

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND  
ENGINEERING**



**An Intelligent GIS Based Physical Model for Improvement  
of Urban Traffic System**

By

Md. Tamim Islam  
Student ID: 084426  
and  
Murshid-Bin-Rahman  
Student ID: 084445

Supervised By

Md. Ali-Al-Mamun  
Assistant Professor

Department of Computer Science and Engineering (CSE)  
Islamic University of Technology (IUT)  
Organisation of Islamic Cooperation (OIC)  
Dhaka, Bangladesh

September, 2012

## **CERTIFICATE OF RESEARCH**

This is to certify that the work presented in this thesis is the outcome of analysis and investigation carried out by the candidate under the supervision of Mr. Md. Ali-Al-Mamun in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

### ***Signature of the Head of the Department***

-----  
Prof. Dr.M.A. Mottalib  
Head  
Department of Computer Science and Engineering

### ***Supervisor***

-----  
Mr.Md. Ali-Al-Mamun  
Assistant Professor  
Department of Computer Science and Engineering, IUT

### ***Authors***

-----  
Md. Tamim Islam

Student Id: 084426

-----  
Murshid-Bin-Rahman

Student Id: 084445

**THIS THESIS IS DEDICATED TO OUR PARENTS WHO BROUGHT US INTO THIS WORLD,  
AND WHOSE LOVES REMAIN IN OUR HEART FOREVER.**

## ACKNOWLEDGEMENT

First and foremost, we are beholden to Allah, the Almighty and the Gracious without Whose patronage and blessings this research work would not have been successfully completed. He gave us ardor, buoyancy, power of fortitude and courage and vanquished all the stumbling rigidity that we faced on the way.

The discrepancy between difficult research and an impossible one are often the valuable advice and counsel of those who have worked in the field and tackled similar researchers before.

We owe a debt to our project supervisor **Mr.Md.Ali-Al-Mamun**, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology, Bangladesh for his benefiting guidance and encouragement, astute advice and endless endurance throughout the steps forward to the work without which this project have not been flourishing.

We are thankful to **Mr. Saifuzzaman**, Lecturer, Department of Geography and Environment, Jahangirnagar University, who encouraged us to make a thesis on this subject. He not only provided us with all the geographic data but also with his advice and precious time to make this project a successful one.

We would also like to thank all other esteemed faculty members of CSE Department especially to **Prof.Dr. M.A. Mottalib**, Head of the CSE department, for helping us from time to time.

Last of all our earnest thanks go to our friends and classmates who has provided us support, precious information and different suggestions that helped for successful completion of the project.

## **ABSTRACT**

This is a thesis on a system which will be able to sense the traffic situation of a road; and with its prior knowledge of neural network it will be able to predict the possibility of traffic congestion so that a better decision can be taken by the authority. The number of vehicles coming from residential, industrial or restricted area will also be taken into consideration to get a better response at the time of prediction. With the help of neural network a model for the entire possible scenario of congestion is trained and this learning has helped to provide the output scenario for any given situation. The output of the system will be the percentage occupation of the road or congestion depending on which authority will take the wise decision to remove the congestion. This can also compare the current situation of different roads to learn the percentage of congestion. Then if the information of the traffic situation is made available through internet then general people can get the idea of congestion and avoid the busy roads to select an alternative free one. Government agencies will be able to take decisions like where to construct new roads, install traffic lights etc. if there is authentic information regarding the behavior of traffic in different parts of the day. Gathering of this information is easily possible from our project

### **Keywords:**

Community area pressure, Geographic information system, Neural Network, Back propagation algorithm, training of input and output

## Table of Contents

Chapter 1: Introduction .....	9
1.1 Thesis context .....	9
1.2 Problem Statement .....	9
1.3 Solution Outline .....	10
1.4 Analytical statement of the solution.....	10
Chapter 2: Related Works .....	11
Chapter 3: Background Study .....	15
3.1 Geographic Information System (GIS) .....	15
• Definition of GIS.....	15
• Components of GIS .....	16
• Functions of GIS .....	16
• GIS Data Types .....	17
• Comparison between raster data and vector data:.....	20
• GIS concepts:.....	21
• Representing geographic space:.....	21
3.2 Neural Network.....	23
• Definition of Neural Network.....	23
• Layers in a Neural Network:.....	24
• Supervised Learning of the Data in AI module .....	25
3.3 Back propagation Algorithm .....	25
• Phase 1: Propagation .....	25
• Phase 2: Weight update.....	25
• Modes of learning.....	26
• Algorithm .....	26
• Batch Training (train) .....	27
Chapter 4: Proposed Approach.....	30
4.1 General Framework .....	30
4.2 Satellite image.....	30
4.3 Road information .....	31

4.4 How to count the number of vehicles .....	32
• Radio Frequency Identification (RFID) .....	32
• Global Positioning System (GPS).....	33
• Close Circuit Camera .....	33
• Metallic plate .....	33
• Air tube method.....	33
• Statistics .....	33
4.5 Parameters.....	34
• Number of vehicles currently on the road.....	34
• Community area pressure.....	34
• Vehicles from the previous node .....	34
4.6 Neural network .....	35
• Training of data .....	35
• Back propagation algorithm.....	35
Chapter 5: Implementation & Output of the system.....	36
5.1 Congestion state of different roads .....	36
5.2 With and without community area pressure.....	37
5.3 Working days' average.....	38
Chapter 6: Limitations & Scope of our system .....	40
Chapter 7: Conclusion .....	42
Chapter 8: Bibliography .....	43
Chapter 9: Appendix .....	45

## List of Figures

3.1.1: GIS data	15
3.1.2: Functions of GIS	17
3.1.3: Raster representation	17
3.1.4: Effect of a raster representation	18
3.1.5: Raster Data representation	18
3.1.6: Vector Data	19
3.1.7: Vector data	20
3.1.8: Comparison between raster and vector data	20
3.1.9: GIS Concepts	21
3.1.10: Image from GIS software	22
3.1.11: The image of the buildings	22
3.1.12: vector layer of roads	22
3.2.1: Human Neuron	23
3.2.2: The neuron Model	24
3.3.3: Layers of NN	24
4.1 General Framework of our thesis	30
4.2 A spy satellite named QuickBird	31
4.3.1: Name of the roads	32
4.3.2: Information of the roads	32
5.1 Comparison of congestion among the connected roads	36
5.2 Congestion state of 4 roads	37
5.3: The difference with and without community area pressure	37
5.4: working days congestion state at different times from 7:00am to 12pm	38
5.5: Average congestion state on working days at different time	39
5.6 Average congestion state on vacation or holidays	39



## **Chapter 1: Introduction**

### **1.1 Thesis context**

In our country the traffic signals are time oriented. After a certain period of time they repeat the signals again and again without considering the load on the road from the surrounding areas. Since no load management and calculation are taken into account, so all types of road get congested very easily.

In order to deal with traffic behavior we cannot ignore the reality of surrounding community area pressure. For any residential area the traffic or the vehicles contributed to the main road is not the same as the area with no population or barren land. Again the number of vehicles is not same throughout the day. At morning people usually go to their respective work places, so the vehicles contribute more to the main roads. This road behaves exactly opposite at the evening when people return for their houses after the work.

For this reason we have incorporated the new idea of community area pressure at the time of forecasting the traffic model. Then with the help of neural network we have trained our system with various input and their corresponding output. Based on this knowledge our system will be able to generate output which will suggest the actual condition of the road so that effective measure can be taken to reduce the congestion. Again this knowledge of traffic behavior will help the traffic surgeons to take better decision, the drivers to select best route with less congestion and the government to take major decisions like where to construct new roads and how to improve the overall traffic condition of the country.

### **1.2 Problem Statement**

We want to sense the number of vehicles on the road through any of the available medium and then with the help of our system we will generate an output showing the different state of the road; for example busy, congested, less busy etc. Then this knowledge will help the traffic police to take the decisions like which roads need longer time of green signal to avoid the unwanted congestion.

### **1.3 Solution Outline**

In order to get the road information we have used the ArcGIS and Google Earth software. Then we have decided some general conventions and three parameters contributing to the congestion of the road. These are the

1. Number of vehicles currently on the road
2. Community area pressure and
3. Vehicles coming from the previous node.

We can make this information dynamic if we can apply any of the recent technologies like RFID counter, Global Positioning System, image processing of the close circuit camera etc. As soon as we get all the number of vehicles of these parameters, then we will put them in the neural network where we have formed a model after much training of data. This physical model from its previous knowledge will suggest us about the current situation of the roads and show the current congestion state as its output. Depending on this output the concerned people will take efficient and further decisions.

### **1.4 Analytical statement of the solution**

After getting the count of vehicles of the three parameters we will take it as the input to our system. These input values will go to the neural network where an equation was formed after the training of data. The equation will now read the input values and show the corresponding output to the user. They will show the output in terms of bar graph so that the user can easily understand the current situation of the road.

We can also store all this information for a week or month so that we can understand the general behavior of the road in different times as well as days. This information will be helpful to take further development decisions.

## Chapter 2: Related Works

Traffic congestion is very severe and growing problem in many urban areas across the world like our country. In the near future the traffic congestion problem is likely to worsen due to automobile and structural trends both in developed and developing country. Moreover everyday more than thousand cars are putted down into the road. So it creates traffic congestion. The roads are not constructed with specific plan and proper criteria. Based on comparison of the increasing number of vehicle roads are not increasing. Therefore again it influences to increase the traffic congestion. In public sector planning using the transportation tools is based on the static flow of network and travel cost planning. Network flows reach a steady state equilibrium based on this concept most transportation and land use forecasting tools.

There are several parameters for traffic congestion. First of all we tried to identify the major influencing entities for traffic congestion and then modeled it for improving the condition. Secondary data are collected in many ways from the authenticated source, say (some selected road junction data are collected), which will be processed and analyzed. How the data would be influenced by the GIS software or what will be the estimated solutions within this data. In Nigeria one paper was processed based on this thing where two things were checked. The first way is the provision of traffic information that enables commuters and motorists to take rational decisions as to which route to take during peak hour travel. The second is the determination of appropriate queries that can evoke graphical response, which could be used to manage traffic congestions.

### **Statement of the Problems:**

In a highly urbanized environment, the automobile is a significant contributor to environmental pollution. The extent of pollution depends on certain variables such as the age of the vehicle, the type of the vehicles and the quality of the fuel used. Some of the effects of pollutants include respiratory diseases, caused by oxides of Nitrogen, and aggravation of asthma on those who are already afflicted with this disease. Apart from this, carbon monoxide,

which is a dangerous pollutant from automobile exhaust, affects blood and prevents easy flow of oxygen from getting to the brain, the heart and other tissues.

