

Digitization of the entire traffic system and mitigation of the ongoing traffic crisis across cities of developing nations

A Thesis

Submitted to the Department of Electrical and Electronics Engineering

Of

BRAC University

By

Mohammad Samin Yasar (12121067)

Md. Tahmid Rashid (12121090)

Supervised by

Dr. Md. Khalilur Rhaman

Associate Professor

Department of Computer Science and Engineering

BRAC University, Dhaka

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

Submitted on

26th April, 2015



DECLARATION

This is to affirm that that this thesis report is submitted by the authors listed for the degree of Bachelor of Science in Electrical and Electronic Engineering to the Department of Electrical and Electronic Engineering under the School of Engineering and Computer Science, BRAC University. We hereby declare that the research work is based on the results found by us and no other. Materials of work found by other researchers have been properly acknowledged or referred to. This thesis, neither in whole nor in part, has been previously submitted elsewhere for assessment.

Signature of Supervisor

Signature of Authors

Dr. Md. Khalilur Rhaman

Mohammad Samin Yasar

Md. Tahmid Rashid

ACKNOWLEDGEMENTS

We would begin by conveying our appreciation to our supervisor Dr. Md. Khalilur Rhaman, Associate Professor, Department of Computer Science and Engineering, BRAC University for encouraging and supporting us throughout the course of our project. We thank all our peers and every other individual involved with us for being supportive to us and lending a hand whenever called for it. Lastly, we are grateful to our university for providing us the necessary resources and funds.

ABSTRACT

This paper focuses on a novel approach for handling the present traffic situation in perspective of Bangladesh. We plan to moderate the ongoing traffic predicament that currently plagues Dhaka city and gradually expand it to the whole country. Road traffic congestion is apparently a borderless ordeal in Dhaka and its adjacent cities and the situation tends to worsen as new cars enter the current stream every day. The aim of the paper is to develop a threefold solution to counter the traffic clogging. The approach taken during the course of this research focused primarily on an experimental evaluation of the small-scale model of the traffic routing algorithms. Among the threefold solution, the first approach is to develop a traffic algorithm to calculate the routes with shortest possible times to destinations. We plan to implement the system's usability by providing feedback to our target users (car drivers) so that they can decide on which route to take. This will be done by means of an overhead display on the car dashboards backed up by an embedded OS or Android. For our input we plan to take the amount of cars that are at any specific route at a time and provide that data to the car driver by the means of modern vehicle density measurement techniques. Travelling times are calculated using Dijkstra's algorithm and the shortest possible time required is provided to the commuters taking into consideration the situation of the roads at any point of time. The second approach is to make use of 24-hour Dynamic Traffic Light Controllers (DTLCs) based on artificial neural networks. The DTLC will be implemented using the Intel NUC in conjunction with the Arduino Mega. The decision making algorithm is designed to replicate, in a meager form, the human brain with the system trained to learn to respond to certain traffic situations. At present the BRTA (Bangladesh Road Transport) employs Static Traffic Light Controllers (STLCs) to handle traffic flow at some intersections while other, less important, ones have manual control in the form of the traffic officer in charge.

Table of Contents

Declaration	2
Acknowledgements	3
Abstract	4
Chapter 1: Introduction	7
1.1 Motivation.....	7
1.2 Related Work	9
1.3 Thesis Outline	11
Chapter 2: Background Study	12
2.1 The Traffic Bandwagon	12
2.2 Viable Options To Slow Down The Convoy.....	15
Chapter 3: System Architecture	17
3.1 System Overview	17
3.2 Simulation and Demonstration	17
Chapter 4: Design Implementation	22
4.1 Methodology	22
4.2 Vehicle Flow Sensor Inputs	23
4.2.1 Traffic Flow Monitoring and Detection	23
4.2.2 Instantaneous Vehicle Count Method	23

4.2.3 Choice of Detector Technology.....	24
4.2.4 Radio-frequency identification (RFID)	25
4.2.5 Floating Car Detector (FCD).....	26
4.3 Algorithms	28
4.3.1 Artificial Neural Networks Algorithm	28
4.3.2 Dijkstra’s and A*Algorithm.....	33
Chapter 5: Experimental Results	35
5.1 Route Calculations	36
5.2 Simulation Results in Vissim.....	39
5.2 Traffic Light Manipulations.....	42
Chapter 6: Discussion	43
6.1 Limitations and Difficulties	43
6.2 Expectations, outcomes and future prospects.....	44
References	45

CHAPTER 1

INTRODUCTION

Traffic control is a rapidly evolving subject, reflective of new developments in Electrical Sensor Technology, and Information and Communication Technology. In Dhaka, a city of about 22 million people [1], traffic congestion has worsened dramatically over the last 4-5 years, in spite of the introduction of automated traffic lights. Expensive and drastic measures, such as the construction of flyovers, the construction of Hatirjheel, have been implemented to counter traffic congestion [1]. However, these measures are yet to produce their desired results. At the very core of the problem is the underlying fact that only 7-8% of the land in Dhaka is used for the construction of roads and highway. A typical megalopolis dedicates around 20% of its land for these roads. Another burning issue is the increase in the number of new cars that are hitting the road. Despite the government imposing high taxes on the purchase of new cars, the condition is as yet to improve. This paper proposes an intelligent traffic system based on two novel takes, in the perspective of Bangladesh, of countering the traffic situation. The first is a dynamic traffic light system and second is the intelligent routing of cars depending on the traffic situation around the city.

1.1 Motivation

Traffic in Bangladesh is seen as one of the major vices that is stifling the country's economic progress, with an average of 2 hours being spent waiting on the roads. This has caused innumerable sufferings and a decline in public welfare and has resulted in massive loss of productivity with the resulting effect being felt on the GDP. At present, BRTA employs static traffic light control (STLC), which has fixed cycle of green and red lights depending on

the historical flow of cars along certain routes. The STLCs, which are based on fixed counters and do not have any sort of provision to consider the present traffic flow, have failed to significantly improve the situation with people still suffering from traffic jams as they used to. In severe cases it is seen that these STLCs are deemed inadequate to handle the traffic with the officer in site manually overriding them [2].

The proposed DTLC will have real time updates on the present traffic situation by means of sensors placed on each road. These sensor inputs would be fed into the neural networks in place, which will evaluate them to provide an optimal drive order. The artificial neural networks are used as they avoid the risk of being locked into local minima and will provide a reasonable robust solution to any condition of the intersection. Because of the computational load that these neural networks impose, we have used the NUC, powered Intel atom to handle the processing and memory loads of the algorithm.

The DTLC is part of the broader vision of an Intelligent Traffic System (ITS), which is in line with the vision of a digital Bangladesh. The aim of is to record and maintain all sorts of transport activities under the digital canopy. It is an integrated system which can handle and predict all sorts of traffic challenges that may present itself on a regular day. DTLC is viewed as a stepping stone to that vision. If successful, it will enable better traffic control in Bangladesh which will use the existing infrastructure to reduce congestion and improve productivity. Since waiting on the road also leads to increased greenhouse emission and pollution, ITS may also help to reduce the carbon footprint and make the environment cleaner. The power system driving the ITS is based on fully renewable energy meaning that there would be no load on the main grid.

With DTLC, there will be an improvement on the traffic regulation by eliminating the redundancy which is normally associated with the STLC. If there is a circumstance where one part of the road is congested while the other part have almost no traffic, DTLC will help to bring stability and improve the situation and allow the congested part to clear out by prolonging the duration of the green light on that route. When the traffic load is low on all sides of the intersection, the duration of green light will be small and there will be a more rapid transition.

Alongside DTLC, which, at its very best, is still a passive and a reactive way of reducing traffic jam, the paper also proposes a more active way in the form assigning routes to cars based on the estimated travelling time. These times are computed using sensors that are placed on the road. They measure the traffic flow and send the readings to a centralized server which then uses these values to calculate the estimated arrival times for a vehicle travelling at a certain speed from a certain point A to point B.

1.2 Related Work

As traffic congestion is a well-documented problem with none of the biggest cities in world immune to it, it is fair to assume that this is a well-researched topic. There have been several attempts at applying the Artificial Intelligence to tackle traffic clogging at intersections or for the intelligent routing of vehicles from one part of the city to another. Anfilets et al.[2] implemented machine learning in the form of artificial neural networks to solve the traffic light cycle paradigm. Their results showed a significant improvement in the traffic queuing at intersections when their algorithm was applied with the queue size decreasing to a satisfactory level. This paper uses a similar approach to theirs to tackle the queuing up of vehicles at these intersections. In our case, we used floating car detection method for taking

the input of the traffic compared to the image processing that was used in their case. Jansson [19] used genetic algorithm to solve the traffic intersection problem by assigning optimal drive orders depending on the present traffic parameters. His work took into consideration the pedestrians' requests as well as trams' request. Our work also involves assigning specific drive orders for clearing of queues but is centered on vehicular requests only. For the rerouting of the vehicles depending on the traffic situation, we used Dijkstra's algorithm.

Oliveira et al. [4] used the Environment Observation Method based on Artificial Neural Networks Controller, EOM-ANN controller to optimize traffic lights cycle depending on dynamic inputs. Their simulation was conducted in SUMO (Simulation of Urban Mobility) whereas our work was simulated in PTV VISSIM. Their results also showed statistical improvements in the traffic conditions when their algorithm was applied. Nanayakkara et al. [5] developed their own GA solution to the route planning for the map of Singapore with their proposed GA using only distance information. Their algorithm is tested to find the shortest distance between any two nodes with a search space of over 1000 nodes, and is compared to ant-based route planning algorithms. Our proposed algorithm does not necessarily prioritize the shortest routes between any two nodes, but instead, the routes which can be traversed in the shortest possible times based on the traffic flow in those routes using Dijkstra's.

To supplement our algorithm, we used real time feed of the traffic which are provided by the Floating Car Detection (FCD) technology. This is a non-intrusive technique of gathering
On the sensor front, we are using Floating Car Detection technology which involves triangulation of the vehicular position using signals pinged from the user's phone. On this issue our work has some similarity with Eichler [6] and Fabritiis et al.[7]. The latter involved

testing on the Rome's Ring Road and used artificial neural networks and pattern matching to predict short-term arrival times of vehicles. Their work used the existing embedded Telematics platform (developed by OCTOTelematics) present in most cars in Europe to gather information about the present traffic. On Board Units (OBO) were installed in cars which relayed information to a central data system, thus providing the system with the requisite information. Due to the lack of such infrastructure in Bangladesh, we had to develop our own FCD system using a simple GPS receiver and a GSM transmitter. Our proposed version of FCD uses the existing telecommunication network to relay information to our central data base. The system analyses the density of mobile traffic to estimate traffic conditions at any point of time.

1.3 Thesis Outline:

The paper is structured as follows:

Chapter 2 talks about the background study conducted which laid the foundation to our work.

Chapter-3 talks about the detailed system architecture of our work. For testing the feasibility of our proposed solutions our work involves the use of sensors which are coordinated and processed over a centralized web server. The chapter contains diagrams of the different parts of our architecture along with one which shows all of the parts working in cohesion.

Chapter-4 talks about the methodology of choice for our proposed version of an intelligent traffic system, in perspective of Dhaka.

2.1 The Traffic Bandwagon

Dhaka, a megacity filled with 131 million people and constricted with only 7.5% of road from its total area (for a standard city, the minimum road requirement is 25%) of which 30% is engaged by hawkers, salesman and shopkeepers is presumably destined to be a slow city [8, 9, 10]. It is needless to mention that the development of the city has staggered against its little available haphazard road network and snail like pace. The growth resembles a canopy without an endpoint. The presumably opportunist behavior of majority of its citizens to take advantage of the absence of law enforcement has only added momentum to the miserable situation. The following table shows us the comparative traffic speed of some selected major cities; it shows that Dhaka is in fact the slowest city of the world [10]. As such if no all-rounded measures are immediately taken then the city may come to a halt in very near future.

