UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING AND INFORMATICS

Automated Route Selection: Short Term Traffic Decision Support For Nairobi

BY

Okioga Edgar Nyamweya
P56/P/8066/03

Supervisor

Dr. Peter. W. Wagacha

June 2006

Submitted in partial fulfillment of the requirements of the Master of Science in Information Systems

Declaration

This Project is my original work and to the best of my knowledge has not been presented for a degree in any other university.

Signed ....

Date ....

Okioga, Edgar Nyamweya P56/8066/03

This project has been submitted in partial fulfilment of requirements for the Degree of Masters of Science in Information systems of the University of Nairobi with my approval as the University supervisor.

Signed ....

Date ....

Dr. Peter. W. Wagacha

Project Supervisor

Signed ....

Date ....

Mr. Eric. Ayienga

Second Supervisor

Acknowledgements

I wish to thank the following for the successful completion of this project. Firstly God, for His immense blessings, mercies and opportunity provided. Respect, gratitude and honour to my dearest wife and best friend Mogoi our daughter Rasoa for their continuous support, encouragement and prayers throughout my MSc studies. My parents and sisters for their moral support and words of wisdom: never ending in encouraging me to follow my goals.

A special note of appreciation to Dr. Eric Aligula for sharing with me KIPPRA data. Much gratitude to Dr. Peter Wagacha and Mr. Eric Ayienga of University of Nairobi, for their advice, critique, guidance and opinions which have shaped this research from an idea to a product.

I would also like to thank the entire MSc 7 group for their support and cooperation. I have no words to explain the role played by the Zebra group comprising of myself, James Kimuyu, Achoka Luduba, James Omwenga and Florence Mwende for the many hours they gave to the study group during my stay at Chiromo.

Last but not least, my appreciation to the entire academic staff of the School of Computing and Informatics for the knowledge they have imparted to me in the course of my studies in the School. I would also like to acknowledge the non-academic staff.

Thank you all.

Abstract

The city of Nairobi is currently grappling with the problem of rapidly increasing traffic, and its management. We have developed a prototype decision support system for short term traffic prediction and subsequent shortest path analysis for this City. We investigated on the use of artificial neural networks in time series predication and the application of the optimal A* search algorithm for the shortest path between two points. A geographical information system was used to visualize both the road network and optimal paths.

Topographical maps of Nairobi were digitised and a GIS topology build to support the A* search routine. For purposes of simulation, historical traffic data collected from Kenya Institute of Public Policy Research and Analysis was formatted, analysed and pre-processed using a sliding window time series and modelled using a feed forward back propagation artificial neural network.

The resulting network was used to predict one step-ahead traffic speeds. With the traffic speed and other road network parameters such as lane width and surface type just to mention, these values were then used to calculate the time taken to traverse a node or a link. In essence the actual length of the road was modified to a virtual length, while the speed determined from the ANN. The resulting time value was used to process the A* search routine resulting to an optimal path visualised on a GIS interface. For purposes of objectivity, the Dijkstra search routine was deployed to compare and contrast the two search routines (A* and Dijikstra) from a naïve perspective. A one week survey of existing road traffic speeds was conducted using a probe car fitted with a GPS. The average speed recorded for Nairobi was approximately 25km/hr.

A.I techniques can be deployed within the framework of GIS based decision support systems to fundamentally predict short term traffic congestion, simulate scenarios to enhance traffic management and help in creating policy for long term sustainability of infrastructure.

The A* search is effective for small networks as seen in Nairobi However, care needs to be taken in developing the heuristic component. If it is small, the A* decomposes to a greedy search and performs similarly to the Dijkstra's algorithm. Other factors need to be considered as identified in this research in fine tuning the A* search in terms of road characteristics and traffic influence for instance surface condition, location, width and gradient.

A critical generic component of a DSS is a visualization system or graphical user interface. As demonstrated in this report, GIS is critical in traffic management as a visual data mining tool. By visualizing the results of the search module, a user is able to asses the maturity of our road network and identify suitable routes to expand or build mechanisms to control traffic. The speed survey carried out identifies roundabouts as most critical bottle necks.

In conclusion, the city of Nairobi needs to deploy a traffic and route management system as proposed by this research. This will cut down the response time of emergency services and also warn people on identified routes of oncoming emergency vehicles and personnel thus create space. It goes without saying that data is not ready accessible in Kenya as experienced by the researchers. It is important for the government and academic institutions to partner in research and surveys to ensure that data collected is readily available for future research and analysis.

We have reported our encouraging findings here.

Table of Contents

List of Appendices

List of Figures

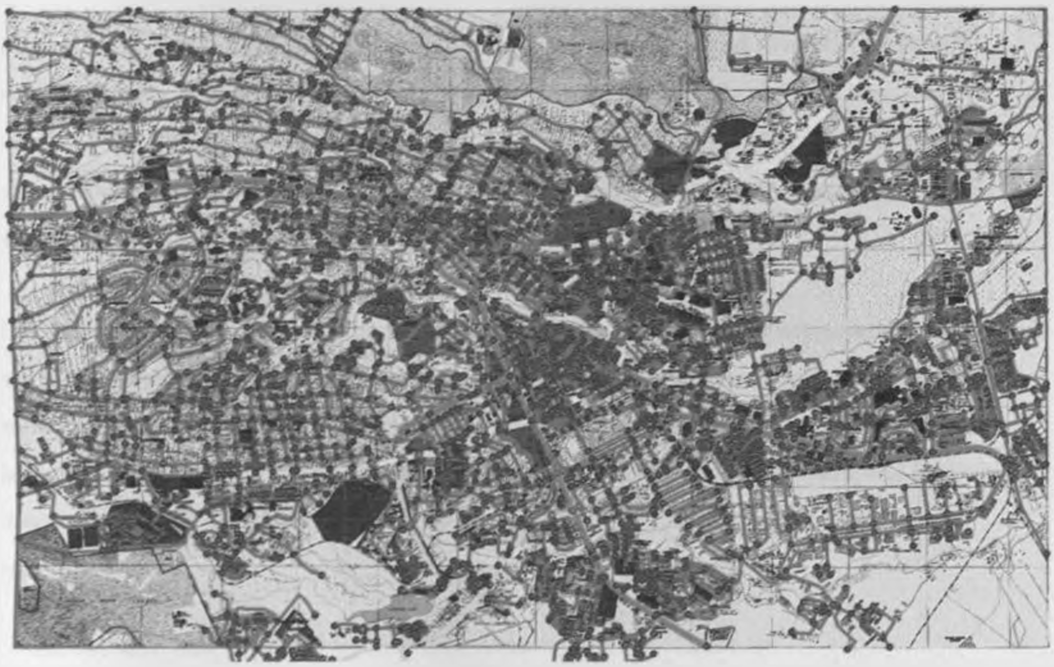
List of Tables

List of Abbreviations

A.I .... Artificial Intelligence
BPN .... Back propagation neural network
KIPPRA .... Kenya Insutute for Public Policy Research and Analysis
GOK .... Government of Kenya
FNN .... Forward feed Neural Network
MLP .... Multilayer Propagation Neural Network
NN .... Neural Network
A* .... A Star Search Algorithm
SQL .... Structured Query Language
VB.NET .... Visual Basic Application
RMSE .... Root Mean Square Error
MAPE .... Mean Absolute Percentage Error
DSS .... Decision Support System
GIS .... Geographical Information Management System
CBD .... Central Business District
TLB .... Transport Licensing Board
GPRS .... General Packet Radio Service
IEEE 802.1X .... Wireless Network Protocol
WAN .... Wide Area Network
LAN .... Local Area Network

1 Introduction

"..Driving from Buruburu to the city centre and back, a round trip of 14 kilometers costs Ksh 300 against a standard rate of Ksh 80. If 300,000 vehicles are involved, the loss is Ksh 66,000,000 daily. In one month, the loss is Ksh 1,980,000,000 and in eight months the loss is Ksh 19,808,000,000[1].."



[1] *Moses Chetambe, Webuye in an article appearing in the Daily Nation*

"….but the worst was yet to come. The city is grinding to a halt, and Kenyans are getting increasingly frustrated. For the poor country to spend 30 per cent of its foreign reserves importing oil, then burn the lot in traffic hold-ups, something must be terribly wrong."[2]

1.1 Background

1.2 Project Overview and background

A driver has to manage constantly changing road impedance (constraints) factors such as congestion, rough road, accidents which can alter the route of choice during travel impacting travel time, speed, and journey comfort to say the least. Optimal route selection (minimum cost, maximum speed, minimum time, and minimum risk e.t.c) by an individual driver on Nairobi road using historical and real-time information will need to account not only for the current impedance conditions but also for future traffic conditions which consequently impact the optimum route to follow.

A driver alerted with traffic forecasts in real-time is more likely bound to select routes with maximum utility of time, speed or opportunity to avoid any form of congestion resulting to decreased vehicle operating costs, pollution, road rage, incidents or a heightened level of risk exposure as in the case of transporting hazardous products though heavily human populated areas.

This project was a research into the prediction of traffic congestion and consequently automated real time optimization of route selection to maximize vehicle movement and minimize adverse impacts within the city of Nairobi. By utilizing machine learning techniques and geographical information systems, a prototype computer application was developed automate and optimize short term routing route selection with the goal of managing congestion in Nairobi central business district.

..."It gets worse when it rains.
Even the mildest of drizzles has the amazing ability to halt traffic on the road.
And when the president is on the move, everyone else has to stop and give way."[2]

1.3 Historical Background

Kenya has been categorized as a low income country with gross national income (GNI) per capita of dollars 765 or less in 2005 UNDP [HDR05]. Nairobi the capital of Kenya, serves as the seat of the government. It houses parliament, and all government ministries, diplomatic offices,

2 Peter Kimani in an article on the Daily Nation

the United Nations Environmental programme, Kenyatta National Hospital (the biggest referral hospital in East Africa) and the University of Nairobi just to mention. Nairobi has served as one of East Africa's important centers for commerce, industry, and tourism for many years. Historically, Nairobi's transport and communication network was developed to link the city to nearby countries, through road (national trunk roads), rail (Kenya Railways), and air (Jomo Kenyatta International Airport and Wilson Airport).

Nairobi is the headquarters of the national rail parastatal and is situated along the Great North Road that links the landlocked Uganda, Sudan, Rwanda, Burundi and other countries to the port of Mombasa. Uganda and South Sudan, in particular rely heavily on this transport corridor for international trade and relief aid to the later. This road link passes right through Nairobi. Plans are under way to develop major bypasses and link roads to decongest Nairobi by diverting commercial trucks away from the central business district.

1.3.1 Urban Growth

Kenya's population has steadily grown from 13.5 million in 1975 to an estimated 32.7 million in 2003. It is projected to be 44.2 million by 2015 [HDR05]. Urban population has also been growing steadily at a higher rate. In 1975, the total urban population was 12.9% with an estimate of 39.3% in 2003 and a projected growth to 51.8% by 2015 as reported by Transport Research Laboratory [TRL02]. This will result to an increased need for an efficient and effective transport network system or management policy for Nairobi and other urban centers.

Nairobi has experienced similar growth if not more resulting from rural urban migration, expansion of fringe areas and development. In 1963, the population was about 350,000 inhabitants, and this number increased to 828,000, 1,325,000 and 2,137,000 in 1979, 1989 and 1999 respectively [TRL02]. The high rate of population growth in Kenya, increasing demand for employment in the region will fuel continued rapid growth in Nairobi during the foreseeable future as reported by Post, Buckley International Inc [PB99]. According to the City Planning Department, Nairobi will continue to increase in population to 3,460,582 in 2007 however, the actual yearly percent increase in population is projected to steadily decline from 5.51% in 1997 to 4.95% in 2007 [PB99]. Within the next 10 years, nearly 1.3 million new residents will settle in the expanding metropolitan area, specifically the fringe areas where land is relatively affordable, generating more and longer trips than presently [PB99]. This is characterized by the rapid and huge housing development projects along Mombasa road (Mavoko/Athi River)

2 Peter Kimani in an article on the Daily Nation

Nairobi serves a large population from the neighboring districts on a daily basis for work, schools and college, hospital, industries and business. This is illustrated in Table 1-2:

Population of Kenya, Nairobi and other nearby districts.							
	1979	1989	1999				
			Male	Female	Total	Area (Km^2)	Density (per/Km^2)
Kenya			14,205,589	14,481,018	28,686,607	581,677	49
Nairobi District	828,000	1,325,000	1,153,828	989,426	2,143,254	696	3,079
Thika District			323,479	322,234	645,713	1,960	329
Kiambu District			369,101	374,909	744,010	1,324	562
Muranga District			164,670	183,634	348,304	930	375
Kajiado District			206,353	199,701	406,054	21,903	19
Machakos District			442,891	463,753	906,644	6,281	144

Table 1-1 Population of Kenya, Nairobi and other nearby districts. Transport Research Laboratory [TRL02]

From table 1-1, neighboring districts are experiencing rapid growth which bears direct impact on the city with regard to heavy traffic flows during peak hours and supply of services which cannot meet daytime demands [TRL02]. Traffic demand outstrips the available resources notably infrastructure, human resource, and traffic control facilities.

1.3.2 Nairobi Road Network

The initial layout of Nairobi main road system was reasonably well planned and spacious, but its' development has not been able to keep up with the explosive growth of population and vehicular traffic [TRL02]. The current government has put forward initiatives to establish and implement a number of bypasses or circular routes through which long distance traffic can avoid the central business district. Going by travel demand forecasts, road network performance, and little or no emphasis on traffic management, the average travel speed will continue to decline even with the implementation of committed road improvements, based on present land use and demographic trends [PB99]. This will result to greater traffic delays, higher vehicle operating costs and more conflict between motorized, street pedestrian and non-motorized traffic. Results of a survey carried out in 1998 by the Ministry of Local Government reveal there are approximately 300km of main road and 850 km of access roads in Nairobi, including unpaved earth roads of which much is in a deteriorated condition.

2 Peter Kimani in an article on the Daily Nation

1.3.3 Nairobi Traffic Volumes and Composition

Data collected along some 6 major entries and exists from Nairobi central business district in May 2002 [PB99] suggests there are less mass-hall vehicles compared to smaller or individual vehicles. Uhuru highway has the most private cars but with the least small to mini buses. On the other end of the spectrum, Muranga road has less private vehicles and more small to mini buses. These statistics have a direct bearing on traffic congestion patters as experienced today.

1.3.3.1 Vehicle Population

Historical vehicle populations in Nairobi are 230,478 in 1989, 274,820 in 1994, and 320,072 in 1989 [TRL02]. A comparison of the historical and future conditions is as presented in Table 1-4 by [PB99]:

Network Operation Characteristics			
Year	**Vehicle Kilometers Traveled (VKT)**	**Vehicle hours of travel (VHT)**	**Congested Speed: network Travel speed (KM/H)**
1998	3,276,910	84,571	50.55
2003	3,155,947	78,922	50.31
2013	3,819,966	107,656	46.32

Table 1-2 Network Operation Characteristics Post, Buckley International Inc [PB99]

From the Table 1-2, it seen that speeds decrease due to greater congestion as characterized by the VKT and VHT. VKT and VHT increase since additional vehicle throughput cannot be handled over the existing network. The net effect is longer period of congested travel over the network, slower travel speeds, higher vehicle operating costs, driver passenger frustrations leading to road rage and increased environmental degradation due to both noise and air pollution. As forecasted by the study, a 2.3% increase in average travel time from 1998 and 2003 and 3.5% increase from 2003 to 2013 are expected. Clearly, the issue of congestion and travel delay is of great concern today and will get worse in the future if not addressed.

1.3.3.2 Transportation Studies

There has been a number of transport planning initiatives for Nairobi City, however little implementation of resulting recommendation undertaken. Past plans are thus of interest primarily from the perspectives adopted and the extent to which these address the needs of the urban majority as shown in table 1-3:

Transport planning studies		
Study	**Year**	**Underlying Principles**
Nairobi Metropolitan Growth Strategy	1973	• Use of Bus ways
Nairobi Long-Term Transport Study	1989	• Improvement of origin destination movement • Improvement of pedestrian journey

2 Peter Kamau in an article on the Daily Nation

Transport planning studies		
		• Current and Future traffic demand management • Environmental impact management • Road network improvement • Organize and coordinate bus and matatus system • Increase traffic transport speeds, reducing congestion and improving safety
Current Urban traffic situation KIPPRA[2]	2004	• Impact of new traffic regulations on vehicles and commuters.

Table 1-3 Previous Transport planning studies Transport Research Laboratory [TRL02]Post, Buckley International Inc [PB99]

1.4 Problem definition

"Once upon a time, when traffic lights governed motorists and roads bore less portholes, motorists' greatest agony was whether they would find lights and side-mirrors of their parked cars intact" [2]

The Nairobi road network comprises more than 2000 km of road ranging from national highways linking Nairobi to other parts of the country, to unpaved earth tracks providing access to individual properties [PB99]. To minimize vehicle-operating costs and maximize utility of movement, a driver would benefit from an automatic route analysis for decision support.

To achieve optimum and environmental friendly vehicle usage, a pre-trip travel advisory, on road route guidance and route selection will be of need. The Ministry of Local Government has embarked on road bypass constructions and privatization of roads of which the taxpayer will pay for. Major investments in new bypasses or other expensive road infrastructure projects would be deferred in favor of a program of traffic management which is not capital intensive or with adverse environmental impact [PB99]. The Table 1-4 identifies affordable urban transport strategies as a basis of the research problem:

Urban Transport Strategy Matrix			
Urban Transport sub-sector	**Issue or problem category**	**Problem diagnosis**	**Recommended Strategy**
Road network	Road congestion and delay	Inadequate management of existing road capacity	Implementation of a traffic circulation and priority plan
	Minimize exposure of hazardous road shipments.	The railways infrastructure is poor and management wanting. Shipments of hazardous products done via road network.	Determining the optimal route for shipment that minimizes both travel distance and population exposure along the route
	Road finance and management priority	Management of rehabilitation sequence and priority	Implement balanced network management and prioritized rehabilitation programme
Traffic Management	Traffic Flow and	Poor management of existing	Implement real time advisory

[2] Kenya Institute of Public Policy Research and Analysis
[2] *Peter Kimani in an article on the Daily Nation*

Urban Transport Strategy Matrix			
Urban Transport sub-sector	**Issue or problem category**	**Problem diagnosis**	**Recommended Strategy**
	Enforcement	road space Lack of equipment for effective traffic management	
	Demand Management	Rigid work rules causes concentrated AM and PM traffic in the CBD Unrestricted peak period vehicle access in major corridors results in high volumes and low service levels	Implement flex-time/staggered work hours that extend the traditional work hours of 8am-5pm to night. Implement bus priority measures.

Table 1-4 Urban Transport Strategy Matrix Post, Buckley International Inc [PB99]

Currently, drivers depend on experience, intuition, radio traffic 'infomercials'[3] and traffic police to direct traffic: all of whom do not have hard facts on the global traffic status nor the future condition. As much as there are attempts by all entities to optimize their local situation, global optimization of the road network can not be achieved.

> *"Nakuru is 200 kilometers away and one needs about two hours to get there, the same amount of time that a commuter needs to get to town from Umoja, one of the estates in Nairobi's Eastlands, just about 10 kilometers away."*[2]

1.5 Proposed Solution

The solution to the problem of automated traffic routing and management is a hybrid artificial intelligence and GIS based decision support system.

This solution has an underlying database managed by a relational database management system (SQL Server 2005). The database contains three basic themes of data: Neural network data, road traffic count survey data and road network data.

A graphical user interface (GUI) with GIS forms the primary means of communication and data exchange. Built within the GUI is a neural network to learn the underlying traffic survey data and thereafter predict traffic speeds based on historical speeds. Closely coupled to this is a route search module which identifies the shortest path between any two points.

[3] Advertising used to send critical information to radio broadcast receivers

2 Peter Kimani in an article on the Daily Nation

In summary, the proposed solution uses the following:

- A geo-referenced road network of Nairobi,
- An artificial neural network to predict vehicle travel speeds at selected road intersections and identified as a node.
- A route analysis module employing the A* search algorithm.
- A GIS subsystem to handle spatial queries, analysis and visualization
- A relational database as a persistent data store for GIS , neural network data and traffic count survey data,
- A reliable operating system which supports multithread applications.

1.6 Project Objectives

"If people elsewhere measure distance by the amount of time needed to drive through, then such yardsticks are useless in Nairobi and its environs. "
Peter Kimani

The research objective was to build a prototype automated route selection system from short-term traffic prediction in Nairobi. The prototype uses data available from KIPPRA to simulate actual operations envisioned. To help achieve this objective, a set of questions were posed to define the boundary of the research area.

Main research questions

1) Is artificial intelligence capable of solving traffic management problems in Nairobi as an alternative to capital investment on road construction?
2) What framework is suitable for building an A.I, GIS decision support system for road network analysis?

Minor research questions

1) How effective are artificial neural networks in predicting road traffic congestion in Nairobi?
2) Is the A* search algorithm an effective and efficient algorithm in automated traffic management and decision support?
3) How critical is visualization and spatial analysis in modern intelligent traffic management systems?
4) What role can artificial intelligence play in urban development and planning?
5) How can the Nairobi City Council better manage emergency services?
6) What is the state and accessibility of data in Kenya with respect to traffic management?.

"Don't mention Jogoo Road, at least not from Landhies Road which serves countryside bus terminus commonly known as Machakos airport. Here, people find it faster to walk than ride or drive into the city. The same is true of side roads that feed into the city; by and by, the city is getting clogged."[2]

[2] Peter Kama

1.7 Project Hypothesis and Theory

Three sets of hypothesis were used to guide the development and subsequent conclusion of the project:

1) *Neural networks as a universal function approximator can predict vehicular speeds on a road network given previous historical speeds and time.*

2) *Travel time as a the cost of traversing a road segment or junction as a function of impedance due to speed, congestion, width, surface type and security can be used as the basis of a heuristic search (A*) in automatic route selection and decision support at a predicated time in the future.*

3) *Automated route analysis can be reduced to a static route analysis problem when time spans are taken and each time span has the road network at equilibrium.*

1.7.1 Theoretical underpinning

- Neural networks are universal function approximators. This property is used in predicating travel speed given historical traffic speed parameters and associated environmental and temporal factors.
- Heuristic functions do not have theoretical underpinning but by experimenting and visualization one can observe phenomena caused by heuristic functions. The properties of the A* search (an admissible function) is used in determining the optimum route of vehicular movement at any time in the future with the aim of reducing congestion. For a good introduction on the A* search algorithm, refer to Russel and Norvig [RN95].
- Studies in equilibrium state that a system will attain equilibrium and a route not in use does not have a higher utility than the current in use.