Illegal parking is also a major problem in urban environment. This is because parking on roadside, which is a common phenomenon, reduces the traffic corridors meant for the efficient movement of automobiles. Thus, it becomes a major problem in cities and especially in the Central Business District (CBD), where multi-storey buildings are common and the land use is devoted mostly to commercial purposes. The resultant effect of such illegal parking, therefore, is traffic congestion. This illegal parking leads to delay in traveling time and increases the cost of traveling because more fuel is used up in the process of accomplishing a delayed journey [8].

Why GIS software is so mandatory for this type of research which is found from this paper. It is mentioned that GIS is not only veritable tools but also provide that it is properly build on a designed database. Again, most of the times the transport administrator searches for something which can enhance the transportation supply in accordance with the environmental condition. A decision support system integrated in Geographical Information System (GIS) is just like that for the analysis and evaluation of different transport policies so that it enhances the efficiency of administration. This DSS (Decision Support System) works on three levels such as the first one performs the network analysis the second assesses the energy consumption and pollutant emissions and the third level evaluates the several policies selected. The model allows the estimation of traffic flow patterns within each link of the road network starting from the knowledge of the network characteristics and traffic demand.

Geographical information system (GIS) technology offers extremely significant power in transport modeling. The spread of GIS use facilitates the efficient and portable spatial data storage, updating and processing. In addition, a GIS system facilitates model accessibility, database maintenance and updating, and cartographic display of model results. This can greatly enhance the role of the transport model as a decision support system (DSS) in transportation planning and policy development. The interface of network-based equilibrium models with a GIS platform offers substantial potential of modeling transportation planning, analysis and control. GIS systems can greatly improve the realistic representation of the multi-modal

transportation network, increased likelihood of database integrity relative to the traditional multi-modal network data model, effective user-interfaces and efficient visualization of network equilibrium solutions.

In this work, a computer-integrated tool for evaluating urban transportation policies is presented. The tool provides estimates of road traffic and evaluates the implications of urban traffic policies. The tool is a prototype GIS-based DSS that involves realistic representation of the multimodal transportation network. The uniqueness of the tool lies in combining transport network and travel demand database management, GIS utilization for policy definition and result presentation, traffic simulation and analysis, energy consumption and pollutant emission modeling, evaluation of environmental impacts and scenario comparison into a seamlessly integrated package. While neither of these elements is new, it is the tight integration that provides the new approach and qualifies the tool as a user-friendly decision support tool. In the following sections of the paper the basic structure of the tool is presented, its operation is outlined and characteristic case studies for the Greater Athens Area in Greece demonstrate the effectiveness of the proposed approach.

The Greater Athens Area is the target region used in case studies of the DSS tool described. This area consists of 83 local authority areas, while the integrated urban character of the region is ensured by the existence of a common/shared infrastructure (public transport system, road network, etc.) connecting all its parts as well as a common administrative framework. The urban activities of this region gravitate towards the centers of the cities of Athens and Piraeus, as well as in the centers of a small number of regional centers clustering the two main centers. The Municipality of Athens is the central part of the region [9].

The parameter of each policy is based on traffic, environmental and energy indicators. This model serves as the repository of the data as well as the user interface of the tool which is integrated in GIS environment. Moreover, Government authority does their work traditionally in most of the countries. Besides the urban traffic congestion is increasing and becomes complex day by day. After all, traditional static process is not so adequate for analyzing road

network flows and searching for minimum cost route. For analyzing or modeling the road networks flows and finding the minimum cost routing one paper is processed based on GIS-based decision support tool. In accordance with this paper, temporal resolutions, capturing dynamic congestion propagation effects are forecasted by this system.

The urbanization is another factor for traffic congestion in developing countries. It is related with some features like population, increasing number of coming people in the urban cities, enough vehicle condition and so on. Ultimately it makes a gathering situation. With taking the help of Geographical Information System and Global Positioning System some guys are trying to mitigate this traffic situation. Within this paper they were done their work finding some parameter and then modeled the system. Parameter are like travel time data, counting the travel speed, road segmentation, data segmentation etc.

## Chapter 3: Background Study

### 3.1 Geographic Information System (GIS)

- **Definition of GIS**

Digital geographic data are numerical representations that describe real-world features and phenomena, coded in specific ways in support of GIS & mapping applications using computer. For geographic data to be useful, they must be encoded in digital form and organized as a geographic database. This allows a range of functions for storing, processing analyzing and visualizing the spatial data whereas conventional paper map represents a general- purpose snapshot of the real world at the given time only. Thus geographic database is dynamic, rather than static.

GIS is a computer-based tool for solving real-world problems. A GIS integrates information in such a way that it helps us to understand and find solutions to problems. The objects are stored in a database and dynamically linked to an onscreen map, which displays graphics representing real-world objects.

A **geographic information system** is designed to capture, store, manipulate, analyze, manage, and present all types of geographical data.

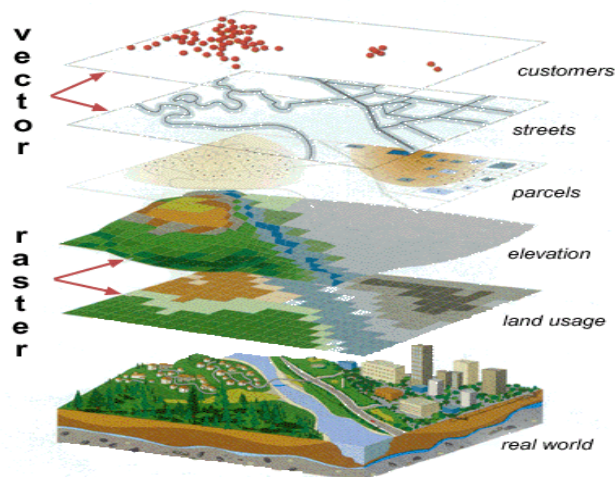


Figure 3.1.1: GIS data

- **Components of GIS**

- 1) Trained manpower- Computer engineer
  - Database expert
  - Spatial planner
- 2) Data - Maps, images
  - Attribute data,
  - Thematic Maps
  - Satellite Images
- 3) Hardware - Like any other hardware
  - with some extra component
  - (big printer, digitizer...)
- 4) Software -Storing data
  - Analysis tools
  - Data capture
  - Data editing, manipulation & processing
- 5) Procedures- Database development and analysis
- 6) Presentation- On paper map, digital

- **Functions of GIS**

The GIS is capable of finding solutions to real-world problems. A GIS has the following capabilities:

- 1) Data Capture
- 2) Store Data
- 3) Data Manipulation and Editing
- 4) Database Management
- 5) Data Integration and Modeling
- 6) Database Query and Analysis
- 7) Customization and Tools Development



## 8) Output Preparation

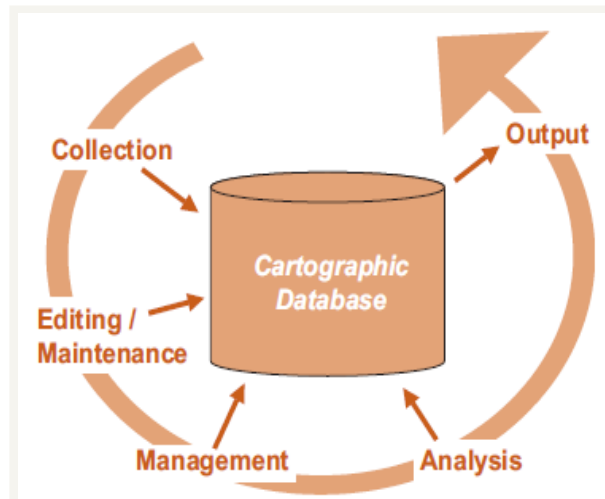


Figure 3.1.2: Functions of GIS

- **GIS Data Types**

The GIS data types can be divided into 2 main categories.

- 1) **Raster data:** Raster representations divide the world into arrays of cells and assign attributes to the cells.

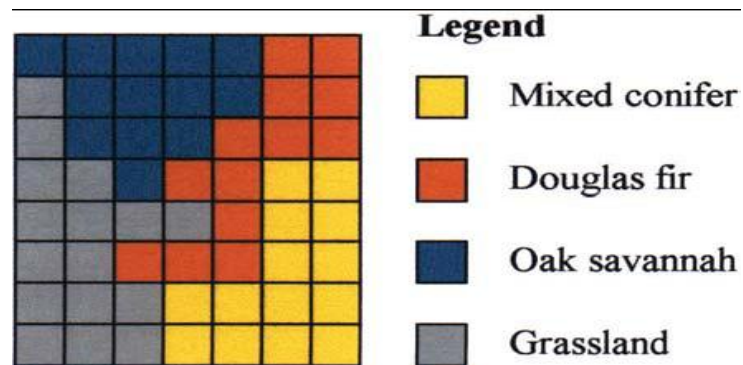
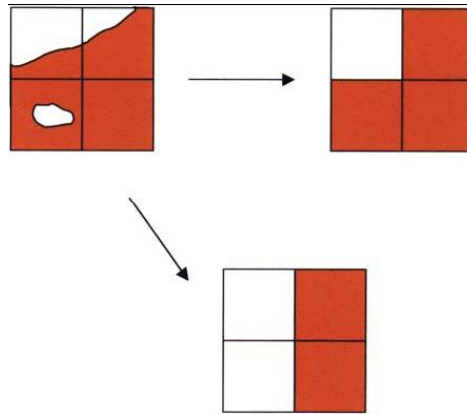


Figure 3.1.3: Raster representation.

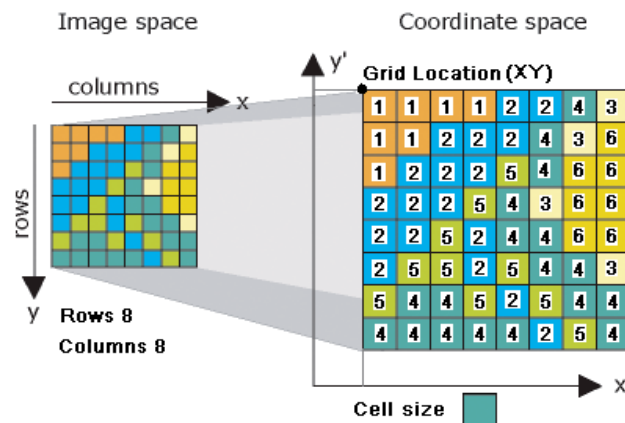
Raster image comes in the form of individual pixels, and each spatial location or resolution element has a pixel associated where the pixel value indicates the attribute, such as color, elevation, or an ID number.

Raster representations divide the world into arrays of cells and assign attributes to the cells. Raster graphics are resolution dependent. They cannot scale up to an arbitrary resolution without loss of apparent quality.



**Figure 3.1.4:** Effect of a raster representation using (A) the largest share rule and (B) the central point rule

In a raster representation space is divided into an array of rectangular (usually square) cells. All geographic variation is then expressed by assigning properties or attributes to these cells. The cells are sometimes called pixels.



