time data to the travel time of different vehicle types and the vehicle composition of the fleet. The results of the verification method indicate that there is a significant Association between the Google travel time data and weighted mean travel time obtained by considering the vehicle composition and mean travel time for each vehicle type. The study does not look into the shortest segment in which the travel time estimate could be obtained with significant accuracy.

By considering all these factors, the verification methods followed, and analysis presented in this study could be identified as comprehensive enough to evaluate the Google travel time data by referring to manual travel time information.

7.4 Evaluation of the Applications

Upon verification of the travel time information obtained from Google Distance Matrix API, it was possible to use this information in transport planning activities. This study elaborates on three major applications which are very important in the traffic Engineering domain. The first application elaborates on using Google travel time data to estimate traffic flow based on machine learning principles. Deviating from traditional probability estimation methods used in past research, this study proposes to use K- Nearest Neighbour clustering-based regression method in the prediction of traffic flow. In developing the machine learning model Google Distance Matrix API data, archived traffic flow data, and geometric data were used. The estimates and the machine learning model developed in this application has a significant accuracy in predicting traffic flow. The study establishes significant accuracy of using K nearest neighbour regression model in traffic flow prediction. A linear correlation of 0.97 is obtained between the predicted value and the observed value, and the cross-validation method shows that the machine learning model does not overfit.

The successful implementation of this traffic flow estimation model has many practical applications. It enables to predict the traffic flow based on the speed parameters for urban single lane road. Hence this model could be used in many planning works and to evaluate the performance of highways and its capacity. This could be an alternative to obtain traffic flow in an environment where traffic sensors and detectors are not available.

The second application discusses the use of Google Distance Matrix API travel information data to identify bottlenecks along road segments. Both static road bottlenecks and moving road bottlenecks could be identified with this method and it is based on the spatiotemporal analysis of speeds along a road segment. The successful implementation of this bottleneck identification methodology enables to analyse arterial roads and identify the low performing links. The ability to conduct the study with less amount of manual data collection and high expenditure is the main advantage of this method.

The third application elaborates on using travel time information obtained from Google Distance Matrix API on the evaluation of transport projects. This application suggests to Conduct before and after evaluation of transport projects-based on the travel time and space mean speed. Two case studies are presented on implementing bus priority lanes and implementing reversible lanes in Colombo Metropolitan Area. With Respective to the implementation of bus priority lane, it is possible to identify how the travel time and space mean speed was badly affected. It was observed that travel time has increased while space mean speed has reduced with the implementation of bus priority lanes. It is important to note that Google only monitors the mobile phone users moving on the road. It was observed in literature that 50% of passengers use public transport in their daily commuting. Which means if there is an increase of speed for mobile phones moving on the busses and equal reduction in speed for mobile phones moving on other vehicles, the average speed for a passenger should not change from the earlier condition. What is observed after the analysis was there is a reduction in speed with the implementation of bus priority lanes. This indicates that the average moving speed of a passenger on the road has reduced. Hence this analysis was evident to identify that the implementation of bus priority lanes has no major contribution to improve the traffic.

A similar analysis was presented with respect to the implementation of reversible lanes. It was identified after the analysis that there is a significant reduction in travel time of 8 minutes for passengers who are travelling towards Colombo while there is a considerable increase in travel time of 3 minutes for passengers who are travelling towards Battaramulla. By considering the traffic flow difference between two opposite directions, it could be identified that the travel time saving of 8 minutes has a significant impact on traffic improvement. The increase of 3 minutes has a lesser impact when compared to the positive impact gain by saving 8 minutes towards Colombo. Hence with this analysis, it was possible to identify that the implementation of reversible lanes in Colombo Metropolitan area during the morning peak hour has a major contribution to improve the traffic.

By referring to these facts, it could be concluded that the applications suggested in the study is comprehensive enough to establish how travel time information obtained from Google Distance Matrix API could be utilized for successful evaluation of transport projects.

7.5 Best practices to follow

This study proposes a novel method of collecting travel time information-based on the Google Distance Matrix API. Based on the experience of the research there are several best practices which need to be followed in order to conduct data collection and analysis. This section will present the proposed best practices to be used in data collection and data analysis which could be helpful in the application of this methodology in future work.

The first instruction to follow in data collection is to understand that this is a data collection method based on a web application and storing the data on a cloud server. The reliability of the data collection and consistency is protected by using the web application and Cloud Servers. Therefore, it is required to understand the difference between collecting data from Distance Matrix API and gathering information from accessing Google map or similar service from user mobile phone.

By implementing data collection method, a large set of traffic information data is gathered on the server If the data collection is conducted for a longer period of time. Therefore, it is required that the user is confident enough in handling big data sets and working with databases to analyzing such large data set. Further, it is recommended that the user has at least basic knowledge of computer programming and web applications to engage in data collection.

It is required to ensure that the maximum execution time of a PHP script is not exceeded in an execution. Therefore, controlling the input file size is recommended to ensure that the API calls are within the maximum execution time of the PHP script. for practical purposes. Therefore, it is suggested that the input file does not have more than 500 segments.

It is recommended to collect data from the API by passing single origin-destination pair for every API call. Passing of multiple origins and destination is not recommended as it generates unimportant data within the free limit of data collection. It is recommended to use GPS coordinates for origins and destination in order to ensure the accuracy. By passing place names and addresses to the API will result in giving wrong information.

Special care has to be taken in the generation of the input file. The GPS coordinates should be placed on the road itself on defining the origins and destinations to collect travel time on roads. Figure 66 shows two methods of defining the GPS coordinates of origins and destinations which could be used for travel time collection. The GPS co-ordinate should be either on a lane of the road, or it should be on the centre median.

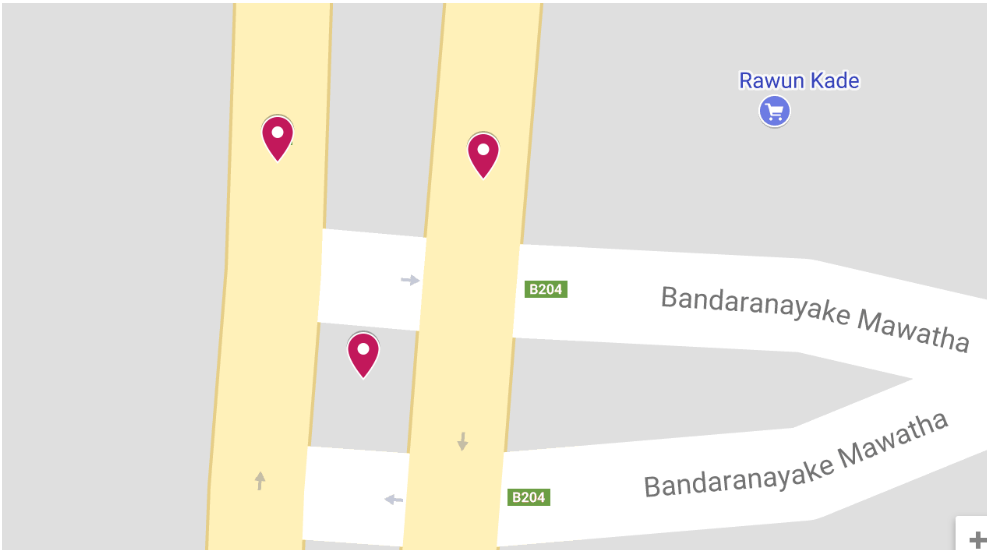


Figure 66:Defining the GPS coordinate of an origin or destination

The distance and the travel time is calculated based on the driving route in between the origin and destination. It is important that the route distance will give the actual distance between two points. Therefore, origin and destination GPS coordinates should be placed on the road or on the centre median. If the origin GPS coordinate is defined on a lane of the road, then the destination should also be on the same lane on the centre median. If the destination was placed elsewhere the distance value and travel time will be wrong. The Figure 66- Case A illustrates how the origin and destination points should be defined in order to get the correct travel route. The Figure 67 – case B illustrates an erroneous route formation due to the consequences of defining origin and destination pair mistakenly.

