DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
R. V. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU),
Bengaluru – 560 059

A Technical Seminar report on

"NEURAL NETWORKS - MULTI LAYER PERCEPTRON IN DYNAMIC TRAFFIC PREDICTION"

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering
In
Computer Science & Engineering

By

Kiran K (1RV10CS042)

Under the guidance of

Prof. Swarnalatha K.S.
Asst. Professor
Department of C.S.E., R.V.C.E.

Carried out at:

Department Of Computer Science & Engineering
R. V. College of Engineering
Bengaluru – 560 059

(Academic Year: 2013-14)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

The automobile industry is booming; vehicles are becoming more affordable than ever to everyone. Today, traffic jams have become a daily affair in the metropolitan cities like Bangalore. Traffic is going to increase with time. So urban management corporations built flyovers, underpasses, multi-lanes and variable traffic signals as a means of eradicating this problem. It worked in some places but failed miserably adding to the woes of daily travelers by worsening the problem.

Traffic prediction is required both for preventive measures as well as to serve as information for future purpose of effective construction of such structures. Accurate traffic reports are essential for congested and overcrowded cities. Hence, traffic management is vital in such cities. Traffic prediction is a key to avoid congestion and preventing a breakdown of normal vehicular movement. The main problem is that the traffic is dynamic in nature. There are various factors that affect the traffic flow and speed, for instance, demographics, time, speed, lane width and length. In civil engineering parlance, every road has a unique identifier that is formed by a combination of features that span the above factors. The amount of parking space, the locality of the road and the type of the road – cross-section/ intersection, tunnels, motorways, highways, subways, multi-lanes or single –lanes are a subset of other features that compound the problem of traffic prediction by emanating different classes of congestions, jams or accidents in different roads.

What makes the problem more complicated is that often, traffic in one road may be a cause for jams in others. This domino effect can quickly bring the entire grid down within minutes, and more quickly during peak hours. It must not be mistaken that traffic prediction is necessary only in urban localities. Traffic prediction is a universal problem. It applies to urban and rural roads alike. Roads are connecting urban to suburbs and rural places, diminishing the factor of distance and time to connect to modernization and development. So it should not be forgotten that traffic chaos can easily propagate to rural roads as well.

1.1 The need for traffic prediction

A few questions need to be asked before one can actually define traffic prediction. What does the word predict mean in this context? Does it imply foreseeing jams or congestion? Does it take into account heterogeneity? What should be the time to predict be? Where should the prediction be? It is not enough to simply count the number of vehicles on a road. Often a certain high number of a class of vehicles can clog up the grid. It is possible for a single truck to stop the traffic flow. A high number of scooters can also do the same.

In this era of smartphones, we constantly rely on Google Maps that predict accurately the time to reach a destination along with the prediction of alternative routes. In this case, the prediction is not restricted to a single point, nor is the time individualistic. Rather, it can be observed that this kind of prediction is of the aggregative type, adding up the traffic time at multiple points that are in between the source and the destination. Hence, traffic prediction is taking into account data from various points to build a holistic view of the traffic in the region specified.

To predict effectively the traffic in the next intersection, or provide a rerouting mechanism, with congestion controls- both short term and long terms, artificial intelligence is being employed. AI is seen as the solution to the limitation of processing power of humans. If the computer can learn to think, it is possible for the computer to solve problems on its own, due to its gazillion computing power unlike the human brain. It is true that computers cannot reason independently. That hasn't stopped us from feeding it useful data that can help them learn on their own. Machine learning has become the term of this decade, changing the meaning of data mining substantially. It has also redefined the limits of what science can do to solve existing problems. Prediction is a part of AI, and data mining technologies like association analysis, decision trees, and artificial neural networks are employed.

1.2 Artificial Neural Networks (ANN)

We are so narcissistic that we have classified our brain to be the most intelligent brain in the world; the dolphins come next. Our brain is a complex structure in itself, made of millions of neurons communicating every second, with every body movement or even with every thought.

The concept of ANN was introduced by McCuloch and Pitts (1943). An artificial neural network (commonly just neural network) is an interconnected collection of artificial neurons that uses a mathematical or computational model of theorized mind and brain activity, simulating the powerful capabilities for knowledge acquisition, recall, synthesis, and problem solving exhibited by the brain.

Hence the neural networks are computational models that can be trained to simulate thinking or learning by experience. There is an input layer and a output layer, and connecting them are a series of hidden layers which could be just one hidden layer or may be thousands of layers. Each layer can have variably number of neurons, simulating the actual neurons. Certain weights connect a neuron of one layer to a neuron of another layer. Neurons within a layer do not communicate with each other.

A neuron computes an output using an activation function that considers the weighted sum of all its inputs. Inputs given are forwarded to successive layers, each neuron having their activation functions [A_f]. A_f is commonly the sigmoid function, but it need not restricted to it. The sigmoid function is defined as:

$$f(x) = \frac{1}{1+e^{-x}}$$

where f(x) is the output of the neuron and x is the weighted sum of all its input.

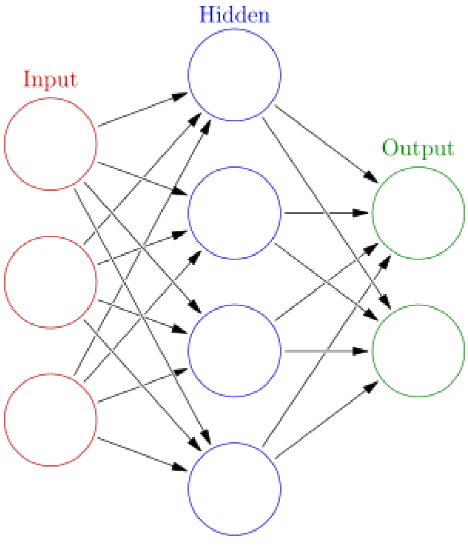

Figure 1.1 Representation of a (1) * (1) * (1) ANN

The activation function introduces non-linearity which results in convergence. Neural networks can be of back-propagating type, where such a convergence is required. This is because the weights given initially are random and may not predict correctly for the known input-output data-set. This known data is called the training set which is fed into the ANN, the errors are propagated backwards, and the weights adjusted accordingly. This may not conclude in one iteration, and hence the requirement for faster convergence arises. Only non-linear functions allow convergence, and hence non-linear functions are employed. Due to the activation functions, ANNs can be employed to model complex, non –linear phenomena.