List of cell values

[11112243112224361222546622254366225244662552544354452544444254]

**Figure 3.1.5:** Raster Data representation

**2) Vector Data:** Vector data are represented by means of coordinates. By using one or more pairs of coordinates, spatial objects are identified and represented by one of the three forms of basic geographical elements:

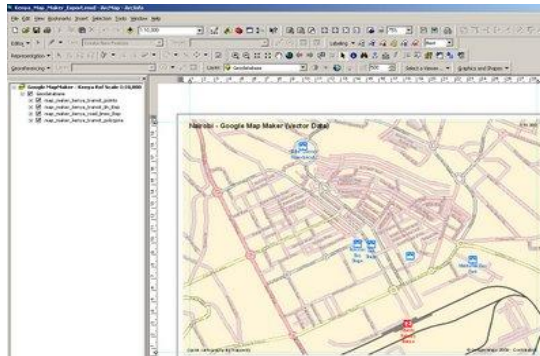
- (1) Points: represented as single pairs of coordinates, are the simplest type of vector data.
- (2) Lines or arcs: represented as the string of coordinates, begin and end with a node.
- (3) Polygons or areas: represented as a closed loop of coordinates.

In a vector representation, all lines are captured as points connected by precisely straight lines (some GIS software allows points to be connected by curves rather than straight lines, but in most cases curves have to be approximated by increasing the density of points). An area is captured as a series of points or vertices connected by straight lines as shown in the following figure. The straight edges between vertices explain why areas in vector representation are often called polygons.



Figure 3.1.6: Vector Data

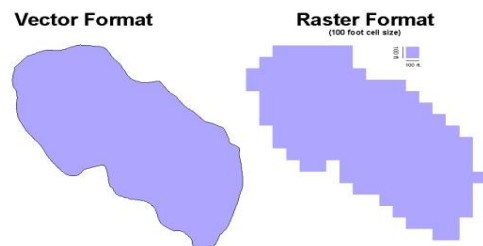
- \* Vector data means the representation of the real world objects in the form of points, lines & polygons.
- \* To analyze a map and save the records in a database vector data is more useful.
- \* Points are stored using the coordinates, for example, a two-dimensional point is stored as  $(x, y)$ . Lines are stored as a series of point pairs, where each pair represents a straight line segment, for example,  $(x_1, y_1)$  and  $(x_2, y_2)$  indicating a line from  $(x_1, y_1)$  to  $(x_2, y_2)$ .



**Figure 3.1.7: Vector data**

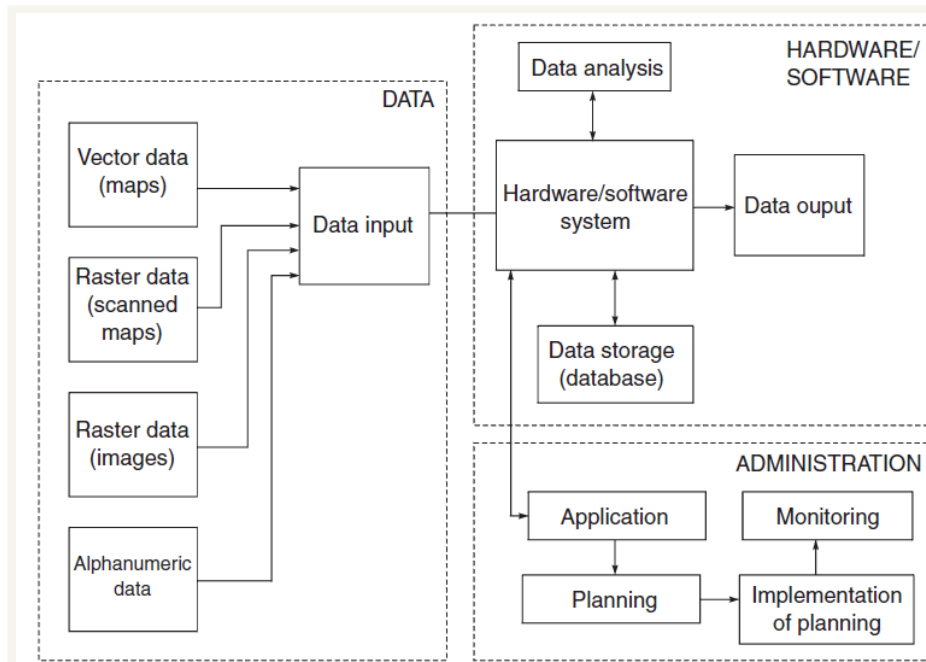
- **Comparison between raster data and vector data:**

- Vector data structure produces smaller file size than raster image because a raster image needs space for all pixels while only point coordinates are stored in vector representation.
- When the graphics or images have large homogenous regions and the boundaries and shapes are the primary interest then Vector Data is more useful.
- Vector data is easier to handle than raster data on a computer because it has fewer data items and it is more flexible to be adjusted for different scale.
- Topology among the graphical objects or items is much easier to be represented using vector form, since a commonly shared edge can be easily defined according to its left and right side polygons. On the other hand, this is almost impossible or very difficult to do with pixels.



**Figure 3.1.8: Comparison between raster and vector data**

- **GIS concepts:**



**Figure 3.1.9: GIS Concepts**

- **Representing geographic space:**

Real-world features exist in two basic forms-

- 1) **Objects:** Objects are discrete and definite, such as buildings, highways, cities, parks etc. They are represented in the geographic database as the Object-based Model.
- 2) **Phenomena:** Phenomena are distributed continuously over a large area such as terrain, temperature, rainfall, noise level and other environmental indices. They are represented in the geographic database as the Field- based Model.

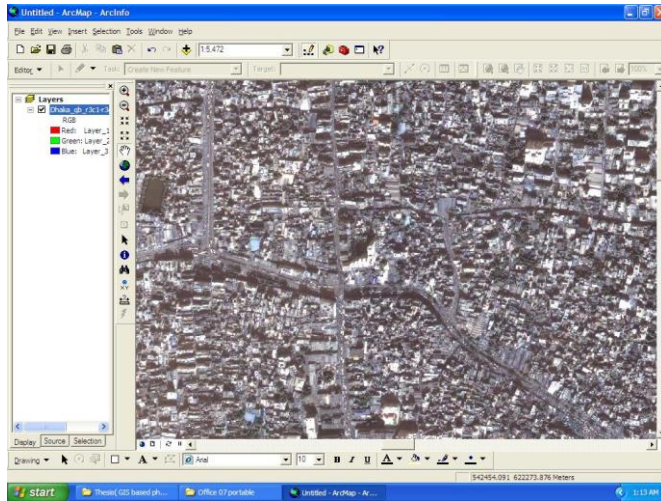


Figure 3.1.10: Image from GIS software

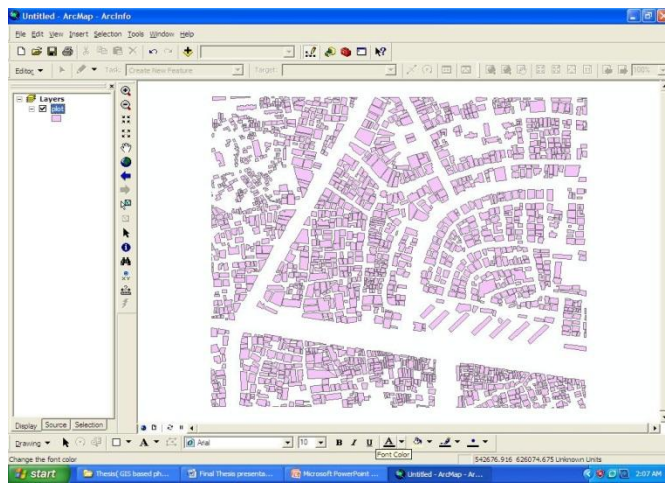


Figure 3.1.11: The image of the buildings

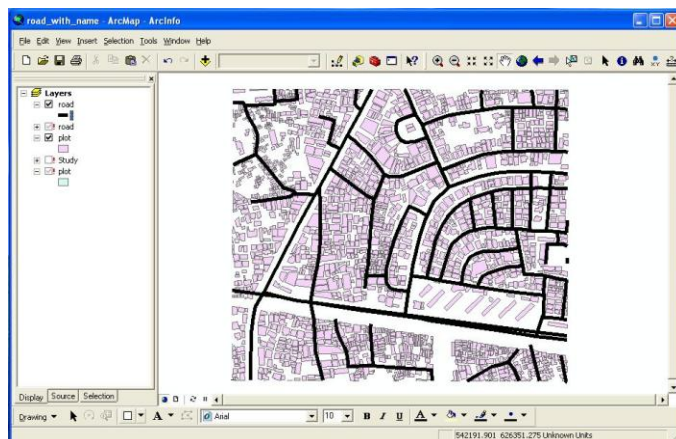


Figure 3.1.12: vector layer of roads

## 3.2 Neural Network

- **Definition of Neural Network**

An artificial neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. Artificial neurons were first proposed in 1943 by Warren McCulloch, a neurophysiologist, and Walter Pitts, a logician, who first collaborated at the University of Chicago.

In a neural network model simple nodes (which can be called by a number of names, including "neurons", "neuroses", "Processing Elements" (PE) and "units"), are connected together to form a network of nodes — hence the term "neural network". While a neural network does not have to be adaptive *per se*, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.

The term **neural network** was traditionally used to refer to a network or circuit of biological neurons. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes.