Table 1: Comparative Traffic Speed of Some Major Cities [6, 11, 12]

Sl. No.	City	Population	Average Speed	Comments
1	Dhaka	13,100,000	4-5 kph	Data is based on inference and observation
2	Bangalore	7,600,000	12 kph	
3	Bangkok	8,750,000	15 kph	Ranked 84th in a study of 84 cities by Dr. Jeff Kenworthy, Professor, Curtin University Sustainability Policy

				Institute (CUSP) in Perth, Australia. ITS Magazine 2008
4	Beijing	13,200,000	18 kph	Ranked 81st see above
5	London	12,300,000	19 kph	London is the slowest city in Europe by Forbes Magazine 2008
6	Jakarta	15,100,000	19 kph	Ranked 79th see above
7	Shanghai	17,900,000	20 kph	Ranked 77th see above
8	Mumbai	22,300,000	22 kph	Ranked 76th see above
9	Berlin	4,275,000	24.2 kph [10]	2nd in Europe after London by Forbes Magazine 2008
10	Singapore	4,700,000	34 kph	Singapore's average speed has increased 16kph after the introduction of Congestion Charge
11	Riyadh	4,950,000	54 kph	Ranked 5th see above
12	Brisbane	1,920,000	50 kph	Ranked 7th see above

Note: Data compiled from different sources. San Diego is the fastest city in the world with an average speed of 61 kph as per the study of Dr. Jeff Kenworthy [10].

Inevitably in Dhaka every day we face unbearable traffic jam and the people here have inherently accepted their fates. The citizens of Dhaka city are forced every single day to schedule their activities accordingly keeping adequate lead time in their hand to face and overcome the traffic jam. Scores of sporadic measures have been taken already to make the life a tad bit easier: separate lane [8] in major roads for the slow moving transports like rickshaws; separate lane for the three wheelers in the VIP road (this time of iron bars and

chains, and it failed too), over-bridges at the high traffic points, making footpaths, closing road openings on high traffic points and divert traffic to other direction and lot more.

Unfortunately, nothing proved to be fruitful so far, and very interestingly most of the undertakings have failed to establish an efficient traffic system and lessen traffic jam on the city roads. The aspects that cannot be overlooked are that these failures are not simply wastage of money, it tells that the measures were: (i) taken without much pre-thinking and without an integrated plan [9], (ii) taken without calling for the awareness of the mass people (both the law enforcing authority, drivers, pedestrians, and passengers etc.); and the measures also: (i) drained away valuable time of the concerned authorities to take necessary steps to design and implements the projects, and (ii) brought unbearable hassles to the citizens, and lot other things.

The eccentric behavior of road users here is further fueled by the inrush of new cars every day. Statistical report from the Bangladesh Road Transport Authority (BRTA) reveals that just short of 25 thousand new vehicles were registered in February 2015, making a total of around 2 and a quarter million registered vehicles in the country [13]. This is the reason most commonly cited by road traffic management authorities in Bangladesh for perennial traffic congestions that plague many parts of this heavily-populated nation. On the other hand, official Government of Bangladesh publications claim over 20 thousand kilometres of urban roads and over 100 thousand kilometres of country roads as part of the national transportation network [14]. This clearly reveals that there are frequently several routes to go from origin to destination. Traffic congestion is inherently created and aggravated by people tending to follow the usual and well-traversed routes rather than taking a real-time decision to swerve into a different course. Historical analysis of common driver practices reveal that most

drivers are unaware of alternate pathways and habitually select routes based on their own knowledge and tend not to venture into new roads.

2.2 Viable Options to Slow Down the Convoy

From time immemorial, governments have been engaged in the endless drive to grab hold of this menacing evil by the name of traffic congestion. Strategies may vary across borders but the main goal remains the same-to combat traffic crisis. On the context of Bangladesh, a score of probable solutions can be brought up based upon historical and geographical contexts. It is important for the country to stop draining valuable money and time on unproductive projects and activities. The following suggestions can be thought of as plausible remedies [9, 6]:

1. Integrated activities for the general public, drivers, transportation laborers, law enforcing authority to abide by the traffic rules and raising awareness among all.
2. Boarding/loading bay on the bus stands.
3. Ensure security on the footpaths and make them usable by cleaning; security within the terminals areas.
4. Drive away the hawkers, addicts and floating people from the over-bridges; but at the same time they need to be rehabilitated somewhere otherwise they will come back again.
5. Ban slow moving vehicles (non-motorized vehicles viz. rickshaw, cart etc.) on the city's main road to eliminate unequal competition between slow and fast moving vehicle to have the same small space. These slow vehicles may be allowed to move only in the locality and their vicinity.

6. Ban parking on the roads, road sides. Parking lots must be built; the govt. can encourage the private sector to come forward.
7. Ban stocking building materials on the roads/road sides.
8. Integrated plans, one action must be complementary to another
9. Increase the road networks and add more parallel roads to existing major roads to form a more efficient network. Creating “bus only” lanes and the lanes should be on the left side to accommodate nature of the service.
10. Make use of available alternative networks like waterways and popularize them.
11. Congestion Pricing can be introduced which is proved to be successful in cities like London, Singapore, Stockholm and Oslo.
12. Manage the importation of cars and levy higher taxes on private cars in order to encourage using public transports. But at that same time the quality of service also has to be improved of the public transports to get the real benefit.
13. Encourage long vehicle/double-decker buses in lieu of mini-buses and three wheelers, as long vehicle/double-decker buses can carry more passengers but take proportionately less space.
14. Installing facilities like Rope-ways in short routes like Farmgate – New Market, Farmgate – Mouchak, Sukrabad – Kawran bazaar etc. as short term solution to avoid traffic on the road surface.

3.1 System Overview

The theme of our thesis is the collection of traffic data which allows our system to understand traffic situation at any point of time and provide users with the best possible route. The first component required for such a task is a way of tracking the traffic flow along any route. This can be accomplished by using modern vehicle count methods. For our pick, we chose Floating Car Detection, a non-intrusive and relatively inexpensive way of collecting vehicular data. The conditions of the roads directly affect the traffic signal cycles which are dynamically attuned to the sensors. To control this, we have used Neural Networks to yield a set of drive orders appropriate for each road intersection. Lastly, to compute the route with the shortest possible time, we are using the standard Dijkstra's algorithm.

3.2 Simulation and Demonstration



Figure 3.1 PTV VISSIM User Interface

The simulation was carried out in PTV Vissim software which is used worldwide for vehicular road network modeling. PTV Vissim is a microscopic multi-modal traffic flow simulation software package. "Microscopic simulation", sometimes called microsimulation, means that each entity (car, bus or person) of reality that is to be simulated is simulated individually, i.e. it is represented by a corresponding entity in the simulation, thereby considering all relevant properties. The software has capabilities of placing road traffic detectors for FCD and incorporating algorithms for vehicle routing, very suitable for our simulation.

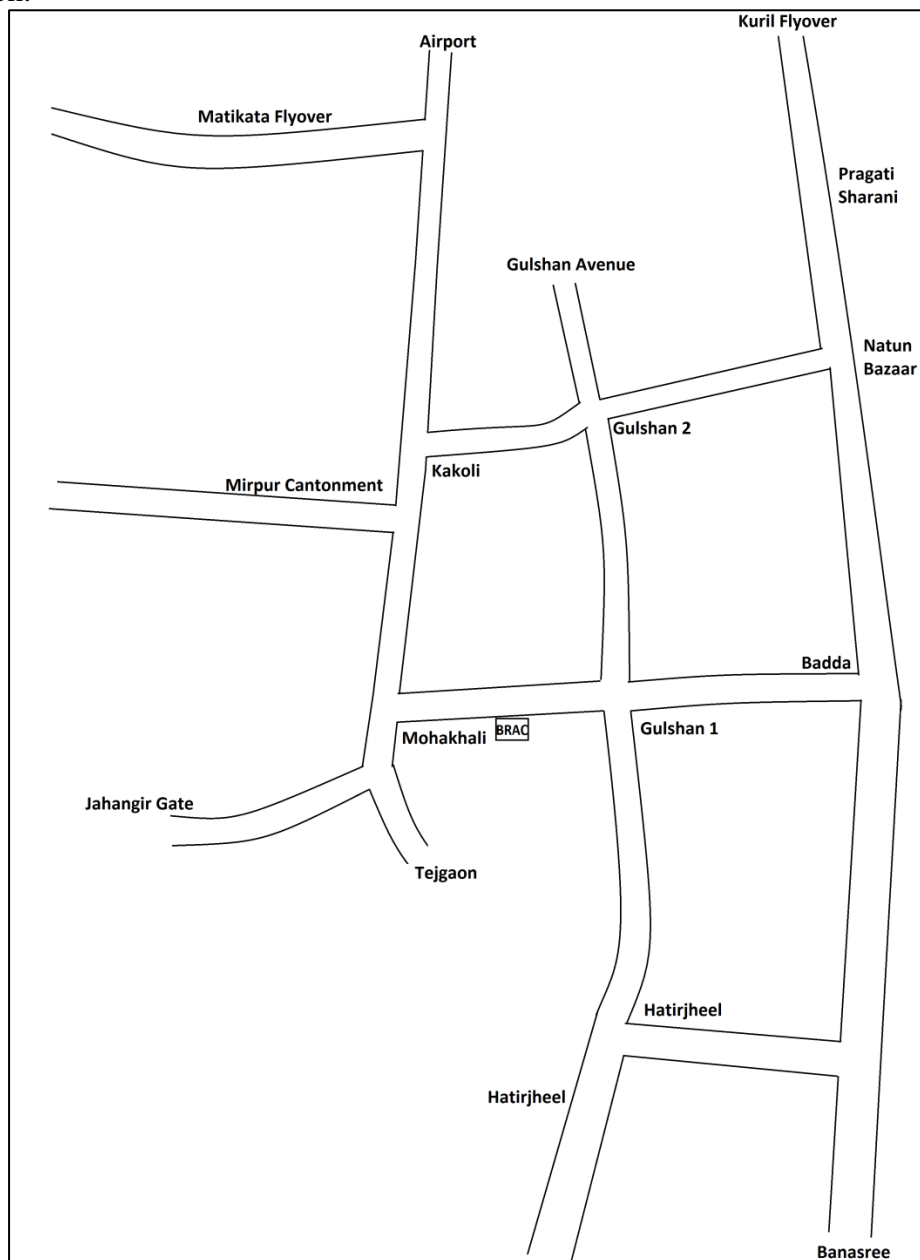


Figure 3.2 Areas of Dhaka City selected for simulation and demonstration

We chose a section of Dhaka city to be the subject of our experiment and the regions include Banasree, Hatirjheel, Gulshan 1, Gulshan 2, Mohakhali, Kakoli, Badda, Pragati Sharani, Mirpur Matikata, Mirpur Cantonment, Jahangir Gate and Tejgaon.

For our demonstration purpose, we imitated the road traffic on a PVC board by the means of 200 white LEDs. A set of red, yellow and green LEDs were used to represent the traffic lights. Sensor readings were emulated using sets of LDRs placed over traffic stops. The readings correspond to pre-calibrated list of vehicle volume index on a scale of 0 to 4 with 0 indicating the most relaxed state and 4 denoting a jam-packed situation. The processing was done in Eclipse Java powered by a computer running Windows OS. The intermediary device used was an Arduino Mega microcontroller development board to interface the computer and the LEDs. Since the algorithms are expansive and heuristic, for both the tasks we used the Intel NUC System-On-Chip (SOC) single board computer due to its satisfactory memory and processing capabilities. Intel Next Unit of Computing (NUC) is a small form factor PC designed by Intel. The model we used was Intel D2820FYKH powered by a dual core Intel CPU and 4GB of RAM adjoined with a 64GB of Solid State Drive for secondary storage. The operating system used was Windows 8.1.