1.8 Project Justification

"And for those using Mombasa Road, a change of clothes might come in handy when everything stops and you are stuck there for days "[2]

Traffic congestion occurs when the volume of traffic on a roadway is high enough to become detrimental to its performance. In congested conditions, vehicle speeds are reduced, increasing drive times. These conditions are also frustrating for drivers leading to road rage and automobile accidents. Furthermore, vehicles burn unnecessary fuel when on idle. A period of extreme traffic congestion is known as a traffic jam . Traffic congestion is synonymous to Nairobi roads. Road rage is not alien to Nairobi as seen in the shooting of a matatu driver in July 2004 by Professor A. Obel inventor of Kemron an AIDS miracle cure.

2 Peter Kimani in an article on the Daily Nation

Road rage also known as road violence is the informal name for deliberately dangerous or violent behaviors under the influence of heightened, violent emotion such as anger and frustration, with regard to the use of automobiles. Frustration with the road condition, along with perceived inconsiderate actions by other drivers results in a heightened emotional response (anger).

The traditional approach to reliving congestion is to build more and larger roads. However, for a variety of reasons as will be explored, this approach is no longer viable. Instead, the focus now is on improving the management of existing infrastructure.

1.8.1 Nairobi Congestion causes and solutions from pundit journalistic view

The print media has highlighted a number of articles and editorials on the issue of road congestion problems facing Nairobi and offered solutions, most of which have no bearing to any authoritative study done in recent history. Politicians have also had a fair share of ideas on the causes of road congestion with solutions not grounded in actual research. The table 1-7 is a summary of articles and views from various online resources on the matter of Nairobi traffic congestion.

2 Peter Kimani in an article on the Daily Nation

Table 1- 7 Selected Public Opinion on Road Congestion

Sample of public opinion on traffic in Nairobi	Cause of Congestion	Solution Offered	Remark
Raila explains delay in road works	• Matatus are the cause of numerous congestion problems • Dilapidated road network	• Road repairs • Expansion of selected roads to dual carriage way • Privatization of road network	The solutions offered are in direct contrast to those recommended by the Post, Buckley International Inc [PB99]which emphasized traffic management not capital investment
Transport Licensing Board in plan to ease congestion		• Land use change to parking • Stop licensing of matatus on already congested routes. • Privatization of bus parks	The solutions have been offered but the cause of the problems has not been explicitly mentioned. Some of the solutions are mentioned in the Nairobi Urban Transport plan 1998.
Taxis cause of parking nightmare	• Taxis waiting for commuters		A solution to the taxi conundrum is not offered. Else where in the world, taxis are not allowed to stop and park in the CBD.
City traffic boss, Masereti speaks out on law violation	• Dilapidated city rods • Number of vehicles has increased. • Traffic lights are only at a few junctions and traffic islands while some need police intervention • Poor enforcement of traffic regulations (Matatu menace, hawking)	• Report the offending motor vehicle to the traffic boss.	The solution aims are curbing road menace but does not address the issue of hawking on the streets, non functional traffic control equipment nor road condition. Post, Buckley International Inc [PB99]report suggests the enforcement of traffic regulations as a means of controlling traffic congestion.
For safety's sake, try double-decker busses	• Careless transport planning and lack of traffic management schemes • Public service vehicles are not scheduled and do not observe	• Capital and visionary long-term planning • Immediate impact by introducing double-deckers to maximize road space. • Bus scheduling • Enforcing traffic regulations and tests	The solution focuses on traffic management of public service vehicles and offers suitable affordable solutions that do not require heavy capital investment. The existing traffic rules are not being enforced as such there is much

2 Peter Kimani in an article on the Daily Nation

Sample of public opinion on traffic in Nairobi	Cause of Congestion	Solution Offered	Remark
	traffic regulation • Competition between non-motorized and motorized transport. • Lack of pedestrian facilities, hawkers, lack of traffic controls, poor education and attitudes. • Human and vehicular Population growth	• Punitive measures for repeat traffic offenders. • Legislature to control road safety and control.	need to realize that the rules do exist but machinery to enforce is missing. There needs to be a critical change of attitude in Nairobi drivers with regard to traffic and road manners.
City transport under threat	• Public service vehicles are not scheduled • Too many small operators • Private vehicles replacing public vehicles	• Promote the use of mass haul public transport • Introduce public transport schedules. • Provision of legal framework for public vehicle operations.	The solutions offered clearly embrace the concept of traffic management as an alternative to capital intensive road upgrade and development programmes.
Roads project must work	• Dilapidated roads	• Construct an elevated highway over Uhuru highway as well as other roads. • Initiate a concession programme and private firms participation in road development	The solutions offered are in direct contrast to those recommended from the Nairobi Urban Transport plan (1998) which emphasized traffic management not capital investment The use of concession might result to increased road use costs from tolls: might lead to further congestion on cheaper roads
Accidents reduced drastically, says report	• Small public transport vehicles • Lack of route management	• Phase out 14-seater matatus and replace with larger capacity vehicles • Introduce route management.	This is a viable solution that lends itself to the category of non capital intensive. As suggested by the Post, Buckley International Inc [PB99] report, traffic management and use of mass transport vehicles will ease traffic congestion both in the short and medium term.
Traffic: Blaming	• Lethargic approach to traffic	• Enforce traffic regulations	The solution offered is on the realm of attitude change of drivers.

2 Peter Kimani in an article on the Daily Nation

Sample of public opinion on traffic in Nairobi	Cause of Congestion	Solution Offered	Remark
minister unfair	management issues by government and policy makers.	• Overhaul traffic control points, islands and junctions.	There is mention of the need to overhaul traffic rules that do not apply or conflict with each other on Nairobi roads.
Traffic jams: "Why I accuse the planners"	• Exponential increase in number of vehicles w.r.t road capacity • Low carrying capacity of roads especially at junctions, lack of ring roads • Little or no compliance to traffic rules and regulations especially for matatu and other public service vehicles.	• Reduce the number of vehicles by ensuring they are more expensive to acquire or by insisting on a decent standard of maintenance. • Introduce slip lanes at junctions, addition of ring roads • Enforce the traffic regulation thereby instituting proper road attitudes and behavior	Majority of vehicles imported are small private vehicles whose duty is more often than not undervalued. Many financial institutions offer facilities for easy vehicle purchase in return keeping the log book as security making the process of owning a vehicle easy As for standards, only commercial vehicles are inspected hence control of private vehicles is not practical. Enforcement of the traffic act and regulation is a solution that can be immediately implemented. The solution offered by this article does indicate the potential of traffic management solutions arguably this might offer short reprieve.
Transport Policy urgent	• No implementation of past policies or recommendations from studies • Lack of a public transport management framework/policy • Urban growth	• Prohibition from venturing into the central business district to some vehicles (minibuses) • Introduce congestion charges • Introduce mass haul vehicles. • Expanding existing infrastructure • Adequate implementation of urban plans and recommendations of studies.	The solutions discourage the proliferation of small public transport systems, which will encourage the use of mass haul vehicles. Introduction of congestion charges will increase the cost of vehicle operation, but in the long run, ease the movement of the same. This has been suggested in the Post, Buckley International Inc [PB99] report. Expanding the existing infrastructure is a long term solution, but is capital intensive for short term gains and studies. There is need of government to implement results of studies and research carried out.

2 Peter Kimani in an article on the Daily Nation

1.9 *Project Motivation*

Nairobi is experiencing a rapid rate of growth which is estimated at 5% per annum with a resulting increase in urban travel leading to greater traffic congestion. Within the next 10 years, nearly 1.3 million people are expected to contribute to this. It is more than necessary to have a plan or methodology of traffic prediction analysis in order to better prepare for optimum resource usage.

- This project is timely in suggesting a short to medium term solution to congestion problems.
- By being able to predict congestion and route traffic, it would be possible to manage public transport routes, rates and schedules as such manage the role of matatus and other forms of public transport.
- Traffic management and congestion analysis results can be targeted toward increasing public transport vehicle operating speeds, reducing overall congestion and improving traffic safety. This will also enable improved traffic laws enforcement, affecting traffic flow and safety.
- Congestion has a number of secondary effects which are symptomatic of poor urban transport management. These include:
    - Pollution as evidenced by dying and dead vegetation by the side of roads,
    - Road rage and other minor traffic offences,
    - Numerous traffic accidents,
    - Spiraling vehicle operating costs,
- Outputs of this study, will avail opportunities for other researches to delve more into the psychological aspects of congestion, road related stress and traffic associated pollution.
- Nairobi is a growing metropolitan city. Its' population of plus 3 million will benefit from short to medium term congestion control with possible vehicle operating costs reduction and efficient road management. The findings can be replicated to other towns and cities.

2 Peter Kimani in an article on the Daily Nation

- Previous research has looked in to the cause of congestion with little emphasis on understanding the nature of congestion as a spatial temporal problem which is predictable and replicable.
- This study is a realization of a previous study Osoro [O05]. With the possibility of using the developed system for simulation.
- As an implementation of recommendations, other researchers can look into the possibility and cost benefit of using local technology to build, operate and maintain automated traffic counting equipment. This technology can be patented and sold as appropriate technology to other 3rd world countries.

1.10 Project Assumptions

"The rest of the sequence happened too fast: the "parking boys" graduated into "street boys," some armed with an arsenal of weapons that included human waste, and segments of the city became unsafe to venture through" 2

1. Road traffic is a dynamic phenomenon. For purposes of simulation, we can consider traffic to be static within some range of time to allow modeling and speed prediction using ANN.
2. The existing traffic count data collected by KIPPRA at 30 minute interval (2004) will suffice to model an artificial neural network to simulate actual working environment,
3. A typical road in Nairobi has a vehicle carrying capacity of 1,200 vehicles per hour.
4. The Greenshield linear model for macroscopic traffic flow based on limited data is a suitable mathematical representation of vehicle travel speed and road capacity.
5. Any generated data will be within 10% confidence of actual observable data for the purpose of building neural networks for road segments where data is available.
6. The traffic in Nairobi shows a stable pattern.
7. The speed of a vehicle within Nairobi is less than or equal to 50 km/hr as indicated in the Traffic Act.
8. In future, data collection will be via sensors sending data to a central database on real time.
9. The node and link resistance values can be randomly generated as a fair representation of the status on ground.

2 Peter Kimani in an article on the Daily Nation

1.11 Report Summary

The report is organized into 6 Chapters. Chapter 1 gives an introduction to the project and in depth historical view of issues afflicting Nairobi traffic management. The project report assumptions, theory, and justification are presented concluding with a succinct report summary.

Chapter 2 introduces forecasting in general with specific focus on vehicular traffic. A brief introduction is given on traffic flow theorem as developed by Greenshield which is the cornerstone of this project. An introduction to neural network models and architecture is presented with a more in-depth look to time series forecasting using neural networks is presented. A more detailed approach to neural network design in presented and looks at the development of neural networks as time series forecasting modules, use of lag windows to forecast time based events neural data processing, network training and control of over fitting. Since the feedfoward back propagation neural network model is used, a brief discussion is presented on BPN networks and the motivation on its selection as compared to other models. A succinct treatment of search algorithms is presented focusing on A* and Dijikstra search. A brief introduction is presented on role of decision support systems and GIS in the context of this project culminating with architecture requirements in road network topology design

Chapter 3 identifies the methodology used to conduct this study. This includes sources of data and actual surveys, questionnaires issued and system architecture. A detailed treatment is given on the process of digitizing the road network, design of the database and neural network. Data flow diagrams to level 3 are presented with an entity relation diagram of the database. A justification is presented on the use of MLP neural network as opposed to other models. Specific emphasis is given to the process of saving neural network weights and recall for the same since this is used in the speed prediction process. As the neural network is a core module of the system, a module test using IRIS dataset is presented with results proving the source code is correct.

2 Peter Kimani in an article on the Daily Nation

Chapter 4 presents the system implementation and integrity tests carried out. The hardware, software and implementation process is reported culminating with a proposed mode of use and training frequency of the neural network.

Chapter 5 presents the results obtained. This includes detailed neural network training results and following the training pattern introduced in chapter 3 and 4.

Chapter 6 concludes the research work and presents recommendations for further research.

Chapter 7 presents the references and bibliographies with appendices A to I containing supporting results and source code.

2 Peter Kimani in an article in the Daily Nation

2 Literature Review

2.1 Introduction

2.2 Traffic Forecasting

Traffic congestion is a predictable man made phenomena. Congestion is not a new pandemic afflicting modern man. Julius Caesar became so frustrated by traffic congestion that he banned the movement of carts during daylight hours Kornhauser et al [KB06]. In a bid to discover volume trends across time (peek hours) in metropolitan Atlanta, Georgia roads, Kornhauser et al draw the conclusion that traffic demand drops drastically during non peak hours of the day, yet providing efficient and affordable public transportation is extremely difficult. Travelers without traffic reports and forecasts are only able to plan a route then hope there is minimal traffic congestion on their chosen path.

In a related study, Florian [F99] presents a flexible modeling approach to alleviate current challenges in urban transport with related pollution. In this study, the issue of congestion is approached from the principle of equilibrium. Florian identifies environmental pollution as a fundamental objective in planning and understanding the phenomena of congestion and traffic [F99].

Sherif et al [SPC00] introduce an approach to optimize the short term traffic prediction performance using multiple topologies of dynamic artificial neural networks and various network-related and traffic related settings by deploying multi-modal approaches under parameters and traffic condition settings.

2.2.1 Forecasting Models

Prediction or forecasting is the process of generating information for the possible future development of a process from data about its past and its present development Kasabov [K98]. Kasabov identifies three different tasks to be distinguished under the generic prediction problem:

1. Short-term prediction – default meaning of the word prediction

2. Modeling, which is finding global underlying structures, models, and formulas, which can explain the behavior of the process in the long run and can be used for long-term prediction as well as for understanding the past.
3. Characterization, which is aimed at finding fundamental properties of the process under consideration, such as degrees of freedom.

Challenges in solving prediction problems requiring resolution before any serious prediction work is carried out [K98]:

1. Determine if the process id predictable.
2. Determine the type of data available and the process subject to prediction.
3. Determine the right features for presenting the prediction problem.
4. Identify amount of historical data required for a good prediction.
5. Identify a methodology which will be used to test the accuracy of the prediction.

No single method is expected to be the best method under all circumstances to provide travelers with individual travel time information for a limited part of a route Versteegt et al [VT03]. By reviewing the state of the art of travel time prediction methods carried out to obtain insight into the strength and weaknesses of existing methods, Versteegt et al [VT03] categorize prediction into two classes: explanatory based (simulation) and extrapolation based (statistical based). In order to justify their approach, Versteegt et al compile a prediction horizon shown in figure 2-1 illustrating the predictive strength of various methods from statistical regression to neural network system.

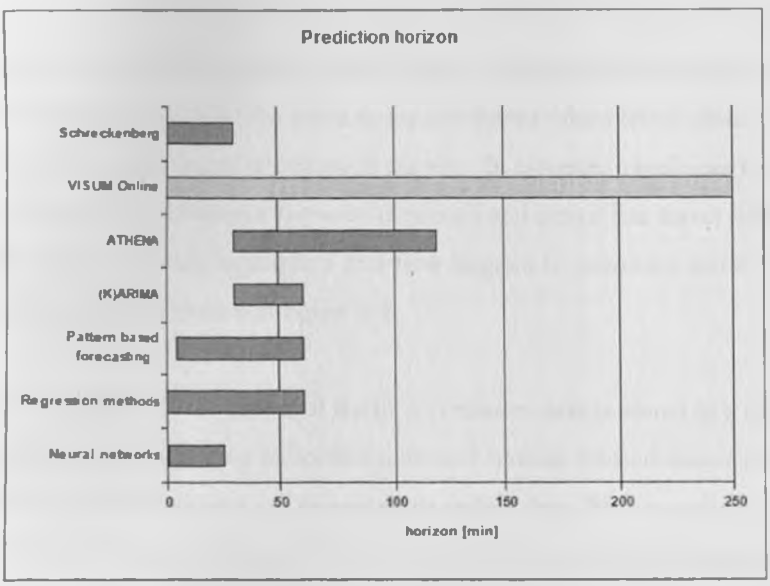


Figure 2-1 Prediction horizon showing predictive strengths of various methods.

From figure 2-1 , Versteegt et al [VT03], conclude that :

- Most methods have a threshold in terms of prediction horizon,
- Horizons overlap by several methods
- No single method is best

Jacobs [J03] deploys a feed forward artificial neural network to predict average speed in Holland from loop detector data. By using a lag window of historical speed from a loop detector, Jacobs is able to predict a single step ahead forecast of traffic. Yasdi [Y99] demonstrates the effectiveness of a neural network system for prediction of congestion using traffic volume. Unlike Jacobs [J03] whose neural network parameters include time, day of week, month, weather, holidays and events, Yasdi [Y99] only uses traffic volumes to predict congestion.

You et al [YK00], propose an architecture and data flow for a system to predict travel time. In addition to this, they review various technologies that have been used for developing a travel time forecasting model with geographic information systems (GIS) technologies to be employed for location based services. You et al [YK00] conclude that data to estimate travel times is delayed information due to the dynamic nature of network traffic as such cannot be guaranteed to be a true representation. Park et al [PSHJ05] shows that for certain future time periods (e g. 60 minutes later from now) travel time forecasting using only the historical profile without real-time profile is better than one using real-time profile or both.

Without exception, a traffic conditions change rapidly and dynamically as time goes by, thus traffic conditions cannot be the same as the conditions when travel times are initially estimated in traffic management information centers. In essence, travel time forecasting models could reduce the difference between estimated and actual link travel times. You et al [YK00] diagrammatically highlight a data flow diagram to predicted travel times and calculate shortest paths as shown in figure 2-2.

From figure 2-2, a database of historical traffic surveillance data is stored in a data warehouse. Online, real-time data collection tools and systems (closed circuit television, GPS probe cars, loop detectors e.t.c) continuously collect data. Both historical and real time data is staged though a process of data fusion and presented to a forecasting system. GPS data also verifies and fine-tunes the prediction system's results. From the traffic prediction results, the shortest path is determined.

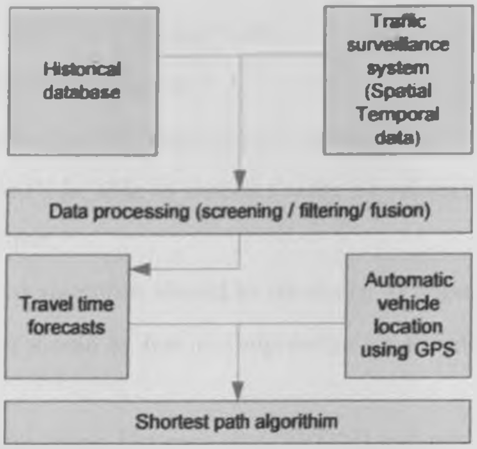


Figure 2-2 Predict time and determine shortest path adopted from [YK00].

Kisgyergy et al [KR02] build a real time travel time prediction model for the freeway network of San-Antonio based on information collected by the loop sensor , GPS system, and later use the predictions for traffic management and advance traveler information systems. Kisgyergy et al [KR02] use a multi-step ahead feed forward neural network for prediction which outperforms as a single step neural network.

Fu-Sheny et al [FJ61] consider use of ANN for modeling and controlling traffic flow on the macroscopic level to accurately model the dynamics of the traffic flow and then control the traffic density hence force it to follow a desired pattern. Lingras et al [LS99] survey automatic traveler information systems and related research including various approaches used to predict traffic volumes in the short-term future. The ability to manage traffic congestion has a direct relationship with how accurate forecasting of traffic conditions in the short-term future. Optimization of driver route selection in response to future real-time road and traffic conditions can help alleviate road congestion and its associated problems. Lingras et al identify critical features in such systems as listed:

1. Route guidance system should use minimum and maximum values of projected travel times in determining the best route for a particular trip.
2. Typical input for traffic prediction should include:
    - Previous traffic data (avoid late night and early morning)
    - Modeling based on data of the same day historically.
    - Impact of events based on similar events in history (previous year)
    - Incident data to be used

3. Characteristics to aim at in the final model:
   - System should be adaptive,
   - System should be able to process time series,
   - System should be able to output a range of values as opposed to a single precise output,
   - The forecast algorithm should be resistance to noise,
   - The system should be fast in computation speed, and result formulation.

The benefits of traffic optimization Praween et al [PVD03] and route management can result in:

1. Reduced road travel time and cut down on unnecessary journeys,
2. Reduced stress levels,
3. Congestion avoidance,
4. Avoidance of unsafe driving conditions,

Finally, Demetsky et al [DMSS98] explore the potential for using case-based reasoning, on emerging artificial intelligence paradigm, to overcome real-time traffic flow routing congestion problems. This research develops a prototype CBR routing system, for the interstate network in Hampton Roads Virginia. CBR has the potential to overcome real-time routing and congestion management thus leading to significant user cost saving. It can be concluded that:

- Managing traffic flow through real-time guidance has emerged as one of the promising approaches to alleviate congestion.
- An effective traffic management decision support system must be able to function in real time. As soon as traffic conditions change such as when an incident occurs, routing strategies must be revised to mitigate the effects.

2.2.2 Types of Data Used

Different researches have used different parameters to predict congestion using a neural network. The table 2-1 shows a list of past research and primary data type:

Table 2-1 Past researches and data used in traffic prediction using neural networks

You et al [YK00]	GPS location data, travel speed and travel time
Kisgyergy et al KR02]	Travel speed, vehicle occupancy, traffic volume, GPS location

	data
Fu-Sheny el al [FJ61]	Vehicle flow rate, jam density, vehicle travel speed
[Yasdi [Y99]	Day of the week, traffic volume
Jacobs[J03]	Time, day of week, month, weather, holiday, events, vehicle speed.
Park et al [PSHJ05]	Historical travel time, cost , travel speed
Kisgyergy et al [KR02]	Vehicle speed, road occupancy, traffic volume, GPS location, time

The common data element collected is travel speed, traffic flow and density. Studies in traffic flow behavior have shown that the three parameters (speed, density and flow) describing uninterrupted traffic stream are pair wise dependent.

2.2.3 Traffic Volume (Flow)

Traffic flow is one of the fundamental measures of traffic on the road system. Since traffic is composed of a number of vehicle types for instance Lorries, trailers, busses, the traffic volume is normally converted into equivalent passenger units (PCU) by using vehicle equivalent factors. The table 2-2 is a sample derived from KIPPRA for converting vehicle counts to equivalent PCU.

factor	Car	Matatu	Bus	Lorry
Car	1.00	1.50	2.00	2.50

Table 2-2 Equivalent passenger units from KIPPRA data

The passenger unit of a vehicle depends upon the size and the speed of the vehicle, type and kind of road environment. They are generally not dependent on the surface type and road width Okioga [O04].

2.2.3.1 Importance of "flow/volume" in highway transport studies

Volume counts are used as parameters to establish:

- Relative importance of a given road in traffic system
- Variations in levels of traffic flow over time
- Extent of the utilization of a facility in terms of its capacity to carry traffic
- The distribution of travel demand in a network.

2.2.3.2 Type of volume counts and their uses

The figure 2-3 shows the different types of counts based on durations of study.