1.3 Multi-layer Perceptron (MLPs)

MLPs are simple feed forward neural networks. Except for the input nodes, all other nodes are neurons with activation functions which are non-linear. MLPs employ technique of Backpropagation to train the neurons accordingly. This involves feeding input to network and calculating the error rates between expected and actual outputs and going back to adjust weights of each layer of MLP. This process is iterated until convergence is reached, or minimum training gradient is attained (or expected output is gained). MLPs are simply backpropagation networks with unidirectional flow of data. The above method of trial/error uses the concept of gradient descent where the iterations are stopped when the errors reach the acceptable minima.

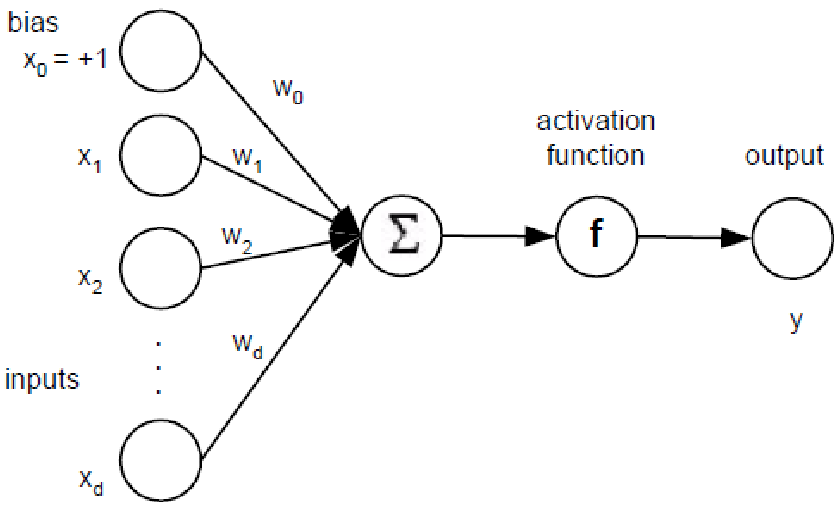

Figure 1.2 A single perceptron

The perceptron is simply the weighted sum of inputs in most cases:

$$y = \sum_{j=1}^{d} w_j x_j + w_0$$

Where the weights are shown above the connecting line between neurons of different layers and **f** is the activation function. This value is fed as **x** into the sigmoid function or any other activation function.

The architecture of the MLPs depend upon the following:

1. Number of hidden layers in the network
2. Number of hidden neurons in the network
3. Connectivity of the input, output and the hidden layers (along with possible bias)
4. Learning rate (this defines the amount of trial/error for correct prediction)
5. Activation functions of each neuron

The initial architecture may vary significantly from the final neural network that provided predictions within acceptable error rates. The initial weights given are as told before, random values which change during backpropagation.

1.4 Transport forecasting

It is defined as a process of estimating number of vehicles, at a given epoch (time), their speed, and other metrics, that could be used to reroute/predict traffic as an estimate for future costs and calculate capacity of infrastructure. Transport forecasting systems today employ Dynamic Traffic Allocation model.

1.4.1 Dynamic Traffic Allocation (DTA)

Dynamic Traffic Allocation (**DTA**) model involves four key steps:

1. **Trip Generation**: This involves the study of the demographics, land use, residences, social and economic attributes of households surrounding the road network in question. This is collection of existing data about road networks.

2. **Trip Distribution**: This involves developing "trip table", a matrix that displays the number of trips going from each origin to each destination within the road network. Suitable data collection tools need to be employed to gather such data, which are explained later in the design chapter.

3. **Mode Choice**: This involved predicting different modes of transportation used by different proportion of trips. Not everyone may prefer to use cars for a certain trip.

Some may prefer public transport like buses, and this preference may vary along with time. These details come under the mode choice.

4. **Route Assignment**: The final step of DTA involves Trip Allocation by a particular mode – a prediction made for the particular trip.

1.4.2 Traffic Stream Properties

Traffic flow (rate), speed, and density are three basic parameters that describe traffic conditions.

1. Flow

Traffic flow can be defined as the total number of vehicles that pass over a given point or section of a lane or roadway during a given time interval. The number can be expressed hourly, daily or in a specific time interval basis. Traffic flow rate is defines as the equivalent hourly rate at which vehicles pass a given point or cross-section during a time-interval. Flow is usually limited by road capacity.

2. Speed

Speed is the rate at which a vehicle passes a given point. The speed is limited by number of vehicles occupying the road at the given point. Heterogeneity can also be included as a limiting factor to speed.

3. Density

Density is the number of vehicles observed and measured over a certain road segment over a period of time and not just a given point.

1.5 Metrics

As listed out in Table [TODO insert table number], the metrics can be both time-variant and static in nature. Traffic state is influenced by metrics both in time and spatial domain. The table is not exhaustive in nature providing some of the basic information that can be fed into the neural network as input.

`

Static Indices	Dynamic Indices
Area Type	Volume by Movement
Lane Width	Percent Heavy Vehicles
Number of Lanes	Operating Speed
Exclusive Left turn/ Right Turn lanes	Peak Hour Factor
Parking Conditions	Cycle Length, % Parking Space

Table 1.1 Various metrics involved in traffic prediction.

Also the road type may vary. The roads in question may be intersection/cross-section. They may also be tunnels, straight segments, motorways or urban networks. The traffic data varies with the road type as well.

1.6 Prediction Problem

There are two problems that are commonly faced while solving traffic congestion problems and traffic prediction.

1.6.1 Wardrop's principle of User Equilibrium

Each driver takes the shortest path, subject to every other driver who is doing the same. Hence an equilibrium is attained because every driver takes the same shortest path which clogs up the path, due to which the longer paths are less clogged and slowly become the shorter path, and this cycle repeats. This can be compared to Nash Equilibrium which comes under the game theory domain where there is a payoff confusion.

1.6.2 Bilevel problem

The other problem faced while trying to resolve traffic related problems is the duality existing in nature. The travel time is a function of demand and the demand is a function of travel time. Resolving one would mean resolving the other first, but the other in turn depends on this parameter, creating a deadlock.

1.7 Why are Neural Networks suitable for traffic prediction

Traffic flow is stochastic in nature as well as traffic prediction characteristics are non-linear. This works well with neural networks. When working with neural networks, no prior assumptions regarding the problem required and they provide automatic Classification

Neural networks are also able to adapt to moderately noisy data. They can scale well with dataset size, which may vary depending on the road network in question.