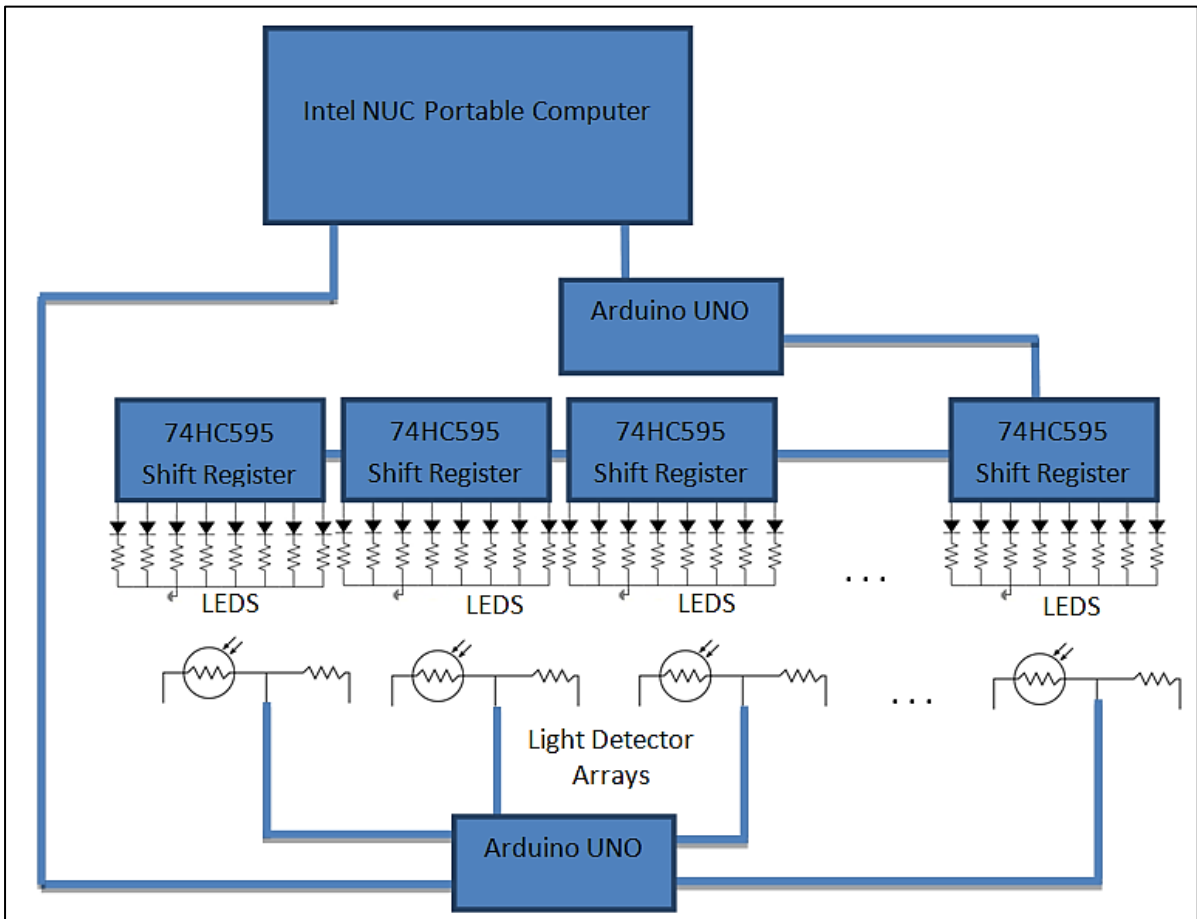


Figure 3.3 Schematic diagram of the demonstration system

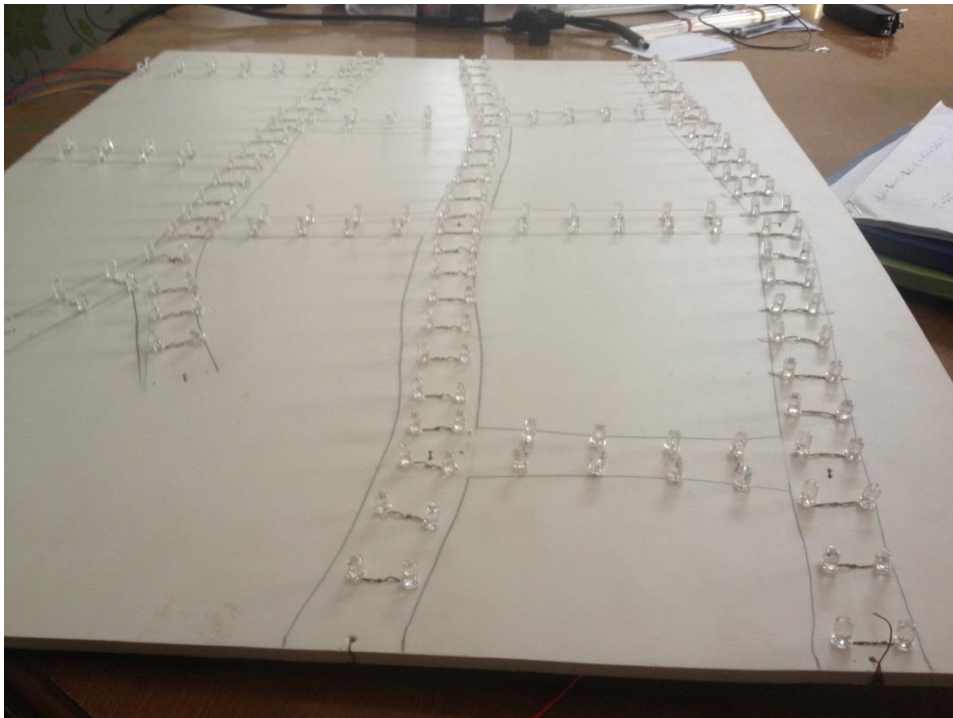


Figure 3.4 PVC board with LED strips to imitate the roads

The last layer of our endeavor was the development of an onboard display for the car dashboards. For this, we used a 7" LCD Panel driven by a Raspberry Pi SoC Computer. The overlying operating system associated with the Raspberry Pi was Android version 4.4 KitKat. The system would dynamically generate routes on maps based on the data processing done on the central backend server and present that on the display. Drivers would be advised to take the routes with shortest time required. The application establishing the link with the backend was developed with Android ADK and Eclipse Java and the connection to the server was channeled through local 3G data service. A 3G dongle was used with the Raspberry Pi for our demonstration. Our choice of system and its associated components originate with the intention to keep the cost at minimum as we aim this operation to be a government level implementation. Since individual installations may prove to be cumbersome and time consuming, a government project would prove to be very useful.

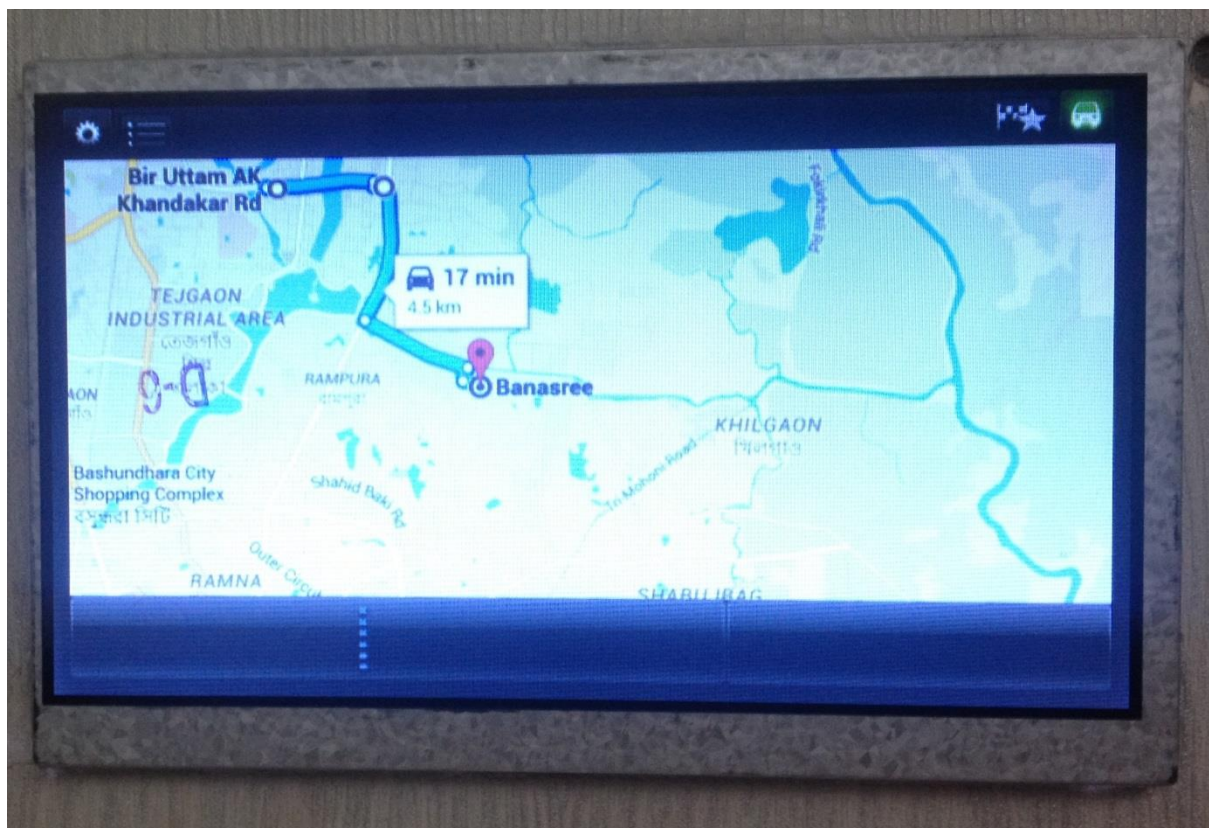


Figure 3.4 LCD Panel for the onboard display for the car

4.1 Methodology

One of the ways that our method is different is the input parameter. Whereas most papers on this area of research have focused on the traffic density, our method is centered on the traffic flow, i.e., how quickly the traffic is clearing out from a busy road. The system takes in car flow rate or in other words, volume of cars displacing a certain point at a given time using Floating Car Detection (FCD), which is a modern vehicle count method. Traffic signals are manipulated using Artificial Neural Networks. With the combination of FCD and the Artificial Neural Networks, we can take into account the traffic flow at any time and accordingly manipulate the traffic signals at the intersection, besides using Dijkstra's algorithm for the rerouting vehicles to maintain traffic equilibrium.



Figure 4.1 Floating car data mechanism demonstrated by BMW

4.2 Vehicle Flow Sensor Inputs

4.2.1 Traffic Flow Monitoring and Detection

In order to combat the ever growing traffic ordeal, it is indispensable to rely on a certain proven technique to measure the stream of cars on the roads. Numerous studies have been conducted on the grounds of *instantaneous vehicle count method* to collect traffic data in real-time in order to determine the nature of traffic.

4.2.2 Instantaneous Vehicle Count Method

Intelligence necessitates information, and information requires data, which is generated by monitoring or surveillance. A fully integrated vehicle detection system includes sensors, communication, and centralized management equipped with processing power. These technologies work in conjunction to each other to assist the automated transportation system as whole. What follows is the organized process of managing:

1. Volume Count
2. Vehicle Classification
3. Vehicle Occupancy
4. Travel Time
5. Delay

In order to determine the number, movements, and classifications of roadway vehicles at a given location traffic volume studies are carried out. These data helps to identify critical flow time periods, determining the influence of large vehicles or pedestrians on vehicular traffic flow. Two methods are available for conducting traffic volume counts: (1) manual and (2) automatic. Manual counts are typically used to gather data for determination of vehicle classification, turning movements, direction of travel, and vehicle occupancy. The automatic count method provides a means for gathering large amounts of traffic data. Automatic counts

are usually taken in 1-hour interval for each 24-hour period. The counts extend for a week, month, or year. When the counts are recorded for each 24-hour time period, the peak flow period can be identified.