Thus the term has two distinct usages:

1. Biological neural networks
2. Artificial neural networks

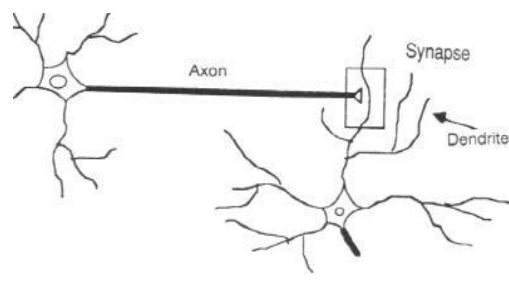


Figure 3.2.1: Human Neuron

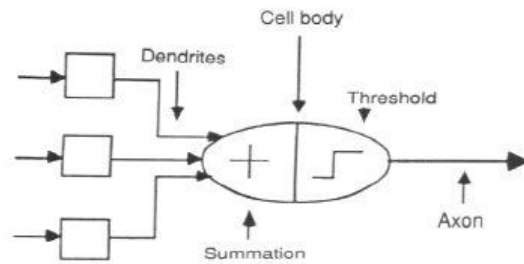


Figure 3.2.2: The neuron Model

- **Layers in a Neural Network:**

The most common type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of "**output**" units. The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

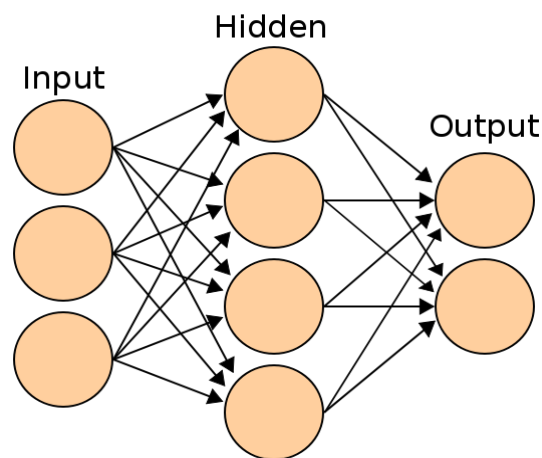


Figure 3.3.3: Layers of NN



- **Supervised Learning of the Data in AI module**

Neural Network is an Artificial Intelligence module used for forecasting, process control, research, data validation, risk management etc. The Road Data, Plot Data & Vehicle Data will be given to the Neural Network for training purpose. The given input and output will be trained in the neural network in such a way that they will generate a weight value within themselves. After that they will form an equation given the weight values. The new equation will be based on the trained value so putting the test value on the equation we will be able to get a solution. Based on this solution we can take further decisions.

The training of data takes 90% of its sample values for training purpose. Then 5% of samples are there for validation and 5% of the samples are for the testing purpose. Based on this samples an equation is formed in the neural network.

### **3.3 Back propagation Algorithm**

For better understanding, the back propagation learning algorithm can be divided into two phases: propagation and weight update.

- **Phase 1: Propagation**

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

- **Phase 2: Weight update**

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the *learning rate*. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

- **Modes of learning**

There are two modes of learning to choose from: One is on-line (incremental) learning and the other is batch learning. In on-line (incremental) learning, each propagation is followed immediately by a weight update. In batch learning, much propagation occurs before weight updating occurs. Batch learning requires more memory capacity, but on-line learning requires more updates.

- **Algorithm**

Actual algorithm for a 3-layer network (only one hidden layer):

```
Initialize the weights in the network (often randomly)
Do
  For each example e in the training set
    O = neural-net-output(network, e) ; forward pass
    T = teacher output for e
    Calculate error (T - O) at the output units
    Compute delta_wh for all weights from hidden layer to output layer; backward pass
    Compute delta_wi for all weights from input layer to hidden layer; backward pass
  continued
  Update the weights in the network
Until all examples classified correctly or stopping criterion satisfied
Return the network
```

As the algorithm's name implies, the errors propagate backwards from the output nodes to the inner nodes. Technically speaking, back propagation calculates the gradient of the error of the network regarding the network's modifiable weights. This gradient is almost always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "back propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent. Back propagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

Back propagation networks are necessarily multilayer perceptron (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer network logistic, the Softmax function, and the Gaussian function.

The back propagation algorithm for calculating a gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode.

s must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Non-linear activation functions that are commonly used include the

There are many variations of the back propagation algorithm. The simplest implementation of back propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written

$$x_{k+1} = x_k - a_k g_k$$

Where,  $x_k$  is a vector of current weights and biases,  $g_k$  is the current gradient, and  $a_k$  is the learning rate.

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode.

In incremental mode, the gradient is computed and the weights are updated after each input is applied to the network.

In batch mode, all the inputs are applied to the network before the weights are updated. The next section describes the batch mode of training; incremental training is discussed in a later chapter.

- **Batch Training (train)**

In batch mode the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases.

### **Batch Gradient Descent (traingd)**

The batch steepest descent training function is `traingd`. The weights and biases are updated in the direction of the negative gradient of the performance function. If you want to train a network using batch steepest descent, you should set the network `trainFcn` to `traingd`, and then call the function `train`. There is only one training function associated with a given network.

There are seven training parameters associated with `traingd`:

- Epochs
- Show
- Goal
- Time
- `min_grad`
- `max_fail`
- `lr`

The learning rate `lr` is multiplied times the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge.

The training status is displayed for every `show` iterations of the algorithm. The other parameters determine when the training stops. The training stops if the number of iterations exceeds `epochs`, if the performance function drops below `goal`, if the magnitude of the gradient is less than `mingrad`, or if the training time is longer than `time` seconds. `max_fail`, which is associated with the early stopping technique, is discussed in [Improving Generalization](#).

The following code creates a training set of inputs `p` and targets `t`. For batch training, all the input vectors are placed in one matrix.

```
p = [-1 -1 2 2; 0 5 0 5];
```

```
t = [-1 -1 1 1];
```

Create the feedforward network.

```
net = newff(p,t,3,{'traingd'});
```

In this simple example, turn off a feature that is introduced later.

```
net.divideFcn = '';
```

At this point, you might want to modify some of the default training parameters.

```
net.trainParam.show = 50;
```

```
net.trainParam.lr = 0.05;
```

```
net.trainParam.epochs = 300;
```

```
net.trainParam.goal = 1e-5;
```

If anyone wants to use the default training parameters, the preceding commands are not necessary.

Now we are ready to train the network.

```
[net,tr]=train(net,p,t);
```

The training record `tr` contains information about the progress of training.

Now we can simulate the trained network to obtain its response to the inputs in the training set.

```
a = sim(net,p)
```

```
a = -1.0026 -0.9962 1.0010 0.9960
```

### Gradient Descent with Momentum

In addition to `traingd`, there are three other variations of gradient descent.

Gradient descent with momentum, implemented by `traingdm`, allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a lowpass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum. With momentum a network can slide through such a minimum.

## Chapter 4: Proposed Approach

### 4.1 General Framework

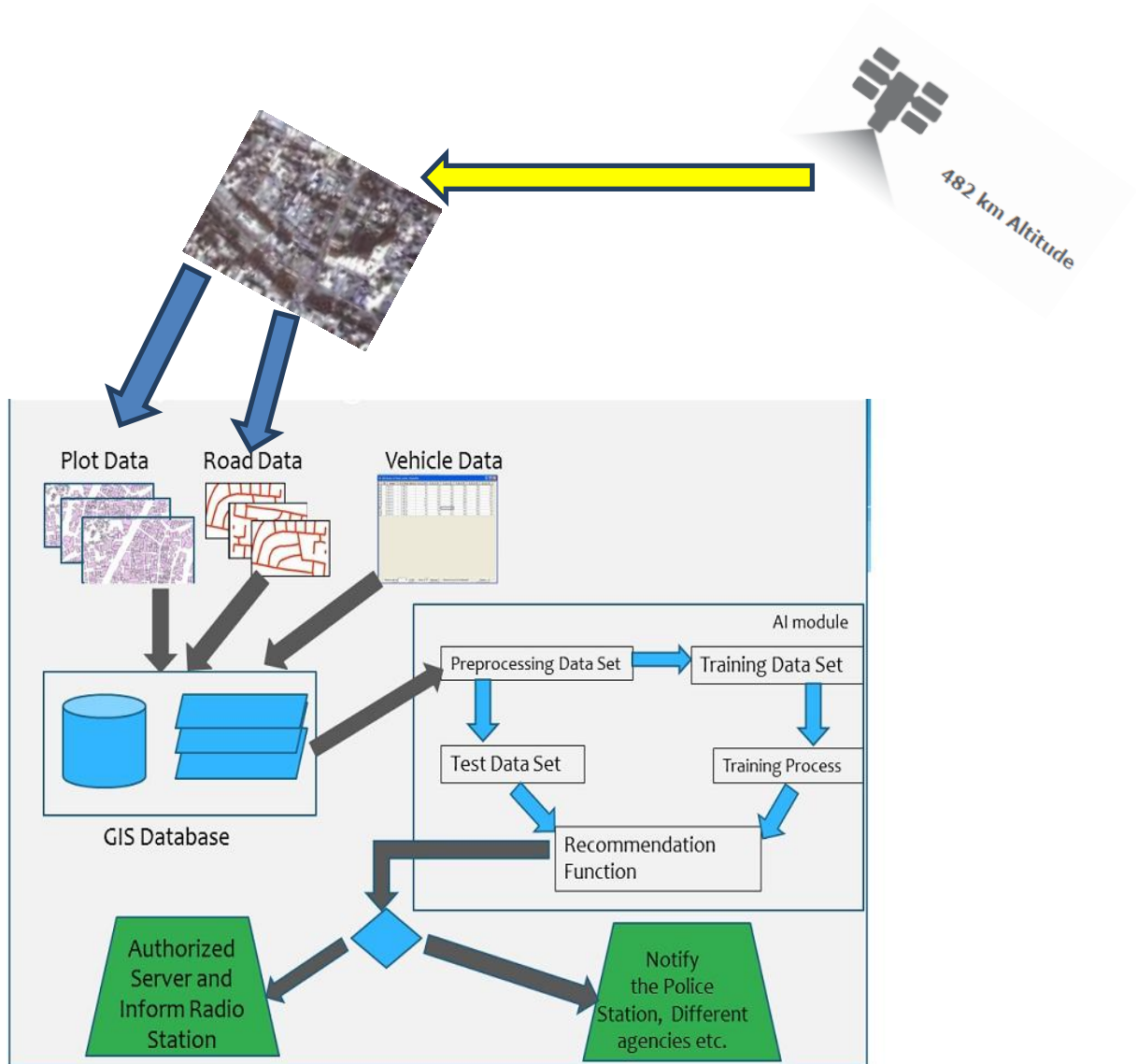


Figure 4.1 General Framework of our thesis

### 4.2 Satellite image

- Captured by: QuickBird Spy Satellite USA.

- This satellite can capture image with accuracy of 15-25 cm (for defense purpose). For commercial purpose they provide accuracy of not less than 50 cm.
- “Digital Globe” commercially provides this image at the cost of 21 dollar per sq. km.
- For study purpose we have collected this image from Jahangirnagar University. This image has an accuracy level of 60 sq. cm per pixel.
- Size of the image a portion of Dhaka city around 2.2 GB.



QuickBird clean room pre-launch preparations. The first of DigitalGlobe’s state-of-the-art high-resolution commercial imagery satellites.

**Figure 4.2 A spy satellite named QuickBird**

### 4.3 Road information

We have selected a traffic control node of Dhaka city randomly. This node is situated at  $23^{\circ}45'51.73''N$  and  $90^{\circ}23'20.45''E$ . We have taken the four roads as North road, East road, South road and West road in the clockwise direction as shown in the figure below.