1.8 Objectives of the research work

- Understand the need and importance of Artificial Neural Networks in DTA (Dynamic Traffic Allocation) system.
- To get to know the use of Multi-layer Perceptrons in Traffic prediction systems
- Analyse the existing MLP models varying over input and network structure.
- Understand data collection and dimensionality reduction of data before it is sent to NN
- Compare the performance of MLP for traffic prediction with other ANN models like RBF
- Compare the performance of MLP for traffic prediction with other models, studying the accuracy rates.

1.9 Organization of Thesis

In this seminar, dynamic traffic forecasting and management using multilayer perceptron (MLP) networks is analysed. MLPs are studied, their use in DTA is researched and effectiveness is compared against other models. Traffic prediction control system must anticipate traffic situations and adjust their strategy on time. Hence the purpose of this research is to study the influence of various factors on the short time prediction of the traffic.

The rest of the seminar report is organized as explained below. Chapter 2 gives a brief description of the literature survey performed for the research work of this seminar. Chapter 3 explains the procedure of design of the prediction model which includes data collection, classification and optimization. Chapter 4 describes the results of the design approaches and also analyses the performance for consistency of each model. Chapter 5 provides the conclusion and Chapter 6 includes the future work possible (or what was excluded from this research frame). The references can be found in Chapter 7.

Chapter 2 Literature Survey

Vehicular traffic has three parameters associated with it - flow, density, and speed. Traffic flow is the study of interactions among vehicles, drivers, and infrastructures. The traffic flow is not uniform and varies both with time and space.

Satu Innama, in [1] describes a model to predict traffic for short-term purposes, based on on-line travel time measurements with video. The paper researched the effect of length of links and location of measurement stations and the structure of the entire measurement system and also set up camera stations at every link to measure the three parameters and predict traffic accordingly. The research focused on comparing results of travel time between short and long links.

Satu Innamaa, in [2] studied the influence of various factors on the results of the short-time prediction of the traffic situation on urban highways. The MLP predictions models were made to predict the speed and flow 15 minutes ahead of the observation period in five-minute periods. The research scope also included separating speed and flow information into two separate models and analyzing the effectiveness of such a separation. The division into 3 models each of the five-minute period was also studied which resulted in the worst performance compared to the aggregated model of 15 minutes. The model also showed results for normalized as well as scaled data, calculating the Root Mean Square Errors (RMSEs) for each result. The errors were less than 20 % for traffic flows based model, while it was less than 4 % for the speed based models.

[2] also employs a novel neural network training algorithm called as Levenberg –Marquardt Algorithm or damped least squares method which uses the RMSEs for non-linear neural network models. It also studies the influence of the number of input parameters on the number of hidden neurons which in turn affects the results of the network.

Bing Wu; et al. in [3] look at the applications of data mining technologies in dynamic traffic prediction. The paper employs a compound backpropagation network which employs decision trees and association analysis along with neural networks. This could be done with any neural network model, as the other data-mining technologies improve the results significantly. The static and dynamic indices are explained in detail as well as the need for an extra correcting class neural network. It also exploits sequential patterns in the data of the whole region. The highlight concepts of the paper are convergence and stability of the neural network.

Pitiphoom Posawang; et al., in [4] utilized an intelligent traffic camera system to classify congestion into three levels – light, heavy and jam. Occupancy Ratio (OR) technique is compared with the deployed MLP prediction model that provides automatic classification. The data collection happens per minute alongside the stream video processing. The input consisted of 10 features – the seven days of the week, the minute of the day, speed and volume. WEKA machine learning tool was used to train the network, optimizing the number of hidden neurons.

Eleni Vlahogianni and Matthew Karlaftis in [5] study local and global iterative algorithms for real-time short-term traffic prediction. The paper focuses on purely temporal structures of neural networks which iteratively predict short-term traffic flow. It draws out a comparison between single-step ahead approach vs. direct approach. The single step ahead approach predicts the (t+1) time traffic using {1,2…..t} time data. The direct approach predicts the (t+h) time traffic using {1,2….t} time data.

Bowu Zhang; et al. in [6], classify all road points into time-variant clusters. Strong correlation of traffic in temporal and spatial domain is displayed in the results. A traffic influence metric – 'Similarity' is defined which indicates the influence of one road's condition on another.

Kranti Kumar; et al. in [7] consider speed of each category of vehicles separately as input variables in contrast to previous studies reported in literature which consider average speed of combined traffic flow. This inclusion of heterogeneity of traffic flow (Indian traffic condition) is consistent with the results of homogeneous neural networks deployed for traffic prediction. The prediction is based on given continuous short –term traffic information feedback. The paper employs sensitivity analysis to get the nine most significant parameters (i.e. dimensionality reduction) to reduce cost of unnecessary processing.

Gregory, Aaron's thesis work [8] compares MLPs with Radial Basis Function Neural Networks (RBFs). The results are explained in chapter 4. A modified form of RBFs worked better than MLPs in traffic forecasting application, according to the thesis. This may not serve as a generic case though. [9] also provides a comparison of various ANN models for hourly traffic prediction.

Chapter 3 Design

In this chapter, the various design approaches to build the required prediction model are explained. The design process consists of the following phases:

1. Data Collection and Extraction
2. Data Classification
3. Data Optimization

Each one of them is explained below in detail. Data optimization is explained as an extension compound neural network model.

3.1 Data Collection and Extraction

Traffic data that is to be fed into the prediction model can be both historical data and real time data. Real time data is mostly stream video processing data captured by the traffic camera systems deployed at cross-sections or regular identified link points in the grid.

3.1.1 Data Retrieval

Data can be collected by setting up 5 minute interval points (or 15 minutes) where cameras are deployed. This could be done at all intersection points.

Following variables are measured for:

- Static and dynamic indices /metrics as explained in introduction
- Speed – Max, Min, Average (depending on model)
- Flow – Max, Min, Average (depending on model)
- Day and Time of Week
- Urban /Non-Urban Data

The objective is to predict the traffic flow status ahead through dynamic indices. It should also take into account the relationships between dynamic and static indices, and also sequential patterns in the data of the whole region.

Table 3.1 lists out data collected by [7] in the stretch of Muzaffarnagar bye-pass, on National Highway-58 (NH-58), Km 115.000 from Roorkee to Delhi, Uttar Pradesh, India at two selected locations (at 116.500 and 128.700). Data samples were collected using video cameras at selected locations for a period of five days from Monday to Friday between 11.00 am to 1.00 pm.