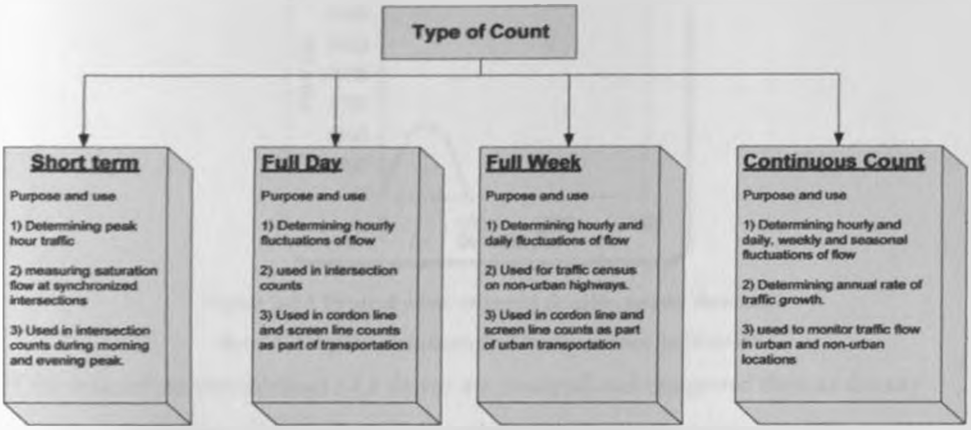
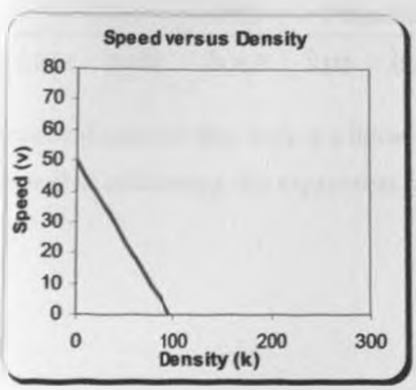


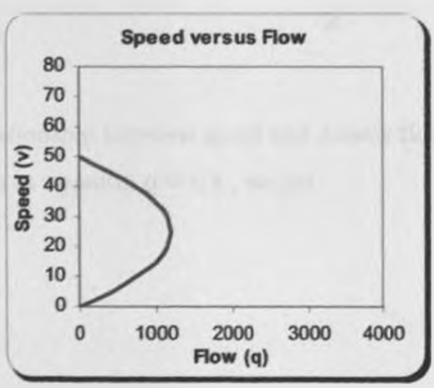
Figure 2-3 Type of Traffic Counts

From figure 2-3, most studies in Kenya are continuous counts and are carried out by the Ministry of Transport. The Kenya Institute of Public Research and Policy carry out short term while the Ministry of Local Government under the Kenya Urban Transport Infrastructure Programme carry out full week. Multilateral donors like the European Community and the Japanese International Aid carry out short term studies for selected studies.

2.2.3.3 Greenshield's Model

Studies in traffic flow behavior have shown that the three parameters (speed, density and flow) describing uninterrupted traffic stream are pair wise dependent. There exists a relationship between speed and density, flow and density and speed and flow Partha et al [PA].Figure 2-3.1 shows typical plots of speed-density, speed-flow and flow-density data.





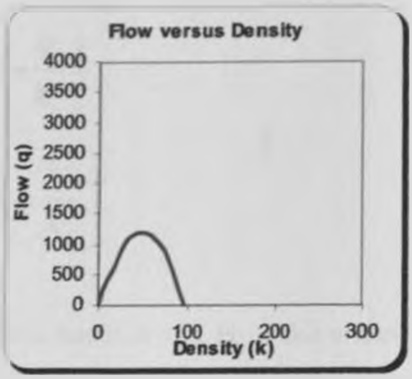


Figure 2-3.1 Typical plots of speed density, speed flow and flow-density for uninterrupted traffic stream in Nairobi

If the immediate surroundings of a driver are cramped and congested then as density increases, drivers for safety reasons reduce their speed and vice versa. In 1930, Greenshield based on limited data was able to develop a model of uninterrupted traffic flow that predicts and explains the trends that are observed in real traffic flows. While Greenshield's model is not perfect, it is fairly accurate and relatively simple linear model. Greenshield made the assumption that, under uninterrupted flow conditions, speed and density are linearly related.

The fundamental traffic theory states that the three basic variables of traffic namely, its flow, speed (space-mean speed) and density have a relationship with each other so that

$$q = uk \qquad 1$$

The units of u if given in vehicles per hour and density if given in vehicles per lane per km, would yield a flow rate given in vehicles per lane per hour.

$$\frac{vehicles}{lane \times \text{hour}} = \frac{kms}{hour} \times \frac{\text{vehicles}}{\text{km} \times \text{lane}} \qquad 2$$

Greenshield assumed that there is a linear relationship between speed and density flow, expressed as substituting this expression for u in equation q = u k , we get

$$u = u_f - \frac{u_f}{k_j}(k) = u_f\left(1 - \frac{k}{k_j}\right)$$

$$q = ku_f\left(1 - \frac{k}{k_j}\right) = u_f\left(k - \frac{k^2}{k_j}\right) \qquad 3$$

Which expresses q as a parabolic function of k. From this it is evident that q_m is a point on the curve where the slope of a line tangent to the curve is equal to zero and where $k = k_m$; therefore, differentiating this equation with respect to k, and setting it equal to zero, we get

$$\frac{dq}{dk} = u_f\left(1 - \frac{2k_m}{k_j}\right) = 0 \qquad 4$$

Since u_f cannot be equal to zero,

$$1 - \frac{2k_m}{k_j} = 0 \quad \text{or} \quad k_m = \frac{k_j}{2} \qquad 5$$

Next, we derive an expression for q as a function of u where,

$$u - u_f = u_f\frac{-k}{k_j} \text{ and } k = k_j\left(1 - \frac{u}{u_f}\right) \qquad 6$$

Therefore from q = u k and substituting this expression for k, we get

$$q = uk_j\left(1 - \frac{u}{u_f}\right) = k_j\left(u - \frac{u^2}{u_f}\right) \qquad 7$$

Which expresses q as a parabolic function of u. This relationship also indicates that for a given value of the flow rate, q there are two corresponding values of k and u.. This represents the two flow conditions, where a flow rate q under free flow condition, is achieved at a higher speed ($u > u_m$) obtained under a lower ($d < d_m$) density which is also equal to the same flow rate q, that achieved at a lower speed ($u < u_m$) at a higher density ($d > d_m$) under congested flow condition. Furthermore, differentiating with respect to u and setting it equal to zero, we obtain

$$\frac{dq}{du} = k_j\left(1 - \frac{2u}{u_f}\right) = 0 \qquad 8$$

Since k_j cannot equal zero at q_m

$$1 - \frac{2u_m}{u_f} = 0 \quad \text{and } u_m = \frac{u_f}{2} \qquad 9$$

Therefore,

$$q_m = u_m k_m = \frac{u_f}{2}\frac{k_j}{2} = \frac{u_f k_j}{4} \qquad 10$$

The following can be derived from Greenshield's model:

- When the density is zero, the flow is zero because there are no vehicles on the roadway.
- As the density increases, the flow also increases to some maximum flow conditions.
- When the density reaches a maximum, generally called *jam density*, the flow must be zero because the vehicles tend to line up end to end (parking lot conditions).
- As the density increases the flow increases to some maximum value, but a continual increase in density will cause the flow to decrease until jam density and zero flow conditions are reached.

2.2.4 Time Series Data

Time series forecasting, or time series prediction, takes an existing series of data $x_{t-n}, \ldots, x_{t-2}, x_{t-1}, x_t$ and forecasts the $x_{t+1}, x_{t+2}, \ldots$ data values. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately. Examples of data series include financial data series (stocks, indices, rates, etc.), physically observed data series (traffic volume, sunspots, weather, etc.), and mathematical data series (Fibonacci sequence, integrals of differential equations, etc.). The phrase "time series" generically refers to any data series, whether or not the data are dependent on a certain time increment. Throughout the literature, many techniques have been implemented to perform time series forecasting. Several difficulties can arise when performing time series forecasting. Depending on the type of data series, a particular difficulty may or may not exist.

A first difficulty is a limited quantity of data. With data series that are observed, limited data may be the foremost difficulty. Limited data may result to a model overfitting the data available resulting to poor generalization or where the model parameter are more than the data, this also leads to the model not being able to learn the data in any way.

A second difficulty is noise. Two types of noisy data are (1) erroneous data points and (2) components that obscure the underlying form of the data series. Two examples of erroneous data are measurement errors and a change in measurement methods or metrics. A technique used in to reduce or remove this type of noise is the moving average. The data series $\ldots, x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t$ becomes $\ldots, [(x_{t-4} + x_{t-3} + x_{t-2})/3], [(x_{t-3} + x_{t-2} + x_{t-1})/3], [(x_{t-2} + x_{t-1} + x_t)/3]$ after taking a moving average with an interval i of three. Taking a moving average reduces the number of data points in the series by $i-1$. A third difficulty is nonstationarity, data that do not have the same statistical properties (e.g., mean and variance) at each point in time. A simple example of a no stationary series is the Fibonacci sequence: at every step the sequence takes on a new, higher mean value. A fourth difficulty is forecasting technique selection. From statistics to artificial intelligence, there are myriad choices of techniques. One of the simplest techniques is to search a data series for similar past events and use the matches to make a forecast. One of the most complex techniques is to train a model on the series and use the model to make a forecast. K-nearest-neighbor and neural networks are examples of the first and second techniques, respectively.

2.2.4.1 Importance of Time Series data

Time series forecasting has several important applications:

- One application is preventing undesirable events by forecasting the event, identifying the circumstances preceding the event, and taking corrective action so the event can be avoided.
- Another application is forecasting undesirable, yet unavoidable, events to preemptively lessen their impact for instance, in this case traffic congestion.
- Finally, many people, primarily in the financial markets, would like to profit from time series forecasting Muhoho [M05].

2.3 Artificial Neural Networks

2.3.1 Neural Network Description

Efraim et al [EAL04] classifies neural networks and other learning algorithms as displayed in figure 2-4.

Figure 2-4 Learning Algorithms

Efraim [E95] presents the difference between natural and artificial intelligence. A list of the advantages of artificial intelligence versus natural intelligence:

1. A.I is permanent: from a commercial and business continuity perspective natural intelligence is perishable,
2. A.I. offers ease of duplication and dissemination,
3. A.I. can be less expensive therefore human effort expensive over the long run,
4. A.I. is consistent and thorough with predicable results. Human beings are irrational and erratic,
5. A.I. is documented: use of trace, logging. Natural intelligence is difficult to reproduce.

Conversely, the Advantages of Natural Intelligence over A.I. are:

1. Creativity,
2. Use of sensory and feed back to improve and adapt,
3. Wide context of experience and logic,

2.3.1.1 What is an artificial neural network

A sampling of definitions from Sarle [S02]:

- a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.
- A neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
  - The network through a learning process acquires knowledge.
  - Interneuron connection strengths known as synaptic weights are used to store the knowledge.
- A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information. Furthermore, each element operates asynchronously; thus, there is no overall system clock.
- Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge.

2.3.1.2 Types of artificial Neural Networks

There are many kinds of ANNs. Table 2-4 is a collection of some of the most well known methods. The two main kinds of learning algorithms are supervised and unsupervised as described in table 2-4.

2.3.1.2.1 Supervised Learning Artificial Neural Networks

In supervised learning, the correct results (target values, desired outputs) are known and are presented to the ANN during training so that the ANN can adjust its weights to try match its outputs to the target values. After training, the ANN is tested by giving it only input values, not target values, and seeing how close it comes to outputting the correct target values [RN95], [S02].

2.3.1.2.2 Unsupervised Learning Artificial Neural Networks

In unsupervised learning, the ANN is not provided with the correct results during training. Unsupervised ANNs usually perform some kind of data compression, such as dimensionality reduction or clustering. The distinction between supervised and unsupervised methods is not always clear-cut. An unsupervised method can learn a summary of a probability distribution, then that summarized distribution can be used to make predictions [RN95], [S02].

2.3.1.2.3 *Network Topologies for Artificial Neural Networks*

Two major kinds of network topology are feedforward and feedback. In a feedforward ANN, the connections between units do not form cycles. Feedforward ANNs usually produce a response to an input quickly. Most feedforward ANNs can be trained using a wide variety of efficient conventional numerical methods [S02]. In a feedback or recurrent ANN, there are cycles in the connections. In some feedback ANNs, each time an input is presented, the ANN must iterate for a potentially long time before it produces a response. Feedback ANNs are usually more difficult to train than feedforward ANNs [RN95], [S02]. Table 2-4 illustrates some well-known kinds of ANNs:

Artificial neural Network		**Pioneers**
Supervised Feedforward	Linear Feedforward	Hebbian - Hebb (1949), Fausett (1994) Perceptron - Rosenblatt (1958), Minsky and Papert (1969/1988), Fausett (1994) Adaline - Widrow and Hoff (1960), Fausett (1994) Higher Order - Bishop (1995) Functional Link - Pao (1989)
	MLP: Multilayer perceptron	MLP: Multilayer perceptron - Bishop (1995), Reed and Marks (1999), Fausett (1994) Backprop - Rumelhart, Hinton, and Williams (1986) Cascade Correlation - Fahlman and Lebiere (1990), Fausett (1994) Quickprop - Fahlman (1989) RPROP - Riedmiller and Braun (1993)
	RBF networks	RBF networks - Bishop (1995), Moody and Darken (1989), Orr (1996) OLS: Orthogonal Least Squares - Chen, Cowan and Grant (1991)
	Classification only	LVQ: Learning Vector Quantization - Kohonen (1988), Fausett (1994) PNN: Probabilistic Neural Network - Specht (1990), Masters (1993), Hand (1982), Fausett (1994)
Supervised Feedback	Feedback	- Hertz, Krogh, and Palmer (1991), Medsker and Jain (2000)
	BAM: Bidirectional Associative Memory	BAM: Bidirectional Associative Memory - Kosko (1992), Fausett (1994)
	Boltzman Machine	Boltzman Machine - Ackley et al. (1985), Fausett (1994)
	Recurrent time series	Backpropagation through time - Werbos (1990) Elman - Elman (1990)

Artificial neural Network		Pioneers
		FIR: Finite Impulse Response - Wan (1990) Jordan - Jordan (1986) Real-time recurrent network - Williams and Zipser (1989) Recurrent backpropagation - Pineda (1989), Fausett (1994) TDNN Time Delay NN - Lang, Waibel and Hinton (1990)
Unsupervised	Unsupervised	- Hertz, Krogh, and Palmer (1991)
	Competitive , Vector Quantization	Grossberg - Grossberg (1976) Kohonen - Kohonen (1984) Conscience - Desieno (1988)
	Competitive Self-Organizing Map	Kohonen - Kohonen (1995), Fausett (1994) GTM: - Bishop, Svensen and Williams (1997) Local Linear - Mulier and Cherkassky (1995)
	Adaptive resonance theory	ART 1 - Carpenter and Grossberg (1987a), Moore (1988), Fausett (1994) ART 2 - Carpenter and Grossberg (1987b), Fausett (1994) ART 2-A - Carpenter, Grossberg and Rosen (1991a) ART 3 - Carpenter and Grossberg (1990) Fuzzy ART - Carpenter, Grossberg and Rosen (1991b) DCL: Differential Competitive Learning - Kosko (1992)
	Dimension Reduction	Diamantaras and Kung (1996) Hebbian - Hebb (1949), Fausett (1994) Oja - Oja (1989) Sanger - Sanger (1989) Differential Hebbian - Kosko (1992)
	Autoassociation	Linear autoassociator - Anderson et al. (1977), Fausett (1994) BSB: Brain State in a Box - Anderson et al. (1977), Fausett (1994) Hopfield - Hopfield (1982), Fausett (1994)

Table 2-4 Various Kinds of Neural Networks available [S02].

2.3.1.3 When to not use a neural network

Neural networks are universal approximators able to map any data to a model Tsai et al [TLW05], Zhang [Z04], Fu-Sheny [Fj61]. However there are inherent shortcomings of computers and neural networks in particular making them impractical to consistently use:

- Cannot formulate on algorithmic solution
- Need to generate lots of examples of the behavior
- Need to pick out the structure from existing data
- Massive parallelism is required,

2.3.1.4 Fundamentals of Neural Networks

Neural networks, sometimes referred to as connectionist models Russel et al [RN95], Bishop [B95], Looney [L00], are parallel-distributed models that have several distinguishing features:

- A set of processing units,
- An activation state for each unit, which is equivalent to the output of the unit;
- Connections between the units. Generally each connection is defined by a weight that determines the effect that the signal of one unit onto another unit;
- A propagation rule, which determines the effective input of the unit from its external inputs;
- An activation function, which determines the new level of activation based on the effective input and the current activation;
- An external input (bias, offset) for each unit;
- A method for information gathering (learning rule);
- An environment within which the system can operate, provide input signals and, if necessary, error signals.

2.3.1.5 Processing unit

A processing unit in figure 2-8 is also called a neuron or node, performs a relatively simple job; it receives inputs from neighbors or external sources and uses them to compute an output signal that is propagated to other units. ANN are modeled after the biological neuron [RN95].

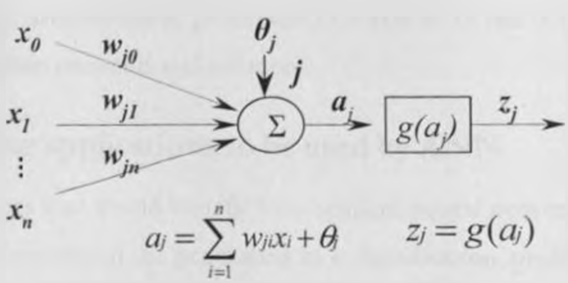


Figure 2-5 Processing Unit

Within the neural systems there are three types of units:

- Input units, which receive data from outside of the network;
- Output units, which send data out of the network;
- Hidden units, whose input and output signals remain within the network.

- Each unit j can have one or more inputs x_0, x_1, x_2 … x_n but only one output z. An input to a unit is either the data from outside of the network, or the output of another unit, or its own output.

Neural networks can provide good results in short time scales but only for certain types of problem, with great deal of care taken over design and input data pre-processing Dti [D94]. ANN should be considered as components within overall application – not as solutions in their own right use in expert systems

2.3.1.6 ANN attributes

Attributes of artificial neural networks that need to be considered before extensive use or deployment:

- Learning from experience: need lots of data
- Generalizing from examples: high levels of generalization rules
- Extract essential information from noisy data: recognize patterns underlying process noise
- Develop solutions faster with less radiance on domain expect; to some extent but require expects in architecture design, inputs especially and result use
- Adaptability: learn "on the job" solutions can be designed to adapt to their operating environment
- Computational efficiency: training requires computational power but once trained, very easy to use. Parameter pruning processing can be used to speed up training
- Non-linearity: are non-linear processors thus able to fit real world problems much easier than conventional solutions

2.3.1.7 Identifying applications to be used by ANN

Identifying applications that would benefit from artificial neural network requires a keen sense of the overall outcome of the processing be it classification, prediction or getting missing values for data. Dti [D94] have developed a process of ANN development to speed up the process of evaluation.

2.3.1.7.1 Technical features

1. The application deals with poor quality or incomplete data,

2. The application requires integration of different types of input data e.g. a combination of computer data and signals from sensors or agents,
3. It is difficult to specify a model for mathematical simulation, or rules for acknowledge based system,
4. The application needs to be adaptive i.e. the neural network must be capable of learning during operation, adopting its responses as the operating environment slowly charges.
5. Input data and target data is available: need to have sufficient data to train, validate and test ANN

2.3.1.7.2 Practical requirements

1. Availability of adequate resources: people, equipment, time and money, learning
2. Need to evaluate safety critical or business critical applications to provide cast-iron proof of ANN. Rules in decision making might be difficult to glean.
3. Costs of obtaining and processing data is much lower than the benefits
    - i) collection of data
    - ii) researching of data / phenomena
    - iii) storage of data and communication

2.3.1.8 Pre-processing and post processing

Pre-processing describes any process that converts inputs into a form suitable for use within a neural network. Post processing describes any process which operates on the ANN output. Dti [D94], Mbugua [M05] identify critical steps in preprocessing include:

1. Transforming the data into a form suitable to the ANN
2. Selection of the most relevant data thus eliminate noisy, irrelevant sets from the data available for modeling,
3. Minimize the number of input to the ANN as found in image processing applications where too much data slows down the learning process or wholly results to over fitting thus poor generalization.

The importance of preprocessing is to reduce network complexity and the computational time. Designing and training neural networks is a computer intensive process. An essential component of the process is focused on reducing network complexity with respect to inputs required and hidden processing elements required to achieve good performance and accurate generalization. Generalization is the ability to give accurate answers on data that it has not seen as part of the training process. The achievement of good generalization is a key design aim, it is achieved by careful choice of neural network architecture and amount of training applied to the artificial neural network. Good generalization results from good performance. Performance is the predictive accuracy when presented with data other than that with which it has been trained. The figure 2-6 illustrates the process of developing an artificial neural network.

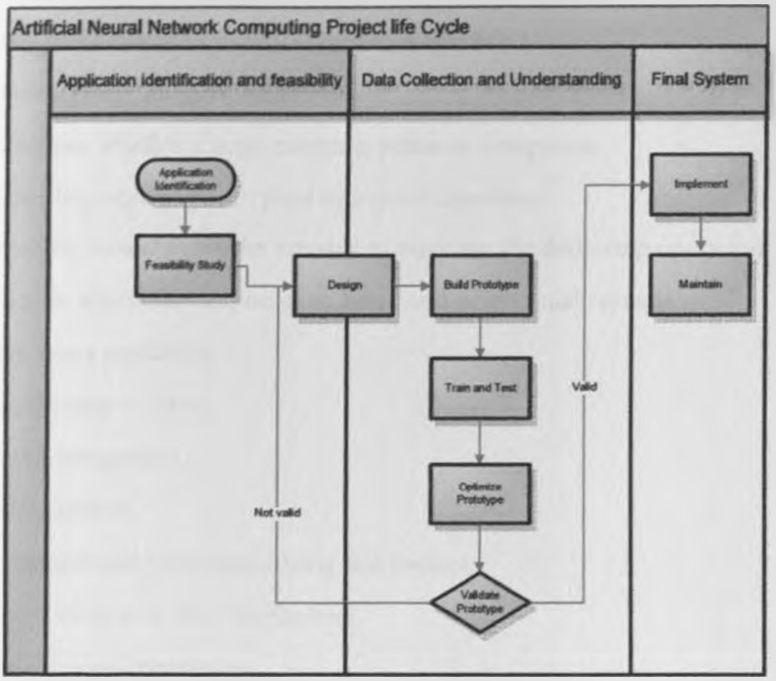


Figure 2-6 Artificial Neural Network computing Project life Cycle

2.3.1.9 Cross validation testing

Cross-validation is a method for estimating generalization error based on "resampling" Sarle [S02], Ian et al [IW99]. The resulting estimates of generalization error are often used for choosing among various models, such as different network architectures. Cross-validation can be used simply to estimate the generalization error of a given model, or it can be used for model selection by choosing one of several models that has the smallest estimated generalization error.