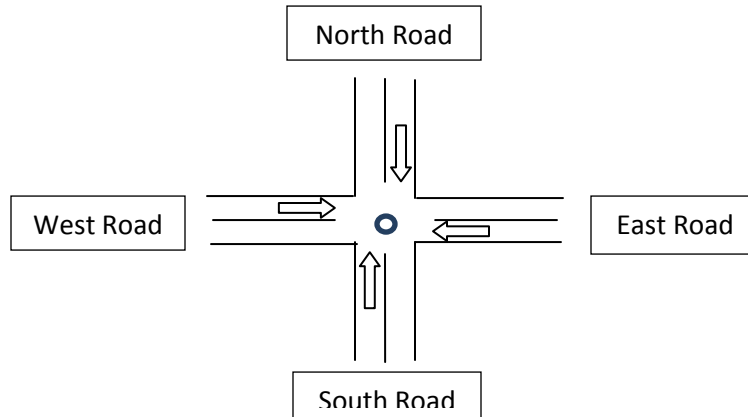


Figure 4.3.1: Name of the roads

Then we have measured the length of the roads from its previous node or the traffic signal to get the following information. Then taking the approximation that 5 meters are occupied by a vehicle we found out the highest number of vehicles that can reside on this road. The table is shown below.

	North Road	South Road	East Road	West road
Length	370 m	550 m	1000 m	532 m
Number of lanes	4	4	2	4
Maximum number of vehicles	296	440	400	425

Figure 4.3.2: Information of the roads.

#### 4.4 How to count the number of vehicles

There are many different ways to keep track of the number of vehicles in a road. Few of them are as follows.

- **Radio Frequency Identification (RFID)**

**Radio-frequency identification (RFID)** is the use of a wireless non-contact system that uses radio-frequency electromagnetic fields to transfer data from a tag attached to an object, for the purposes of automatic identification and tracking. The tag contains electronically stored information which can be read from up to several meters (yards) away. Unlike a bar code, the tag does not need to be within line of sight of the reader and may be embedded in the tracked object. This is the method where an access point will remain connected with the server. There are some RFID tags which remain connected with the moving objects. Whenever the object comes closer to the RFID access point some information is sent to the server. This information helps in computation of the vehicles.



So if we can keep an access point at the beginning of the road then this will contain the number of vehicles getting in to the road. Again the access point will tell us the number of vehicles going out from the road. Subtracting these numbers we can easily get the number of vehicles currently present on the road.

- **Global Positioning System (GPS)**

Global positioning system devices have a direct connection with the satellite. With this device we can find the exact location on the earth. If we want to use GPRS for counting the number of vehicles then a device needs to be connected to every vehicle which will send information to the cloud. Then from the cloud one can get the idea about the number of vehicles currently residing on the road.

- **Close Circuit Camera**

A close circuit camera is connected to every end of the roads. In this camera one can see the picture of the situation of the road. With some further enhancement one can use the technology of Image Processing to isolate the vehicles and count the number of them. In this way one can get the count of vehicles. But this technology requires a lot of calculation & processing.

- **Metallic plate**

This is a modern technology introduced in our country. Here a metallic plate of sensors remain laid down on the road. Each time a wheel crosses the metallic bar, its counter is incremented to one. In this way the number of vehicles crossing the sensor can give the vehicle count.

- **Air tube method**

This is the oldest method of automatic traffic counting. The tube is hollow and sealed on one end, with the other leading to a box off the side of the road. Every time the tube is compressed by a wheel, the pulse of air generated is registered by a pressure switch.

- **Statistics**

Another source of car count can be got from the people of statistics department. They go directly to the location and count the number of vehicles manually on demand. Then they can provide us with their statistical output. It is done by many statistical research organization and Academic Institutions.

Besides all these there may be other methods and technologies which can give the number of vehicles count on a road dynamically. For our thesis we have taken the car count from the statistics department.

## 4.5 Parameters

We have taken three parameters to get the idea of the road condition. This can be increased or decreased as per the situation of the road and behavior of the traffic.

- **Number of vehicles currently on the road**

This is a number which indicates the number of vehicles that are staying on the road due to signal or congestion. After a red signal is given, many cars come & wait for the green signal. These numbers of vehicles occupy the road from the beginning.

- **Community area pressure**

Community area pressure gives the number of vehicles which will contribute to the traffic congestion from the nearby and surrounding residential, industrial, commercial etc. type of area. In few minutes of time the people coming out from their living area are taken into consideration so that the exact traffic condition is understood.

An area with many higher storied buildings will contribute more vehicle and pressure on the road. Then again a densely populated area will have higher contribution to the road than a less densely populated area. Last but not the least, the area that has only one connecting road will generate more pressure than the area having more than one connecting road.

- **Vehicles from the previous node**

This is another number representing the count of vehicles coming from the last traffic signal. These vehicles will fill up the road and create congestion after some time. So this parameter is also taken in to consideration so that accurate vehicular information is found.

## 4.6 Neural network

- **Training of data**

After getting the road information and the vehicular information we have trained our model the input and output scenario of the roads. We have taken many samples to train the model in Matlab. Depending on this prior knowledge the system can tell the user about the current situation of the roads. For our model we have kept the training data on the text files like North\_road\_data, East road data, South road data & West road data.

Then we have used some neural network functions using “nftool” in Matlab for our training purpose. Few of them are like

```
newfit(east_road_input,east_road_target,numHiddenNeurons)
```

```
train(net_east,east_road_input,east_road_target);
```

```
sim(net_east,x_east);
```

For the implementation purpose we have concentrated on the fact that if we can create the functions, train with the practical input and generate the output for only one node then we can extend that for the rest of the nodes.

- **Back propagation algorithm**

The algorithm we have used for our system modeling is the Marquardt-Levenberg back propagation algorithm of neural network. This algorithm can predict the output from a well-trained input condition. In mathematics and computing, the **Levenberg–Marquardt algorithm (LMA)**, also known as the **damped least-squares (DLS)** method, provides a numerical solution to the problem of minimizing a function, generally nonlinear, over a space of parameters of the function. These minimization problems arise especially in least squares curve fitting and nonlinear programming. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach.

## Chapter 5: Implementation & Output of the system

### 5.1 Congestion state of different roads

From the training of the situation in the neural network we have created a new equation. Now if we give any input in this equation then we can get the desired output situation. If our system is given an input with the parameters mentioned above then this generates an output showing the percentage of road occupied or filled up with vehicles. After getting the percentage of all the roads of a node we can easily take a decision about which roads need green signal for longer times. This helps to give priority to the busier roads than comparatively less busy roads.

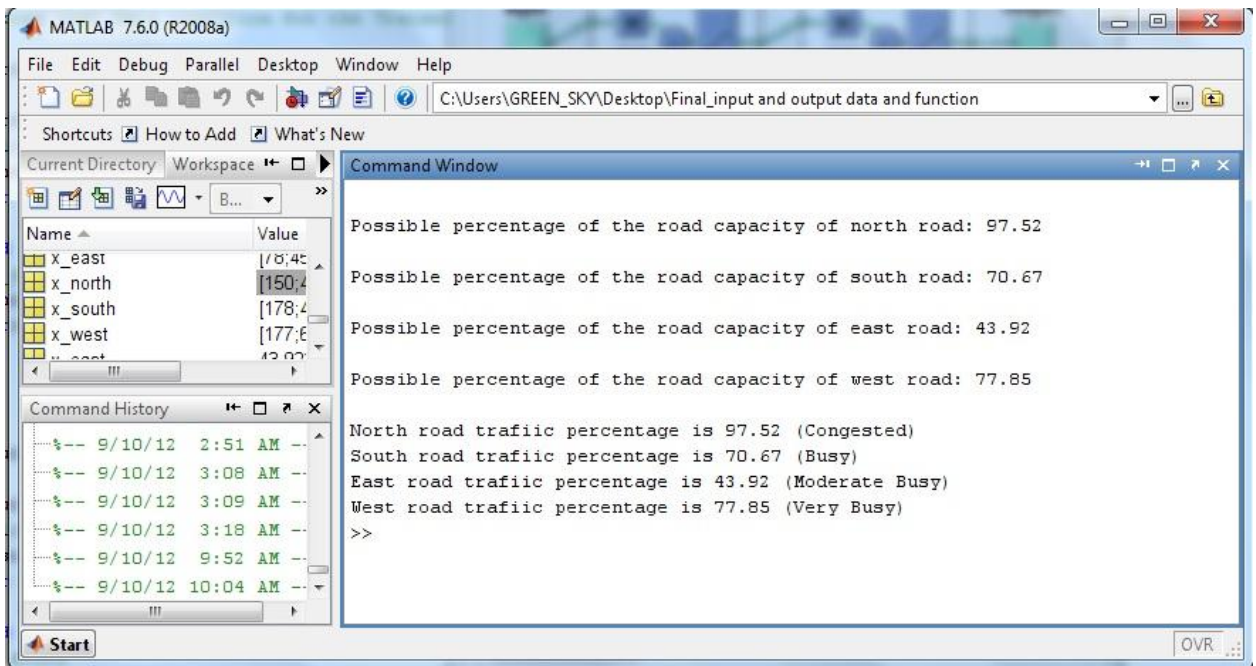


Figure 5.1 Comparison of congestion among the connected roads

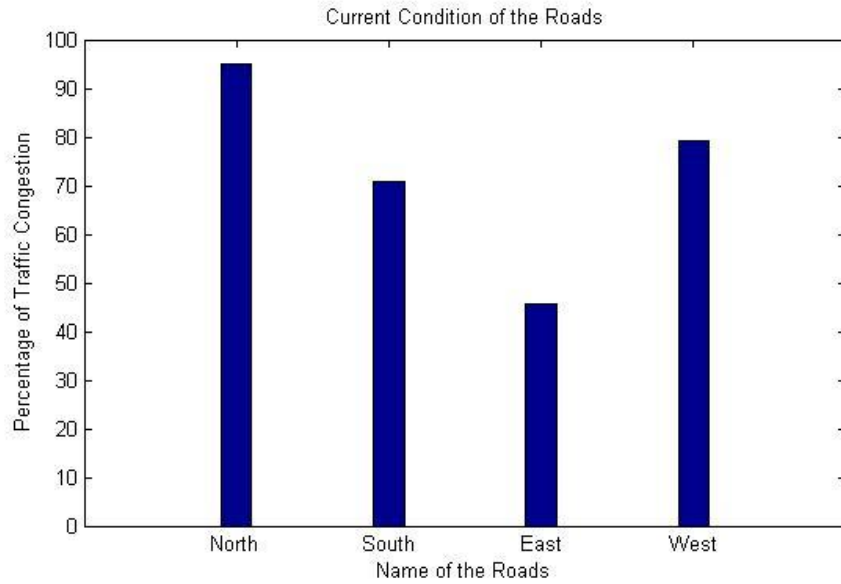


Figure 5.2 Congestion state of 4 roads

## 5.2 With and without community area pressure

This figure shows the effect of community area pressure in the occurrence of traffic congestion with and without community area pressure analysis. This shows that for a better approximation we must take the community pressure into account as they play a vital role in traffic congestion formation. So for a better approximation of the road congestion state we cannot ignore the influence of community area pressure.