We try to present a short overview of some of the latest automated traffic data collection technologies. Strictly speaking, traffic count technologies can be differentiated into two categories: the intrusive and nonintrusive methods. The intrusive methods basically consist of a data recorder and a sensor placing on or in the road. Non-intrusive techniques are based on remote observations. An emerging, more effective process of automated traffic data technology is global positioning system (GPS) based floating car data (FCD). FCD is an alternative or rather complement source of high quality data to existing technologies. They will help amplify safety, efficiency and reliability of the transportation system and are becoming crucial in the development of new Intelligent Transportation Systems (ITS). In the near future they will be the 'de facto' technology to enhance travel time predictions.

4.2.3 Choice of Detector Technology

Up recently, there has been an emergence of non-intrusive vehicle techniques for determining the link travel time where they make use of GPS based FCD technology, as discussed earlier. This is due to its simplicity and low cost of implementation. However, the major problem with GPS is that the accuracy of a typical GPS receiver is about 10 meters. This makes it difficult to pin-point a crossing for the purpose of congestion measurement. Secondly, it has been noticed that GPS sends erroneous velocity data even when the vehicle is stationary.

Other notable ITS probe FCD vehicle technique includes:

- a) Sign-post based Automatic Vehicle Location (AVL)
- b) Automatic Vehicle Identification (AVI).
- c) Ground-based Radio Navigation.

d) Cellular Geo-location.

Our system aims to make use of RFID enabled probe vehicles to monitor traffic and detect congestion. Several technologies have been existent for quite a while for congestion detection, such as the inductive loop, magnetometer, infrared, acoustic, ultrasonic, visual camera, radar etc. These techniques suffer from a common drawback - high implementation cost and maintenance cost.

4.2.4 Radio-frequency identification (RFID) or Transponder Systems

Radio-frequency identification (RFID) is an automated identification approach, depending upon stocking in addition to retrieving info from remote areas using devices called RFID labels or transponders. The technology requires some degree of synergy of the RFID reader in addition to a great RFID label. An RFID tag is an object that can be applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. A rudimentary RFID system includes about three elements:

1. An antenna or coil
2. A transceiver (with decoder)
3. A transponder (RF tag)

Typically, an RFID tag consists of a microchip to collect information and an antenna that transmits this data wireless to a reader. In its simplest form, the chip will contain a serialized identifier, or license plate number, that uniquely identifies that item. Typically, processed data would be used to provide revised scheduling and arrival time information to the general public, via variable information signs. Transponder systems are also used with Selective Vehicle Detection (SVD) systems which are designed to allow priority at traffic signals or cordon points for public transport or emergency service vehicles. The key advantage of RFID

is that RFID tags can be read through materials without line of sight. RFID tags can be read automatically when a tagged product comes past or near a reader.

4.2.5 Floating Car Detector (FCD)

In modern times there has been an advent of floating cellular data which is gradually becoming a top choice to collect vehicle densities on roads. Our approach focuses on using this technique due to its simplicity and the fact that it doesn't need many additional components [18]. This method uses cellular network data (CDMA, GSM, UMTS, and GPRS) and thus no special devices are necessary. Instead of installing beacons or tags onto the cars, this method uses the cars themselves as floating sensors to generate information on vehicle flow. Virtually every switched-on mobile phone turns into a traffic probe and is as such an anonymous source of information. Using either GPS triangulation or the hand-over data stored by the network operators' towers, the location of the mobile phone is determined [18].

Since GSM localization is less accurate than GPS based systems, numerous phones must be tracked and sophisticated algorithms be implemented to extract high-quality data [18]. For instance, care must be taken not to misinterpret cellular phones on a high speed railway track near the road as incredibly fast journeys along the road. However, the more congestion, the more cars, the more phones and thus more probes. The distances between cell sites are lower in metropolitan areas where traffic data have high demand and thus precision increases there. FCD's strength over GPS-based or conventional methods such as cameras or street embedded sensors lies in the fact that it requires no infrastructure or hardware in cars or along the road. It is much less expensive, offers more coverage of more streets, it is faster to set up (no work zones) and bears lower maintenance [18]. A vital feature pertaining to FCD-systems is the employment of impulsive security mechanisms. The sensor data given by any floating car

could possibly be used to track a certain vehicle unless the information can be furnished anonymously. FCD ensures that by randomizing any identifiable tags.

FCD (Floating Car Detector) sensors provide a way to measure traffic flow in any road. These readings are then sent to the main server which then processes them and assigns values to each route accordingly. These assigned values are then used to form the input to our Artificial Neural Networks (NN). The NN then decodes the input, evaluates them and then comes up with an array of solution. The solution which fits the present situation is then chosen. The NN is run several times for different routes to provide the route which allows the user to reach his destination at the shortest possible time.

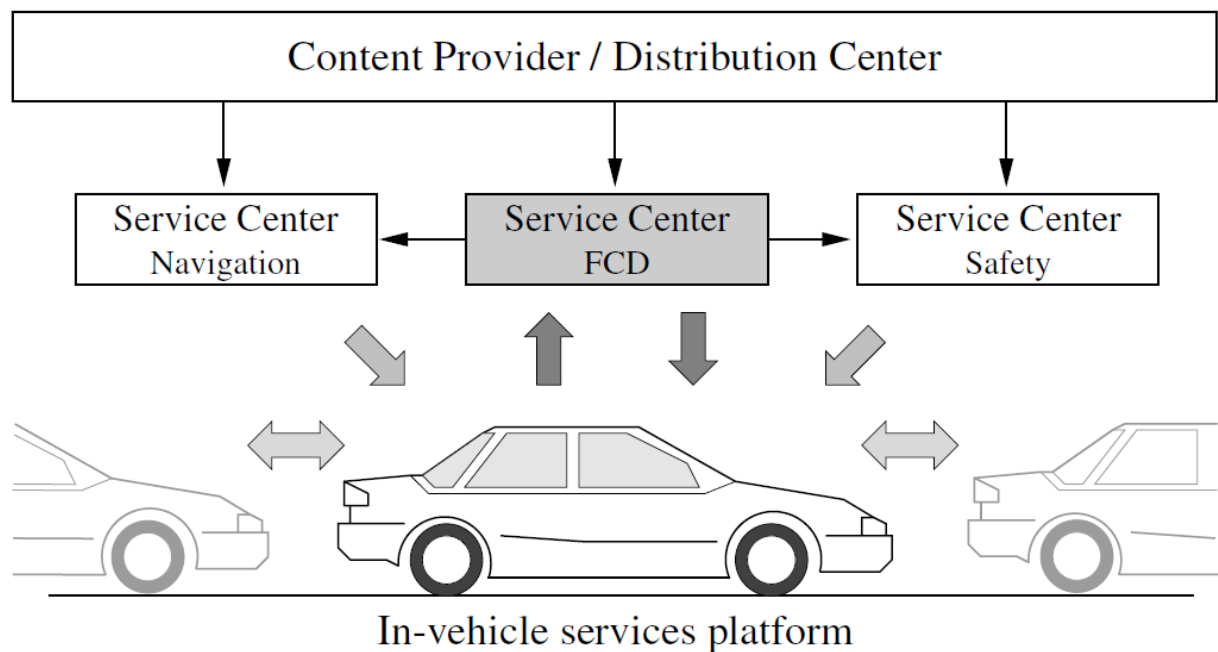


Figure 4.2 Floating Car Data Principle

4.3 Algorithms

4.3.1 Artificial Neural Networks Algorithm:

Artificial Neural Networks falls in the category of AI that deals with machine learning, i.e., the study of systems that can learn and adjust itself from the data provided. They are complex adaptive systems, meaning they can change their internal structure based on the information flowing through it [3]. The aim is to replicate the functions of a human brain by making the algorithm learn and adjust to the changing conditions. At its very elementary stage, neural networks are composed of artificial neurons, which receive inputs, process these inputs and generate outputs [15]. There is an activation stage for each neuron which determines whether the said neuron will fire its output to the next layer of neurons or not. For our work, the activation stage involved a sigmoid curve. The choice for using the sigmoid curve is because it has a continuous output over the region of (0, 1) non-inclusive and satisfies a property between the derivative and itself such that it is computationally easy to perform.

$$\frac{d}{dt} \text{sig}(t) = \text{sig}(t)(1 - \text{sig}(t))$$

where, sig(t) is the sigmoid function.

Each layer of neurons fire an output depending on the activation function and the outputs are propagated through to the final node. The inputs and outputs are normalized so that they are continuous in the range of 0 to 1.

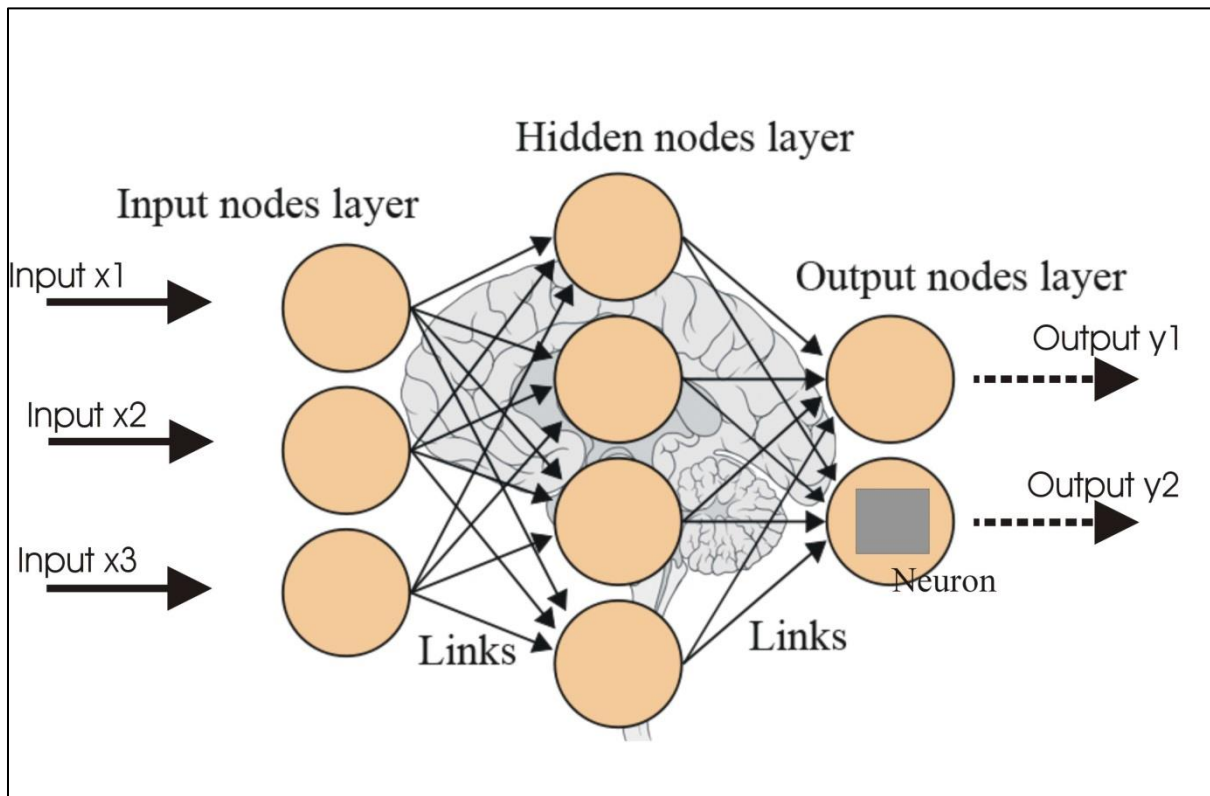


Figure 4.3 Neural network algorithm hierarchies

There are weights associated to each neuron such that the inputs of these neurons are multiplied by the weight before being propagated forward. These weights are what give our neural network its adaptive ability as it is not feasible to change the internal structure of the network. For the initial run of the system, the weights of each node are randomly generated in the range of $[-1,+1]$. These weights are then adjusted so that the output is closer to the expected output. For this to happen, we implemented the supervised learning technique on our network.