	Delhi-Haridwar								Haridwar-Delhi							
	Volume				Average Speed (Km/hour)				Volume				Average Speed (Km/hour)			
Vehicle Category	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD
C/J/V	13	71	37.75	20.86	71.5	78	75.56	2.11	14	57	41.13	14.14	71.5	78	75.18	2.15
S/M	25	53	33.25	13.28	55	59	57	1.58	16	48	29.25	14.17	52.5	60	57.75	3.57
LCV/M	0	3	1.75	1.5	60	66	63.4	2.19	1	6	3	2.44	62	66	64.33	1.86
B	2	8	5	3.46	49.5	53	51.21	1.46	2	8	5	2.58	49.5	52.5	51.28	1.04
T	8	21	13	5.59	53	57	55.57	1.24	8	18	14.75	4.72	55.5	60	56.75	1.46
TW	0	1	0.75	0.5	33	34.5	33.5	0.86	0	5	1.75	2.36	35.5	39	37.33	1.76
T/T	1	5	2.75	1.71	28	32	30.36	1.97	0	5	1.75	2.22	29.5	32	31	1.32
H/B	1	2	1.25	0.5	6.23	9.25	7.54	1.46	0	3	1.25	1.5	7.24	8.32	7.76	0.59

Table 3.1 Summary of traffic volume and speed measurement for 15 minutes interval at the selected location as recorded by Kranti Kumar et al. in [7]

Another collection method is shown in Table 3.2 recorded by [4] which collects data for every minute, and later is aggregated to one hour/ one day depending on the amount of dimensionality reduction required. This data was used to build a congestion predicting MLP.

SAMPLES OF TRAINING DATASET

DW	MT	SP	VOL	CL
Mon	471	12.78	5	3
Mon	472	12.88	2	3
Mon	473	11.94	4	3
Mon	474	15.37	4	3
Mon	475	17.18	3	3

Table 3.2 Samples of training dataset recorded by [4]

The metrics are:

DW= Day of the Week

MT = Minute of the Day

SP = Speed

VOL = Volume (Traffic Flow)

CL = target Congestion Level (as output)

3.1.2 Data preprocessing

This step is necessary because the amount of data collected is huge, and feeding all data into ANN may not be feasible and performance enhancing. Also possible is the huge amount of data may cause the ANN to not learn correctly.

A possible alternative to cleaning of data after it is gathered would be to decide on how much data is being collected. This would mean to decide on the stretch of the road which needs to be observed and where the traffic has to be predicted. This could mean deciding on the span of test and on the number of intersections /cross-sections. However this is not always feasible; even a reduction of the span may not mean reduction in irrelevant data. Though ANN can moderately adopt to noise, feeding irrelevant data makes no sense in terms of cost and viability.

Possible scaling of data to [-1,1] may be necessary to maintain mean of 0 and standard deviation of 1 which is considered to be standard distribution of data. This standardization is useful for the ANN's learning.

Normalization / Dimensionality Reduction is a must considering the number of features that accompany data that spans even 5 minutes. [7] uses sensitivity analysis about the mean to reduce the number of input parameters to be fed into the ANN. Figure 3.1 shows the sensitivity analysis carried out by [7] for the incoming data.

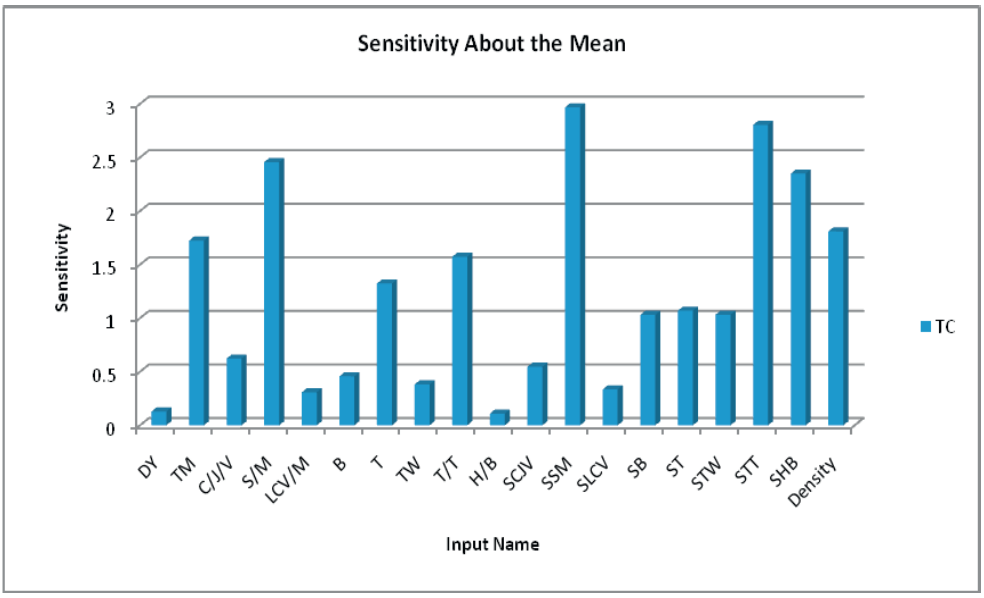

Figure 3.1 Sensitivity Analysis of various input parameters about the mean

It is not necessary to go into the details of the abbreviations on the X-axis of the figure. Vehicles were classified in eight categories namely Car/Jeep/Van, Scooter/ Motorcycle, Light Commercial Vehicle (LCV)/Minibus, Bus, Truck, 3-wheeler, Tractor and Horse-cart/ Bullock-cart/ other animal drawn vehicles.

[7] adopts sensitivity analysis to perform dimensionality reduction, selecting the nine most significant input parameters.

The nine most significant inputs parameters are clearly shown in the figure, which are speed of Scooter/Motorcycle (SSM), speed of Tractor/Trailor (STT), number of Scooter/Motorcycle (S/M), speed of Horsecart/Bullockcart (SHB), traffic density, time (TM), number of Tractor/Trailor (T/T), number of Trucks (T) and speed of Truck (ST). The reduction of 19 parameters to 9 most significant parameters was cost-reducing and also effective in terms of better learning of the ANN.