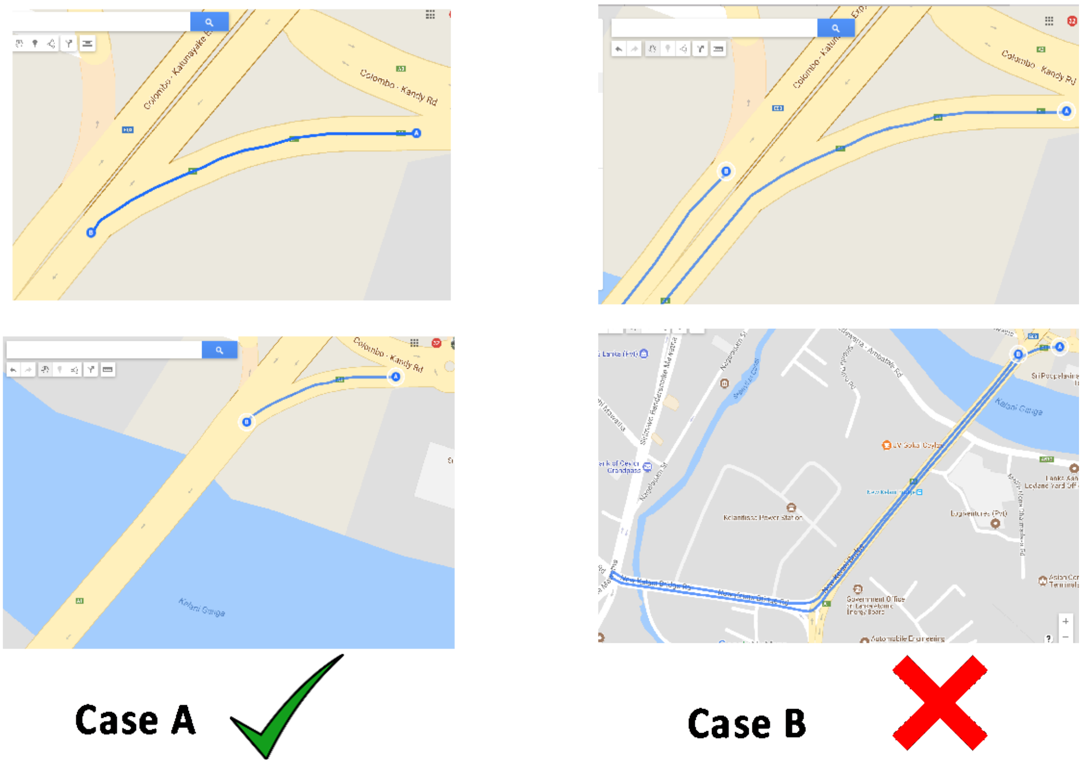


Figure 67 : Marking origin and destination correctly on the map

When an origin-destination pair is passed down to the API, Google en-route between the origin and destination by considering the shortest travel time path. Figure 68(left) shows such an example in which the travel time is required along The AA002 Galle - Colombo Road (Ash colour road), but the route is selected along the Marine Drive Road (Blue colour road) . If the route does not lie on the road segment in which the travel time is required, then an intermediate point between the origin and destination should be added as shown in Figure 68 (right). Hence necessary care should be taken defining origins - destinations that the Distance Matrix API select the required segment in routing.

It is possible to use the Google Distance Matrix API to consider junction delay near an intersection. If the junction delays to be included in the travel time, the destination should be selected as it is shown in the Figure 69(left) . If the junction delays should be avoided, then the destination should be selected as shown in Figure 69(right).

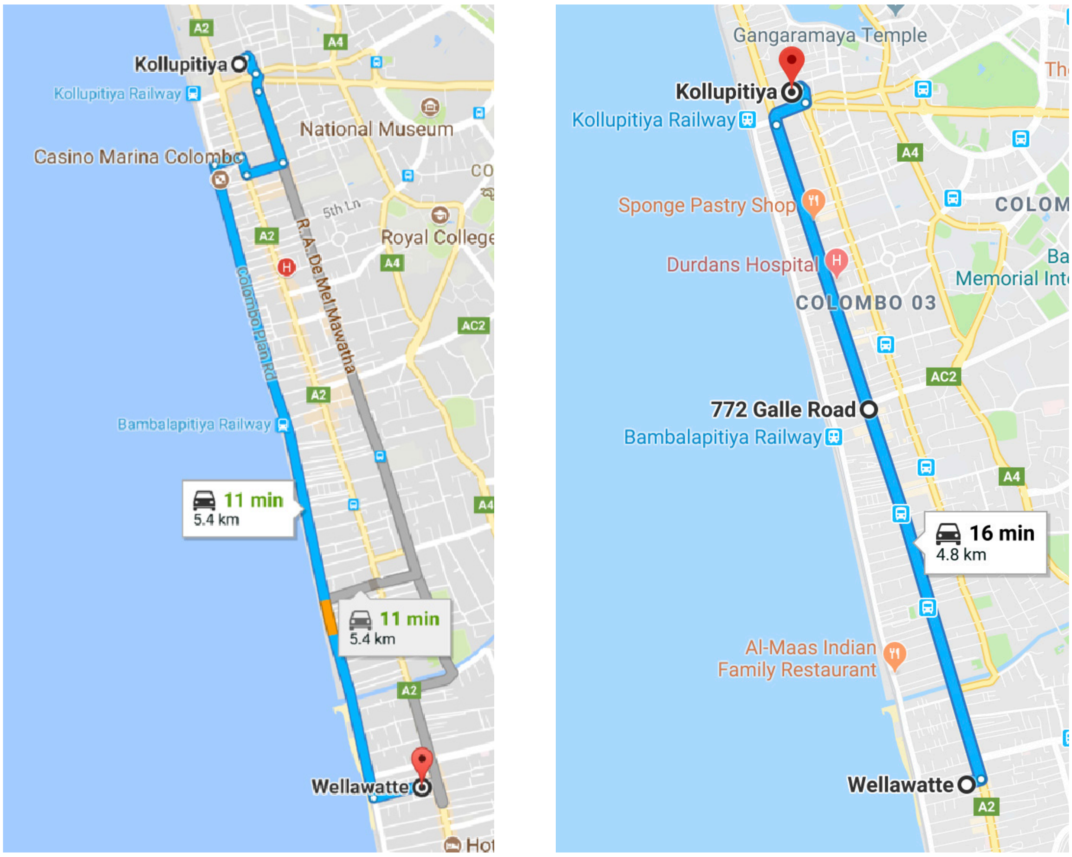
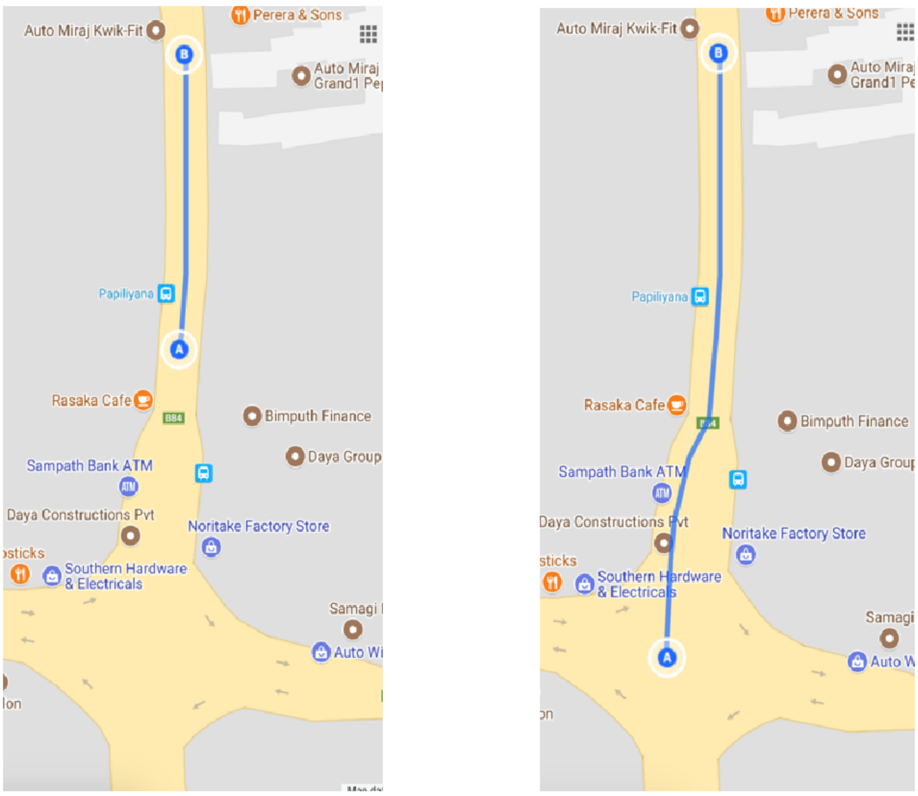