Supervised learning is essentially a strategy that involves the user to be smarter than the network that he is teaching [16]. The user has to provide the set of sample inputs for the system to run through, with the outputs being known by the user. The network will make its guesses which will be compared with the outputs of the user and the difference between them will allow the network to make the adjustments. The above process is known as back

propagation, where the outputs of the network are generated in the same pattern as a neuron. The difference here is that the error, that is the difference between the actual output and the expected, correct output, is passed through additional layer of neurons and are hence propagated backwards. This results in the weights being adjusted to produce something close to our expected output. For each set of inputs, the network was allowed to undergo 500 iterations before approximating towards the expected output.

The output of the network was normalized and thus came in the region of 0 to 1. The output was interpreted to provide the drive order, which gave the duration of the green light cycle.

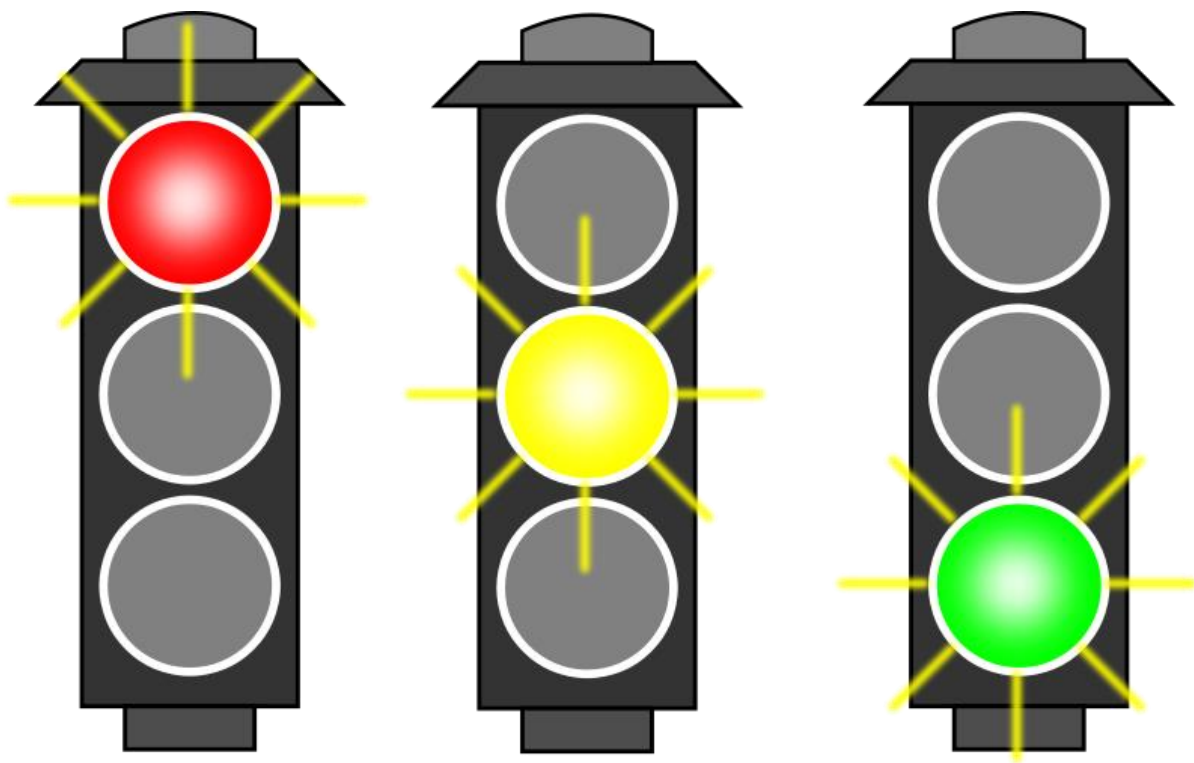


Figure 4.4 Typical traffic light cycle

In our neural network, we used two hidden layers, one input layer and one output layer. There were four input nodes corresponding to the four levels of threshold (to be discussed later) in each roads. These inputs are connected to the nodes of the hidden layers. The first hidden layer contains 4 nodes while the second layer contains two nodes. The final layer has the singular node which will provide the drive order for the traffic intersection. To ensure that our algorithm does not fall prey of being bottled in a local minima, we used biased weights in each of the hidden layers.

Normally, for static traffic light controllers, STLCs, there is rarely an optimal green cycle meaning that there are large queues of cars to be cleared. This can be mathematically defined as:

$$Q_L = Q_{tot} - Q_{clc}$$

Where, Q_{tot} is the total number of cars in queue,

Q_{clc} is the number of cars cleared and

Q_L is the number of cars remaining

Thus an additional length of the green light cycle is required to clear the Q_L amount of cars [15]. By the time of the next cycle, this value is slated to increase meaning that it is likely that $Q_L > 0$ even after the next cycle. However, with the deployment of dynamic traffic light controllers, DTLCs, the value of Q_L is expected to fall with the system actively assigning drive orders to clear the queues.

Our desired outputs will thus be a combination of drive orders for each intersection. We have tried to demonstrate the possible behavior of road traffic because of the selection of drive orders on the PVC board with LED strips to show the limited scale viability of the approach.

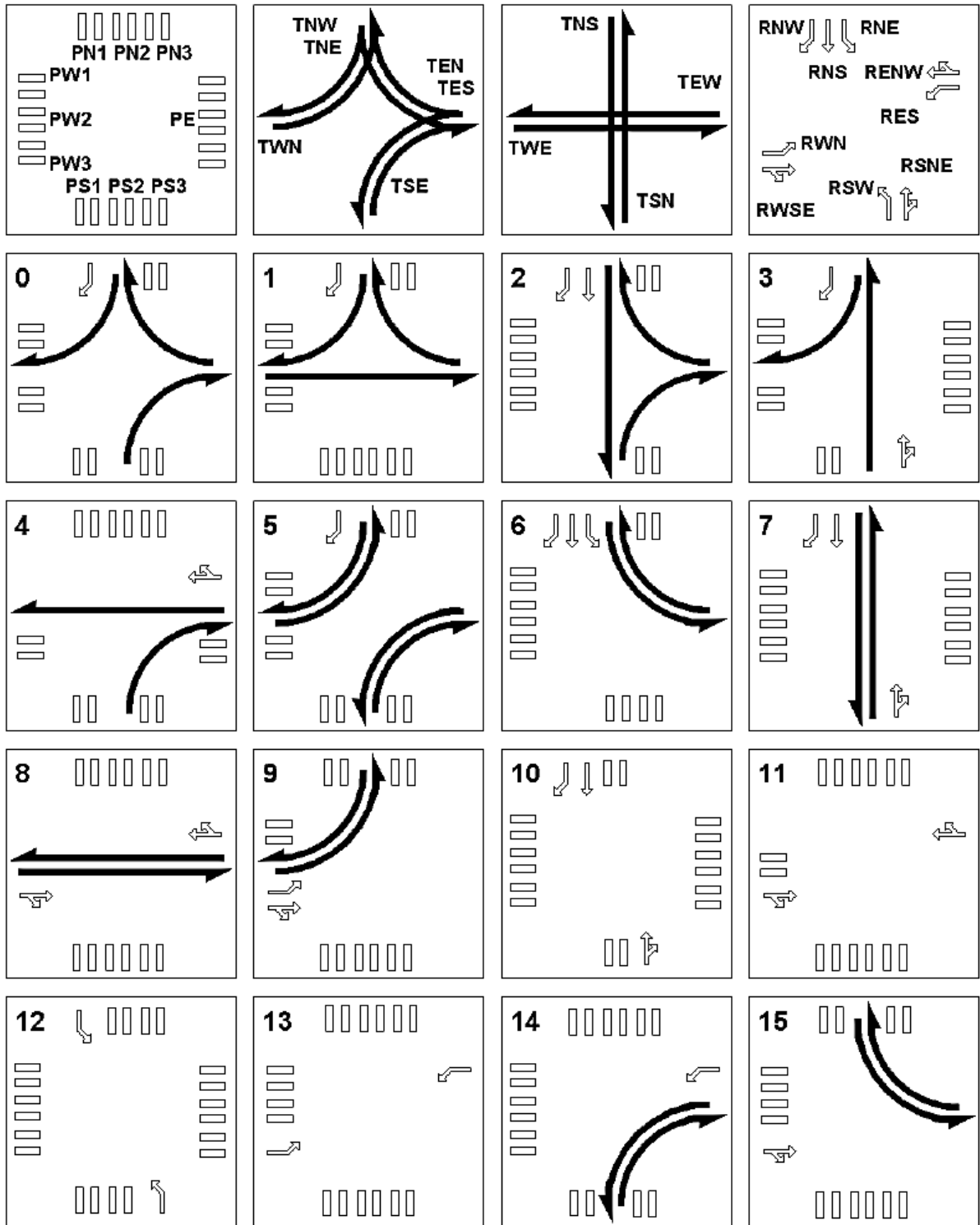


Figure 4.5 Combinations of some possible drive orders [19]

4.3.2 Dijkstra's and A*Algorithm:

For the routing of vehicles, we used a variant of the Dijkstra's algorithm. Dijkstra's is normally used for computing the shortest length in any given graph. However, the input to our algorithm was the traffic flow at any given route. The product of the traffic flow and the historical average speed of vehicles in that route was the weight assigned to that route.

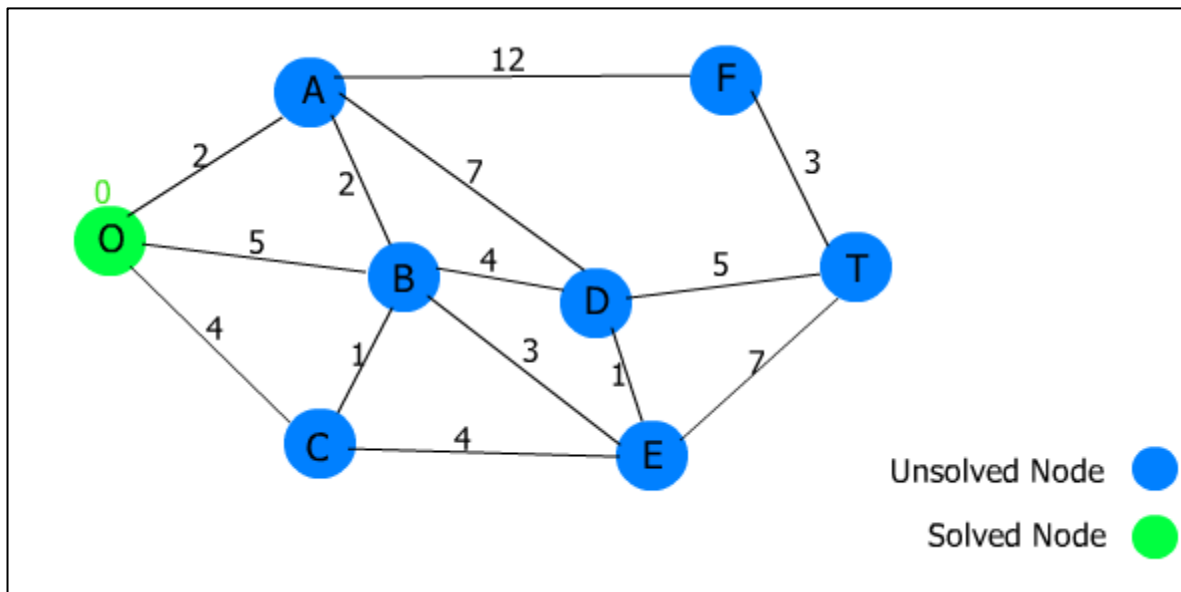


Figure 4.6 Dijkstra's algorithm node solving process

Dijkstra's algorithm works by beginning from the root Node and visiting all of its neighbors, ranking them based on the minimal sum of the cost of visiting edge connecting it to the root node [14]. The original variant of the algorithm stops when the end node is reached. The modification to our algorithm is that it calculates the minimum traveling time from the source node to each of the nodes in the network, thus allowing users to make informed choices based on the traffic. It even suggests the host of intermediary nodes that needs to be traversed for reaching a certain destination node in the shortest possible time.