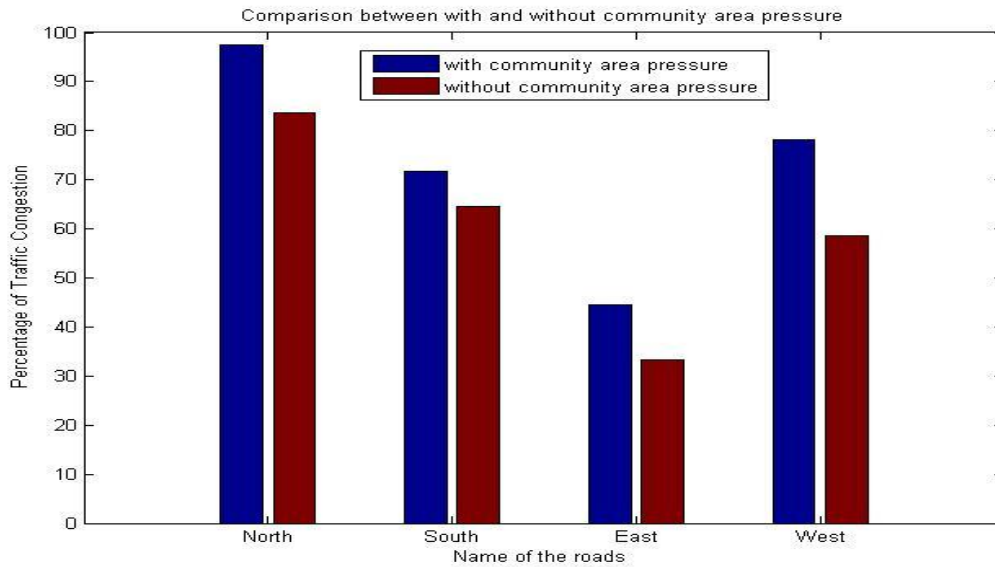


Figure 5.3: The difference with and without community area pressure

**5.3 Working days' average:**the result shows us the percentage congestion of the north road by the vehicles at different times of working days of a week.

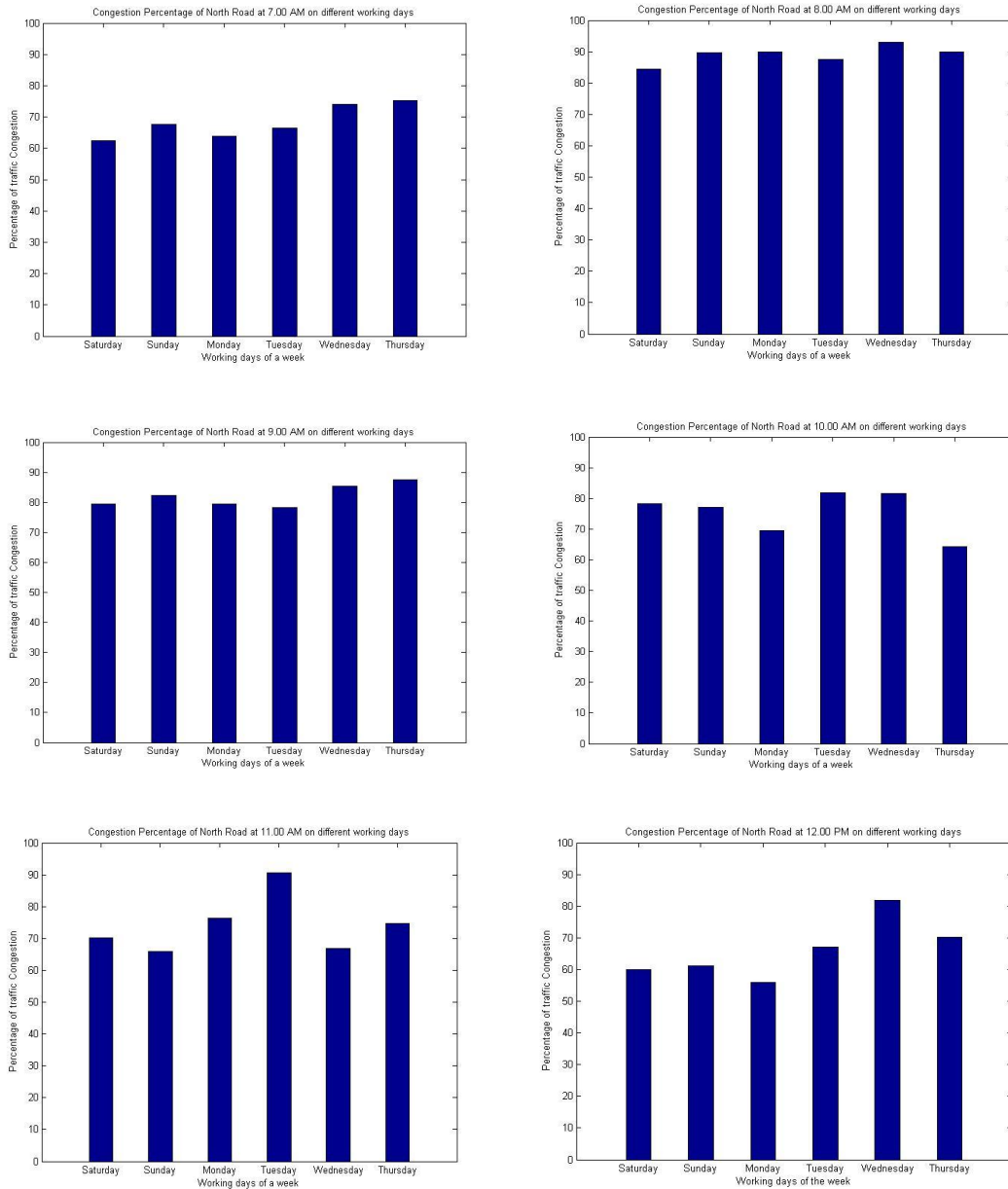


Figure 5.4: Working days congestion state at different times from 7am to 12pm

From this information we can find the average congestion state of a particular road in different time of a day. After getting the pressure on the road given by the vehicles in few days' time we can get the average congestion scenario of the road. Then depending on this result further decisions can be taken.

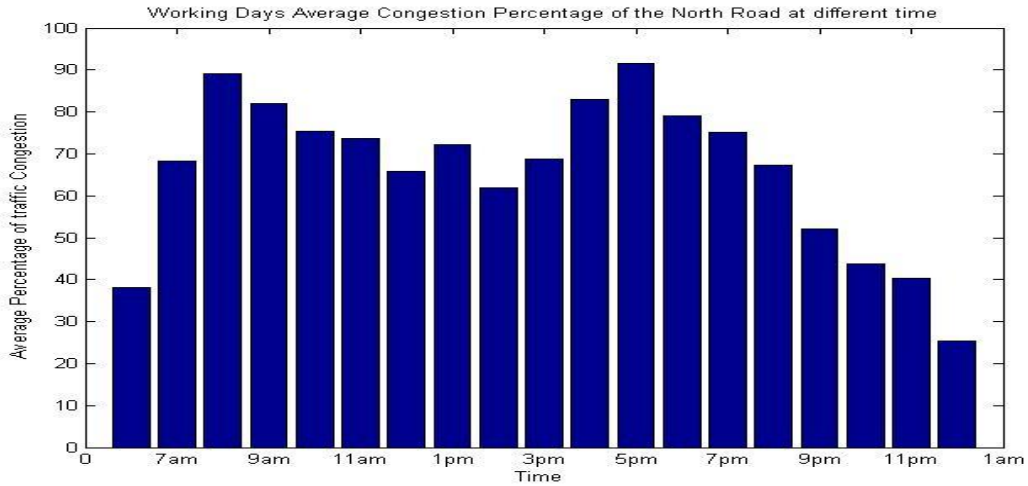


Figure 5.3: Average congestion state on working days at different time

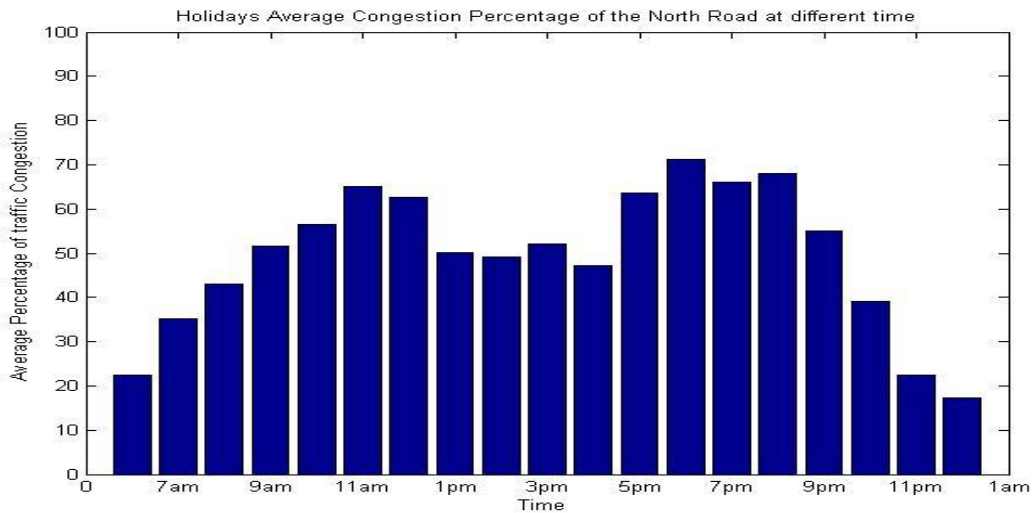


Figure 5.4 Average congestion states on vacation or holidays.

At the time of national holidays, different festivals or for some unwanted accidents the traffic behavior may change. Though the days are like the working days but the number of vehicles is reduced dramatically. In such cases they should be considered as the weekends. Though our system will work perfectly on these complex situations but this may provide a wrong impression about the roads and behavior of the traffic if not considered as weekend. If this is kept in mind during the formation of database, then the information will not mislead and give the more accurate picture of the roads.

We can find the congestion states of other roads in the similar way.

## Chapter 6: Limitations & Scope of our system

Highest efficiency from this system can be achieved if we can make the system dynamic. Due to the unavailability of necessary infrastructure or technological support we could not make it dynamic. As a result there are some errors in our approximation. We had to use some data which are not precise or accurate. Unless we can use any of the dynamic vehicle counting method this data will not be accurate either.

Another limitation is regarding the parameters we have chosen as the cause of traffic congestion. We have taken three parameters namely, Number of vehicles currently on the road, Community area pressure, Vehicles from the previous node. We have taken these parameters by our assumption that they are responsible for the traffic congestion. But there may be more parameters responsible which we have ignored.