2.3.2 Neural Network Application Areas

Artificial neural networks can be used in areas where hard and fast rules (such as those that might be used in an expert system) cannot easily be applied as seen earlier. Almost any finite-dimensional vector function on a compact set can be approximated to arbitrary precision by feedforward neural network (which are the type most often used in practical applications) given there is enough data and computing resources.

There are also many other important problems that are so difficult that a neural network will be unable to learn them without memorizing the entire training set, such as:

- Predicting random or pseudo-random numbers.
- Factoring large integers.
- Determine whether a large integer is prime or composite.
- Decrypting anything encrypted by a good algorithm.

Not withstanding, neural networks are able to carry out the following operations:

- Function approximation on both linear and polynomial systems
- Time series prediction,
- Classification of data,
- Pattern recognition,
- Fault diagnosis,
- Equipment and plant monitoring and control,
- Pattern analysis in data warehouses,
- Image & signal processing
- Process modeling.

Zhang [Z04] gives a general overview of neural networks, design consideration, previous work done on forecasting, dataset consideration and performance testing. His overview identifies a number of application areas where ANN has been deployed as seen research by Zhang. In addition Zhang prescribes a series of steps and conditions to be taken when developing any neural network model for forecasting purposes.

2.3.3 Advantage of Artificial of Neural Networks

Versteegt et al [VT03], identify the following advanteges using artifual neural networks:

1. Computationally fast to give result once trained,
2. Relatively fast to implement relative of other learning systems,
3. No behavioral knowledge required to model data using neural networks,
4. Learning ability is data and architecture based,
5. Auto organization: Can create its own representation of the data given in the learning process

6. Tolerance to faults: Because ANN store redundant information, partial destruction of the neural network do not damage completely the network response
7. Flexibility: ANN can handle input data without important changes like noisy signals or other changes in the given input data.
8. Real time: ANN are parallel structures thus can benefit from multiprocessor
9. Scalability: ANN can be easily be parted to fit any problem from a particular area.
10. Data representation: Can take discrete numeric / non numeric and or continuous data as input / output.
11. Can work with both continuous and discrete data.

2.3.3.1 Disadvantage of artificial neural networks

Versteegt et al [VT03] identify the following disadvantategs of using artifual neural networks:

1. Training requires lots of data,
2. The individual relations between the input variables and the output variables are not developed by engineering judgment so that the model tends to be a black box or input/output table without analytical basis.
3. No reliable outcome for situations the network is not trained for.
4. Minimizing overfitting requires a great deal of computational effort

Zhao et al [ZCK03], Adya et all [AC98] in a related study attempt to reproduce past research results on ANN to corroborate findings highlight a major disadvantage of artificial neural networks: development and application of neural networks is no easy feat. From this study, they conclude the following:

- It is harder to obtain substantial improvements in extrapolative forecasting with ANN than might be assumed reading earlier studies.
- It is important for researchers to fully document and publish research details and data for reproduction and replication. (Replication should be done as soon as published material is available to note missing assumptions / data).
- When ANN are effectively implanted and validated, they show potential for forecasting and prediction.
- Significant portion of ANN research in forecasting and prediction lacks validity

- It is not easy to replicate studies as such the need for systematic approach in dealing with data mining Mbugua [M05].

2.3.4 Forecasting using Neural Networks

Tang et al [TF93] report on neural nets as models for time series forecasting inspired by the inconsistency of reported neural network performance. By conducting a series of forecasting experiments using neural networks and comparing the results with the conventional Box-Jenkins method they are able to demonstrate that neural nets outperform the Box-Jenkins method hence are suitable for forecasting problems. Frank et al [FD01] observe the importance of correctly specifying the sliding window size for a forward feed neural network based forecaster. Better forecasting (reduced error) is obtained by embedding a correct sliding window size. Large variation of the window diminishes performance. Corani et al [CCG03] apply a neural network on PM10 time series data to predict a 1 day ahead PM10 for Milan. By visualization they are able to observe the cyclic nature of data. Niraj [N03] uses similar framework model to model air pollution. In a related study to demonstrate the use of ANN to predict the seasonal and monthly rainfall over the series as inputs, Sahai et al [SSS00] use a 5 year sliding window with one step ahead forecasting to forecast rainfall.

Yao [YIPT98] implement a neural network to discover an effective market decision support system by discovering artificial variables that influence sales performance of color televisions. Unlike previous researches presented thus far, this particular research focuses on the need to know how attributes combined effect impacts sales as such build a neural network to predict sales effectively.

2.3.5 Feed forward Multilayer Back propagation Neural Network

Back-propagation is the most commonly used method for training multi-layer feed-forward networks. It can be applied to any feed-forward network with differentiable activation functions. This technique was popularized by Rumelhart, Hinton and Williams Russel et al [RN95]. BPN is a layered, feed forward network that is fully interconnected by the layers. There is no feedback connections and no connections that bypass one layer to go directly to a later layer Freeman et al [FS91]. Looney [L00] identifies pertinent advantages of back propagation multi layer feed forward neural network as listed:

- Learning is independent of the order in which training data is presented.

- The architecture can be manipulated for better results
- They able to run on parallel processors

However, he also points out their shortcomings as :

- Training may converge to a local minimum that is shallow so that learning is not robust.
- The learning rate cannot be predicted in advance. This results to iterative learning which can lead to oscillation if not small.

For most networks, the learning process is based on a suitable error function, which is then minimized with respect to the weights and bias. If a network has differential activation functions, then the activations of the output units become differentiable functions of input variables, the weights and bias. If we also define a differentiable error function of the network outputs such as the sum-of-square error function, then the error function itself is a differentiable function of the weights. Therefore, we can evaluate the derivative of the error with respect to weights, and these derivatives can then be used to find the weights that minimize the error function, by either using the popular gradient descent or other optimization methods. The algorithm for evaluating the derivative of the error function is known as back-propagation, because it propagates the errors backward through the network.

The functionality of a neural network is determined by the combination of the topology (number of layers, number of units per layer, and the interconnection pattern between the layers) and the weights of the connections within the network. The topology is usually held fixed, and the weights are determined by a certain training algorithm. The process of adjusting the weights to make the network learn the relationship between the inputs and targets is called *learning*, or *training*. Many learning algorithms have been invented to help find an optimum set of weights that results in the solution of the problems. They can roughly be divided into two main groups:

- Supervised Learning - The network is trained by providing it with inputs and desired outputs (target values). These input-output pairs are provided by an external teacher, or by the system containing the network. The difference between the real outputs and the desired outputs is used by the algorithm to adapt the weights in the network (Figure 2-7). It is often posed as a function approximation problem - given training data consisting of pairs of input patterns

x, and corresponding target t, the goal is to find a function $f(x)$ that matches the desired response for each training input.

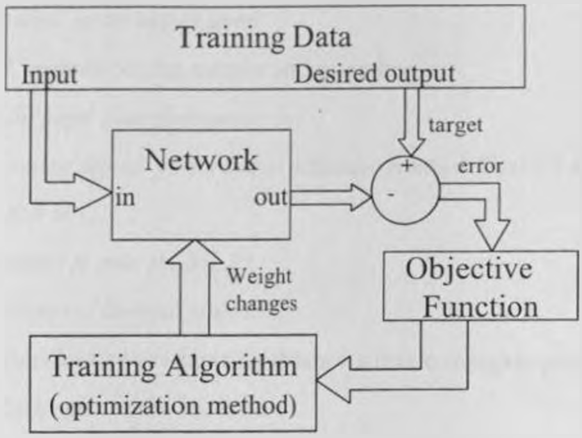


Figure 2-7 Supervised learning model

- Unsupervised Learning - With unsupervised learning, there is no feedback from the environment to indicate if the outputs of the network are correct. The network must discover features, regulations, correlations, or categories in the input data automatically. In fact, for most varieties of unsupervised learning, the targets are the same as inputs. In other words, unsupervised learning usually performs the same task as an auto-associative network, compressing the information from the inputs.

Looney [L97] discusses neural networks and learning algorithms used in artificial intelligence. Similar to other data mining methodologies, multilayer propagation architecture must be designed properly for the particular dataset to assure that the network will learn robustly and will be reasonable efficient. Looney [L97] presents a series of questions as a guide to developing multi layer propagation neural networks:

1. *How many layers of neurons should a neural network have?*
   i) Hidden layer and output layer of neurons are sufficient, provided that there are enough neurons in the hidden layer.
   ii) To reduce the number of neurons in the hidden layer two hidden layers can be used.
   iii) One hidden layer is sufficient as it avoids complications.
2. *How many input nodes should we use?*
   i) This depends on the feature vector and attributes.
3. *How many neurons in the hidden layer?*

i) This is based on rules of thumb. Cyntia [C03] lists a number of formulas.

4. *How many neurons in the output layer?*

    i) Depends on the output and encoding

5. *What should the target (identifiers)vectors be?*

    i) *Scaling depends on the output activation function. Tanh (-1 to 1) and for sigmoid (0.9 to 1)*

6. *How can we proceed to train the MLP?*

    i) Steepest descent method,

    ii) Accelerated gradient methods such as conjugate gradients,

    iii) Strategic search methods,

7. How can we test to determine whether or not the MLP is properly trained?

    i) 60% training set

    ii) 25% validation

    iii) 15% test set

8. *How do we select parameters (such as learning rate and momentum), and speed up and improve the learning?*

    i) Use heuristic methods,

    ii) Manually iterate while changing (learning rate, momentum, weight , hidden nodes, activation function) parameters.

9. *What should be the range of weights and the network inputs and output?*

    i) Start with -0.5, 0.5: but some weights need to move to (-6, 6)

Fildes et al [FL03] propose an effective and computationally viable approach to objectively specifying the structure of neural network and thereafter evaluate its success (neural network) by examination of performance compared to various alternative statistical forecasting methods. Earlier research has established that the performance of a neural network model depends quite critically on the process by which it is constituted: Input nodes, hidden layer, learning rate, weights and bias, Transfer function. However no systematic procedure has been developed. After the selection of input nodes based on cross-validation, a 3-stage approach is proposed here which consists sequentially as shown in figure 2-8 generally seen as:

1. Selecting the learning rate and momentum,
2. Select the number of hidden nodes,
3. Select the initial weights

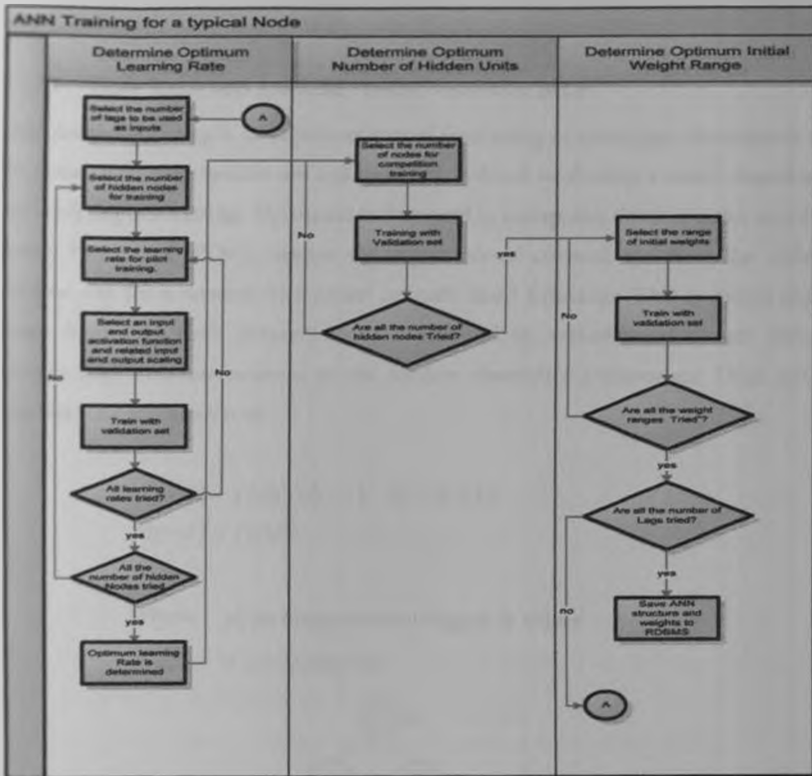


Figure 2-8 Complete Process flow for training ANN using Lags.

Some performance hints as identified by Looney [L97]:

- A single hidden layer is often sufficient,
- A basic simple performance evaluation between models is Root Mean square Error (RMSE),
- Test using an out of sample only once the winning model is selected,
- Start with 3 auto-regressive lags selected as a default.
- Logistic function used as the transform with input scaled between (0.35 – 0.65)
- Too large a structure will lead to over fitting.
- Learning rate of 0.9 performs well and 0.01 poorly.

- 200 iterations with max 1000

2.3.5.1 Time Series and Artificial Neural Networks: MLP

Time series forecasting is an important area of forecasting in which past observations of the some underlying variable are collected and analyzed to develop a model describing and analyzing relationship. The model is then used to extrapolate the time series into the future. Frank et al [FD01] observe the importance of correctly specifying the sliding window size for a forward feed neural network based forecaster. This is critical since better forecasting (with reduced error) is obtained by embedding a correct sliding window size. Incorrect variation of the window diminishes performance. Time series generalization can be seen as:

$$x(t+d) = f(x(t), x(t-1) \ldots x(t-N+1))$$
$$x(t+d) = f(y(t))$$

Where: $y(t)$ is N-ary vector of lagged x values

d is normally one

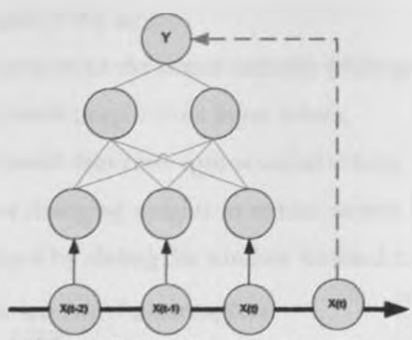


Figure 2-9 Sampling of a time series

From figure 2-9, discrete steps can be used to generate a set of training data for feed-forward network. Successive values of the time dependent variable X(t), given by (X(t-d+1),….x(t), form the inputs to a feed forward network, and the corresponding target value is given by X(t+1).

Frank et al [FD01] use heuristic methods to determine the appropriate window size. In this study, they use the false nearest neighbor method and the singular-value analysis. Kaitani et al [KHM05] examine the forecasting performance of FNN models compared to other competing models, when the signal to noise ratio is small. They use a one step ahead neural network model as it is easier to calibrate and easier to compare among competing models. Cortex et al [CMJ01] attempt to use a genetic algorithm to modify the structure and parameter of ANN as such evolve the ANN topology, enhancing forecasting and generalization of time series data. From this study, the following can be concluded:

1. The architecture of an ANN in prediction requirements is sensitive to weight, architecture, learning and training data.
2. Sliding window range has a positive and negative input on prediction quality with too large resulting to over fitting and too narrow poor generalization.
3. Performance of an ANN model can be evaluated by measuring the forecasting accuracy using, Root Mean Squared.

Tang et al [TF93], in a related study on neural networks as models for time series prediction highlight the benefit of using the sliding window approach. They also site the performance of ANN is highly dependent on the structure, training and data.

Crone [C05] presents a detailed treatment on the process of modeling time series data using a feed forward artificial neural network with a sliding window. The critical steps are identified (illustrated on figure 2-10) as:

1. Present input data pattern to the neural network (sliding window data),
2. Calculate neural network output from input values,
3. Compare neural network forecasts against actual values,
4. Backpropagate error changing weights to reduce output forecast errors,
5. Present new data input by sliding the window forward to show next pattern set.

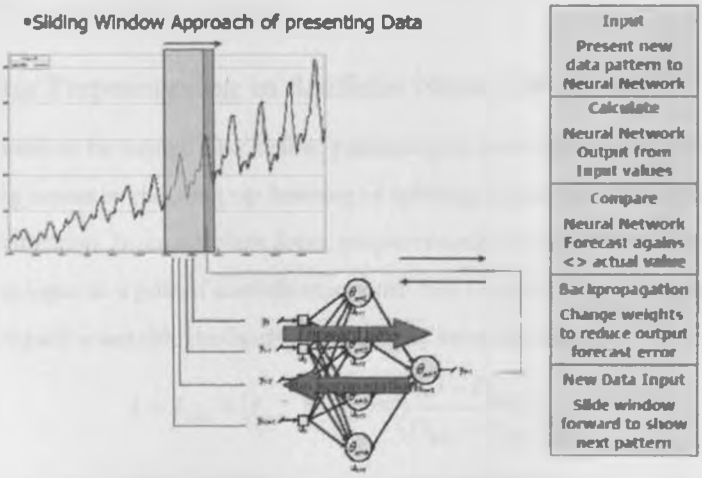


Figure 2-10 Neural Network Training on Time Series adopted from C05]

The decisions in neural network modeling all require expert knowledge. They can be identified as shown in figure 2-11:

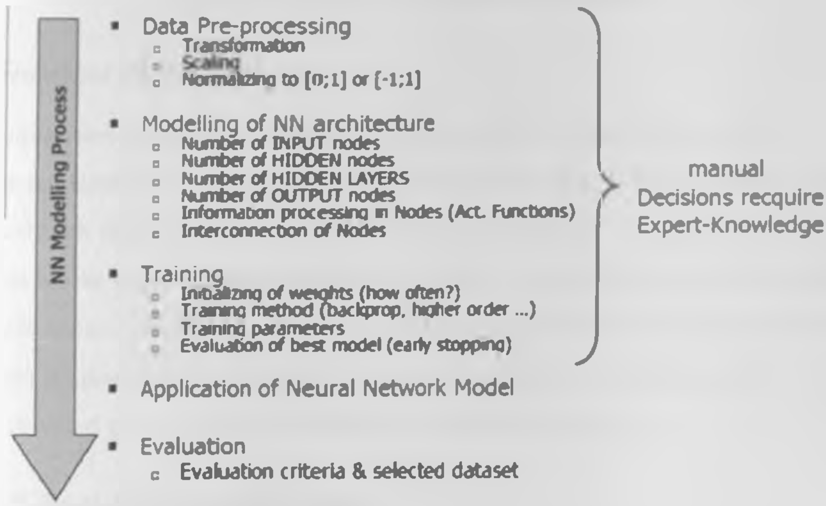


Figure 2-11 Decisions in Neural Network Modeling adopted from [C05]

The Advantage of time series forecasting using neural networks can be summarized as :

1. ANN can forecast any time series pattern (t+1) without preprocessing nor extensive model selection.
2. ANN offers many degrees of freedom in modeling.

Some of the disadvantages of time series modeling with artificial neural networks are:

1. Experience of knowledge is required,
2. Selection of the modeling window requires heuristics of extensive comparison of RMSE of various models with respect to validation error RMSE.
3. Explanation and interpretation of ANN weights is impossible or meaningless. ANN is a black box technique.

2.3.5.2 Data Preprocessing in Artificial Neural Networks

Input data needs to be treated first before presenting to an artificial neural network. Preprocessing assists in speeding up learning or splitting data to simpler attributes to assist in classification. In its simplest form, preprocessing can involve scaling down input or converting input to a pair of coordinates in the case of cyclic data Mbugua [M05]. The figure 2-12 depicts a suitable method to scale input between a range.

$$I = I_{min} + (I_{max} - I_{min}) \times \frac{(D - D_{min})}{(D_{max} - D_{min})}$$

D_{min} and D_{max} is computed for input basis

I_{min} and I_{max} is the range to normalize by

Figure 2-12 Data Preprocessing by normalization

2.4 Shortest Path Problem

The computation of shortest path has been extensively researched since it is a fundamental issue in the analysis of transportation networks. There are many factors associated with shortest path algorithms. First, there is the type of graph on which the algorithm works-directed or undirected, real-valued of integer link costs, and possibly-negative link costs. Furthermore, there is the family of graphs on which an algorithm works- cyclic, planar and connected. The shortest path algorithms presented in this thesis assume directed graphs with non-negative real-valued link costs.

2.4.1 Dijkstra's Search Algorithm

Dijkstra's algorithm is a breadth first search thus it would search all points within a fixed circular radius, gradually expanding this circle searching further away from the point. It Best when you do not know where the destination is as it is a least cost path. Dijkstra's algorithm can solve single source shortest path problems by computing the one-to-all shortest path tree from a node to all other nodes.

2.4.2 A* Search Algorithm

It is not always feasible to use Dijkstra's algorithm to compute the shortest path from a single start node to a single destination since this algorithm does not apply any heuristic. It searches by expanding out equally in every direction and exploring a too large and unnecessary search area before the goal is found. It has a high computing cost. This has led to the development of heuristic searches. A* is a graph search algorithm that finds a path from a given initial node to a given node (or one passing a given goal test) It employs a "heuristic estimate" that ranks each node by an estimate of the best route that goes through that node. It visits the nodes in order of this heuristic estimate. The A* algorithm is an admissible search algorithm Russel [RN95]. In general, a search algorithm is called admissible if it is guaranteed to always find the shortest path from a start node to a goal node. If the heuristic employed by the A* algorithm never overestimates the cost, or distance, to the goal, it can be shown that the A* algorithm is admissible Russel. Invariably, an admissible heuristic is an informed guess that never overestimates the true cost of a solution

A* considers no more nodes than any other admissible search algorithm, provided that the alternative algorithm does not have a more accurate heuristic estimate. In this sense, A* is the computationally most-efficient algorithm that is guaranteed to find the shortest path. Husdal [H05] identifies measures for optimality in routing. These are listed below:

1. shortest time,
2. shortest distance,
3. least total cost,
4. most secure,

A dynamic network is one where cost of traversing the network varies over time. In transportation network representation, the weight of the links and nodes can be assigned as the cost of traverse along the link. Changes in traffic conditions are considered as changes in link and node weights.

2.4.2.1 Reasons for using the A*

According to Russel [RN95], the reasons for using A* are:

1. It is an optimal algorithm,
2. It is an admissible search routine,
3. It's complexity (space and time) is reasonable,
4. It can support hints to enable faster processing,

Compared to Dijkstra algorithm, the A* does not search all directions which is a favorable characteristic hence a decreased computation time.

The A* search can be summarized as $f(n) = h(n) + g(n)$ where f(n) is the A* search solution, h(n) is the heuristics cost , an admissible function and g(n) us a greedy search cost. The greedy search is neither optimal nor complete but can be very efficient Russel [RN95]

2.4.3 Other Search Algorithms

Arroyo et al [AK05] analyze data from drivers using in-vehicle route guidance systems to empirically analyze the behavior of travel times on US road network. This paper focuses on identifying the proper functional forms to desirable travel time distributions. At any current location, the estimate time of arrival to a fixed destination is more properly described by a probability distribution associated with each of the choices. Algorithms like Dijkstra, Bellman-ford and its variations have proven to be efficient on calculating the shortest path on complex road network of linear deterministic single objective type. In vehicle route guidance software implements (above algorithms) rapidly enough to react in real time.