Certain input parameters cannot be reduced simply because their sensitivity about the mean was low. It is highly possible that their association with other higher sensitivity parameters is more contributing to the learning of the ANN than the most significant parameters alone. In such cases, it is useful to transform a N-dimensional space into M-dimensional space, where N=M or N<M or N>M, depending on what is more useful. In most cases, M<N which reduces the dimension of the transformation. Also values may be mapped to binary values reducing storage space.

For example, average delay can be a value from 0- 100. If the 10 ranges ([0,9], [10-19]……) can be mapped onto a respective Level of Service (A,B,C,…..), only ten binary values (in terms of range) can be stored instead of 101 binary values.

3.2. Data Classification / Design Models

Different approaches and sub-approaches have been taken to understand performance of MLPs under different conditions.

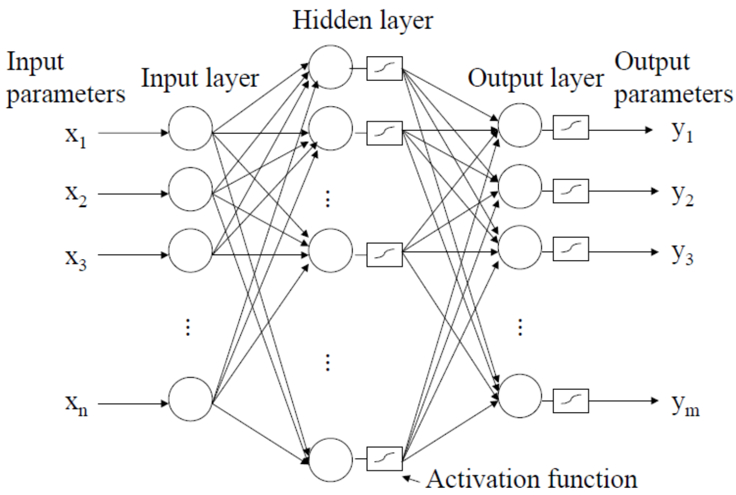

Figure 3.2 A generic MLP network

From the figure 3.2, it can be realized that there is no specific model with rules for traffic prediction. The model can change based on how much degree of freedom the analyst is ready to give to the prediction model build.

As already explained in section 3.1, different approaches may take different methods to do data pre-processing. There is no standard, but the results of all must be highly consistent – able to predict traffic conditions.

3.2.1 Design Approach

Now that we have selected the optimum number of input parameters, and we also know the number of output parameters, the remaining aspect is deciding on the number of hidden layers and the respective number of hidden neurons. For simplification purposes, a single hidden layer is opted, making the MLP (N)*(1)*(M) neural network, where N is the number of input parameters and M is the number of output parameters.

[2] chooses a road of 3 km stretch with 7 cross sections, where the 5th cross-section is where we want to predict the traffic (considering the traffic before and after the target cross-section) Data collected every five minutes in 3 spans. This data is aggregated separately as 15 minute input data as well which consists of mean speed and flow information.

Calculating N_h is the first task. To calculate N_h, the number of input and output parameters should also be decided. The weights that connect a neuron of one layer to another also depend on this set-up, though initially they can be chosen randomly. MLPs work with gradient descent rule where the errors are backpropagated until the errors fall below the acceptable minima, simultaneously adjusting the weights. Hence, an appropriate guess work is suitable for the weights initially.

Considering, N_i,N_h, and N_o are number of neurons in input, hidden and output layers respectively.

W = no. of parameters (weights and biases)

Tmin = Number of training samples required for the MLP to learn and predict accurately.

The maximum number of hidden layer neurons possible is given by the formula for ($N_{h\,max}$)

$$T_{\min} = 10 \cdot W; \quad W = (N_i + 1)N_h + (N_h + 1)N_o$$
$$\Rightarrow \quad N_{h_{\max}} = \frac{0{,}10 \cdot T - N_o}{N_i + N_o + 1}$$

Levenberg –Marquardt Algorithm or also called as damped least squares method is used to solve non-linear least square problem. The RMSEs calculated during backpropagation can be used to change the weights using the above algorithm. The idea of the Levenberg-Marquardt algorithm is that the training method shifts from the gradient method towards Newton's method making use of the advantages of both methods. (because, the model is non-linear)

The benefit that is gained by reducing the input parameters is the possibility to have more hidden neurons in the network with the same training set. This is why most approaches prefer to reduce the input space.

The second task is the selection of the activation function. The activation function combinations used were a logistic function and a hyperbolic tangent for the hidden layer, and a hyperbolic tangent and a linear function for the output layer.

3.2.2. Design Studies

The following design studies were made:

Design Study 1: Divide into 2 models – flow model and speed model.

Design Study 2: Divide into 3 models – each for the 5-min interval data

Design Study 3: Divide into 2 models – one with normalized data, the other is scaled to [-1,1]

Design Study 4: Study as one model where data is aggregated to a 15 minute interval.

Each of the four studies were experimented with and the results are explained in the next chapter.

3.2.3. A Compound Backpropagation Model

Bing Wu; et al. in [3] proposed a compound neural network model which uses other data mining technologies along with the neural network for dynamic traffic prediction.