One of the limitations that is associated with Dijkstra's is its computational time. The original Dijkstra's algorithm had a time complexity of $O(|N|^2)$ where N is the number of nodes. Our

variant is based on a minimum Priority Queue, which helps in improving the time complexity from $O(|N|^2)$ to close to $O(|E|+|V|\log|V|)$. However, a faster variant of Dijkstra's is known as A* (A star) which is based on best first search. A* is an algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently traversable path between points, called nodes. Noted for its performance and accuracy, it enjoys widespread use. However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance, although other work has found A* to be superior to other approaches. A* achieves better time performance by using heuristics.

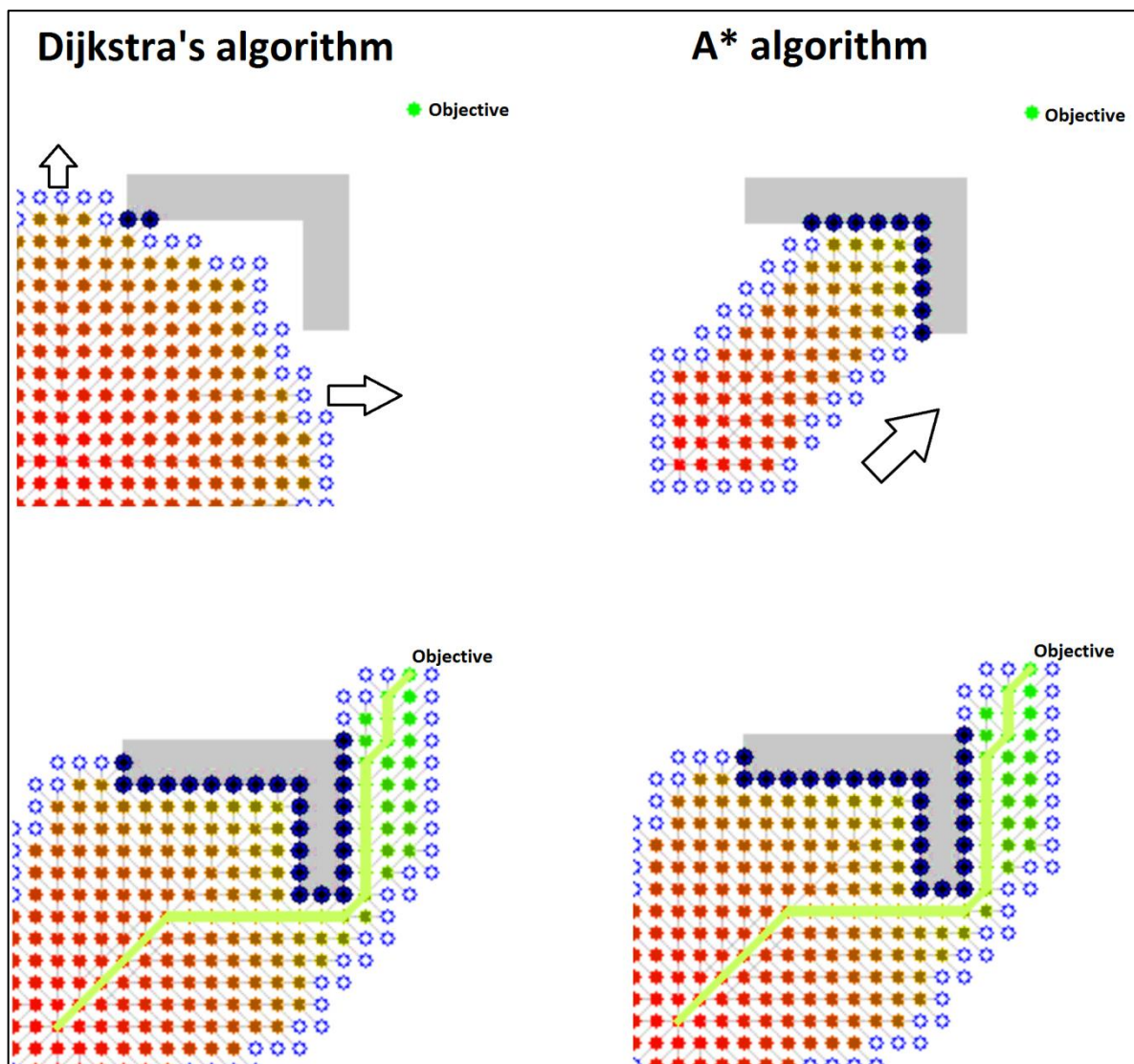


Figure 4.7 Dijkstra's algorithm and A* algorithm comparison

As A* traverses the graph, it follows a path of the lowest expected total cost or distance, keeping a sorted priority queue of alternate path segments along the way. It uses a heuristic function, $h(x)$ for each node x plus the cost from the source node to n [14]. To ensure that A* always finds the shortest path, $h(x)$ must for each node return a value that is lower than the actual minimal cost to the target node. A* has the advantage of speed over Dijkstra's but it lags due to its heuristics requirements. We used A* algorithm along with Dijkstra's for performance analysis and time complexity.

5.1 Route Calculations

The route finding algorithm was implemented in Eclipse IDE using Java programming language. The end user's inputs were 'source' and 'destination'. Basically the user needs to tell the system where he or she is presently residing and where he or she is intending to go. The system automatically calculates the route based on the traffic conditions of the nodes. Additional inputs to the system were the average speed of cars and the sensor output readings from the nodes. Comparative measurements were taken from Dijkstra's and A* algorithms' compute times. It was conducted to present a performance evaluation of the two algorithms and ultimately decide which is more efficient and convenient for our purpose.

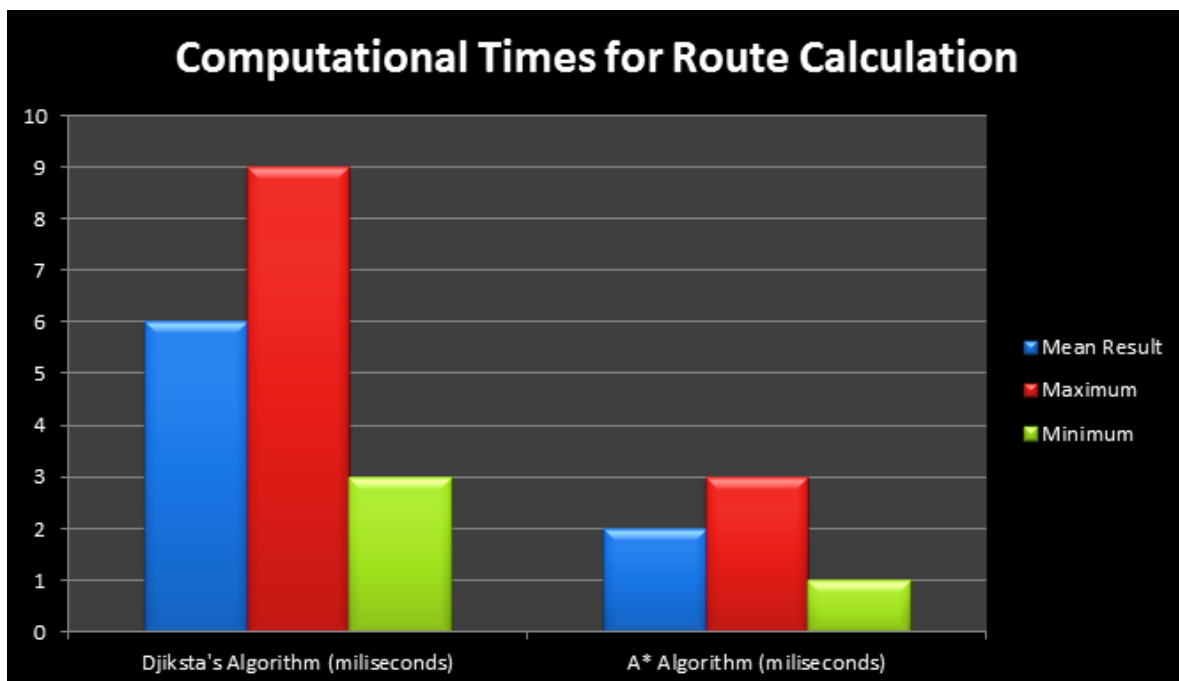


Figure 5.1 Chart showing comparison of computational times

From the Eclipse IDE's internal timer, it has been found that the time required for Dijkstra's was shorter, amounting to an average of 6 milliseconds based on 10 samples. A* on the other hand required 2 milliseconds, forking upon on the same number of computations. This

accounts to the fact that Dijkstra's uses breadth first method of finding out routes and tends to evaluate all possible routes. Hence it requires more time to reach the desired path. A* on the other hand is a 'greedy' algorithm with searching for objective in mind. It uses heuristics to make inferences for determining alternate means to find the shortest routes. Thus, it requires significantly less time to compute. However, there is one big catch for A* which deems it very less usable for road path finding. A* relies on heuristics and has one additional cost function which requires additional randomized assumptions. Dijkstra's doesn't suffer from this as it incorporates every single route possible and has guarantees 100% accuracy of choosing the path requiring shortest time. This means that A* is less reliable while calculating the correct choice in spite of it bearing the lowest time required tag.

The other measurement was determination the shortest time requiring route itself. For our sample test we wanted to calculate the route requiring shortest path from Banasree to Mirpur. On the Java program, Banasree was selected as 'source' and Mirpur was selected as the 'destination'. Gulshan 2 node was assigned with a vehicle volume index of 4, Gulshan 1 node was assigned with a vehicle volume index of 2, Mohakhali was assigned with a vehicle volume index of 2, Cantonment was assigned with a vehicle volume index of 3 and Matikata Flyover (Airport Road) was assigned with a vehicle volume index of 1. When the program was run, the path requiring the shortest time was generated and displayed on the LCD which was Banasree-Gulshan 1-Mohakhali-Mirpur Matikata- Mirpur.