2.5 Decision Support Systems

2.5.1 Description of a Decision Support System

Turban et al [EAL04] defines it as "an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights." The decision maker can interact with the system directly as a user or through an intermediary who acts as the operator, queries the system and interprets the results to the decision maker.

The underlying themes in most DSS definitions are the concepts of interactive computer based systems that utilize data and models to solve semi-structured problems. However, they all emphasize that decision making is still dependent on the user as he retains control over the entire process. This allows his intuition and judgment to be factored into the development of the solution, which however introduces bias and subjectivity into the decision. DSS cannot make judgment they provide the user with various alternatives based on existing models, algorithms, data and scenarios built into them. Their role is to improve the quality of decisions by improving the response time of decision makers, discouraging premature decisions, exploring and testing multiple problem resolution strategies, and generating alternatives. Apart from decision support they can provide additional benefits such as organizational memory, and improved understanding of the problem context, as knowledge and practices built in are drawn from experts.

2.5.2 Components of a Decisions Support System

To respond to the decision maker's cognitive limitations caused by the decision makers biases and compensate for his short term memory, Mwangi [M06] points out that a DSS should mimic the positive human skills of inference and provide systematic approaches of organizing and retrieving information. To adequately address these requirements DSS are built to comprise the following generic subsystems Turban et al [EAL04]:

- Database Management – This contains data required by the system and is managed by a database management software.
- Model Management: This includes the models that are used to solve or simulate problems. These models assist in solving problems especially those based on quantitative data or are tactical in scope Mwangi [M06].
- User interface – It is the medium by which the user communicates with the DSS, whether through a web browser or other Graphical User Interface (GUI). The user interface needs to be well designed and simple to understand to help reduce levels of resistance. The level of technical skills required is influenced to a large extent by the design of the user interface.
- Knowledge Management (KM): This module contains rules, past knowledge, constraints etc that provides decisional guidance. KM calls for ways of leveraging information and individuals experiences for the benefit of the organization .

The four should be used as a guideline, when designing the basic structure of the system.

2.5.3 Intelligent Adaptive Decision Support System

Intelligent adaptive decision support system is best suited to deal with unstructured data. Some of the characteristics of an unstructured system are:

- they are novel, unstructured,
- no cut-and-dried method for handling the problem exists,
- calls for intelligent, adaptive, problem-oriented action,
- nonprogrammable is a better concept,

2.5.4 Successful Decision Support Systems Applications

Mwangi [M06] conducts research into what it takes to develop a successful decision support system (DSS) implementation strategy. This practical case study of one such DSS known as Rapid Emergency response and Contingency planning Tool (REACT) was then carried out. REACT is owned and under development at World Food Programme (WFP) Kenya. She concludes that DSS is definitely different to other information systems and its successful implementation is limited though they have been present in the Information Systems arena for decades. They are currently highly technocentric and further research needs to be carried out on how to improve their success rates.

2.6 *Geographical Information Systems*

A GIS makes use of geographical and attribute data. Attribute data, addresses, populations, etc., is associated with geographical data. Geographical data may be represented as points, lines or polygons. Attribute data can be handled easily using a conventional database management system (DBMS). It is the handling of the geographical data, such as the existence of rivers, roads or contour lines that requires the use of the special techniques that characterize the use of GIS. A GIS, as distinct from a mapping program, will have a database of geographic data, allowing linkages between different types of data and the ability to query this spatial data. For example a GIS database query might allow identification of all roads within a certain distance of a river. Therefore, while traditional database approaches can support queries on the attribute data, GIS is defined by its ability to cater for spatial queries.

2.6.1 Design of GIS databases

Miller et al [MH99] report on a GIS based decision support system for dynamic congestion modeling and shortest path routing in time critical logistics. From this research, GIS provides effective decision support through:

1. Database management capabilities
2. Spatial query language
3. Graphical user interface
4. Cartographic visualization and modeling of the earth

2.6.1.1 Transportation Network Data Model

A transportation network is a type of directed, weighted graph . The use of GIS for transportation applications is widespread and a fundamental requirement for most transportation GIS is a structured road network. In developing a transport network model, the street is represented by a series of nodes and links both with associated weights. This representation (impedance, cost of travel) is an attempt to quantify the street system for use in a mathematical model. Inherent in the modeling effort is a simplification of the actual street system. The network nodes represent the intersections within the street, an event like traffic control section or bridge. Two nodes make a link. The weights on both links and nodes represent a characteristic friction in moving from one node to another.

2.6.1.2 Topological rules for lines

Listed are properties a transportation GIS database system needs to observe [CD95]:

1. Lines are a single part, hence need to be un-split using ArcInfo,
2. There are no duplicate lines, hence build topology using Arcinfo,

The benefits of building a network topology are:

- Lines do not self overlap,
- Lines do not overlap other lines,
- Lines intersect only at nodes, and nodes anchor the ends of all lines ,
- Lines do not overshoot or undershoot other lines they are supposed to meet and intersect,

Figure 2-8 illustrates this process in more detail.

2.6.1.3 Process of Image to vector conversion

Listed is the process of converting an analog data source to a vector digital GIS database.

- Using Arcinfo software, maps are geo-referenced by registering the image to known points using known survey ground stations.
- On screen digitizing to capture road network feature from aerial photos, scanned images e.t.c,
- Validation with satellite image to ensure consistency,
- Validation plots made to inspect digitizing quality and code assignment (visual checks)

2.6.1.4 Attribute enhancements

Additional attributes are added to each segment include:

1. Name of the road or node,
2. Highway type or road class
3. Length of the road.

Un-required attributes are removed from segments to keep the overall database small.

2.6.1.5 Process of vector lines in Arcview

A complete topological GIS database has the following properties:

1. lines are single part,

2. Cleaned database: arcinfo clean function is repeatedly used following edit to verify topology and enforce minimum distance between vertices hence control fuzzy and dangle errors.
3. No dangles or slivers: due to the topology verification, there may be minor differences in feature geometry between certain features

2.6.1.6 Digitizing Process

1. Inspection of each segment to ensure continuity with deliberate overshooting while digitizing
2. Shape file converted to Arcinfo coverage to enforce topology, integrity and build coverage
3. Lines un-split to begin and end at intersection with other routes,
4. Results to reduced features in large networks e.g. (200,000 to 2,000) Connecticut
5. Arcinfo clean to verify topology and enforce fuzzy and dangle rules
6. Coverage charged back to shape file once it is determined that it has captured all necessary geometry and conformed to the topology rules

2.6.2 Geographic Information Systems and Decision Support Systems

Michael et al [MRS03] determine the optimal route for shipments that minimizes both the travel distance of the shipment and population exposure along the route. This demonstrates GIS can be used to visualize and develop a route that minimizes the impacts of hazardous waste incidents along the roadway network and mange risk proactively. Praween [PDV03] develop an advanced traveler information system for Hyderabad City in a GIS environment. This system is able to provide the shortest path and closest facility based on distance and drive time. Praween [PDV03] follow a sereis of steps as illustrated in figure 2-13.

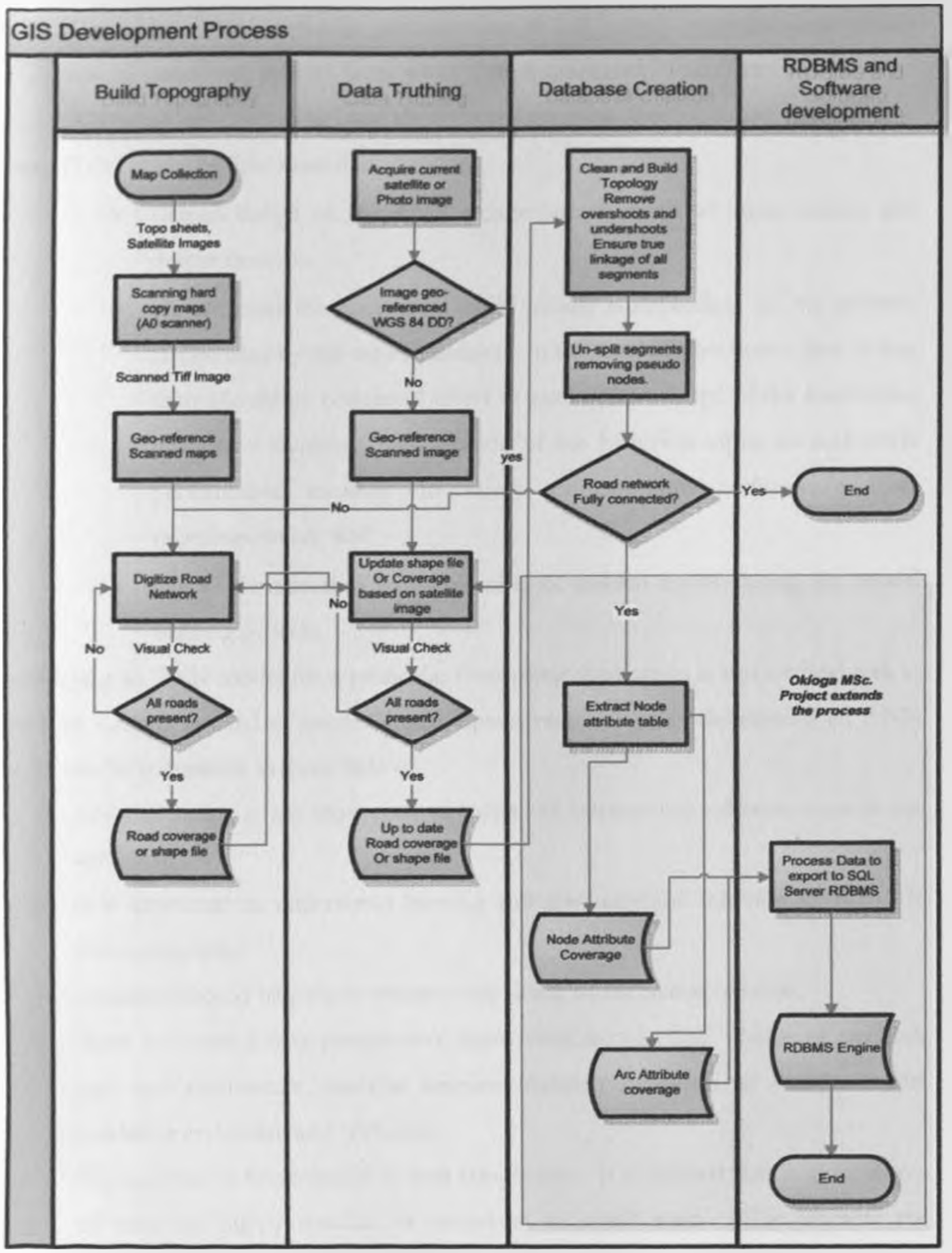


Figure 2-13 GIS Development Process

2.7 Discussion

Zhang [Z04] and other researchers have scientifically introduced ANN as tools suitable for forecasting when the data is static in various business fields with specific emphasis to non-linear relationships. They identify features that make ANN suitable for forecasting.

1. Data driven non-parametric methods that do not requre restricted assumptions on the underlying process from which data is generated: 'learn from data'
2. ANN has been shown to have the universal function approximating capability.

Zhang [Z04] points out the need for:

- Careful design of the ANN architecture in terms of input, hidden and output neurons.
- He identifies the success of ANN mainly is dependent on the patterns represented by the input variables. In the case of time series forecasting, there should be concerted effort in gaining knowledge of the forecasting problem / domains, identification of the historical input set and ANN performance measure to select an optimum architecture from experimentation, and
- Lastly, identification of the number of hidden nodes during the model building process.

Developing an ANN model for a particular forecasting application is not a trivial task as noted by various researches cited. Critical aspects required when developing an ANN model can be summaries as listed below:

1. ANN modeling is a combination of an art and science and software alone is not sufficient.
2. It is important to understand learning and generalization inherent in all ANN forecasting tasks.
3. Attention should be paid to address over fitting of the neural network.
4. Need for careful data preparation, input variable selection, choice of network type and architecture, transfer function, training algorithm, as well as model validating evaluation and seclusion.
5. Thought has to be accorded to data sample size. It is realized that large amounts of data are highly suitable as compared to small sizes. This impacts the generalization ability of the ANN and over fitting. For non-linear modeling, larger sample sizes should be more desirable (Zhang).
6. Data splinting is critical. Data should be split to training, validation, testing. The training and validation is taken as in sample whilst testing as out of sample. Zhang identifies ratios as quoted in literature 70%: 30%, 80%: 20%, or 90%: 10%. The critical factor is data availability and spread in learning, validating and testing.

Data preprocessing is also touched by Zhang but not in details. A brief note is made on the importance of normalization as part of data pre-processing. An argument is put forward on the benefits accrued by data preprocessing as it belittles the ANN as a universal approximator. To justify the need of pre-processing, Zhang identifies studies that have demonstrated the pre-processing is indeed beneficial in improving ANN forecasting performance. Emphasis is given to the process of de-trending and de-seasonalization.

7. Lastly, the issue of ANN architecture is addressed. Zhang asserts the need to build robust experiments to identify suitable ANN architecture. Good domain knowledge is necessary as this is an area of heuristic ability. Zhang emphasizes the need to design a good experiment but does not go to detail. No mention of the performance tests is made however; he has identified literature for future address the issue of modeling.

To build a successful model, a proposed checklist for ANN is presented:

1. Forward feed architecture is by for the best developed and most widely applied model for forecasting.
2. Size of the output layer is determined by the nature of the problem.
3. Both single step and multiple step forecasts are typical requirements that can be achieved by using one output node or more than one output node.
4. As noted in earlier literature and this multiple-step forecast can be achieved by iterating through a single step model.
5. The input layer is more important than the hidden layer in time series forecasting problem as such this needs considerable attention.
6. Most applications use one hidden layer however they should be determined by experimenting with a number of choices then selected by performance criterion like RMSE. Zhang notes that previous studies have demonstrated that the performance forecasting is not very sensitive to hidden nodes number.
7. Choice of transfer functions for hidden nodes as logistic or hyperbolic while output as liner or identity. Zhang correctly asserts that this should not adversely impact the performance of ANN and justifies that of the output layer is normalized into the range (0,1) then logistic function can be used for the output layers.

8. Training, validating and testing is required to offer the best forecasting performance. It attributes success to the use of different learning algorithms can be beneficial.
9. Zhang concludes by identifying ways of selecting an appropriate ANN model. He notes that the model to be selected is one whose performance is based on the testing sample.

In conclusion, researches like Zhang point out the importance of comparing the performance of ANN to traditional statistical methods. As noted in a previous study, this will justify the value of ANN in the problem solution. In addition, three evaluation criteria are postulated:

1. Comparing ANN to a well-accepted traditional model
2. Using true-out of samples
3. Ensuring enough sample size in the out-of-sample for classification problems and time series problems.

3 Methodology

The techniques used in conducting the research include questionnaires, speed survey, data analysis, system analysis and literature review.

3.1 Questionnaires

Questionnaires were prepared for the project. The function was to identify the various causes of traffic snarls in Nairobi. The result was used to identify the least to the most significant causes of traffic snarls on Nairobi roads. Use was made of mainly open-ended questions as they allow the user to fully express their views or opinions.

3.2 Speed Survey

A speed survey was conducted using a hand held GPS for a week. This involved collecting the coordinates at every one second interval from Embakassi to Gigiri every morning and evening. The results were to establish a general average speed of travel and identification of problem spots on the roads.

3.3 Data analysis

Data from KIPPRA was collected and analyzed to train the neural networks to be developed. Since real time traffic data is not available due to lack of sensors, the data from KIPPRA is used to simulate how the system would behave.

3.4 System investigation

To develop an effective DSS, attention has to be given to system analysis and design phase, resulting to better understanding, modular iterative construction and less errors. The system investigation involved the following design methods:

1. Development of data flow diagrams, context diagrams, and entity relations diagrams.
2. Iterative development with unit testing
3. Rapid application design,

3.5 Literature Review

Particular focus on literature that covers theories and research on neural networks for time series prediction and route search was evaluated. Primary sources were published research papers and secondary were books, doctorate and masters thesis written by previous researchers in the aforementioned topics.

3.6 Is the project tractable?

Kasabov [K98] identifies difficulties in solving the prediction problem requiring addressing before any serious prediction is carried out:

- *Is the process predictable at all?*
    - From literature review, traffic congestion is a seasonal phenomenon which can be modeled and thereafter predicted.
- *What is the type of data available and the process subject to prediction?*
    - Within the context of this research, the only data available is traffic volumes as collected by KIPPRA in 2004. This is sufficient to model traffic speeds given Greenshield's model parameters, and estimating.
- *What are the right features for presenting the prediction problem?*
    - The right features for presenting prediction problem in this context is traffic speeds at 30 minutes interval on critical locations or sections.
    - Link travel time is used as both the link and node cost Park et al [PSHJ05]
- *Defining how much past data are required for a good prediction?*
    - Within the context of this study, not enough data is available. As such the problem of over fitting is critical in evaluation of a suitable model.
- *Defining a methodology to test the accuracy of the prediction.*
    - A GPS reading of traffic speeds is used to validate the results of the model to test the accuracy. Where not applicable, validation error is measured to ensure that the model is general.

The above support the project is indeed tractable and a prototype can be developed. Lingras et al [LS99] identify critical features which were included in this prototype:

1. Typical input for traffic prediction should include:
    - Previous traffic data (avoid late night and early morning)
2. Characteristics to aim at in the model (These are critical requirements)
    - Adaptive,
    - Able to process time series,
    - Able to output range of values as opposed to a single precise output,

- Resistance to noise,
- Fast computation speed,

3.7 Project Data

The primary source of data used is based on KIPPRA traffic survey 2004. This data was imported into SQL server. Using Greenshields theorem, the traffic volume is converted to speeds for purposes of prediction using a neural network.

3.7.1.1 KIPPRA Traffic Count

The Kenya Institute of Public Policy and Research undertook a full week traffic count of various locations in Nairobi in 2004 during the month of January, May and June. Data availed for this research is for the locations as depicted on the map shown in figure 3-1.

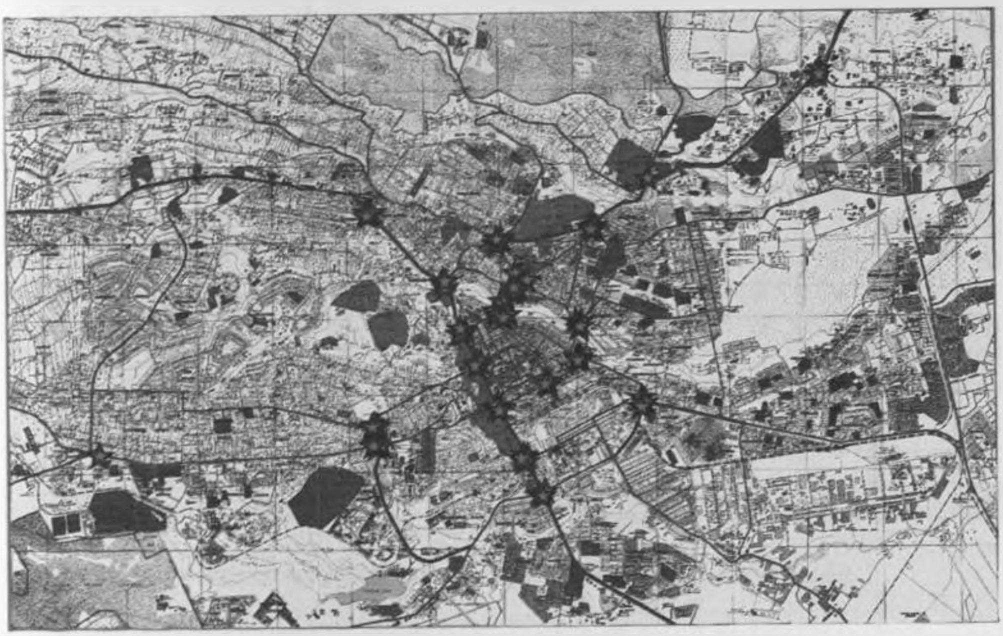


Figure 3-1 KIPPRA data collection sites

Appendix H has a detailed list of all the locations.

3.7.1.2 KIPPRA traffic Survey methodology

Manual methods were used to obtain the traffic counts, whereby all the vehicles entering the marked section were counted and respective volumes obtained at 30 minutes intervals. The vehicles were then classified into different categories as detailed in Table 3-1. The same exercise was carried out for different days of a month and year.

STATION 15: UHURU HIGHWAY - BUNYALA RD R/A

Tuesday January 27, 2004

SURVEY TYPE 1: ARM 1 - UHURU HIGHWAY NORTH

Period	Cars			Matatus			Buses			Lorries		
	APP	DEP	Total	APP	DEP	Total	APP	DEP	Total	APP	DEP	Total
7:00 - 7:30AM	709	1315	2024	213	207	420	15	14	29	65	25	90
7:30 - 8:00AM	767	1147	1914	284	247	531	17	4	21	52	33	85
8:00 - 8:30 AM	742	809	1551	208	205	413	6	0	6	53	34	87
8:30 - 9:00 AM	718	971	1689	226	261	487	9	2	11	54	65	119
9:00 - 9:30 AM	817	992	1809	210	198	408	7	3	10	80	72	152
9:30 - 10:00 AM	813	951	1764	204	262	466	2	2	4	91	40	131
10:00 - 10:30 AM	737	1093	1830	178	258	436	5	2	7	49	64	113
10:30 - 11:00 AM	707	1055	1762	167	269	436	4	0	4	69	69	138
11:00 - 11:30 AM	643	1181	1824	167	238	405	3	0	3	54	49	103
11:30 - 12:00 AM	677	1182	1859	219	287	506	2	3	5	74	54	128
12:00 - 12:30 PM	632	1306	1938	170	278	448	1	1	2	56	35	91
12:30 - 1:00 PM	671	1320	1991	189	253	442	2	3	5	84	38	122
1:00 - 1:30 PM	567	1161	1728	218	209	427	8	1	9	53	36	89
1:30 - 2:00 PM	657	1108	1765	164	250	414	2	1	3	51	48	99
2:00 - 2:30 PM	662	1219	1881	136	297	433	2	0	2	74	39	113
2:30 - 3:00 PM	649	1198	1847	159	218	377	0	0	0	51	48	99
3:00 - 3:30 PM	634	1346	1980	176	345	521	2	4	6	50	62	112
3:30 - 4:00 PM	610	1223	1833	180	415	595	3	2	5	45	33	78
4:00 - 4:30 PM	662	1227	1889	188	267	455	6	10	16	60	45	105
4:30 - 5:00 PM	895	1327	2222	259	343	602	5	9	14	38	44	82
5:00 - 5:30 PM	978	1186	2164	299	325	624	17	5	22	41	33	74
5:30 - 6:00 PM	835	1315	2150	278	418	696	6	13	19	33	34	67
6:00 - 6:30 PM	966	1626	2592	299	598	897	9	13	22	36	33	69
6:30 - 7:00 PM	1041	1262	2303	309	428	737	8	4	12	34	29	63
Day Total	17789	28520	46309	5100	7076	12176	141	96	237	1347	1062	2409

Table 3-1 Sample KIPPRA Data Set

The volume counts are converted to equivalent speeds using Greenshield model.