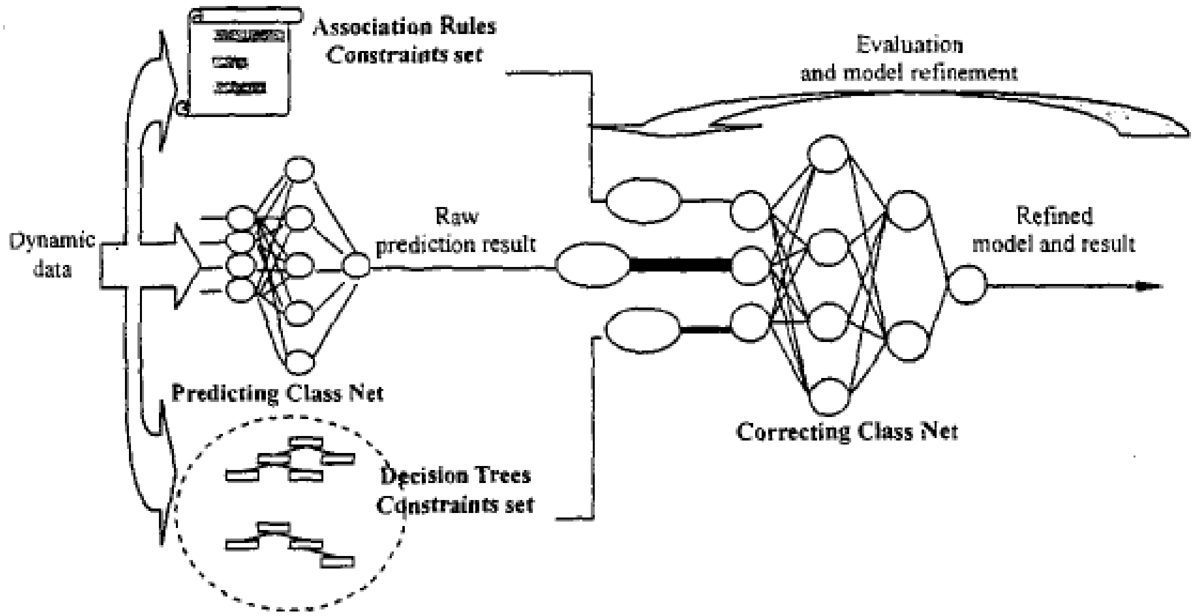

Figure 3.3 Compound Backpropagation model schema

In certain periods of time, changes in parking conditions and bus operation can affect the metrics near the collecting point. To analyze this type of influences, decision trees are used. Neural networks cannot manage such data without manual intervention, while decision trees provide decision paths to process such data. Association analysis results in useful knowledge about relationships between parameters which may be also used for transformation as explained in data preprocessing section of section 3.1. A correcting class network is used for refinement purposes.

Chapter 4 Experimental Results and Analysis

This section covers the results of the various design models (studies) undertaken, and then the design models are compared with each other. MLPs are also compared with a variant of RBFs and accordingly the analysis of the MLP performance to the given problem is carried out.

4.1 Choice of Activation Function

To make the basic model for the prediction, the first task was to study different activation functions of the neural network. For the models based on both the scaled data and the normalized data, the best combination of functions was a hyperbolic tangent for the hidden layer and a linear function for the output layer.

The activation functions need not be sigmoid function and need not be a single non-linear function.

4.2 Results for design studies

Table 4.1 shows the results of various metrics calculated during design study 1 which separates out traffic flow and speed as two different models.

Error	Data	q_{0-5}	q_{5-10}	q_{10-15}	v_{0-5}	v_{5-10}	v_{10-15}
Mean squared error	Scaled	36 000	44 000	51 000	6	7	7
	Normalized	43 000	51 000	60 000	5	6	7
Mean error	Scaled	-11	-18	-16	0	0	0
	Normalized	-7	-9	-3	0	0	0
Mean absolute value of error	Scaled	150	160	170	1	2	2
	Normalized	160	180	190	1	1	2
Mean relative error	Scaled	-1 %	-1 %	-1 %	0 %	0 %	0 %
	Normalized	0 %	0 %	0 %	0 %	0 %	0 %
Mean absolute value of relative error	Scaled	8 %	8 %	9 %	2 %	2 %	2 %
	Normalized	9 %	9 %	10 %	2 %	2 %	2 %

Table 4.1: Error values for design study 1

Error terms of the sub-models for speed and flow prediction are shown in table 4.1 The flow is denoted by **q**, and the mean speed by **v**. Sub-indices stand for the prediction period in minutes from the observation moment. Table 4.1 also includes results for design study 2 and 3 which includes scaling and normalization and also the three five-minute interval data as three separate models.

The flow model based on normalized data gave smaller mean squared errors, mean relative errors, and mean errors. It can be observed though that flow model based on scaled data gave better results when considering the different absolute values of errors.

Thus the flow model based on scaled data gave forecasts that were realistic but the results of the flow model based on normalized data were more evenly distributed around the real values. The speed model based on normalized data gave better results than the speed model based on scaled data.

It can be observed that for the model based on normalized data, the reduction of the input parameters seemed not to be advantageous. However, for the model based on scaled data, it was useful. This is probably because of the smaller relative increase in the number of hidden neurons for the model based on normalized data (from nine to ten) than for the model based on scaled data (from two to four). When the number of hidden neurons is very small, every new neuron is crucial in improving the results, as quoted by [2]. 99 percent of the predicted speeds were within the range of ten-percent relative error for one month data.

4.3 Results for architectural decision studies

The number of hidden layer neurons, as discussed earlier can be increased by reducing the number of input parameters. But from Figure 4.1, it can be observed that the RMSEs remain constant after a threshold value of number of hidden nodes. Hence, an overly increase of the nodes will not increase performance substantially.

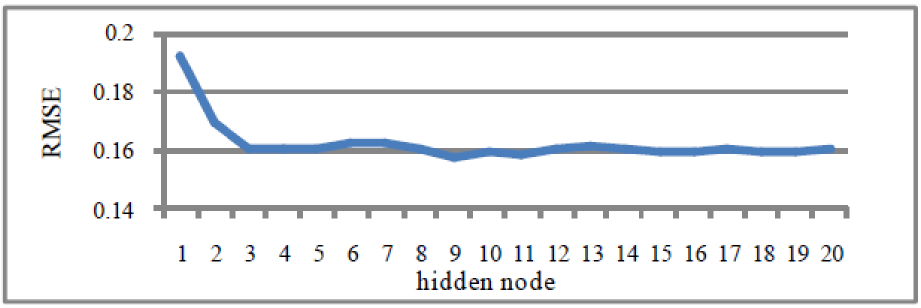

Figure 4.1 The relations between the number of nodes in the hidden layer and the root mean square error

The momentum parameter is the weight of each training cycle. The momentum was tuned by values ranging from 0.1 to 0.9. The optimal RMSE and accuracy were used to decide the best value, which according to Figure 4.2 is 0.2.