Figure 68 : Origin and destination selection to avoid alternative route



Doesn't include junction delay **Includes junction delay**

Figure 69 : Selection of origin and destination to include junction delay

7.6 Directions to future work

Use of Distance Matrix API travel time data for Transport planning activities is elaborated in this study. The study proposes a novel method of data collection-based on the API and the collected data was verified with the actual observed data. Then the study proposes several applications which use Google Distance Matrix API travel time data and the success of these applications were discussed. Following future work could be identified as the way forward in improving the study.

Currently, the method is only available to get information of driving mode. It is possible to extend to other modes such as walking cycling and public transportation. With the availability of public transportation schedules and frequencies, the travel time information could be obtained for transit mode. In order to get cycling travel time information, it is required to define cycling lanes on the map.

During the verification process, it was verified that this method could be used to analyze very short segments. Google Distance Matrix API could be used to estimate the average queue length at an intersection. Using the data, it is possible to graph the temporal variation of queue length on the intersection. This could be used in junction analysis and improvement of traffic signal systems.

Currently, the methodology is based on a web application and a user should have a basic knowledge of computer programming to collect data efficiently. Therefore, it is proposed to develop a graphical user interface based on the web application to collect travel time data from Google Distance Matrix API. In that way, the accessibility and convenience of using the methodology could be increased.

The traffic flow prediction model was developed based on the database in which travel time data and traffic flow data was available. It is proposed to collect travel time data from Google Distance Matrix API whenever the traffic flow survey is being conducted. The database could be expanded with this method, and traffic flow estimation for other types of roads such as multi Lane roads rural roads could be taken by expanding the machine learning model.

In the verification process, it was observed that there are road segments which have similar distance but significantly different travel times. Hence it is possible to develop an accessibility indicator-based on travel time for several locations of the city. Using the same principle, it is possible to identify the traffic movement in the whole city with respect to space and time.

This method could be implemented to build a travel time database in which the historical travel time variation could be analyzed. By collecting data for a long time, it is possible to identify increasing patterns of road traffic and efficiency of controlling methods and estimate average time spent by passengers on the road.

7.7 Concluding Remarks and recommendations

This study proposes a novel method of using travel time information obtained from crowdsourced data in transportation planning activities. The Google Distance Matrix API is used to collect travel time information in which crowdsource data is given out as processed information. A data collection platform was developed by the study and the obtained travel time estimates were verified with the actual observations under several categories. Several applications which use the travel time estimates given by the Google API are presented. It could be concluded that the use of travel time information given by Google Distance Matrix is a reliable, consistent and economical method of collecting travel time information. The low cost of investment involved in this method is an attractive factor. It is recommended that public authorities and organizations responsible in managing city traffic, use this tool to improve traffic management plans and transport policies.

Reference List

[1] Statista, 'Cumulative five-year transport infrastructure spending breakdown globally between 2016 and 2020, By region', 2017. [Online]. Available: https://www.statista.com/statistics/728969/worldwide-cumulative-spending-on-transport-infrastructure-by-region/.

[2] T. N. Dhamala and U. Pyakurel, 'Significance of Transportation Network Models in Emergency Planning of Cities', *Cities People Places An Int. J. Urban Environ.*, vol. 1, no. 2, pp. 58–76, Apr. 2016.

[3] E. D'Andrea and F. Marcelloni, 'Detection of traffic congestion and incidents from GPS trace analysis', *Expert Syst. Appl.*, vol. 73, pp. 43–56, May 2017.

[4] R. Samarajiva, S. Lokanathan, K. Madhawa, G. Kreindler, and D. Maldeniya, 'Big data to improve urban planning', *Econ. Polit. Wkly.*, vol. 50, no. 22, pp. 42–48, 2015.

[5] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, 'Crowdsourcing with Smartphones', *IEEE Internet Comput.*, vol. 16, no. 5, pp. 36–44, Sep. 2012.

[6] Ministry of Megapolis and Western Development, 'Western Region Megapolis Transort Master Plan', Colombo, Sri Lanka, 2016.

[7] R. Gudishala, C. Wilmot, and A. Mokkapati, 'Travel Time Estimation Using Bluetooth', Louisiana, 2016.

[8] S. M. Turner, W. L. Eisele, R. J. Benz, and J. Douglas, 'Travel time data collection handbook', Washington DC, 1998.

[9] X. Liu, S. Chien, and K. Kim, 'Evaluation of floating car technologies for travel time estimation', *J. Mod. Transp.*, vol. 20, no. 1, pp. 49–56, 2012.

[10] E. Emam and H. AI-Deek, 'Using Real-Life Dual-Loop Detector Data to Develop New Methodology for Estimating Freeway Travel Time Reliability', *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1959, pp. 140–150, Jan. 2006.

[11] R. Ewing and K. Bartholomew, 'Best Practices in Metropolitan Transportation Planning', 2011.

[12] S. I.-J. Chien and C. M. Kuchipudi, 'Dynamic Travel Time Prediction with Real-Time and Historic Data', *J. Transp. Eng.*, vol. 129, no. 6, pp. 608–616, 2003.

[13] D. M. Scott, D. C. Novak, L. Aultman-Hall, and F. Guo, 'Network

Robustness Index: A new method for identifying critical links and evaluating the performance of transportation networks', *J. Transp. Geogr.*, vol. 14, no. 3, pp. 215–227, 2006.

[14] E. P. Dennis, R. Wallace, and B. Reed, 'Crowdsourcing Transportation Systems Data', Michigan, 2015.

[15] S. Amini, I. Gerostathopoulos, and C. Prehofer, 'Big data analytics architecture for real-time traffic control', in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2017, no. Tum Llcm, pp. 710–715.

[16] V. Frias-Martinez and E. Frias-Martinez, 'Spectral clustering for sensing urban land use using Twitter activity', *Eng. Appl. Artif. Intell.*, vol. 35, pp. 237–245, 2014.

[17] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence, 'Improving traffic prediction with tweet semantics', in *IJCAI International Joint Conference on Artificial Intelligence*, 2013, pp. 1387–1393.

[18] D. Mchugh, 'Traffic Prediction and Analysis using a Big Data and Visualisation Approach', 2015.

[19] S. Spyratos, D. Stathakis, M. Lutz, and C. Tsinaraki, 'Using Foursquare place data for estimating building block use', *Environ. Plan. B Urban Anal. City Sci.*, vol. 44, no. 4, pp. 693–717, Jul. 2017.

[20] Y. Sun and M. L, 'Investigation of Travel and Activity Patterns Using Location-based Social Network Data: A Case Study of Active Mobile Social Media Users', *ISPRS Int. J. Geo-Information*, vol. 4, no. 3, pp. 1512–1529, 2015.