3.7.1.3 Application of Greenshield Model to Nairobi Traffic

Based on the equations derived based on speed, density and flow relationships, the following assumptions are made:

1. The flow (q) has a value of 1200 vehicles/hour as reported in a report by Okioga [O04] from a survey of Uhuru Highway.
2. The free flow speed of traffic is taken at 50 km/hour based on the Traffic act of Kenya governing the safe speed of moving vehicles within the central business districts and Nairobi environs which is the scope of this study.

Using Microsoft Excel, a scenario is build to determine the jam density of Nairobi Roads. The table 3-2 depicts the values at various jam densities:

Tabulation of Density, Speed and Flow for Nairobi Roads			
% of Jam Density	Density (k) k=%*(A/B)	Speed (v) v=A-B*K	Flow (q) q=k*v
0%	0	50	0
10%	9.6	45	432
20%	19.2	40	768
30%	28.8	35	1008
40%	38.4	30	1152
50%	48	25	1200
60%	57.6	20	1152
70%	67.2	15	1008
80%	76.8	10	768
90%	86.4	5	432
100%	96	0	0

Table 3-2 Jam density values

From the speed and flow columns as shown in the table 3-2, a generalized equation is formed from the graph using linear regression. This assumption is in line with Greenshields' linear model relationship.

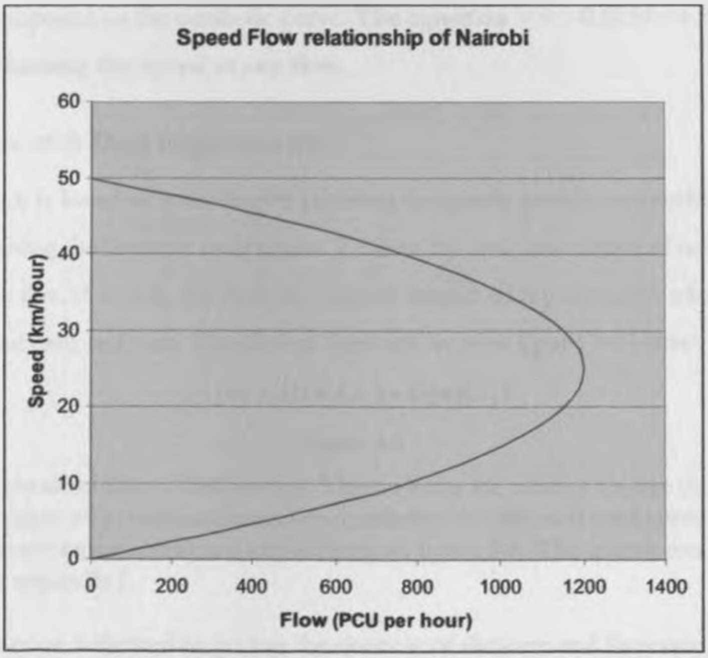


Figure 3-2 Speed Flow Relationship for Nairobi

A generalized liner equation is then assumed for the top half of the parabola as shown on the graph figure 3-3:

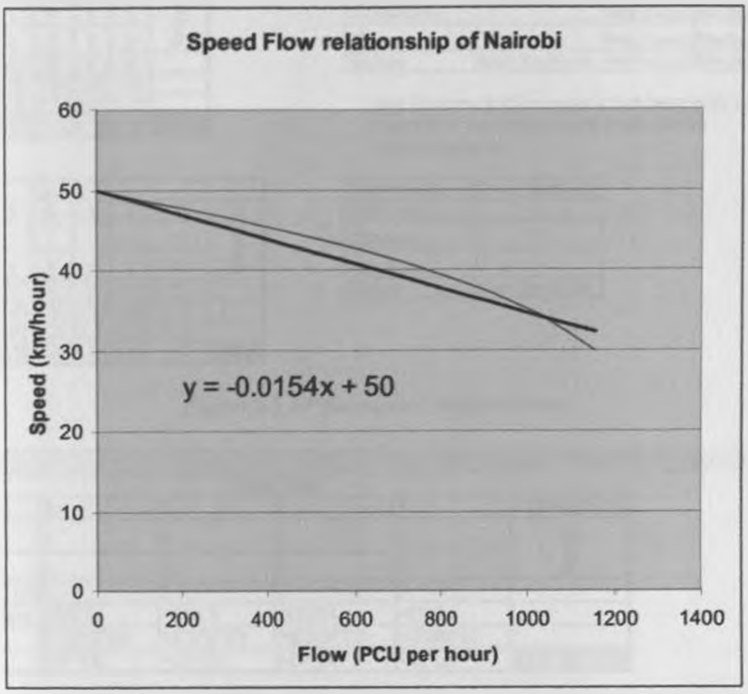


Figure 3-3 Speed Flow relationship of nairobi

To ease computation, a liner regression model is used to generalize the line into an equation as imposed on the parabolic curve. **The equation** $v = -0.0154x + 50$ **will be used in evaluating the speed at any flow.**

3.7.2 A* search Data requirements

The A* search is based on road lengths (forming the greedy search) and euclidian distance forming the heuristic component. To have the real time nature of traffic embedded in the A* search, the basic formulae is jittered with parameters whose values are updated at near real time. The general form will be as in figure 3-4 below

$$f(n) = H(n)(J_1) + G(n)(J_2)$$

Figure 3-4

J_1 and J_2 are obtained from a field survey. They indicate the relative change due to temporal changes of a roads attributes hence redefine the physical road network. The results of questionnaire survey are summarized on figure 3-5. The questionnaire is presented in appendix I.

The cost function is derived by getting the quotient of distance and forecasted travel speed. By using time as a measure of cost, the network becomes near Euclidian. This is because of the varying speeds of roads in the network.

Node Data					
Details	4	3	2	1	10
Min Item	1	1	1	1	Range
Max Item	5	3	3	3	
Value	6	3	3	3	
Significance	1.00	1.00	1.00	1.00	
Weight	0.40	0.30	0.20	0.10	
Factor	0.40	0.30	0.20	0.10	[illegible]

			Max	Min
1	Police	Least Significant	Max impact	Min Impact
2	Observation		Max impact	Min Impact
3	Security		Max impact	Min Impact
4	Slope	Most Significant	Max impact	Min Impact

* Max (1) means that the impact is high hence while min (0.28) means
*the impact is low hence the road length reduces
*D=D(1+impedance)

Line						
Details	5	4	3	2	1	15
Min Item	1	1	1	1	1	Range
Max Item	3	3	3	3	2	
Value	3	3	3	3	2	
Significance	1.00	1.00	1.00	1.00	1.00	
Weight	0.33	0.27	0.20	0.13	0.07	
Factor	0.33	0.27	0.20	0.13	0.07	[illegible]

1	Observation	Least Significant
2	Security	
3	Surface	
4	Drainage	
5	lanes	Most Significant

Figure 3-5 A* parameter compensators

	A	B	C	D	E	F	G
1	Node Data						
2	Details	4	3	2	1	=SUM(B2:E2)	
3	Min Item	1	1	1	1	Range	
4	Max Item	5	3	3	3		
5	Value	5	3	3	3		
6	Significance	=B5/B4	=C5/C4	=D5/D4	=E5/E4		
7	Weight	=(B2/$F$2)	=(C2/$F$2)	=(D2/$F$2)	=(E2/$F$2)		
8	Factor	=B7*B6	=C7*C6	=D7*D6	=E7*E6	=SUM(B8:[illegible]	
9							
10							
11	Line						
12	Details	5	4	3	2	1	=SUM(B12:F12)
13	Min Item	1	1	1	1	1	Range
14	Max Item	3	3	3	3	2	
15	Value	3	3	3	3	2	
16	Significance	=B15/B14	=C15/C14	=D15/D14	=E15/E14	=F15/F14	
17	Weight	=(B12/$G$12)	=(C12/$G$12)	=(D12/$G$12)	=(E12/$G$12)	=(F12/$G$12)	
18	Factor	=B17*B16	=C17*C16	=D17*D16	=E17*E16	=F17*F16	[illegible]
19							

Figure 3-6 Parameter formula relationship

The node values (Slope, security, observation, police enumerated as 4 to 1) were randomly generated as an actual survey was not carried out. Table 3-4 illustrates the code used to randomize the parameters as found in SQL server database. The total impact of node friction is a weighted sum of individual items, with the slope being most significant and police with the least.With respect to the node, figure 3-5 says:

1. The effect of the node resistance value of maximum value (1) is to increase the actual length between two nodes. The maximum value is a multiplier effect of 1 while the least is 0.28 when there is no effect of the aforementioned node attributes.
2. Apart from slope with 5 states, all the remaining attributes have a range of 3 with 1 having the effect of reducing the relative impact and 3 increasing the effect.

The same can be mentioned for the link status. As seen in figure 3-5,

1. The number of lanes (enumeration 5), is most significant in determining the level of resistance in traversing a link. The value (randomly generated) has a maximum state of 3 and a minimum state of 1.
2. Similar to the node attributes, the drainage parameter has 5 viable states while the others have only 3.
3. A driver's observation of traffic ahead is least significant in assessing the resistance of traversing a link.

Since time is the overall cost, the evaluated heuristic distance and path length are divided by calculated neural network speed and path speed and summed.

```
SELECT @X1=XCord, @Y1=YCord , @Z1 =ZCord FROM dbo.TblNode WITH ( NOLOCK ) WHERE
NodeID=@Stopnode
SELECT
        StartNode AS NodeId,
        ((Sqrt(Square(@X1-XCord) +Square(@Y1-YCord)) * @Meter) * (1+ ((1/COS(ATAN(
(@Z1-Altitude) / Length)))+((dbo.tblnode.[Security]/3)*(3/10) +
(dbo.tblnode.[Observation]/3) * (2/10)+ (police/3) * (1/10) + (4/10)*(Zcord/5)) ))/
Neuralnetwork AS NodeCost,
        ((Length/1000) * (1+((1/COS(ATAN( (@Z1-Altitude) /
Length)))+(((lanes/3) * (5/15))+((Drainage/3) * (4/15))+((Surface/3) * (3/15))+((dbo.TblLi
ne.[Security]/3) * (2/15))+((dbo.TblLine.Observation/2) * (1/15)))))) / Speed AS PathCost
,
        ((Sqrt(Square(@X1-XCord) +Square(@Y1-YCord)) * @Meter) * (1+ ((1/COS(ATAN(
(@Z1-Altitude) / Length)))+((dbo.tblnode.[Security]/3) * (3/10) +
(dbo.tblnode.[Observation]/3) * (2/10)+ (police/3) * (1/10) + (4/10)*(Zcord/5)) ))/
Neuralnetwork) + (((Length/1000) * (1+((1/COS(ATAN( (@Z1-Altitude) /
Length)))+(((lanes/3) * (5/15))+((Drainage/3) * (4/15))+((Surface/3) * (3/15))+((dbo.TblLi
ne.[Security]/3) * (2/15))+((dbo.TblLine.Observation/2) * (1/15) )))) / Speed) AS
TotalCost,
        RecID AS ArcID
FROM dbo.tblLine
WITH
        ( NOLOCK) INNER JOIN dbo.tblNode
WITh
        (NOLOCK)
ON
        StartNode =NodeID
WHERE
        EndNode=@StartNode
```

Table 3-3 Implementation in SQL Server

Table 3-4 is a snippet of the implementation of the same in SQL Server Database. A detailed treatment of the source code is available in the appendix.

```
--Update the security with random numbers Range 1-3
DECLARE @RecID int
DECLARE @Random float
DECLARE Node_Security CURSOR FOR
SELECT NodeID
FROM tblNode
FOR UPDATE of [Security]
OPEN Node_Security
FETCH NEXT FROM Node_Security INTO @Recid
WHILE @@FETCH_STATUS = 0
BEGIN
        SET @Random = 1+2*rand()
        UPDATE tblNode SET [Security] = @Random where nodeID = @RecID
        FETCH NEXT FROM Node_Security INTO @RecID
END
CLOSE Node_Security
DEALLOCATE Node_Security
```

Table 3-4 Code to randomize the security status of a node.

3.7.3 Digitize GIS Data

A network consists of a number of line segments that are interconnected in some way. Each line segment is defined by start and end nodes, both of known locations. A segment may contain intermediate points of known locations between start and end nodes. Each intermediate point refines the shape of the segment and is called a vertex. The difference between a vertices and nodes is that the nodes carry information about the topological relationships in the network, while vertices exist to simply delineate the segment. The topological relationships defined by the node determine the connectivity of a network. The figure 3-5 below shows the structure of a typical network as found in Nairobi.

Figure 3-7 Section of Nairobi Road Network

3.7.4 Evaluation of Nairobi Network

Nairobi network is evaluated using the γ index to measure the fundamental properties of the network complexity .The γ index is defined by the equation shown:

$$\lambda = \frac{l}{l_{max}} = \frac{l}{3(n-2)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

Where: γ Ratio of the actual number of links to maximum possible number of links in the network which is equal to $3(n\text{-}2)$

n the number of nodes

l is the number of links in the network

Table 3-5 depicts the number of possible links.

γ for Nairobi	
Number of nodes (*n*)	2975
Number of links (from the Database)	3986
Number of links (based on graph theory)	8919
Value of $0 < \gamma < 1$	0.44

Table 3-5 Nairobi Road network Analysis

A value closer to 0 indicates a simpler network structure with fewer links. A lager value close to 1 indicates a better connected network with more links. From the evaluation above, Nairobi has a relatively poor network. The calculated number of links differs from the actual links in the database due to the following reasons:

1. The Nairobi network is not fully connected. There also exists dangle nodes.
2. There exist some digitizing errors with resultant links that do not connect to the overall network.

The figure 3-6 shows the above scenario on the GIS database.

Figure 3-8 Missing arcs

From figure 3-8, a digitizing error exists between node 1249 and 1258 (University way link) next to central police. Both nodes are not fully connected. Using the avenue script (appendix F), all links which are not connected are identified for future correction.
The result of running the correction script is as shown in the figure 3-7 below:

Figure 3-9 Dangle link detection

To resolve the problem of dangle links, the connection link must be digitized and the database topology rebuilt.

3.7.4.1 Digitizing the road network

The main means of converting analogue data into digital data are manual digitizers and scanners. A topographical map and photo image of Nairobi is used in the digitizing process to acquire road network. The topographical map is the primary source of data while the photo image is a secondary source to verify the accuracy of the topographical map. By juxtaposing the digitized topographical map with the photo image, any salient changes in road network are captured, since the topographical map is older than the photo image. Figure 3-10 describes the steps in building the road network end to end.

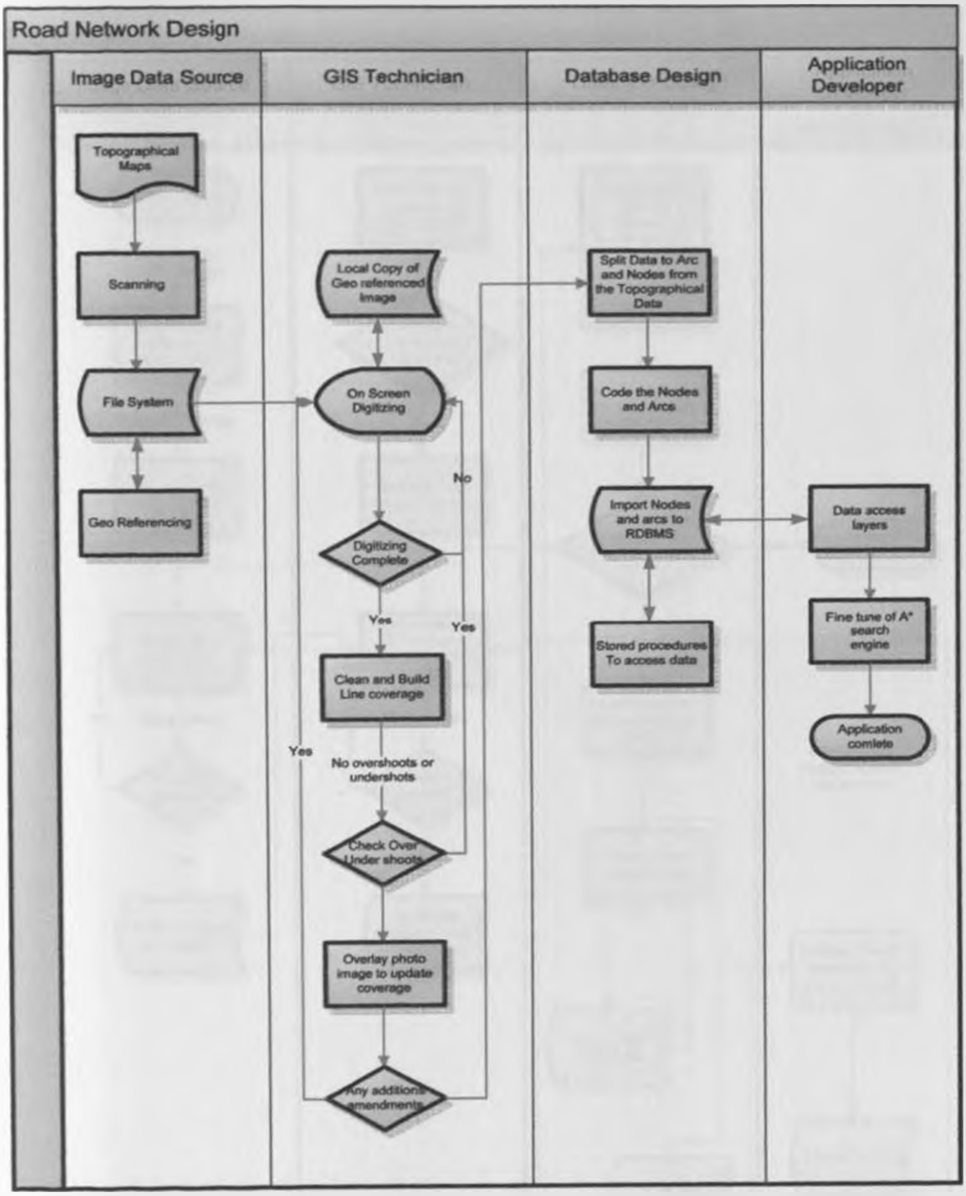


Figure 3-10 Building a complete Road network

Figure 3-10 depicts the detailed process flow of developing the GIS component. From figure 3-10, the GIS technician iteratively digitizes the images to derive the vector data. Since a custom build route analysis system is required and a strong coupling to the neural network is needed according to the system architecture, the road network is imported to a database management system.

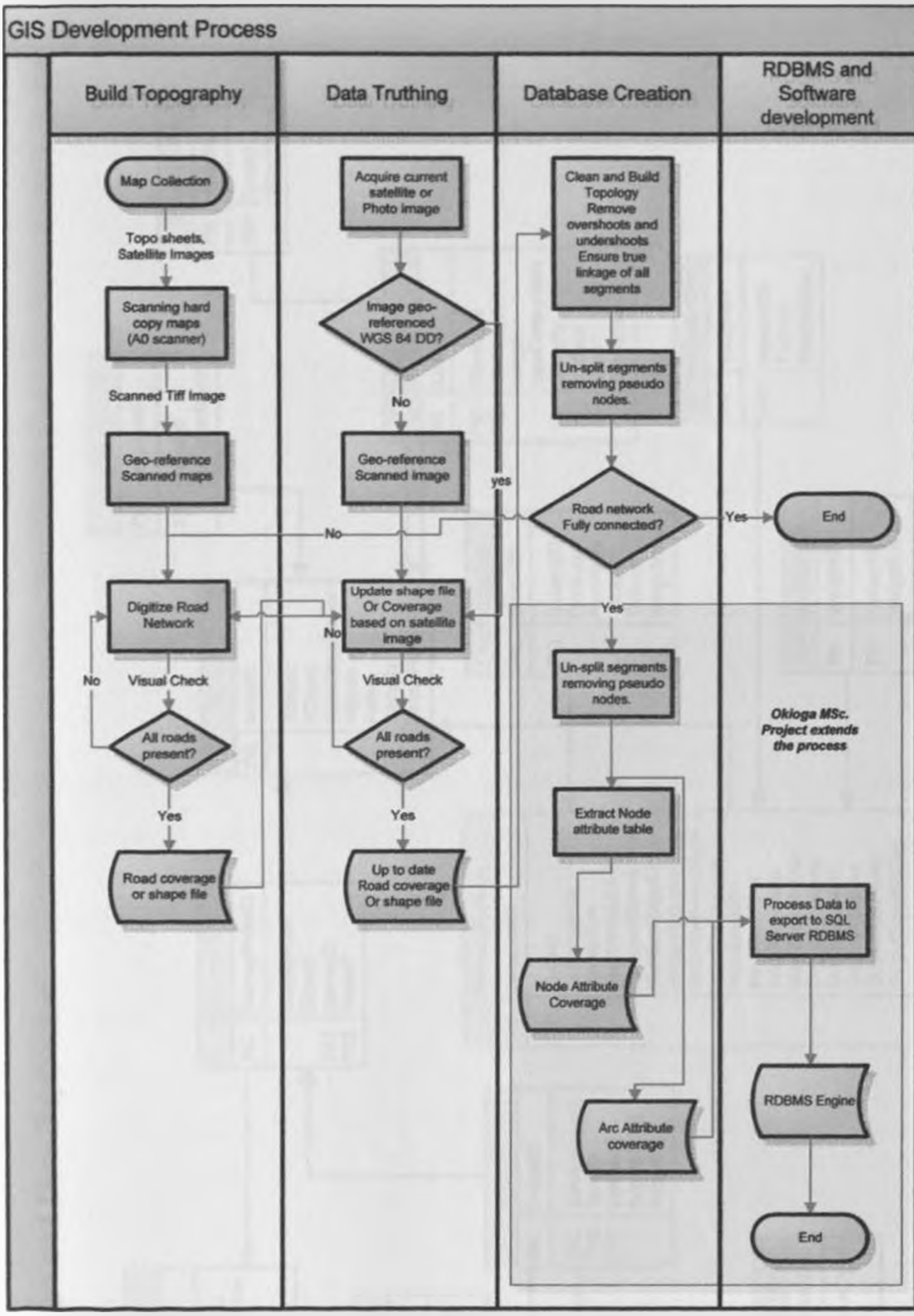


Figure 3-11 GIS Development Process

From figure 3-11, the process of creating the topologically correct road network is described in more detail. Building the topology is a manual process which can be partially automated when it comes to verifying the topology and connectedness of the network. The two most critical components required are the node and arc data.