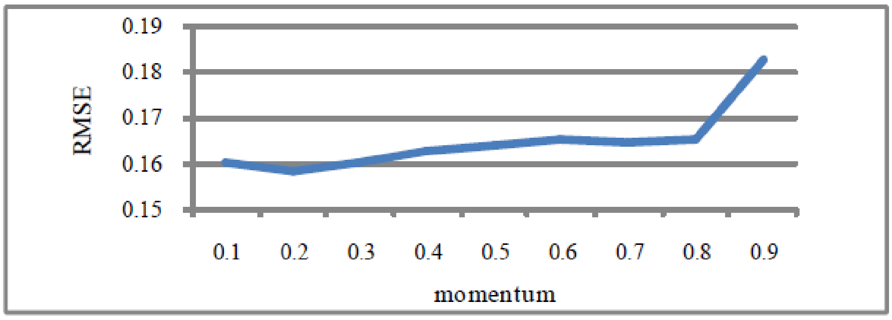

Figure 4.2 The relations between the momentum in the hidden layer and the root mean square error

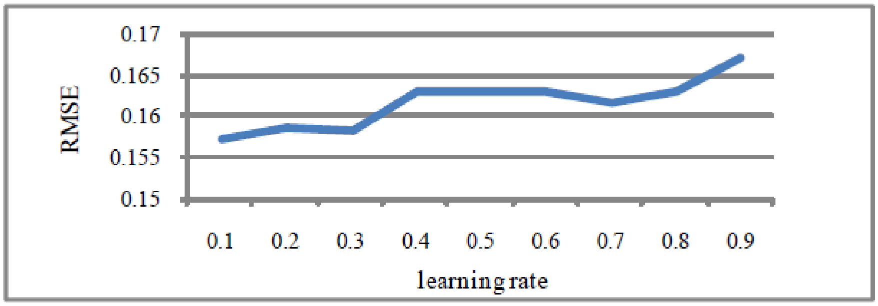

Figure 4.3 The relations between the learning rate and the root mean square error.

From figure 4.3, the optimal learning rate is 0.3. Learning rate is also called as 'alpha' in neural network parlance. This value indicates the amount of trial/error that may be required to adjust the weights to achieve minima during gradient descent. Higher the learning rate, the weights get adjusted faster. But a too high value would degrade the neural network and it will not generalize (i.e. learn the training set by-heart)

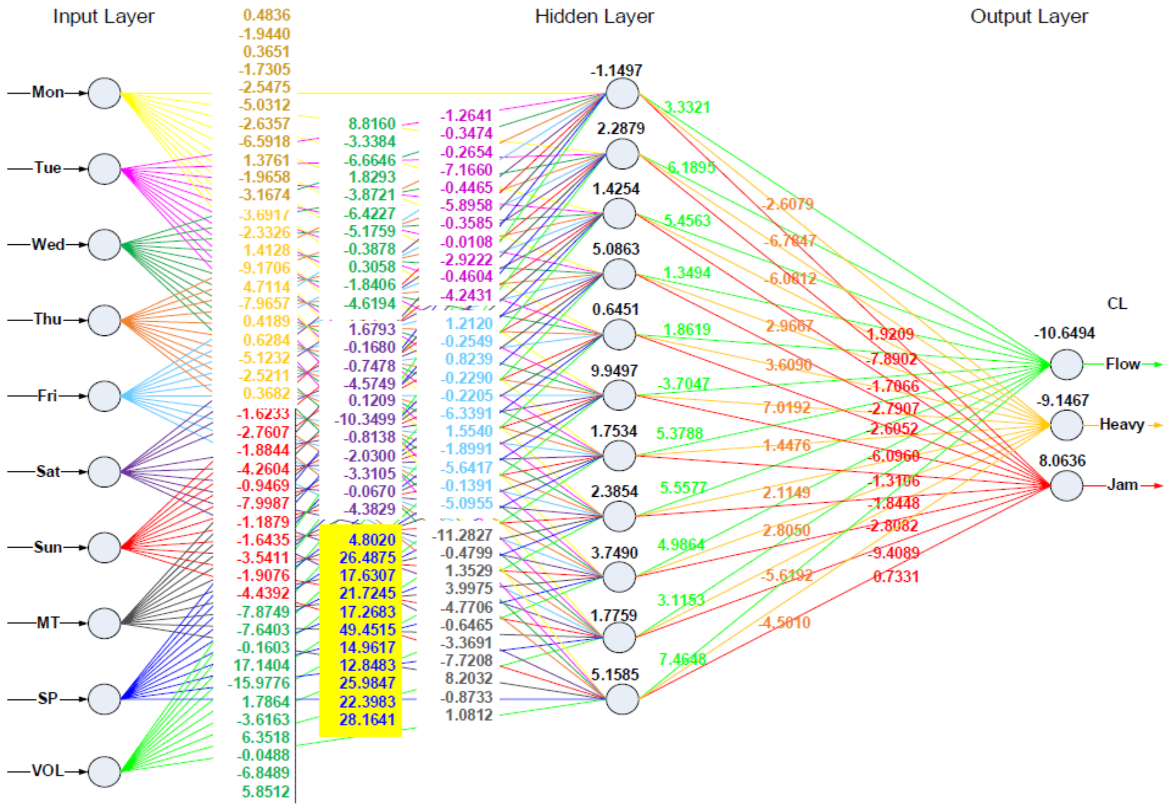

Figure 4.4 A learned MLP network to predict congestion level

Figure 4.4 depicts a learned MLP network that also shows the weights (momentum) that ranges from 0.1 to 0.9 in value. The weights which were initially set to random values were adjusted during backpropagation. The learning rate (alpha) was set to 0.3

THE OPTIMIZED NEURAL NETWORK CLASSIFICATION ACCURACY

Accuracy	Correctly Classified %			RMSE
	Light	Heavy	Jam	
94.99	99.60	72.40	82.30	0.1583

Table 4.2 Results for MLP network of Figure 4.4

Table 4.2 displays the RMSE and accuracy of the congestion predicting network described in figure 4.4. The accuracy achieved was 95%. Table 4.3 Confusion matrix is shown below.

		Artificial Neural Network (ANN) Classification (%)		
		Light	Heavy	Jam
Occupancy Ratio (OR) Classification (%)	Light	83.66	0.30	0.11
	Heavy	2.43	0.00	0.00
	Jam	12.04	0.41	1.05

Table 4.3 depicts the confusion matrix between the classified congestion levels using the MLP prediction model and the occupancy ratio (OR) technique. OR implies the volume of traffic flowing at certain instance. The results of both approaches were fed into the confusion matrix for similarity based analysis. 84.71% of traffic was correctly predicted, which is the sum of the values in the shaded cells.