Table 2 Vehicle Volume Index Data

Checkpoint	Vehicle Volume Index
Gulshan 2	4
Gulshan 1	2
Mohakhali	2
Cantonment	3
Matikata Flyover (Airport Road)	1

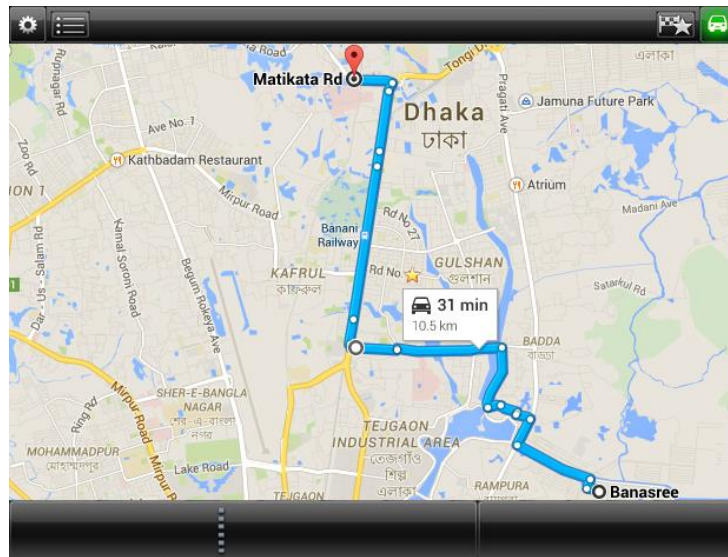


Figure 5.3 Selected route that is to be displayed

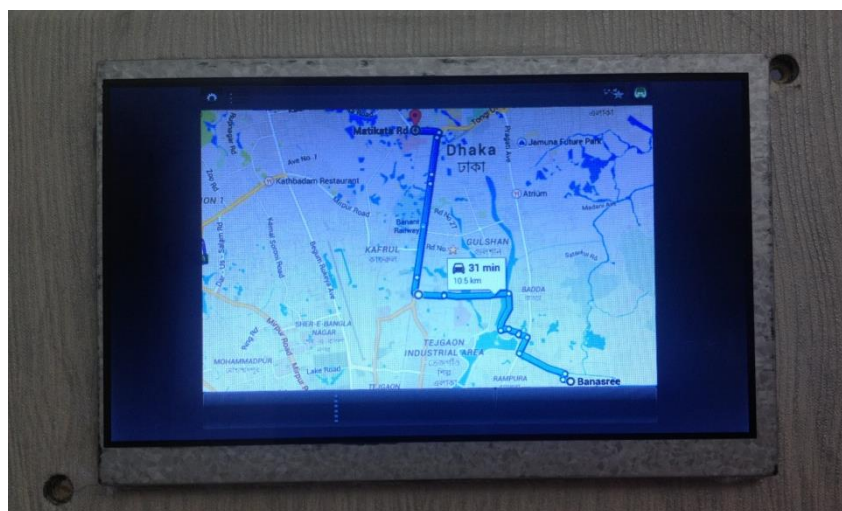


Figure 5.4 Route displayed on LCD

5.2 Simulation Results in Vissim

To verify and support our findings, the first three shortest time requiring routes were noted down from the Java program in the order of their time required. In the next step, the Vissim road traffic simulation software was loaded with our predefined model of the selected regions of Dhaka city and the specified vehicle volume indices were inputted to replicate the traffic conditions. Next each vehicle route that was noted down earlier was analyzed individually and the vehicle travel times and vehicles cleared every hour were noted down for every case. To reduce anomalies and randomizations, 10 runs were performed for each route in the course of 10 minutes for each and the outcomes were compiled in a graph for comparison. As we can clearly see from the graph below, of the three routes, Banasree-Gulshan 1-Mohakhali-Mirpur Matikata- Mirpur took the shortest travel time and also relatively cleared many vehicles.

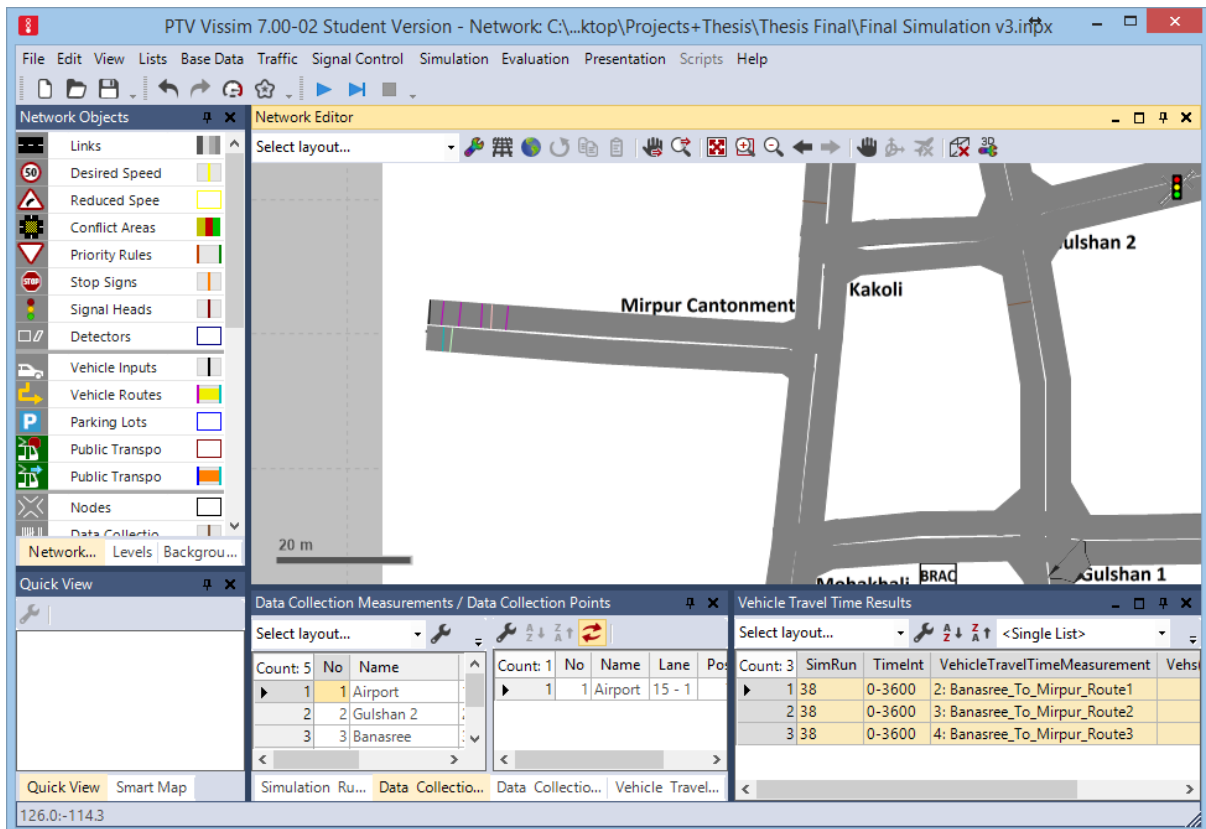


Figure 5.5 Simulation running in Vissim

Table 3 Vehicle Travel Times Data

	Banasree-Gulshan 1-Mohakhali-Cantonment-Mirpur (seconds)	Banasree-Gulshan 1-Mohakhali-Cantonment-Mirpur Matikata-Mirpur (seconds)	Banasree-Gulshan 2-Mirpur (seconds)
Score	18	6	34
Mean Deviation	4.3	1.11	6.9
Positive Outlier	20	11	36
Negative Outlier	3	1	22

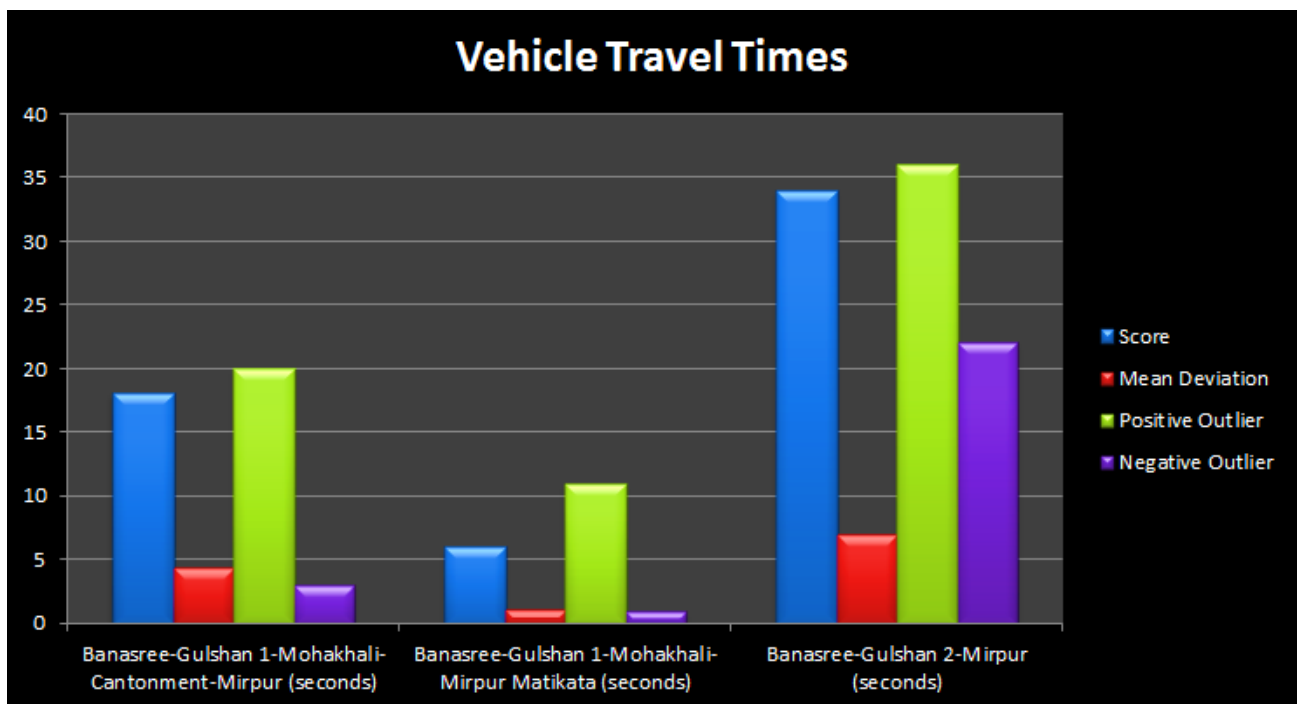


Figure 5.6 Vehicle Travel Times Comparison Chart

Table 4 Vehicles Cleared Every Hour Data

	Banasree-Gulshan 1-Mohakhali-Cantonment-Mirpur (seconds)	Banasree-Gulshan 1-Mohakhali-Cantonment-Mirpur Matikata-Mirpur (seconds)	Banasree-Gulshan 2-Mirpur (seconds)
Score	896	840	211

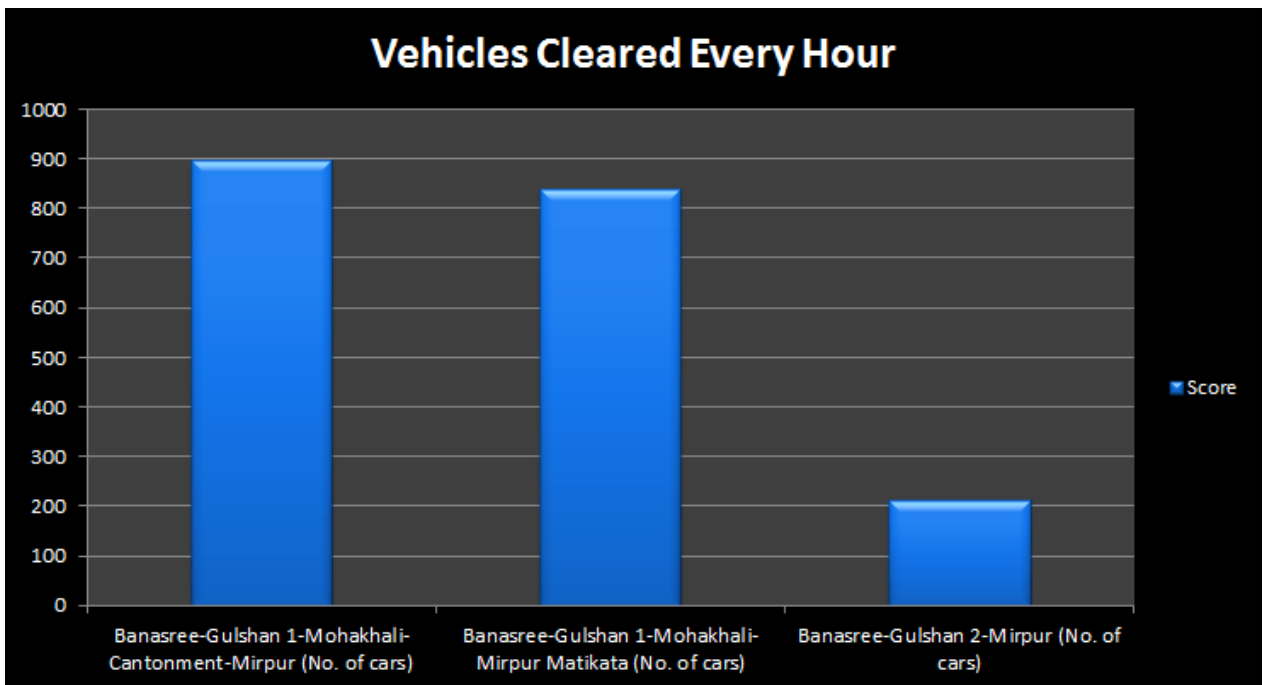


Figure 5.7 Vehicles Cleared Every Hour Comparison Chart

5.3 Traffic Light Manipulations

Selecting the drive order was the next part of our solution to combat the traffic jam condition evident. Based on the vehicle volume index inputs, another Java program was developed based on the Neural Networks algorithm discussed earlier to generate drive order for each intersection. The vehicle volume indexes were put inputted into the program as the input nodes layer along with the intersection for which we wish to extract the drive order. The results were generated as drive orders from the output nodes layer. The drive order was displayed on the Graphical User Interface (GUI) on the screen along with the indication of traffic lights. Accordingly, the traffic lights would change their patterns.