[21] Waze Mobile Limited, 'Privacy Policy', 2018. [Online]. Available: https://www.waze.com/en-GB/legal/privacy. [Accessed: 25-May-2018].

[22] INRIX Inc, 'Privacy Policy', *INRIX Traffic App*, 2017. [Online]. Available: http://inrix.com/privacy-policy/. [Accessed: 21-Jul-2017].

[23] HERE Global B.V., 'Privacy Policy', *HERE Global B.V.*, 2018. [Online]. Available: https://legal.here.com/en-gb/privacy. [Accessed: 25-May-2018].

[24] Cellint Traffic Solutions PVT LYD, 'Cellint Traffic Sense', *Cellint Traffic Solutions*, 2018. [Online]. Available: http://www.cellint.com/trafficsense/#SystemInterfaces. [Accessed: 04-Jan-2018].

[25] Tiramisu Transit LLC, 'Tiramisu - The Real-time Bus Tracker', *Tiramisu Transit LLC*, 2017. [Online]. Available: https://www.transitwiki.org/TransitWiki/index.php/Tiramisu. [Accessed: 06-Sep-2017].

[26] U.S. Air Force., 'Official U.S. government information about the Global Positioning System (GPS) and related topics'. [Online]. Available: https://www.gps.gov/.

[27] S. Zirari, P. Canalda, and F. Spies, 'WiFi GPS based combined positioning algorithm', *Wirel. Commun. Netw. Inf. Secur. (WCNIS), 2010 IEEE Int. Conf.*, no. Ea 4269, pp. 684–688, 2010.

[28] Z. Farid, R. Nordin, and M. Ismail, 'Recent Advances in Wireless Indoor Localization Techniques and System', *J. Comput. Networks Commun.*, vol. 2013, pp. 1–12, 2013.

[29] M. Moeglein, D. Rowitch, W. Riley, J. James Douglass Deloach, and L. Sheynblat, 'Wireless Network Hybrid Positoning', USOO9335419B2, 2003.

[30] F. Alizadeh-Shabdiz, 'System and method for using a satellite positioning system to filter WLAN access points in a hybrid positioning system', US8130148B2, 2008.

[31] C. Post and S. Woodrow, 'Location is everything-balancing innovation, convenience, and privacy in location-based technologies', vol. 2008, no. December, p. 61, 2008.

[32] N. D. Lane *et al.*, 'Google Announces Launch of Google Maps Mobile with "My Location" Technology', *News from Google*, vol. 48, no. November 2005, pp. 140–150, 2010.

[33] R. (Google) Leiteritz, 'Google's submission to several national data protection authorities on vehicle-based collection of wifi data for use in Google location based services', 2010. [Online]. Available: https://static.googleusercontent.com/media/www.google.com/en//googleblogs/pdfs/google_submission_dpas_wifi_collection.pdf.

[34] Z. Ji, R. Jain, and Google mobile team, 'Google Mobile Blog', 2008. [Online]. Available: http://googlemobile.blogspot.com/2008/06/google-enables-location-aware.html. [Accessed: 07-Sep-2017].

[35] E. U. Brandt, 'Mobile State Determination of Location Aware Devices', US8886457B2, 2011.

[36] Google Inc, ‘Privacy Policy’, 2018. [Online]. Available: https://policies.google.com/privacy. [Accessed: 17-Apr-2017].

[37] Y. Wang, Y. Zheng, and Y. Xue, ‘Travel time estimation of a path using sparse trajectories’, *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’14*, no. 5, pp. 25–34, 2014.

[38] B. Dewulf *et al.*, ‘Examining commuting patterns using Floating Car Data and circular statistics: Exploring the use of new methods and visualizations to study travel times’, *J. Transp. Geogr.*, vol. 48, no. August, pp. 41–51, 2015.

[39] W. Zhao, E. McCormack, D. J. Dailey, and E. Scharnhorst, ‘Using Truck Probe GPS Data to Identify and Rank Roadway Bottlenecks’, *J. Transp. Eng.*, vol. 139, no. 1, pp. 1–8, 2013.

[40] B. Zhao and B. Liu, ‘High-Granularity Dynamic Traffic Flow Prediction Model Based on Artificial Neural Network’. 2017.

[41] Statistica, ‘Global market share held by leading smartphone vendors from 4th quarter 2009 to 1st quarter 2018’. [Online]. Available: https://www.statista.com/statistics/271496/global-market-share-held-by-smartphone-vendors-since-4th-quarter-2009/.

[42] E. E. Lopes, ‘Proposta Metodológica para Validação de Imagens de Alta Resolução do Google Earth para a Produção de Mapas Edésio Elias Lopes’, 2009.

[43] S. K. Atukorala and H. Liyanage, ‘Keeping Children in Sri Lanka Safe and Empowered Online’.

[44] Hootsuites Inc, ‘The global state of digital in 2018’, 2018.

[45] Telecommunication RegulatoryCommission of Sri Lanka, ‘Annual Statistics 2017’, 2017.

[46] GSMA Intelligence, ‘Mobile Connectivity Index’, 2017. [Online]. Available: http://www.mobileconnectivityindex.com/widgets/connectivityIndex/pdf/ConnectivityIndex_V01.pdf. [Accessed: 20-Jan-2018].

[47] D. Helbing, ‘From microscopic to macroscopic traffic models’, in *A Perspective Look at Nonlinear Media*, Springer Berlin Heidelberg, 2012, pp. 122–139.

[48] S. Chandra, ‘Capacity Estimation Procedure For Two-Lane Roads Under Mixed Traffic Conditions’, *J. Indian Roads Congr.*, vol. i, no. 498, pp. 139–167, 2004.

[49] C. Antoniou and H. Koutsopoulos, ‘Estimation of Traffic Dynamics Models with Machine-Learning Methods’, *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1965, no. 1, pp. 103–111, Jan. 2006.

[50] A. Cheng, X. Jiang, Y. Li, C. Zhang, and H. Zhu, ‘Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method’, *Phys. A Stat. Mech. its Appl.*, vol. 466, pp. 422–434, Jan. 2017.

[51] X. Dai, R. Fu, Y. Lin, L. Li, and F.-Y. Wang, ‘DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction’, Jul. 2017.

[52] L. U. Laboshin, A. A. Lukashin, and V. S. Zaborovsky, ‘The Big Data approach to collecting and analyzing traffic data in large scale networks’, *Procedia - Procedia Comput. Sci.*, vol. 103, no. October 2016, pp. 536–542, 2017.

[53] C. Xu, Z. Li, and W. Wang, ‘Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming’, *Transport*, vol. 31, no. 3, pp. 343–358, Jul. 2016.

[54] J. B. Elsner and A. A. Tsonis, ‘Nonlinear Prediction, Chaos, and Noise’, *Bull. Am. Meteorol. Soc.*, vol. 73, no. 1, pp. 49–60, Jan. 1992.

[55] J. Bao, W. Chen, and Z. Xiang, ‘Prediction of Traffic Flow Based on Cellular Automaton’, *2015 Int. Conf. Ind. Informatics - Comput. Technol. Intell. Technol. Ind. Inf. Integr.*, pp. 88–92, 2015.

[56] Q. Shang, C. Lin, Z. Yang, Q. Bing, and X. Zhou, ‘A hybrid short-term traffic flow prediction model based on singular spectrum analysis and kernel extreme learning machine’, *PLoS One*, vol. 11, no. 8, pp. 1–25, 2016.

[57] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, ‘An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction’, *Procedia - Social and Behavioral Sciences*, vol. 96, no. Cictp. pp. 653–662, 2013.

[58] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, ‘Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions’, *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[59] J. Zhao and S. Sun, ‘High-Order Gaussian Process Dynamical Models for Traffic Flow Prediction’, *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2014–2019, Jul. 2016.

[60] IBM Corp, ‘IBM SPSS Modeler for Windows’. IBM Corp, 2016.