There are many ways by which we can enhance the effectiveness of our system. Few of them can be as the following.

- From the output of our system we can observe the traffic behavior of different roads in different time of the day. With this information in hand the government can take major decisions like construction of new roads, installation of traffic signals, traffic signal modifications, guard rail installations, flashing beacon installations and bike and pedestrian projects. The Department of Transportation also provides state and local transportation agencies with traffic count information for congestion management and highway performance monitoring.
- We can store the traffic congestion behavior in the cloud so that any person can get the information for his need. For project purpose or for personal use one will be able to search the cloud to get accurate information about the roads and traffic. This will be helpful for the people to select an alternative road if the information is available.
- Government agencies like BRTA can take care of the information by preserving them in a database so that they can take efficient decisions with very dependable data at hand.



- We can send this information time to time to any radio station so that they can broadcast the current congestion situation of the roads. It can also be made information by demand so that any one asking through SMS can get the current update of the roads.
- Another important enhancement can be achieved if we can show the traffic behavior of the roads to the drivers by electronic message boards installed beside the lanes. Watching the current condition in those boards the drivers will be able to choose the optimal path of their destination.
- Intelligent traffic signal lights with variable length of time span can be made with this system. This type of signal lights will check the congestion state of the roads. Then they will make a priority list where most congested road will get longer time of green signal say 5 minutes, then the next less congested road say 4 minutes & the free road for 2 minutes. A round robin will take place so that every road gets the green signal but with different time span according to its need.

## Chapter 7: Conclusion

In this research we have studied the effect of community area pressure on the congestion of traffic. We have also learnt the behavior of traffic in different part of day time. The Geographic Information System software provided the road information. Then with the neural network module of Artificial Intelligence we have trained our system with many sample input with its corresponding scenario as output. This training has helped to take the further decision from any new input to show the traffic congestion picture. Different types of outputs are shown in this project which will be highly efficient in taking major traffic related decisions. The result if stored in the cloud and accessible by the general people on demand can be made, then the benefit would be highest. With the information generated from our system government can take further decisions like where to construct roads or implement signals. Then if the general people can see the behavior of the roads then they can easily find an alternative road avoiding the one with excessive traffic. Our project could be highly efficient if we could take the input data dynamically. But it was not possible in our part to take the dynamic data due to unavailable technological infrastructure and technologies in our country. But we are hopeful that our system will give maximum benefit and efficiency if it is incorporated with any dynamic system which can count the number of vehicles.

## Chapter 8: Bibliography

- [1] David J Maguire, Michal F Goodchild and David W Rhind, Geographical Information System, Volume 1 & 2, 1994.
- [2] Concepts & Techniques of Geographical Information System: C.P.LO & Albert K W Yeung : India, 2005.
- [3] Introductory reading in Geographical Information System: D J peuquet& D F Marble:London, 1993.
- [4] Andre DANTAS and Yaeko YAMASHITA et al, "An Artificial Intelligent Geographical Spatial Model for Urban Transportation Travel Forecast", 1999,
- [5] Kieran Kirwan, Transportation and Traffic Engineering - Faculty of Civil Engineering and Geosciences, TU Delft,"XML for Distribution Real-Time Traffic Information: A Practical Application".the 9th Mini-EURO Conference,2002.
- [6] Mr. R. Chandra prathap, Mr. A. Mohan Rao, Dr. B. KanagaDurai, Dr. S. Lakshmi, "GIS Application in Traffic Congestion Management", ME Student Division of Transportation Engineering, Anna University Chennai, India.
- [7] Yi-Hwa Wu, Harvey J. Miller<sup>2</sup>, Ming-Chih Hung, "A GIS-based Decision Support System for Analysis of Route Choice in Congested Urban Road Networks " ,Department of Geography, University of Utah.
- [8] Dr. E. F. Ogunbodede, Department of Geography & Planning Sciences, AdekunleAjasin University, "ASSESSMENT OF TRAFFIC CONGESTIONS IN AKURE (NIGERIA) USING GIS APPROACH: LESSONS AND CHALLENGES FOR URBAN SUSTENANCE."Conference,2002.

- [9] G. Arampatzis, C.T. Kiranoudis et al. "A GIS-based decision support system for planning urban transportation policies." *European Journal of Operational Research* 152 (2004) 465–475.
- [10] Yi-Hwa Wu, Harvey J. Miller, Ming-Chih Hung, Department of Geography, University of Utah, "A GIS-BASED DECISION SUPPORT SYSTEM FOR ANALYSIS OF ROUTE CHOICE IN CONGESTED URBAN ROAD NETWORKS." *Proc. of Int. Conference on Recent Trends in Transportation, Environmental and Civil Engineering* 2011,
- [11] BogdanTatomir, Leon J.M. Rothkrantz, Adriana C.Suson, Delft University of Technology, "TRAVEL TIME PREDICTION FOR DYNAMIC ROUTING USING ANT BASED CONTROL."
- [12] An Introduction to GEOGRAPHICAL INFORMATION SYSTEM: Ian Heywood, Sarah Cornelius, Steve Carver.
- [13] A. Mohan rao, "Traffic Forecasting on National Highwayusing Time series Modelling – A case study," Department of Civil Engineering, Regional Engineering College, Warangal, 1998
- [14] E.F. Ogunbodede, "Assessment of Traffic Congestions in Akure (Nigeria) Using GIS Approach: Lessons and Challenges forUrban Sustenance," Department of Geography & Planning Sciences,Ajasin University, Ondo State, Nigeria, 1999.
- [15]Harvey J. Miller, "GIS based dynamic traffic congestion modelling to support time-critical logistics," Department ofEconomics, University of Alberta, Canada, 1999.
- [16] John K. Abraham, Najib E1-Khatib and TapanK. Datta, "Congestion management using GIS," Wayne State University, USA,2006.
- [17] Kaltwasser J., Hübner D. (2000), *European Traffic Information Backbone: The Datex Lesson*, ITS Conference Torino2000.

## Chapter 9: Appendix

**We have done our simulation in Matlab. Before training our system we have chosen some parameter so that based on this training data, we can take any decision through the test data.**

Training data of North road

```
function y_north = check_north(x_north)

north_road_input = [67 78 87 115 127 179 115 110 158 147 53 113 89 75
84 156 145 60 55 108 96 89 101 105 55 85 82 139 43 57 85 76 163
59 80 70 75 60 50 40 47 30 19 35 67 83 95 121 103 75 36 45
76 84 36 41 148 121 76 100 10 0 0 20 15 12 30 15 13 0 0
17 21 55 75 95 83 93 100 20 15 25 35 120 75 79 87 88 127 85
31 45 40 40 50 88 75 73 59 38 72 69 63 58
50 45 42 55 50 53 45 43 38 35 40 45 47 28 58 65 80 76 55
96 105 85 96 19 61 75 45 51 67 75 80 73 99 86 50 95 84 30
39 60 65 76 86 55 75 79 5 5 0 15 20 11 30 0 12 17 5
19 25 20 30 55 43 40 35 75 105 125 20 100 127 100 175 0 15 19
18 55 48 58 75 96 105 106 66 100 82 15 53 98
87 78 82 84 79 36 50 16 25 27 45 62 60 66 10 5 10 26 45
76 96 100 25 75 9 85 75 86 100 105 115 125 27 36 96 50 20 25
39 15 95 59 40 33 20 30 10 5 0 10 20 17 45 0 0 18 3
23 27 19 47 60 76 53 62 80 30 75 160 30 15 25 33 15 140 24
];

north_road_target = [39.1 60.1 59.1 71.9 85.1 95.6 99.6
97.6 95.6 96.2 69.9 66.5 69.2 78.0 74.6 94.2 90.8
67.2 62.1 66.5 64.5 50.0 55.4 56.4 47.2 64.8 63.8
78.7 37.5 42.9 59.1 60.1 88.8 78.0 94.9 86.1 66.2
52.0 40.5 67.5 56.4 56.4 62.8 72.6 88.5 94.9 74.6
82.0 84.1 74.3 47.2 33.7 52.0 53.7 66.2 59.4 92.5
70.6 57.7 70.6 8.4 3.3 0 15.2 18.5 13.5 35.4 5.0 8.4
11.8 2.7 19.9 24.6 31.7 51.3 70.9 68.2 62.8 66.5
59.1 50.6 76.0 72.6 84.4 73.3 68.9 99.6 34.7 95.2
43.2
];

numHiddenNeurons = 5; % Adjust as desired
net_north = newfit(north_road_input,north_road_target,numHiddenNeurons);
net_north.divideParam.trainRatio = 70/100; % Adjust as desired
net_north.divideParam.valRatio = 15/100; % Adjust as desired
net_north.divideParam.testRatio = 15/100; % Adjust as desired

% Train and Apply Network
net_north= train(net_north,north_road_input,north_road_target);
% outputs_north = sim(net_north,north_road_input);
```

```
desired_output_north = sim(net_north,x_north);
y_north = desired_output_north;
```

## Training data of South road

```
function y_south=check_south(x_south)
```

```
south_road_input = [
    67 78 87 115 127 179 215 210 208 197 203 213 189 175 184 156 145 160
155 158 196 189 201 205 205 185 182 189 193 197 185 176 169 159 80 70 75
60 50 40 47 30 19 35 67 83 95 121 153 175 136 145 176 184 136 141
148 151 176 100 10 0 0 20 15 12 30 15 13 0 0 17 21 55 75
95 83 93 100 20 15 25 35 120 175 129 87 88 127 185
    31 45 40 40 50 88 75 73 59 63 72 69 63 58 50 45 42 55
50 53 45 43 38 35 40 45 47 53 58 65 80 76 85 96 105 85 96
119 161 75 45 51 67 75 80 73 99 86 100 95 84 30 39 60 65 76
86 75 75 79 5 5 0 15 20 11 30 0 12 17 5 19 25 20 30
55 43 40 35 175 105 125 120 100 127 150 175 0 15 19
    18 55 48 58 75 96 105 106 116 125 132 115 103 98 87 78 82 84
79 86 105 116 125 127 159 162 160 159 160 155 105 96 85 76 96 100 125
175 109 85 75 86 100 105 115 125 127 136 136 100 120 125 139 115 95 59
40 33 20 30 10 5 0 10 20 17 45 0 0 18 3 23 27 19 47
60 76 53 62 180 130 175 160 30 15 25 33 15 140 24
];
south_road_target = [26.3 40.4 39.7 48.4 57.2 64.3 89.7
88.4 87.0 87.5 92.5 90.2 80.6 75.2 72.9 63.4 61.1
67.9 64.5 67.5 78.6 79.0 82.7 83.4 91.8 89.0 88.0
91.1 93.4 94.7 84.0 79.0 77.0 73.1 63.8 57.9 67.2
80.4 72.7 45.4 37.9 37.9 42.2 48.8 59.5 63.8 72.9
77.9 88.4 84.0 77.2 68.1 80.4 81.5 67.2 62.7 62.2
58.8 61.5 47.5 5.6 2.2 0 10.2 12.5 9.0 23.8 3.4 5.6 7.9
1.8 13.4 16.5 21.3 34.5 47.7 45.9 42.2 44.7 85.2
56.8 73.8 71.5 56.8 72.0 69.0 67.0 23.4 64.0
51.8];
```

```
numHiddenNeurons = 5; % Adjust as desired
net_south = newfit(south_road_input,south_road_target,numHiddenNeurons);
net_south.divideParam.trainRatio = 70/100; % Adjust as desired
net_south.divideParam.valRatio = 15/100; % Adjust as desired
net_south.divideParam.testRatio = 15/100; % Adjust as desired
```

```
% Train and Apply Network
net_south= train(net_south,south_road_input,south_road_target);
%outputs_south = sim(net_south,south_road_input);
```

```
desired_output_south = sim(net_south,x_south);
y_south = desired_output_south;
```