3.8 Database Design and Data Preparation

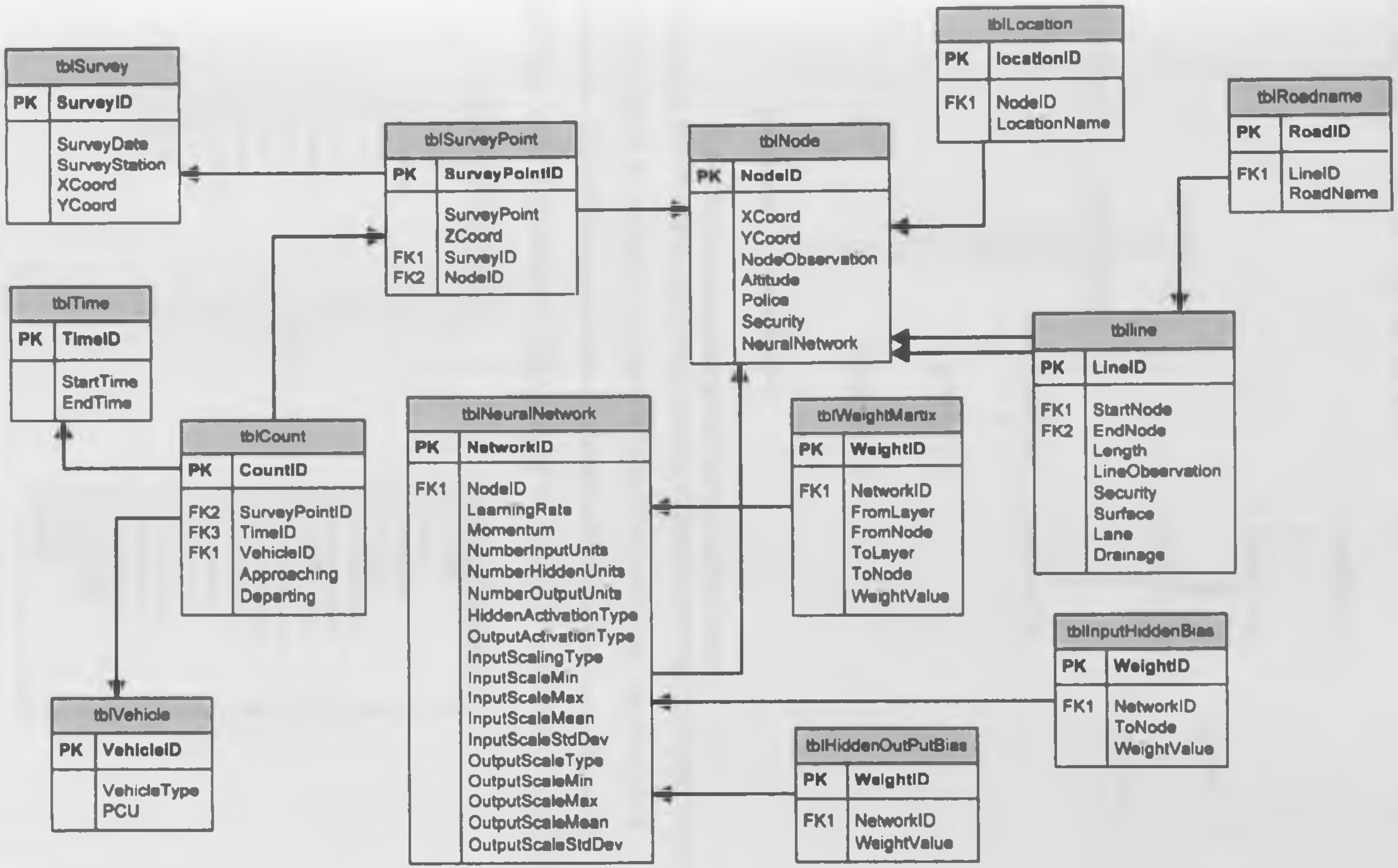


Figure 3-12 System's Complete Database Design Entity Relation Diagram

3.8.1 Database Design and Data Preparation

Figure 3-12 illustrates the complete data ERD while figure 3-13 illustrates the actual implementation of the database on SQL Server 2005 .

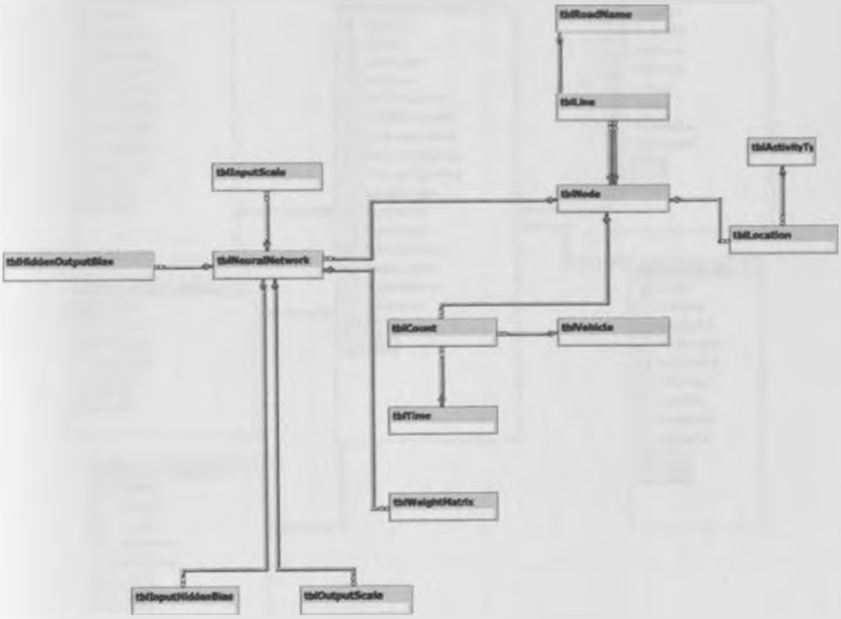


Figure 3-13 Actual implemented Database on SQL Server

Figure 3-13 illustrates the relationship between a node and neural network. Each node has an associated neural network or networks. The table tblNeuralNetwork describes the neural network architecture while tblNode represents an actual road intersection or event on the physical world. Each node is considered a brain suitable for prediction when data is available.

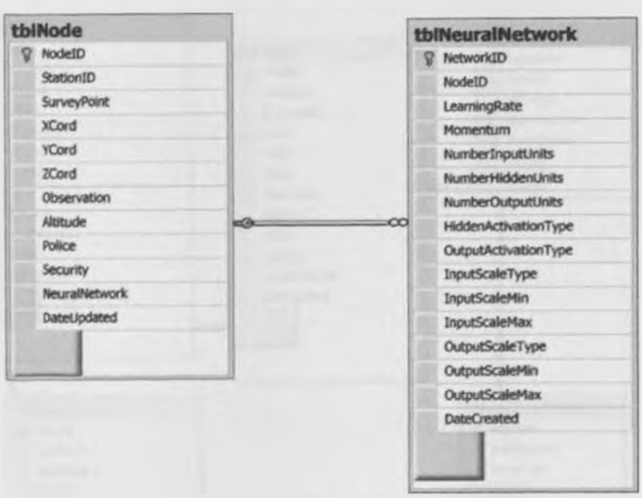


Figure 3-14 Neural Network Association with Road network

Figure 3-14 illustrates the complete neural network ERD. This is made up to 6 tables. Two tables tbloutputscale and tblinputscale store the networks output and input data scaling parameters.

The tblWeightMatrix stores the neural network weights for input-hidden and hidden-output connections while tblHiddenoutputBias and tblinputHiddenbias store the bias weights for the input and output bias values.

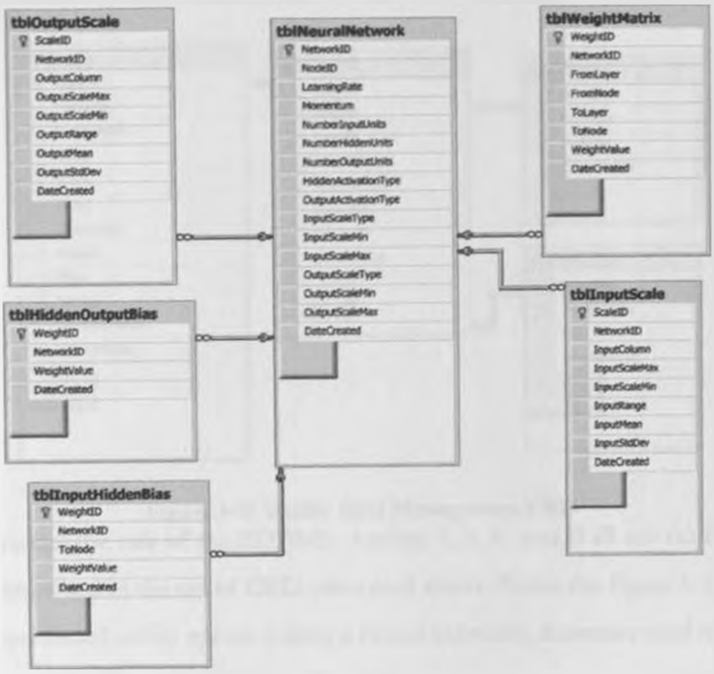


Figure 3-15 Complete Neural Network ERD

Figure 3-15 depicts the route analysis implementation. Both the A* and Dijkstra search depend on the two tables tblLine (which represents roads) and tblnode (which represents any road intersection point). On every update of node speed, all connected arcs from and to the node get updated by the same speed. SQL server take care of this.

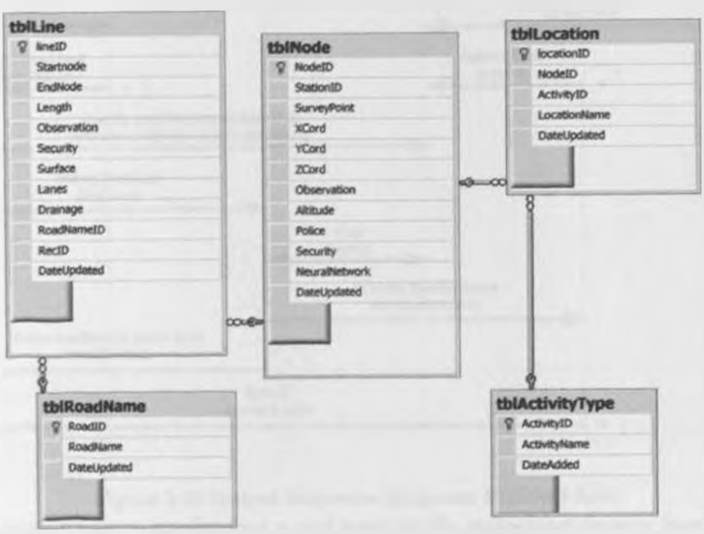


Figure 3-16 Route Analysis ERD

The table tblRoadName identifies all instances of tblline by their known names. The table tbllocation identifies known nodes as buildings, traffic intersection points, speeds bumps, bridges or general location. Table tblactivity categories the locations into general groups. Figure 3-15 illustrates the traffic management tables ERD. The management of traffic data is handled by four tables. The table tblNode identifies all nodes found on the database network. The table tblCount, tbltime and tblVehicle assist in managing any traffic data collected either by a real time traffic sensor or historical data. A complete set of data flow diagrams is presented later in this chapter.

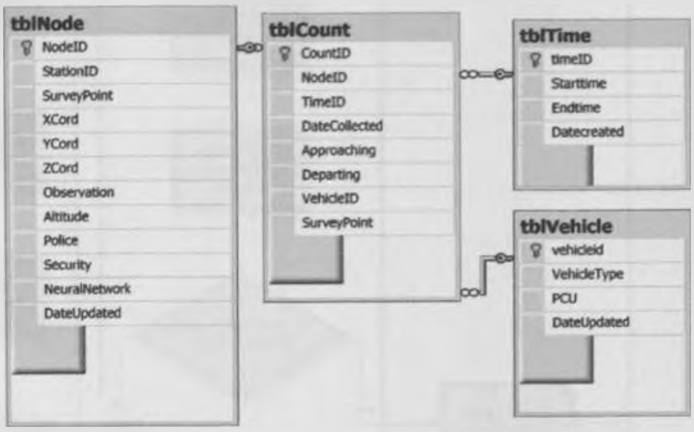


Figure 3-17 Traffic Data Management ERD

Figure 3-17 illustrates the role of the RDBMS. Activity 3, 5, A, and B all rely on full database or some tables as described in the set of ERD presented above. From the figure 3-16, the RDBMS is used to store predicted traffic speeds (using a neural network), maintain road network data, traffic data and provide data to the search algorithm.

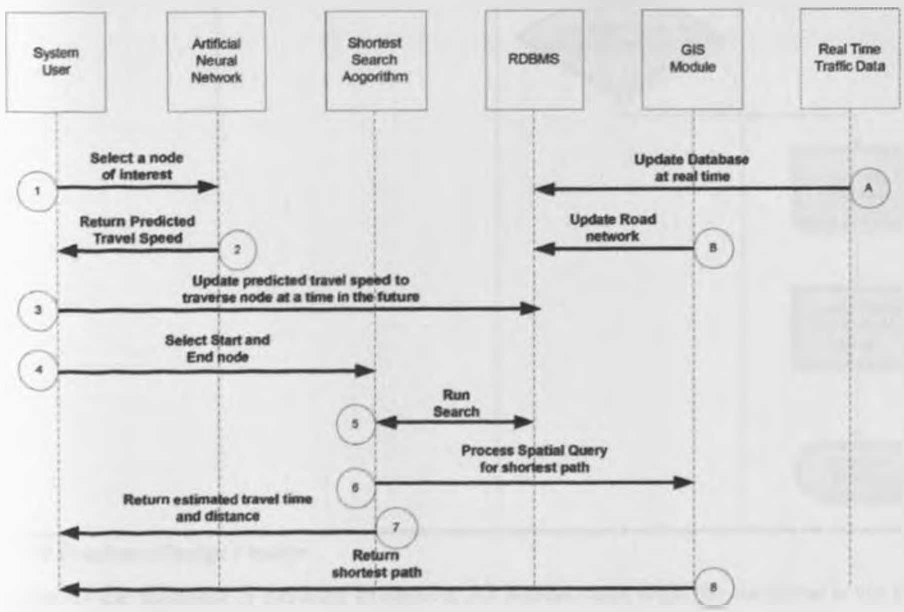


Figure 3-18 System Sequence Diagram: RDBMS Role

This research was not able to implement a real time traffic collection system, hence the use of historical data from KIPPRA to simulate actual speeds. Figure 3-19 summarizes the database design process.

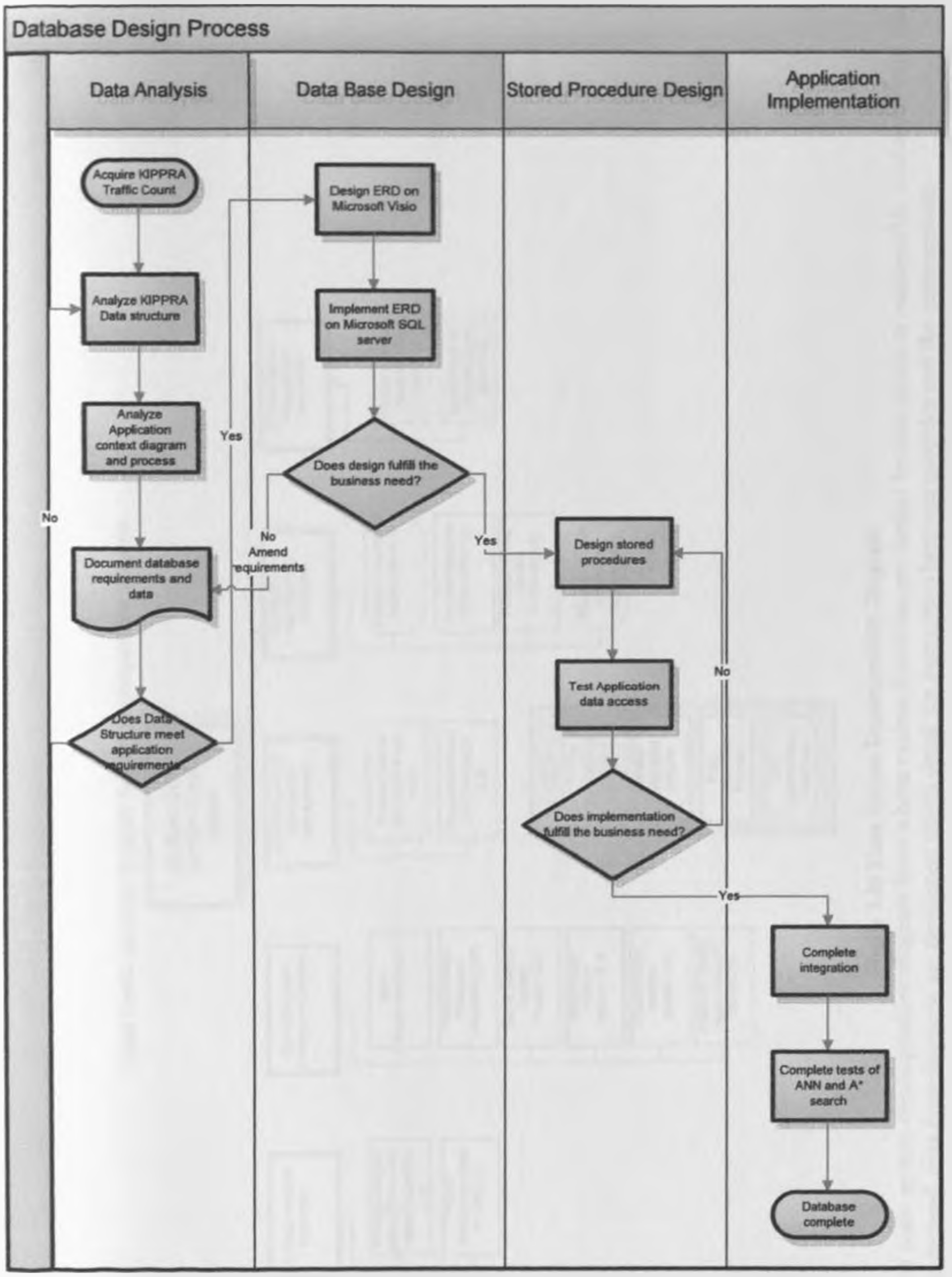


Figure 3-19 Database Design Process

The design of the database is iterative in nature. All interactions with the database is via stored procedures. This makes the database very flexible to update and maintain as all logic related to data access, storage, deletion and update is maintained in a central system independent from application developed.

3.9 System Decomposition Diagram

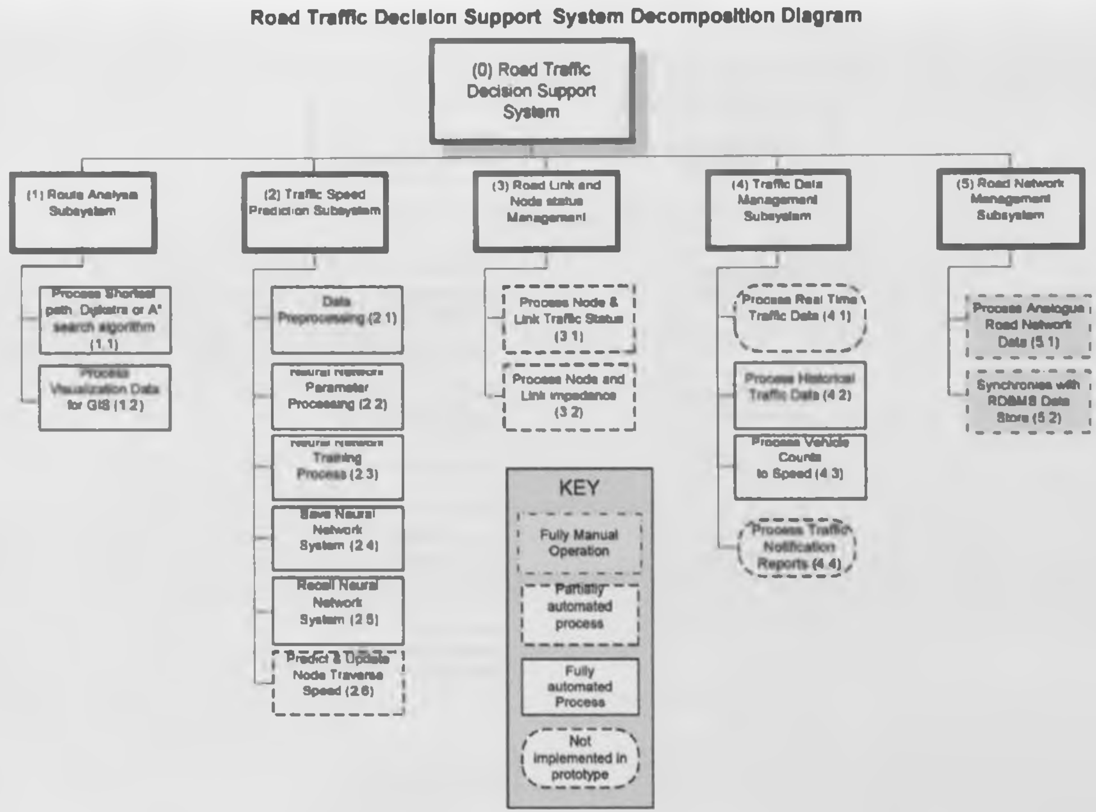


Figure 3-20 Main System Decomposition Diagram

Figure 3-20 illustrates the main system decomposition diagram from where various functions are further broken down to manageable modules. From the decomposition diagrams derived, data flow diagrams are developed which detail the interaction between modules and the environment.

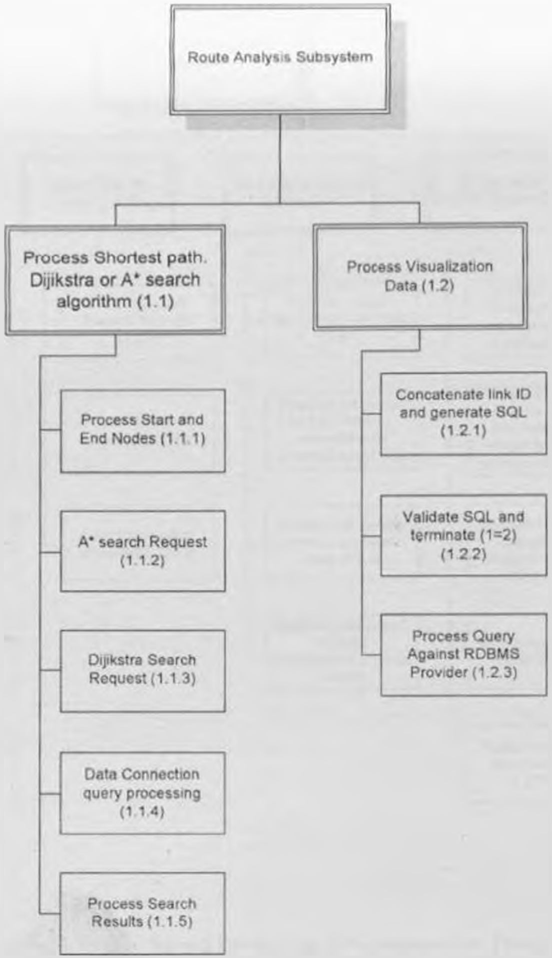


Figure 3-21 Route analysis Decomposition Diagram

The route analysis subsystem determines the shortest path between any two points and sends the resulting paths to the GIS system for visualization.