[61] R. P. G. K. S. Rajapaksha and J. M. S. J. Bandara, ‘Effect of traffic composition on capacity of two-way two-lane, roads under mix traffic condition’, *Int. Conf. Adv. Highw. Eng. Transp. Syst.*, p. 20, 2017.

[62] J. Zhong and S. Ling, ‘Key Factors of K-Nearest Neighbours Nonparametric Regression in Short-Time Traffic Flow Forecasting’, 2015, pp. 9–12.

[63] T. Wendler and S. Gröttrup, *Data Mining with SPSS Modeler*. Cham: Springer International Publishing, 2016.

[64] C. F. Daganzo, *Fundamentals of transportation and traffic operations*. Pergamon, 1997.

[65] J. Andres L. and R. Alvaro, ‘Exploratory Methodology for Identification of Urban Bottlenecks Using GPS Data’, *J. Traffic Transp. Eng.*, vol. 4, no. 5, pp. 280–290, Oct. 2016.

[66] L. Zhang and D. Levinson, ‘Some Properties of Flows at Freeway Bottlenecks’, *Transp. Res. Rec. J. Transp. Res. Board No 1883*, pp. 122–131, 2004.

[67] D. Hale, A. Hajbabaie, J. Ma, J. Hu, H. Park, and J. Bared, ‘Proposed Data-Driven Performance Measures for Comparing and Ranking Traffic Bottlenecks’, in *Transportation Research Procedia; International Symposium on Enhancing Highway Performance Performance*, 2016, vol. 15, pp. 483–494.

[68] C. Chen, A. Skabardonis, and P. Varaiya, ‘Systematic identification of freeway bottlenecks’, *Transp. Res. Rec. J. Transp. Res. Board*, no. January, 2004.

[69] R. L. Bertini and A. M. Myton, ‘Use of Performance Measurement System Data to Diagnose Freeway Bottleneck Locations Empirically in Orange County , California’, *Transp. Res. Rec. J. Transp. Res. Board No 1925*, pp. 48–57, 2005.

[70] X. Ban, L. Chu, and H. Benouar, ‘Bottleneck Identification and Calibration for Corridor Management Planning’, *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1999, pp. 40–53, Jan. 2007.

[71] Z. Jia, P. Varaiya, C. Chen, K. Petty, and A. Skabardonis, ‘Congestion , excess demand , and effective capacity in California freeways’. pp. 1–38, 2000.

[72] M. J. Cassidy and R. L. Bertini, ‘Some traffic features at freeway

bottlenecks', *Transp. Res. Part B*, vol. 33, pp. 25–42, 1999.

[73] J. Wieczorek, R. Fernández-Moctezuma, and R. Bertini, 'Techniques for Validating an Automatic Bottleneck Detection Tool Using Archived Freeway Sensor Data', *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2160, pp. 87–95, Dec. 2010.

[74] Japan International Coporation Agency; Oriental Consultants Co LTD, 'Ubnan Transport System Development Project For Colombo Metropolitan Region'.

[75] Japan International Coporation Agency; Oriental Consultants Co LTD, 'Ubnan Transport System Development Project For Colombo Metropolitan Region'.

[76] A. Janecek, K. A. Hummel, D. Valerio, F. Ricciato, and H. Hlavacs, 'Cellular Data Meet Vehicular Traffic Theory : Location Area Updates and Cell Transitions for Travel Time Estimation', in *ACM Conference on Ubiquitous Computing*, 2012, pp. 361–370.

[77] L. Zheng and Y. Chang, 'Risk assessment model of bottlenecks for urban expressways using survival analysis approach', *Transp. Res. Procedia*, vol. 25, pp. 1544–1555, 2017.

[78] Transportation Research Board's, *Highway Capacity Manual 2010*. 2010.

[79] M. Elhenawy, H. Chen, H. A. Rakha, and C. E. Via, 'Traffic Congestion Identification Considering Weather and Visibility Conditions Using Mixture Linear Regression', vol. 4983, no. 540, pp. 1–20, 2015.

[80] J. J. Wu, H. J. Sun, Z. Y. Gao, and H. Z. Zhang, 'Reversible lane-based traffic network optimization with an advanced traveller information system', *Eng. Optim.*, vol. 41, no. 1, pp. 87–97, 2009.

Appendix A : List of Supported Languages of Google Distance Matrix API

Language Code	Language	Language Code	Language
ar	Arabic	lt	Lithuanian
be	Belarusian	lv	Latvian
bg	Bulgarian	mk	Macedonian
bn	Bengali	ml	Malayalam
ca	Catalan	mr	Marathi
cs	Czech	my	Burmese
da	Danish	nl	Dutch
de	German	no	Norwegian
el	Greek	pa	Punjabi
en	English	pl	Polish
en-Au	English (Australian)	pt	Portuguese
en-GB	English (Great Britain)	pt-BR	Portuguese (Brazil)
es	Spanish	pt-PT	Portuguese (Portugal)
eu	Basque	ro	Romanian
fa	Farsi	ru	Russian
fi	Finnish	sk	Slovak
fil	Filipino	sl	Slovenian
fr	French	sq	Albanian
gl	Galician	sr	Serbian
gu	Gujarati	sv	Swedish
hi	Hindi	ta	Tamil
hr	Croatian	te	Telugu
hu	Hungarian	th	Thai

id	Indonesian	tl	Tagalog
it	Italian	tr	Turkish
iw	Hebrew	uk	Ukrainian
ja	Japanese	uz	Uzbek
kk	Kazakh	vi	Vietnamese
kn	Kannada	zh-CN	Chinese (Simlified)
ko	Korean	zh-TW	Chinese (Traditional)
ky	Kyrgyz		

Appendix B : Arduino Code for retrieval of GPS Data

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "inetGSM.h"
//#include "sms.h"
//#include "call.h"
#include "gps.h"
//To change pins for Software Serial, use the two lines in GSM.cpp.
//GSM Shield for Arduino
//www.open-electronics.org
//this code is based on the example of Arduino Labs.
//Simple sketch to start a connection as client.
InetGSM inet;
//CallGSM call;
//SMSGSM sms;
GPSGSM gps;
char lon[15];
char lat[15];
char alt[15];
char timev[20];
char vel[15];
char charGpsData[100];
const char dataURL[] = "/yungIOT/gps.php?"; //17
const char lonName[] = "longitude="; //10
const char latName[] = "&latitude="; //10
const char timeName[] = "&localTimes="; //12
int a;
char temC;
int index;
int stat;
char msg[50];
int numdata;
int i=0;
boolean started=false;
void setup()
{
    //Serial connection.
    Serial.begin(9600);
//    Serial.println("GSM Shield testing.");
    //Start configuration of shield with baudrate.
    //For http uses is raccomanded to use 4800 or slower.
    if (gsm.begin(2400)) {
//        Serial.println("\nstatus=READY");
        gsm.forceON();  //To ensure that SIM908 is not only in charge mode
        started=true;
    } //else Serial.println("\nstatus=IDLE");
```