## Training data of East road

```
function y_east=check_east(x_east)
east_road_input = [
    67 78 87 115 127 179 215 210 208 197 153 213 189 175 184 156 145 160
155 158 196 189 201 205 155 185 182 139 143 157 185 176 169 159 80 70 75
60 50 40 47 30 19 35 67 83 95 121 153 175 136 145 176 184 136 141
148 151 176 100
    31 45 40 40 50 88 75 73 59 63 72 69 63 58 50 45 42 55
50 53 45 43 38 35 40 45 47 53 58 65 80 76 85 96 105 85 96
119 161 75 45 51 67 75 80 73 99 86 100 95 84 30 39 60 65 76
86 75 75 79
    18 55 48 58 75 96 105 106 116 125 82 115 103 98 87 78 82 84
79 86 105 116 125 127 109 162 160 109 110 105 105 96 85 76 96 100 125
175 109 85 75 86 100 105 115 125 127 136 136 100 120 125 139 115 95 59
40 33 20 30
];
east_road_target = [29.0 44.5 43.7 53.2 63.0 70.7 98.7
97.2 95.7 96.2 76.7 99.2 88.7 82.7 80.2 69.7 67.2
74.7 71.0 74.2 86.5 87.0 91.0 91.7 76.0 98.0 97.2
75.2 77.7 79.2 92.5 87.0 84.7 80.5 70.2 63.7 74.0
88.5 80.0 50.0 41.7 41.7 46.5 53.7 65.5 70.2 80.2
85.7 97.2 92.5 85.0 75.0 88.5 89.7 74.0 69.0 68.5
64.7 67.7 52.2];
```

```
numHiddenNeurons = 5; % Adjust as desired
net_east = newfit(east_road_input,east_road_target,numHiddenNeurons);
net_east.divideParam.trainRatio = 70/100; % Adjust as desired
net_east.divideParam.valRatio = 15/100; % Adjust as desired
net_east.divideParam.testRatio = 15/100; % Adjust as desired
```

```
% Train and Apply Network
net_east= train(net_east,east_road_input,east_road_target);
%outputs_east = sim(net_east,east_road_input);
```

```
desired_output_east = sim(net_east,x_east);
y_east = desired_output_east;
```

## Training data of West road

```
%% Checking the condition of a road in a week%%
```

```

function y_west=check_west(x_west)
west_road_input = [
    67 78 87 115 127 179 215 210 208 197 203 213 189 175 184 156 145 160
155 158 196 189 201 205 205 185 182 189 193 197 185 176 169 159 80 70 75
60 50 40 47 30 19 35 67 83 95 121 153 175 136 145 176 184 136 141
148 151 176 100
    31 45 40 40 50 88 75 73 59 63 72 69 63 58 50 45 42 55
50 53 45 43 38 35 40 45 47 53 58 65 80 76 85 96 105 85 96
119 161 75 45 51 67 75 80 73 99 86 100 95 84 30 39 60 65 76
86 75 75 79
    18 55 48 58 75 96 105 106 116 125 132 115 103 98 87 78 82 84
79 86 105 116 125 127 159 162 160 159 160 155 105 96 85 76 96 100 125
175 109 85 75 86 100 105 115 125 127 136 136 100 120 125 139 115 95 59
40 33 20 30
];
west_road_target = [27.2 41.8 41.1 50.1 59.2 66.5 92.9
91.5 90.1 90.5 95.7 93.4 83.5 77.8 75.5 65.6 63.2
70.3 66.8 69.8 81.4 81.8 85.6 86.3 95.0 92.2 91.5
94.3 96.7 98.1 87.0 81.8 79.7 75.7 66.1 60.0 69.6
83.2 75.2 47.0 39.2 39.2 43.7 50.5 61.6 66.1 75.5
80.7 91.5 87.0 80.0 70.5 83.2 84.4 69.6 64.9 64.4
60.9 63.7 49.1];

numHiddenNeurons = 5; % Adjust as desired
net_west = newfit(west_road_input,west_road_target,numHiddenNeurons);
net_west.divideParam.trainRatio = 70/100; % Adjust as desired
net_west.divideParam.valRatio = 15/100; % Adjust as desired
net_west.divideParam.testRatio = 15/100; % Adjust as desired

% Train and Apply Network
net_west= train(net_west,west_road_input,west_road_target);
%outputs_west = sim(net_west,west_road_input);

desired_output_west = sim(net_west,x_west);
y_west = desired_output_west;

```

This segment is used for checking the road traffic congestion percentage:

```

function [] = checkbusy( road, val)

if(val>= 85), fprintf('%s road traffic percentage is %.2f
(Congested)\n',road,val);
fprintf('Open the signal 6 minutes for the %s road\n',road);
elseif (val>=75 &&val< 85), fprintf('%s road traffic percentage is %.2f (Very
Busy)\n',road,val);
fprintf('Open the signal 5 minutes for the %s road\n',road);
elseif (val>=60 &&val< 75), fprintf('%s road traffic percentage is %.2f
(Busy)\n',road,val);
fprintf('Open the signal 4 minutes for the %s road\n',road);
elseif (val>=40 &&val< 60), fprintf('%s road traffic percentage is %.2f
(Moderate Busy)\n',road,val);
fprintf('Open the signal 3 minutes for the %s road\n',road);
elseif (val>=25 &&val< 40), fprintf('%s road traffic percentage is %.2f
(Normal)\n',road,val);

```



```

fprintf('Open the signal 3 minutes for the %s road\n',road);
elsefprintf('%s road traffic percentage is %.2f (Free)',road,val);
fprintf('Open the signal 3 minutes for the %s road\n',road);
end
fprintf('\n');

```

This portion shows the congestion percentage with respect to the input condition for every separate road.

```

fid_north = fopen('north_test_data.txt','r');

Noi_north = fscanf(fid_north, '%d ', 1);
for i=1:Noi_north
x_north = zeros(3,1);
for j=1:3
x_north(j,1) = fscanf(fid_north, '%d ', 1);
end
% x_north
y_north = check_north(x_north);
% fprintf('\nPossible percentage of the road capacity of north road: %.2f
\n', y_north);

end

fid_south = fopen('south_test_data.txt','r');

Noi_south = fscanf(fid_south, '%d ', 1);
for i=1:Noi_south
x_south = zeros(3,1);
for j=1:3
x_south(j,1) = fscanf(fid_south, '%d ', 1);
end
% x_south
y_south = check_south(x_south);
% fprintf('\nPossible percentage of the road capacity of south road: %.2f
\n', y_south);

end

fid_east = fopen('east_test_data.txt','r');

Noi_east = fscanf(fid_east, '%d ', 1);
for i=1:Noi_east
x_east = zeros(3,1);
for j=1:3
x_east(j,1) = fscanf(fid_east, '%d ', 1);
end
% x_east
y_east = check_east(x_east);
%fprintf('\nPossible percentage of the road capacity of east road: %.2f \n',
y_east);

```

```

end

fid_west = fopen('west_test_data.txt','r');

Noi_west = fscanf(fid_west, '%d ', 1);
for i=1:Noi_west
x_west = zeros(3,1);
for j=1:3
x_west(j,1) = fscanf(fid_west, '%d ', 1);
end
% x_west
y_west = check_west(x_west);
fprintf('\nPossible percentage of the road capacity of west road: %.2f
\n\n', y_west);

end

%junction_percentage = zeros(50,50);
junction_percentage=[y_east;y_west;y_north;y_south];

checkbusy('North',y_north);
checkbusy('South',y_south);
checkbusy('East',y_east);
checkbusy('West',y_west);
%junction_percentage_output = sortrows(junction_percentage);

%plot(y_north,y_south,y_east,y_west);

%junction_percentage_output = sort(junction_percentage,1,'ascend');
fprintf('possible percentage in ascending order: %.12f \n \n',
junction_percentage_output);
fprintf('Show the possible percentage in ascending order: %.12f \n \n',
sort(junction_percentage,1,'ascend'));

graphshow = zeros(4,2);
for i = 1:4
graphshow(i,1) = i;
graphshow(1,2) = y_north;
graphshow(2,2) = y_south;
graphshow(3,2) = y_east;
graphshow(4,2) = y_west;
end
stem(graphshow(:,1),graphshow(:,2));
xlim([0 5])
ylim([0 100])
title('Current condition of the Roads');
xlabel('Name of the roads ');
ylabel('Percentage of Traffic Congestion');

```

Before going to this code we took the seven working days road input parameter (say at 6.00pm) and then this snippet of code will show the average congestion percentage of north road at 6.00 pm.

```

fid_north = fopen('north_test_data_weekly.txt','r');

Noi_north = fscanf(fid_north, '%d ', 1);
graphshow = zeros(6,2);
for i=1:Noi_north
x_north = zeros(3,1);
for j=1:3
x_north(j,1) = fscanf(fid_north, '%d ', 1);
fprintf('%d',x_north(j,1) );
end
%x_north
y_north = check_north(x_north);

graphshow(i,1) = i;
graphshow(i,2) = y_north;

fprintf('\n Possible percentage of the road capacity of north road: %.2f \n',
y_north);
Average_north(i,1)=y_north;

end
fprintf('\n Average percentage of this road at this perticular time :: % .2f
\n', mean(Average_north));
%plot(graphshow(:,1),graphshow(:,2));
% graphshow(7,1) = 7;
% graphshow(7,2) = mean(Average_north);

bar(graphshow(:,1),graphshow(:,2));
xlim([0 7])
ylim([0 100])
title('North Road Percentage at 6.00 Pm on different working days')
xlabel('Day');
ylabel('Percentage');

```

Similarly we can take the east, west, south road condition at different time to perform the similar action.