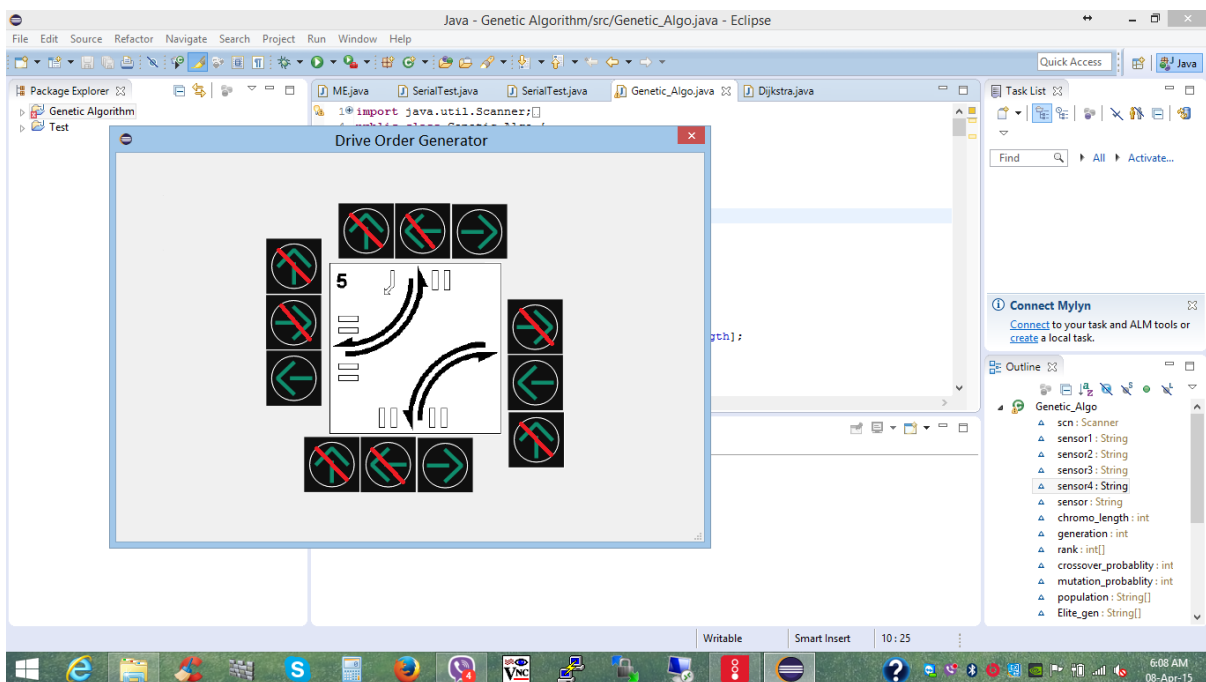


Figure 5.8 GUI showing drive order

6.1 Limitations and Difficulties

During the course of our research, numerous changes had to be introduced and our design had to go through major overhauls considering many practical aspects. Initially, we planned to implement the traffic light controllers with the help of field programmable gate array (FPGA) which is widely used in industries to implement logic circuits. We conducted our experiment with FPGA and it turned out that the scope of FPGA is very limited in term of performance and processing capabilities. In order to fully incorporate algorithms like Artificial Neural Networks, a computer with adequate processing power, memory and input output capability is required. FPGA cannot integrate such algorithms as its meager RAM and flash memory. Moreover, the only means of providing inputs to the FPGA was through push buttons or logic highs and lows. Given the number of traffic intersections whose data we had to input to the system, this method of input is very unsuitable. To the bare minimum, a system-on-chip microcomputer like the Raspberry Pin can be used that can take inputs from a wide range of devices and process information much faster with greater temporary storage support.

Another dilemma we came across was the choice of the sensor for the vehicle flow measurement. Previously, we chose RFID for taking readings of vehicle density. It turned out that RFID requires invasive measures and all vehicles need to be fitted with RFID modules. Moreover, the range of RFID revealed to be very limited, extending only a meter at maximum. After numerous trials, we finally moved to FCD which is far more convenient and reliable as it can operate using existing infrastructure, provide accurate measurement and is widely used for vehicle rate detection all over the world. Consequently using a technology

with industry proven reliability called for no other questions. Another argument going for FCD is that FCD counts vehicles anonymously and has no way of tracking an individual. Appreciating privacy was another aspect we were concerned of while choosing a method to count cars. After our experimental evaluations, FCD turned out to be the ‘de facto’ standard for this mode of work.

6.2 Expectations, outcomes and future prospects

Curtailing road traffic congestions requires extensible effort from all ends and it is never a viable deal to stick to only a handful of approaches to aid in this process. What we have presented was a mere demonstration how introducing two different approaches can help to digitize the entire traffic flow system and in the process bring down the magnitude of traffic jams significantly across developing countries like Bangladesh. Our solution helps drivers by guiding them to take the route with lesser congestion and save their valuable time. On the broader perspective the process offloads the pressure from the jam packed road channels and hence reduces the overall traffic ordeal. Situations like gridlocks can also be avoided as the drivers are notified beforehand what is the state of the road ahead. It is also apparently clear that the much of the work of traffic police have also been reduced in this package as traffic light sequences are fully automated in this system and the traffic light controllers no longer need to be manipulated manually to cater to abnormal traffic situations.

We have tried and tested the plausibility of our system in the small scale and we are confident enough by our findings that this system will be able to scale up to the real world scenario and efficiently deal with traffic congestions spontaneously. We are hopeful and feel passionate to extend our work further and also see a practical implementation of our system in the near future.

In the future our proposed solution package will include alternate modes of travel to ease the load from the exhausted roads. Since we cannot just decrease the number of cars, what we can do is have vehicles like aerial lifts that use fixed tracks to for commuting purposes. Having such vehicles not only would contribute to bring down traffic jam but also lend a hand to building a greener world as emissions would be brought down by large fractions.

REFERENCES

- [1] Mahmud S. M. S., Hoque M.S. (2008), "Deficiencies of Existing Road Network in Dhaka Metropolitan City", Publication in 10th Pacific Regional Science Conference Organization (PRSCO) Summer Institute 2008, Hosted by Bangladesh Regional Science Association (BRSA), 15-17 May 2008, Dhaka, Bangladesh, P.N. - 52. pg- 86 (Abstract), Paper 6b4, (CD proceedings)
- [2] Anfilets S. V., Shuts V. N. (2010), "Artificial Neural Networks For Adaptive Management Traffic Light Objects At The Intersection", Proceedings of the 10th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'10), 20–23 October 2010, Riga, Latvia, p. 457-462. ISBN 978-9984-818-34-4 Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia
- [3] Eichler S. (2006), "Anonymous and Authenticated Data Provisioning for Floating Car Data Systems," Communication systems. ICCS 2006. 10th IEEE Singapore International Conference on , vol., no., pp.1,5, Oct. 2006
- [4] de Oliveira, M.B.W., de Almeida Neto, A. (2014), "Optimization of traffic lights timing based on Artificial Neural Networks," Intelligent Transportation Systems (ITSC), IEEE 17th International Conference on , vol., no., pp.1921,1922, 8-11 Oct. 2014
- [5] Nanayakkara S.C., Srinivasan D., Lai W. L., German X., Taylor E., Ong S.H. (2007) "Genetic Algorithm based route planner for large urban street networks," Evolutionary Computation. CEC 2007. IEEE Congress on , vol., no., pp.4469,4474, 25-28 Sept. 2007
- [6] E. S. Alam, "Traffic problem in Dhaka", *Financial Express*, August 22, 2009

- [7] De Fabritiis, C., Ragona R.; Valenti G. (2008) "Traffic Estimation And Prediction Based On Real Time Floating Car Data," Intelligent Transportation Systems. ITSC 2008. 11th International IEEE Conference on , vol., no., pp.197,203, 12-15 Oct. 2008
- [8] "Speech of the Inspector General of Police (IGP), Traffic Jam in Dhaka City: Challenges and Possible Solution" American Chamber of Commerce in Bangladesh, Dhaka, Bangladesh, January 15, 2008, interview. Accessed: 27 Mar. 2014. [Online]. Available: http://www.amchambd.org/amcham2/all_files/Speech%20on%20Traffic%20Jam%20By%20IGP.doc
- [9] M. A. S. Azad (2011), "The Traffic System of Dhaka City and Effective Utilization of Road Space", Bangladesh Development Debate, August 2011.
- [10] Khan S. (2007), "Automated versus human traffic control for Dhaka and cities of developing nations," Computer and information technology, 2007. ICIIT. 10th international conference on , vol., no., pp.1,4, 27-29 Dec. 2007
- [11] A. Downie, "The World's Worst Traffic Jams", Time Magazine, April 21, 2008
- [12] "Major Agglomerations of the World", City Population, January 2015, statistical summary. Accessed: 27 Mar. 2014. [Online]. Available: <http://www.citypopulation.de/world/Agglomerations.html>
- [13] G. V. Das, "Bangalore's Traffic Mess – The Solution", February 2009, presentation.
- [14] "Number of Registered Motor Vehicles in Bangladesh (Year wise)," Bangladesh Road Transport Authority (BRTA), Dhaka, Bangladesh, February 2015, statistical summary. Accessed: 27 Mar. 2014. [Online]. Available: http://www.brta.gov.bd/images/files/statistics/stat_bd_02-15.pdf
- [15] Board of Investment. Organization Homepage (2014). Accessed: 01 Jun. 2014. [Online]. Available: www.boi.gov.bd/index.php/about-bangladesh/transportation-network#road

- [16] McCulloch W. S., Pitts W., (1943) "A logical calculus of the ideas immanent in nervous activity", The bulletin of mathematical biophysics December 1943, Volume 5, Issue 4, pp 115-133 A logical calculus of the ideas immanent in nervous activity
- [17] Shiffman, D. (2012). The nature of code book project. New York: Daniel Shiffman.
- [18] Eichler S. (2006), "Anonymous and Authenticated Data Provisioning for Floating Car Data Systems", Communication systems. ICCS 2006. 10th IEEE Singapore International Conference on , vol., no., pp.1,5, Oct. 2006
- [19] Jansson G. (2010), "Traffic Control with Standard Genetic Algorithm", M.Sc. dissertation, Department of Applied Information Technology, Chalmers University of Technology, Gothenburg, Sweden, 2010