```
if(started) {
//    //GPRS attach, put in order APN, username and password.
//         //If no needed auth let them blank.
//         if (inet.attachGPRS("internet.wind", "", ""))
//             Serial.println("status=ATTACHED");
//         else Serial.println("status=ERROR");
//         delay(1000);
   //GPS attach
        if (gps.attachGPS())
//             Serial.println("status=GPSREADY");
//         else Serial.println("status=ERROR");
        delay(30000);  //Time for fixing
        //If you use the new SIM808 please use the following code
        stat = gps.getStat();
//         Serial.println(stat);
//         if(stat == 0)
//            Serial.println("FIXED FAIL");
//         else if(stat == 1)
//             Serial.println("FIXED OK");}};
void loop()
{
//    //Read for new byte on serial hardware,
//    //and write them on NewSoftSerial.
//    serialhwread();
//    //Read for new byte on NewSoftSerial.
//    serialswread();
 gps.getPar(lon,lat,alt,timev,vel);
//  char gpsData[100];
//  strcat ( gpsData, lat );
//  strcat ( gpsData, "_" );
//  strcat ( gpsData, lon );
//  strcat ( gpsData, "_" );
//  strcat ( gpsData, timev );
//  Serial.print("**1**");
//  Serial.print(gpsData);
//  Serial.println("**2**");
//  String tmp = "cat";
//  char charGpsData[55];
//  strcpy(charGpsData, tmp.c_str());
//  Serial.print("**1**");
//  Serial.print(charGpsData);
//  Serial.println("**2**");
//  String strLat(lat);
//  String strLon(lon);
//  String strTime(timev);
//  String dataURL = "/yungIOT/gps.php?";
//  String strGpsData = dataURL + "longitude=" + strLon + "&latitude=" + strLat +
"&localTimes=" + strTime;
//  Serial.println(lon);
//  Serial.println(lat);
```

```
//  Serial.println(timev);

//  strncpy (charGpsData, dataURL, 17);
//  strncpy (charGpsData, lonName, 10);
//  strncpy (charGpsData, lon, 15);
//  strncpy (charGpsData, latName, 10);
//  strncpy (charGpsData, lat, 15);
//  strncpy (charGpsData, timeName, 12);
//  strncpy (charGpsData, timev, 20);
//  size_t destination_size = sizeof (charGpsData);
//  charGpsData[destination_size - 1] = '\0';
  index=0;
  for(a=0; a<17; a++){
    temC = dataURL[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    }  }
  for(a=0; a<10; a++){
    temC = lonName[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    } }
  for(a=0; a<15; a++){
    temC = lon[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    }  }
  for(a=0; a<10; a++){
    temC = latName[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    } }
  for(a=0; a<15; a++){
    temC = lat[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    }}
  for(a=0; a<12; a++){
    temC = timeName[a];
    if(temC != '\0'){
```