4.4 Comparison of MLPs with RBF networks

RBF network are also a type of neural networks that has a feed-forward structure consisting of two layers, nonlinear hidden layer and linear output The hidden layer activation function is Gaussian kernel. In this section, a variant of RBF networks are discussed. RBF network training has two-stage procedure. In the first stage, the input data set is used to determine the center locations (*cj*) using unsupervised clustering algorithm such as the *K*means algorithm and choose the radii (*σj*) by the *k*-nearest neighbor rule. The second step is to update the weights (**W**) of the output layer, while keeping the (*cj*) and (*σj*) are fixed. (feed-forward). The Gaussian Kernel takes into account the center of the neuron

The RBFs work on the principle of locality, and the basic intuition that examples with similar attributes usually are more alike than examples with dissimilar attributes. Traditional RBF networks have a constant σ for each RBF neuron. This σ is similar to the learning constant, and is based on the experience of the network builder. The k-means algorithm makes a deterministic number of clusters indicated by a command line argument. The centers of each cluster are fed into Gaussian Kernel). The stopping criterion for the k-means clustering algorithm was 3000 iterations or convergence whichever came first. Classical RBF networks do not perform as well as MLPs for this use-case, but k-means based RBFs had a slightly better performance. The k-means center based RBF network performed marginally better than the multilayer perceptron networks. The best performing multi-layer network had an accuracy of 24.14 percent, while the k-means center based networks had an accuracy of approximately 27 percent.

Chapter 5 Conclusion

When considering real-time short-term prediction systems, the effectiveness depends, mostly, on predicting traffic information in a timely manner. A prediction system should be able to generate accurate single step ahead predictions of traffic flow. SP (Speed) was observed to be a deciding factor in congestion perception and hence the AI learning technique depends heavily on SP.

The dynamic nature of given problem works well with neural networks as they can adapt well to moderately noisy data and need no prior information apart from the training data set to start automatic classification. Once optimum values are chosen for the network architecture parameters, the neural network performs its magic.

The following points were observed in this study of suitability of MLPs in dynamic traffic prediction.

- The ability for a neural network to generalize to the problem space helps assure that this network reacts well to unseen data, and does not over fit to the training data set. This is maintained provided the learning rate is not set to a high value, or the number of neurons is very low.
- The most efficient training algorithm proved to be the Levenberg-Marquardt Backpropagation, as it surpassed the others not only in quality of results but also in speed.
- The number of nodes used in the hidden layer depends on the type and the complexity of the time series, and on the number of inputs. Reducing the number of inputs directly provided the advantage of increasing the number of hidden neurons.
- One of the major limitations associated with ANN modeling is its black box type nature. In time series modeling and fuzzy logic we can find cause and effect of each of independent variable, but in ANN framework, this is not possible. It is not possible to find the mutual interrelation between the variables in ANN modeling.
- The weighting priorities of the input were speed (km/h), traffic volume (car/min), the time of day, and the day of week, to get better accuracy in prediction.

- Based on the results, it was better to increase the number of hidden neurons by reducing the input parameters by decreasing the number of cross-sections rather than by shortening the time-series.

- The models that were divided into two sub-models – one for the mean speed forecasts and the other for the traffic flow forecasts – gave better results than one single model predicting both variables simultaneously. In fact aggregating into a fifteen minute model provided better results than handling five minute interval data which fared poorly.

- For 90 percent of the predicted flows the relative error was 20 percent at most, and for 90 percent of the predicted speeds it was four percent at most.

- Hence it can be concluded that, many other transportation data prediction studies can be implemented easily and successfully by using the different ANN architectures.

Chapter 6 Future Work

Neural networks provide a wide variety of freedom to explore for any given problem, considering the number of parameters of the network architecture that need be decided upon. A difference of even 0.1 in the learning rate can significantly alter the results.

It should be mentioned that these prediction models were done with only a small amount of data which span up to a maximum of three months period of observation. It should be noted that the number of peak period observations should be greater in order to make the model learn them better.

The performance also depends upon various factors like weather, festival season, accidents, local characteristics and driver's behavior. This data cannot be fed into the neural network directly. Other parameters like weather condition, seasonal variation in traffic flow and extreme condition (like accident or traffic jam) have not been taken into consideration in the present study. Since data considered in this study consists only off-peak hour's traffic which does not represent total variability in the traffic flow. A compound neural network partly solves the problem by taking this variability load into other data mining technologies like decision trees and association analysis, but many other factors would be left out in spite of this inclusion.

The influence of the factors related to the model itself has been studied. The factors related to the measurement arrangements and prediction period could be taken up as future work. Future study will also focus on collection of data sets over large periods to cover all possible realistic conditions associated with traffic flow.

Chapter 7 References

[1]. Satu Innamaa, Short Term Prediction of Highway travel time using MLP –Neural Networks, 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, 30 Sept. – 4 Oct. 2001

[2]. Satu Innamaa, Short Term Prediction of Traffic Situation using MLP- Neural Networks, Finnish National Research and Development Program on Transport Telematics Infrastructure (TETRA) 1998 – 2000

[3]. Bing Wu; Wen-Jun Zhou; Wei-Dong Zhang, "The applications of data mining technologies in dynamic traffic prediction," Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE , vol.1, no., pp.396,401 vol.1, 12-15 Oct. 2003

[4]. Pitiphoom Posawang; et al., Perception-based Road Traffic Congestion Classification using Neural Networks, Proceedings of the World Congress on Engineering 2009, Vol I WCE 2009, July 1 - 3, 2009, London, U.K.

[5]. Eleni Vlahogianni and Matthew Karlaftis (2010). Local and Global Iterative Algorithms for Real-Time Short-Term Traffic Prediction, Urban Transport and Hybrid Vehicles, Seref Soylu (Ed.), ISBN: 978-953-307-100-8, InTech

[6]. Bowu Zhang; Kai Xing; Xiuzhen Cheng; Liusheng Huang; Rongfang Bie, "Traffic clustering and online traffic prediction in vehicle networks: A social influence perspective," INFOCOM, 2012 Proceedings IEEE , vol., no., pp.495,503, 25-30 March 2012

[7]. Kranti Kumar, M. Parida, V.K. Katiyar, "Short Term Traffic Flow Prediction for a Non Urban Highway Using Artificial Neural Network", 2nd Conference of Transportation Research Group of India, Procedia - Social and Behavioral Sciences, Volume 104, 2 December 2013, Pages 755-764, ISSN 1877-0428

[8]. Gregory, Aaron L., "Prediction of commuter choice behavior using neural networks" (2004). Graduate School Theses and Dissertations. http://scholarcommons.usf.edu/etd/1056

[9]. Vedat Topuz (2010). Hourly Traffic Flow Prediction Using Different ANN Models, Urban Transport and Hybrid Vehicles, Seref Soylu (Ed.), ISBN: 978-953-307-100-8, InTech