```
      charGpsData[index++] = temC;
    }else{
      break;
    }  }

  for(a=0; a<20; a++){
    temC = timev[a];
    if(temC != '\0'){
      charGpsData[index++] = temC;
    }else{
      break;
    }  }
  charGpsData[index] = '\0';
//  Serial.println(alt);
//  Serial.println(vel);
  sendGPSData(charGpsData);
    delay(2000);
};
void sendGPSData(const char* message){
  if(started) {
        //GPRS attach, put in order APN, username and password.
        //If no needed auth let them blank.
        if (inet.attachGPRS("internet.wind", "", ""))
//              Serial.println("status=ATTACHED");
//          else Serial.println("status=ERROR");
        delay(1000);
//          //Read IP address.
//          gsm.SimpleWriteln("AT+CIFSR");
//          delay(5000);
//          //Read until serial buffer is empty.
//          gsm.WhileSimpleRead();
        //TCP Client GET, send a GET request to the server and
        //save the reply.
//          char charGpsData[55];
//  strncpy(charGpsData, message.c_str(), 100);
//  Serial.print("**1**");
//  Serial.print(charGpsData);
//  Serial.println("**2**");
//  "/yungIOT/gps.php?longitude=3&latitude=5&localTimes=8"
//          Serial.print("**1**");
//          Serial.print(message);
//          Serial.println("**2**");
        numdata=inet.httpGET("www.titansmora.org", 80, message, msg, 50);
        //Print the results.
//          Serial.println("\nNumber of data received:");
//          Serial.println(numdata);
//          Serial.println("\nData received:");
//          Serial.println(msg);
        inet.dettachGPRS();
    }}
```

Appendix C : Spatio temporal variation of space mean speed with the implementation of bus priority lanes

Appendix D : Data set for verification of API travel time with GPS travel time – Short Distance

Peak Time Traffic				
Segment	**Distance (km)**	**GoogleAPI TT(min)**	**GPS_TT (min)**	**DateTime**
Borelasgamuwa - Maharagama	3.1	8	5	1/19/2017 7:00
Borella - Pettah	3.8	29	27	1/18/2017 17:30
Galleface - Bambalapitiya	3.8	39	31	1/19/2017 17:30
Kirulapana - Borella	4.1	32	38	1/25/2017 8:01
Dehiwala - Borelasgamuwa	4.4	12	17	2/18/2017 6:30
Kirulapana - Rajagiriya	5.1	36	39	1/27/2017 18:30
Galleface - Borella	5.2	38	30	1/19/2017 18:00
Gothatuwa - Borella	5.3	27	28	1/18/2017 8:01
Borelasgamuwa - Dehiwala	5.8	26	20	1/18/2017 7:31
Borella - Galleface	5.9	30	38	1/24/2017 18:01
Bambalapitiya - Dehiwala	6.1	27	21	1/20/2017 18:01
Borelasgamuwa - Dehiwala	6.3	28	20	1/19/2017 7:31
Kandana - Kadawatha	9.6	26	31	1/24/2017 17:00
Kandana - Kadawatha	9.6	35	43	1/26/2017 17:30
Kandana - Kadawatha	9.6	37	42	1/26/2017 18:01
Kollupitiya-Hunupitiya	11.8	44	48	3/13/2017 17:30
Jaela - Minuwangoda	13.7	26	28	1/18/2017 8:01
Jaela - Minuwangoda	17.5	33	30	1/20/2017 18:30
Kaduwela-Kollupitiya	17.9	50	54	9/18/2017 16:30
Peliyagoda 1-Mt. Lavinia	19.0	78	81	3/20/2018 17:30

Kothalawala-Boralesgamuwa	20.0	40	43	11/16/2017 18:00
Torana Junction-Maharagama	21.0	52	52	1/22/2018 8:30
Torana Junction-Maharagama	21.5	66	72	12/08/2017 18:00
Piliyandala-Malabe	22.2	41	40	9/13/2017 7:30
Kottawa-Mt. Lavinia	23.0	62	65	2/08/2017 18:00
Kottawa-Mt. Lavinia	23.0	59	65	5/14/2017 7:30
Mabola-Boralesgamuwa	23.4	83	79	2/21/2017 18:30
Pore-Boralesgamuwa	24.4	47	52	3/19/2018 18:30
Pannipitiya-Torana Junction	25.9	58	59	3/08/2018 18:30
Malabe-Gampaha	32.5	47	47	3/17/2017 17:00
Kadawatha-Mt. Lavinia	37.3	65	66	1/18/2017 16:30
Kadawatha-Mt. Lavinia	37.3	74	76	2/08/2017 18:30
Gampaha-Battaramulla	38.0	71	68	1/10/2018 8:30
Gampaha-Battaramulla	38.0	56	60	5/17/2018 6:30
Mattakkuliya-Kottawa	39.3	57	50	1/15/2018 8:30
Maharagama-Katunayake	40.1	68	72	3/19/2018 6:30
Mabola-Boralesgamuwa	40.9	82	74	11/20/2017 18:00
Hunupitiya-Maharagama	46.2	85	86	5/11/2017 18:00
Hunupitiya-Maharagama	55.3	80	72	2/16/2017 7:30
Katubedda-Udugampola	56.3	88	88	1/21/2017 17:00
Katubedda-Udugampola	56.3	81	83	2/12/2018 7:00
Katunayake-Piliyandala	59.0	80	84	1/22/2017 8:30
Katunayake-Piliyandala	59.0	79	77	12/19/2017 16:30
Hunupitiya-Piliyandala	62.0	85	86	12/17/2017 17:30

Off- Peak Time Traffic				
Segment	**Distance**	**GoogleAPI TT**	**GPS_TT**	**DateTime**
Biyagama - Kaduwela	0.8	2	8	1/17/2017 14:01
Biyagama - Kaduwela	0.8	3	8	2/6/2017 19:00
Pettah - Galleface	2.2	5	6	1/18/2017 22:30
Borelasgamuwa - Maharagama	3.1	5	4	1/18/2017 23:00
Bambalapitiya - Borella	3.7	19	21	1/17/2017 13:31
Dehiwala - Kohuwala	3.7	9	5	1/17/2017 23:00
Galleface - Bambalapitiya	3.8	18	21	1/17/2017 13:31
Borella - Pettah	3.8	15	17	1/17/2017 13:31
Borella - Bambalapitiya	3.8	10	10	2/18/2017 23:30
Borella - Kirulapana	4.3	28	30	1/20/2017 15:30
Kaduwela - Ambathale	4.4	14	11	1/17/2017 16:01
Ambatale - Kaduwela	4.4	9	17	1/17/2017 21:30
Maharagama - Borelasgamuwa	4.9	14	17	1/17/2017 16:01
Bambalapitiya - Galleface	5.2	16	10	1/17/2017 13:31
Borella - Gothatuwa	5.4	19	26	1/17/2017 19:00
Borelasgamuwa - Kohuwala	5.5	11	12	2/18/2017 21:30
Pettah - Borella	6.0	25	32	1/24/2017 14:01
Bambalapitiya - Dehiwala	6.0	31	39	1/27/2017 19:01
Katunayake - Minuwangoda	9.1	14	21	1/17/2017 15:30
Kandana - Kadawatha	9.6	25	18	2/3/2017 12:30
Kadawatha - Kandana	9.6	22	21	2/17/2017 21:00
Jaela - Katunayaka	12.2	12	13	2/18/2017 1:00
Jaela - Minuwangoda	13.7	23	17	2/18/2017 23:00

Jaela - Minuwangoda	13.7	21	22	2/18/2017 5:30
Pannipitiya-Torana Junction	17.4	42	41	4/21/2017 15:00
Peliyagoda 1-Mt. Lavinia	20.2	60	62	1/18/2018 19:00
Orugodawatta-Piliyandala	20.7	72	67	11/10/2017 19:00
Kottawa-Mt. Lavinia	23.0	43	41	12/15/2017 5:30
Mabola-Boralesgamuwa	23.4	62	67	3/13/2017 20:30
Pore-Boralesgamuwa	24.4	40	35	12/11/2017 15:00
Kottawa-Mt. Lavinia	26.1	52	54	2/20/2017 21:00
Mattakkuliya-Makumbura	27.4	51	51	5/22/2017 20:00
Pannipitiya-Panadura	31.2	45	51	1/19/2018 15:00
Malabe-Gampaha	32.5	44	43	4/12/2017 15:00
Gampaha-Battaramulla	32.7	66	68	12/08/2017 19:30
Torana Junction-Kottawa	34.5	44	39	2/13/2018 15:00
Gampaha-Battaramulla	37.4	53	52	11/15/2017 22:00
Kiribathgoda-Piliyandala	38.8	55	63	9/11/2017 15:00
Kiribathgoda-Piliyandala	39.2	53	49	2/17/2017 21:00
Maharagama-Katunayake	40.1	73	78	1/20/2018 19:30
Mabola-Boralesgamuwa	41.2	69	71	11/21/2017 15:00
Hunupitiya-Maharagama	43.0	56	59	5/18/2017 22:30
Hunupitiya-Maharagama	43.0	65	63	3/20/2018 6:00
Katunayake-Piliyandala	44.5	65	73	5/09/2017 21:30
Battaramulla-Minuwangoda	46.1	78	74	4/18/2018 19:30
Battaramulla-Minuwangoda	46.5	70	62	11/09/2017 20:00
Battaramulla-Minuwangoda	46.5	66	60	12/12/2017 15:00

Maharagama-Katunayake	53.9	70	64	11/17/2017 16:00
Ambagahawatta-Delkanda	54.3	84	79	5/09/2018 15:00
Katunayake-Piliyandala	59.0	77	81	11/08/2017 16:00
Udugampola-Panadura	64.4	78	82	1/20/2017 21:30
Panadura-Katunayake	73.0	83	81	12/21/2017 21:30

Appendix E : Data set for verification of API travel time with GPS travel time – Long Distance (50km-260km)

Segment	Distance (km)	GDM APT TT (min)	GPS TT (min)	Date and Time
Dambulla-Kurunegala	56.40	77	50	12/17/2017 15:00
Kurunegala-Dambulla	56.40	76	80	12/16/2017 11:00
Kurunegala-Gampaha	71.20	113	130	9/17/2017 5:30
Gampaha-Kurunegala	71.40	115	120	9/16/2017 8:00
Piliyandala-Rathnapura	72.90	122	120	4/21/2017 7:00
Rathnapura-Piliyandala	75.50	125	115	4/21/2017 19:30
Rathnapura-Bandarawela	101.60	170	110	12/22/2017 9:30
Bandarawela-Rathnapura	101.80	173	210	12/25/2017 10:00
Kurunegala-Anuradhapura	105.80	130	120	12/16/2017 11:00
Anuradhapura-Kurunegala	105.80	132	165	12/17/2017 13:00
Battaramulla-Hikkaduwa	108.20	108	90	3/17/2018 13:00
Hikkaduwa-Battaramulla	108.30	110	100	3/18/2018 13:00
Battaramulla-Kandy	121.60	215	185	8/23/2017 7:00
Kandy-Battaramulla	123.20	210	270	8/23/2017 19:30
Katharagama-Matara	129.30	179	130	4/16/2018 13:00
Matara-Katharagama	130.50	182	200	2/18/2018 3:30
Battaramulla-Balangoda	130.80	208	195	11/17/2017 5:00
Pothuvil-Katharagama	131.10	156	125	8/19/2017 7:00
Balangoda-Battaramulla	131.60	221	295	11/19/2017 10:00
Katharagama-Pothuvil	135.60	164	105	8/20/2017 5:30
Battaramulla-Malate	150.20	241	310	1/12/2018 5:00

Battaramulla-Nuwara Eliya	151.20	302	390	8/11/2017 8:00
Malate-Battaramulla	159.70	240	300	1/14/2018 13:00
Katharagama-Galle	168.50	214	225	4/18/2018 13:00
Galle-Katharagama	170.40	220	200	2/18/2018 2:00
Pothuvil-Balangoda	173.60	259	250	11/19/2017 5:30
Balangoda-Pothuvil	173.60	256	325	11/17/2017 9:30
Nuwara Eliya-Battaramulla	175.50	290	275	8/13/2017 13:00
Ampara-Hambantota	200.80	241	380	7/22/2017 14:30
Hambantota-Ampara	205.10	234	315	7/23/2017 6:00
Rathnapura-Ampara	268.70	341	455	7/22/2017 5:30
Ampara-Rathnapura	269.00	334	440	7/23/2017 17:00

Appendix F : Statistical evaluation of Google traveltime data and GPS traveltime data

```
REGRESSION
  /DESCRIPTIVES MEAN STDDEV CORR SIG N
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS CI(95) BCOV R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /ORIGIN
  /DEPENDENT Google_TT_Offpeak
  /METHOD=ENTER GPS_TT_Offpeak
  /RESIDUALS HISTOGRAM(ZRESID).
```

Regression

Notes

Output Created		12-JUL-2018 21:42:18
Comments		
Input	Active Dataset	DataSet0
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	52
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /DESCRIPTIVES MEAN STDDEV CORR SIG N /MISSING LISTWISE /STATISTICS COEFF OUTS CI(95) BCOV R ANOVA /CRITERIA=PIN(.05) POUT(.10) /ORIGIN /DEPENDENT Google_TT_Offpeak /METHOD=ENTER GPS_TT_Offpeak /RESIDUALS HISTOGRAM(ZRESID).
Resources	Processor Time	00:00:00.26
	Elapsed Time	00:00:00.00
	Memory Required	2560 bytes
	Additional Memory Required for Residual Plots	312 bytes

Descriptive Statistics[a]			
	Mean[b]	Root Mean Square	N
Google_TT_Offpeak	40.1154	47.44835	52
GPS_TT_Offpeak	40.6923	47.69494	52

a. Coefficients have been calculated through the origin.

b. The observed mean is printed

Model Summary[c,d]				
Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.995[a]	.991	.991	4.56091

a. Predictors: GPS_TT_Offpeak

b. For regression through the origin (the no-intercept model), R Square measures the proportion of the variability in the dependent variable about the origin explained by regression. This CANNOT be compared to R Square for models which include an intercept.

c. Dependent Variable: Google_TT_Offpeak

d. Linear Regression through the Origin

ANOVA[a,b]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	116009.103	1	116009.103	5576.848	.000[c]
	Residual	1060.897	51	20.802		
	Total	117070.00[d]	52			

a. Dependent Variable: Google_TT_Offpeak

b. Linear Regression through the Origin

c. Predictors: GPS_TT_Offpeak

d. This total sum of squares is not corrected for the constant because the constant is zero for regression through the origin.

Coefficients[a,b]								
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Offpeak	.990	.013	.995	74.678	.000	.964	1.017

a. Dependent Variable: Google_TT_Offpeak

b. Linear Regression through the Origin

Residuals Statistics[a,b]					
	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	3.9612	81.2056	40.2981	24.87793	52
Residual	-7.83530	8.60066	-.18269	4.55718	52
Std. Predicted Value	-1.461	1.644	.000	1.000	52
Std. Residual	-1.718	1.886	-.040	.999	52

a. Dependent Variable: Google_TT_Offpeak

b. Linear Regression through the Origin

Notes		
Output Created		12-JUL-2018 21:44:54
Comments		
Input	Active Dataset	DataSet0
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	52
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /DESCRIPTIVES MEAN STDDEV CORR SIG N /MISSING LISTWISE /STATISTICS COEFF OUTS CI(95) BCOV R ANOVA /CRITERIA=PIN(.05) POUT(.10) /ORIGIN /DEPENDENT Google_TT_Long /METHOD=ENTER GPS_TT_Long /RESIDUALS HISTOGRAM(ZRESID).
Resources	Processor Time	00:00:00.25
	Elapsed Time	00:00:00.00
	Memory Required	2560 bytes
	Additional Memory Required for Residual Plots	312 bytes

```
REGRESSION
```

```
/DESCRIPTIVES MEAN STDDEV CORR SIG N
/MISSING LISTWISE
/STATISTICS COEFF OUTS CI(95) BCOV R ANOVA
/CRITERIA=PIN(.05) POUT(.10)
/ORIGIN
/DEPENDENT Google_TT_Peak
/METHOD=ENTER GPS_TT_Peak
/RESIDUALS HISTOGRAM(ZRESID).
```

Regression

Notes

Output Created		12-JUL-2018 21:45:48
Comments		
Input	Active Dataset	DataSet0
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	52
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /DESCRIPTIVES MEAN STDDEV CORR SIG N /MISSING LISTWISE /STATISTICS COEFF OUTS CI(95) BCOV R ANOVA /CRITERIA=PIN(.05) POUT(.10) /ORIGIN /DEPENDENT Google_TT_Peak /METHOD=ENTER GPS_TT_Peak /RESIDUALS HISTOGRAM(ZRESID).
Resources	Processor Time	00:00:00.19
	Elapsed Time	00:00:01.00
	Memory Required	2560 bytes
	Additional Memory Required for Residual Plots	312 bytes

Descriptive Statistics[a]

	Mean[b]	Root Mean Square	N
Google_TT_Peak	51.5227	56.06713	44
GPS_TT_Peak	51.9773	56.59043	44

a. Coefficients have been calculated through the origin.

b. The observed mean is printed

Model Summary[c,d]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.997[a]	.993	.993	4.67444

a. Predictors: GPS_TT_Peak

b. For regression through the origin (the no-intercept model), R Square measures the proportion of the variability in the dependent variable about the origin explained by regression. This CANNOT be compared to R Square for models which include an intercept.

c. Dependent Variable: Google_TT_Peak

d. Linear Regression through the Origin

ANOVA[a,b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	137375.435	1	137375.435	6287.104	.000[c]
	Residual	939.565	43	21.850		
	Total	138315.00[d]	44			

a. Dependent Variable: Google_TT_Peak

b. Linear Regression through the Origin

c. Predictors: GPS_TT_Peak

d. This total sum of squares is not corrected for the constant because the constant is zero for regression through the origin.

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Peak	.987	.012	.997	79.291	.000	.962	1.012

a. Dependent Variable: Google_TT_Peak

b. Linear Regression through the Origin

Residuals Statistics[a,b]					
	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	4.9369	86.8896	51.3214	22.35254	44
Residual	-7.52051	8.93374	.20131	4.67000	44
Std. Predicted Value	-2.075	1.591	.000	1.000	44
Std. Residual	-1.609	1.911	.043	.999	44

a. Dependent Variable: Google_TT_Peak

b. Linear Regression through the Origin

```
REGRESSION
  /DESCRIPTIVES MEAN STDDEV CORR SIG N
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS CI(90) BCOV R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /ORIGIN
  /DEPENDENT Google_TT_Long
  /METHOD=ENTER GPS_TT_Long
  /RESIDUALS HISTOGRAM(ZRESID).
```

Regression

Notes		
Output Created		12-JUL-2018 21:48:29
Comments		
Input	Active Dataset	DataSet0
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	52
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.

Notes		
Syntax		REGRESSION /DESCRIPTIVES MEAN STDDEV CORR SIG N /MISSING LISTWISE /STATISTICS COEFF OUTS CI(90) BCOV R ANOVA /CRITERIA=PIN(.05) POUT(.10) /ORIGIN /DEPENDENT Google_TT_Long /METHOD=ENTER GPS_TT_Long /RESIDUALS HISTOGRAM(ZRESID).
Resources	Processor Time	00:00:00.31
	Elapsed Time	00:00:00.00
	Memory Required	2560 bytes
	Additional Memory Required for Residual Plots	312 bytes

Descriptive Statistics[a]

	Mean[b]	Root Mean Square	N
Google_TT_Long	192.4375	204.81882	32
GPS_TT_Long	211.8750	238.24226	32

a. Coefficients have been calculated through the origin.

b. The observed mean is printed

Model Summary[c,d]

Model	R	R Square[b]	Adjusted R Square	Std. Error of the Estimate
1	.980[a]	.961	.960	40.96725

a. Predictors: GPS_TT_Long

b. For regression through the origin (the no-intercept model), R Square measures the proportion of the variability in the dependent variable about the origin explained by regression. This CANNOT be compared to R Square for models which include an intercept.

c. Dependent Variable: Google_TT_Long

d. Linear Regression through the Origin

ANOVA[a,b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1290396.23	1	1290396.23	768.864	.000[c]
	Residual	52027.774	31	1678.315		
	Total	1342424.0[d]	32			

a. Dependent Variable: Google_TT_Long
b. Linear Regression through the Origin
c. Predictors: GPS_TT_Long
d. This total sum of squares is not corrected for the constant because the constant is zero for regression through the origin.

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	90.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Long	.843	.030	.980	27.728	.000	.791	.894

a. Dependent Variable: Google_TT_Long
b. Linear Regression through the Origin

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Long	.843	.030	.980	27.728	.000	.781	.905

a. Dependent Variable: Google_TT_Long
b. Linear Regression through the Origin

Coefficients[a,b]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	99.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	GPS_TT_Long	.843	.030	.980	27.728	.000	.759	.926

a. Dependent Variable: Google_TT_Long
b. Linear Regression through the Origin

Residuals Statistics[a,b]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	42.1442	383.5122	178.5860	93.29476	32
Residual	-79.29588	77.28278	13.85148	38.47418	32
Std. Predicted Value	-1.462	2.197	.000	1.000	32
Std. Residual	-1.936	1.886	.338	.939	32

a. Dependent Variable: Google_TT_Long
b. Linear Regression through the Origin