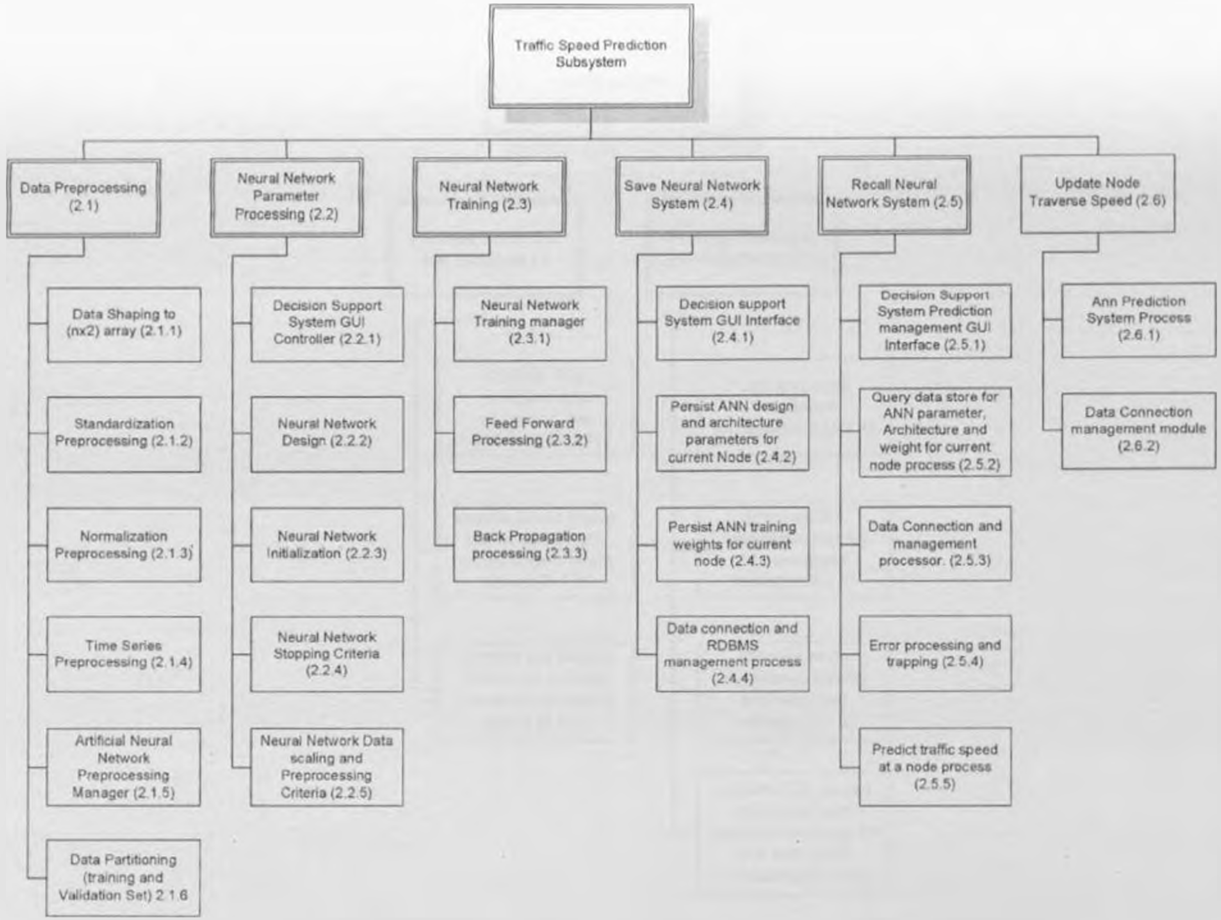


Figure 3-22 Traffic Speed Prediction Decomposition Diagram

Figure 3-22 illustrates the traffic speed prediction subsystem whose role is to predict the speed at one step ahead in the future, given historical data or real time data.

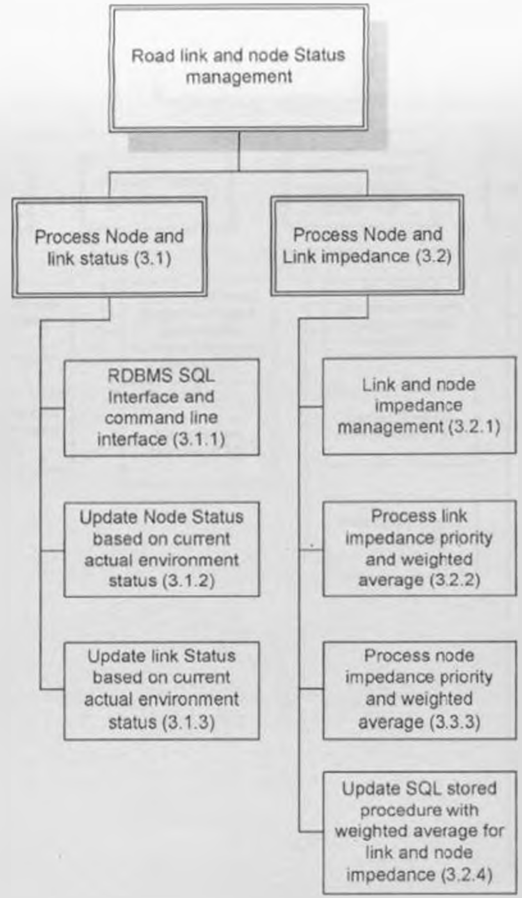


Figure 3-23 Road link and node status management Decomposition diagram

Figure 3-23 is the road link and node status decomposition diagram whose role is to update the link and node attribute status. For instance number of lanes on a road, the drainage status or security , e.t.c

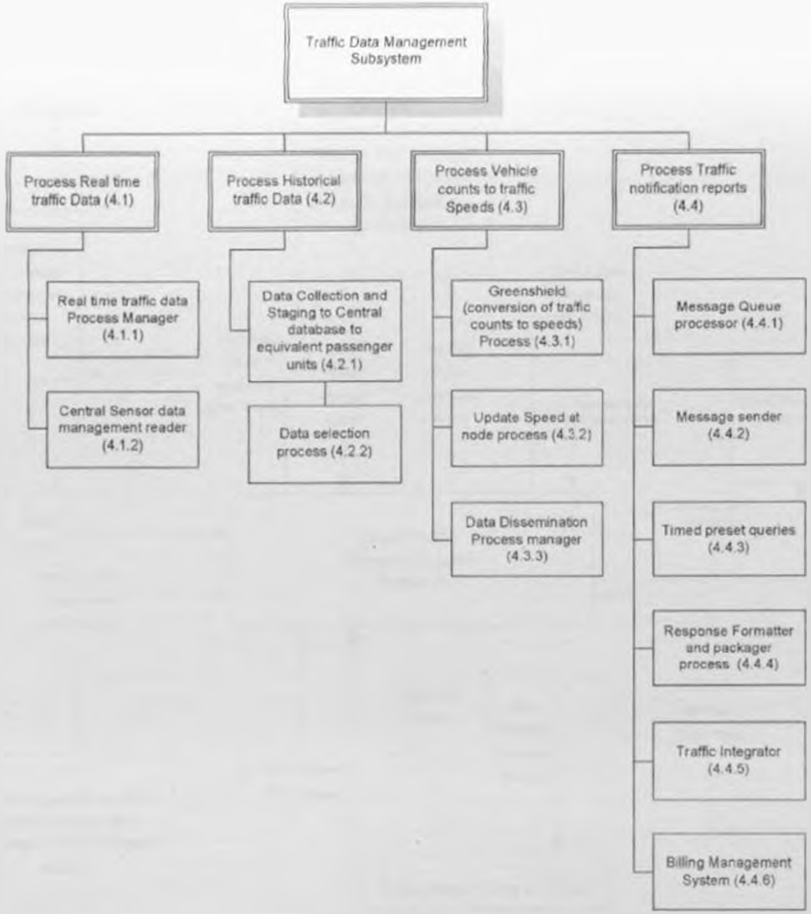


Figure 3-24 Traffic Data Management Subsystem Decomposition Diagram

Figure 3-24 illustrates the Traffic Data management subsystem whose role is to import data to the system. It is assumed that the system will be extended to use real time sensor data in the future however for purposes of simulation, KIPPRA data is used.

3.10 System Data Flow Diagrams

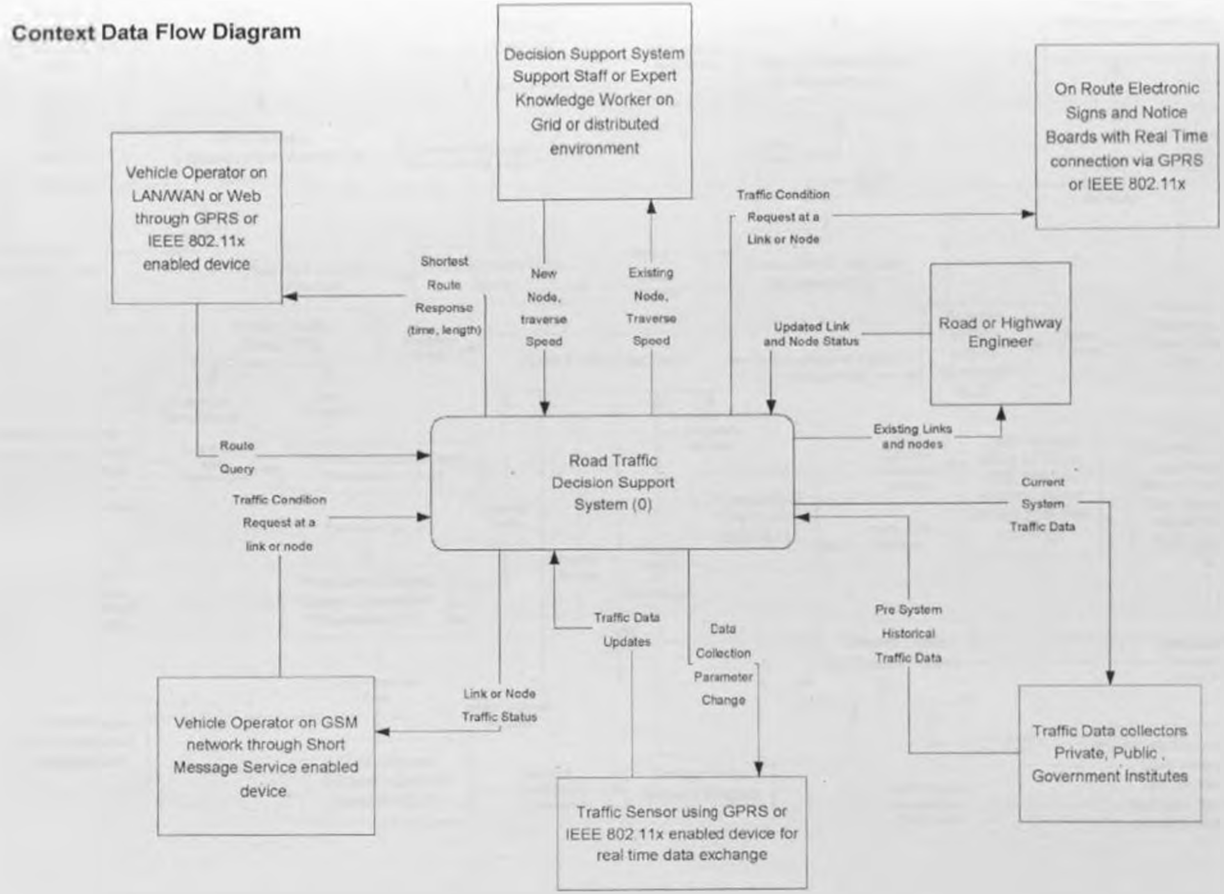


Figure 3-25 System Design Context Diagram

From the decomposition diagram, a series of data flow diagrams were developed to enable finale coding.

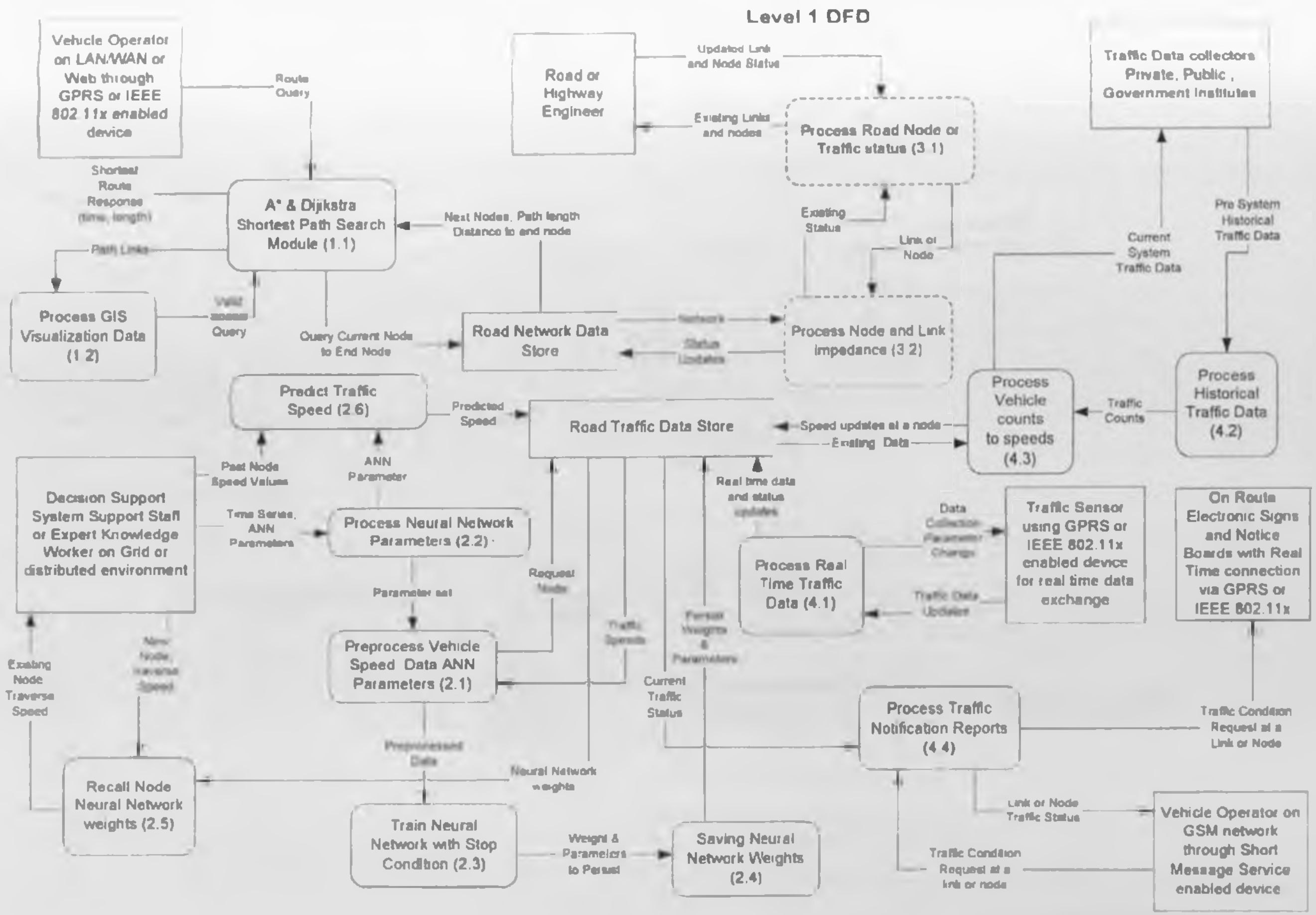


Figure 3-26 Level 1 Data Flow Diagram of the Automated Route Analysis Decision Support System

Figure 3-26 illustrates a detailed overview of the system. Each process is further detailed in subsequent data flow diagrams.

Level 1.1 DFD

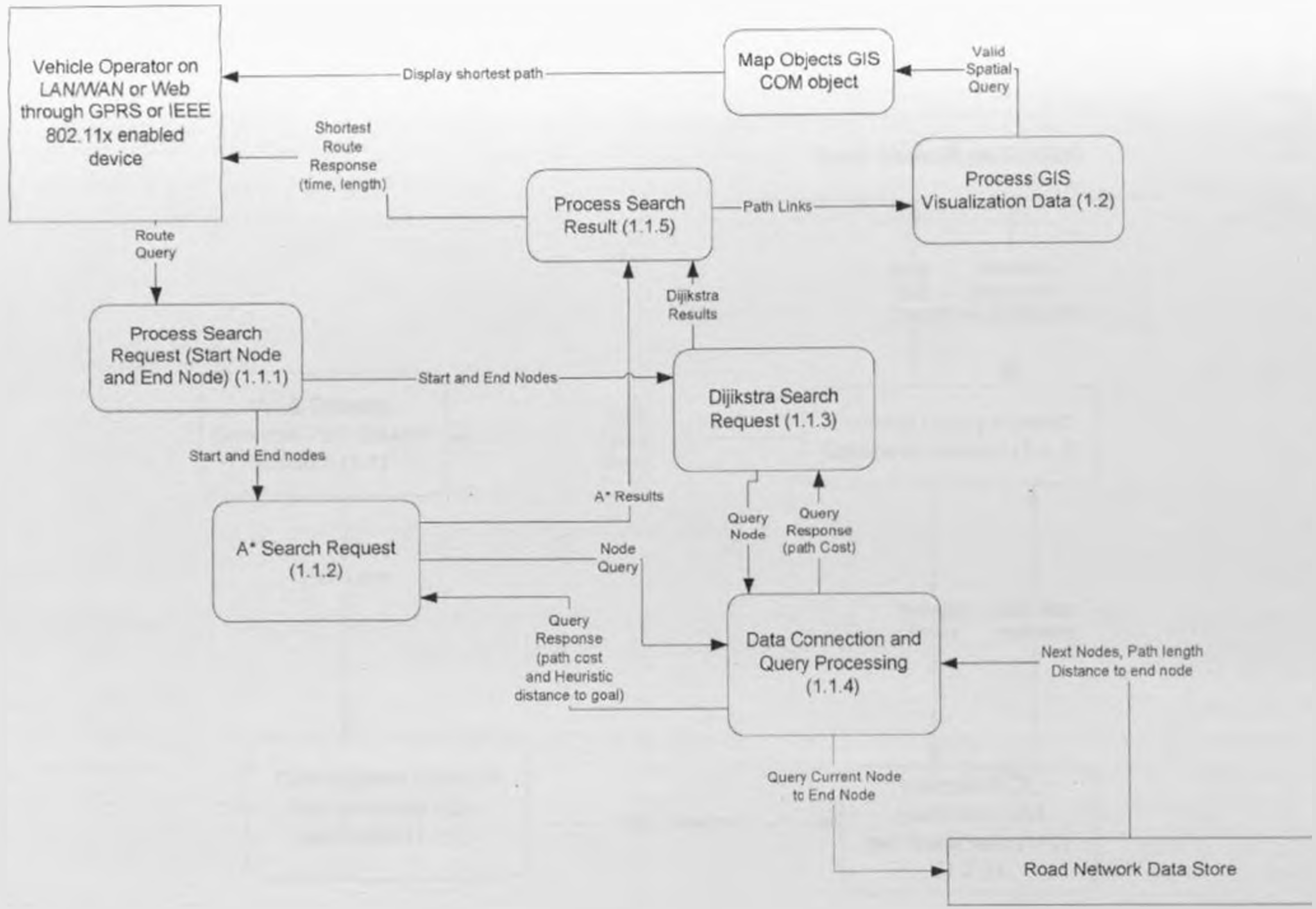


Figure 3-27 Level 1.1 Data Flow Diagram

Level 1.2 DFD

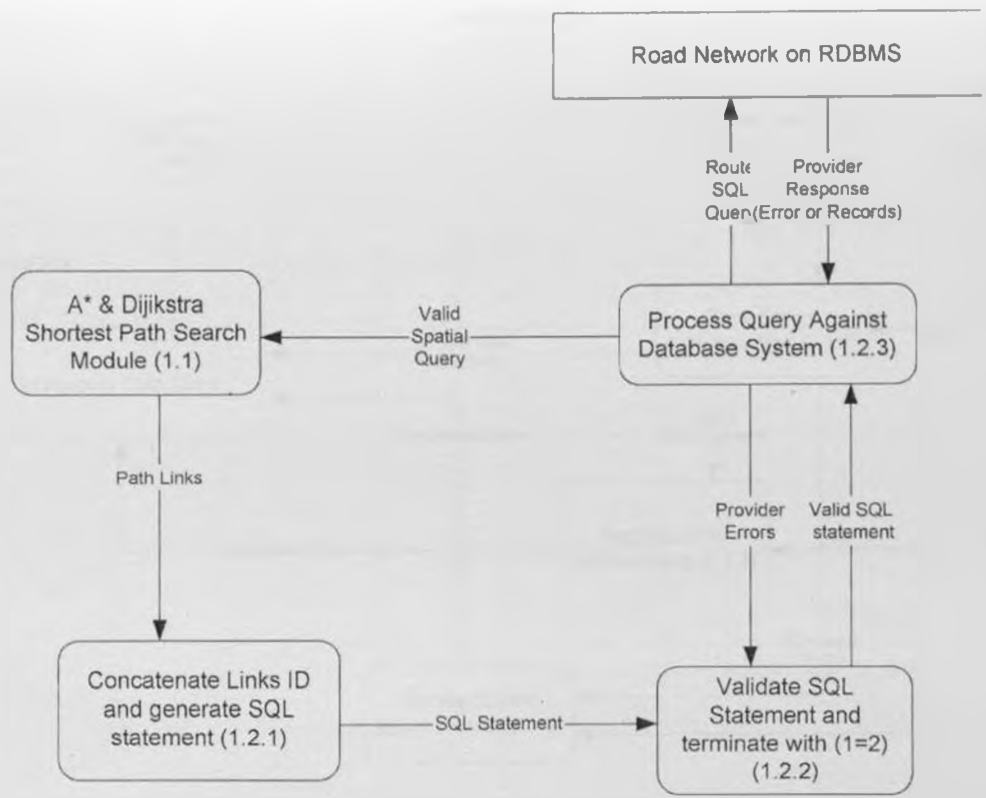


Figure 3-28 Level 1.2 Data Flow Diagram

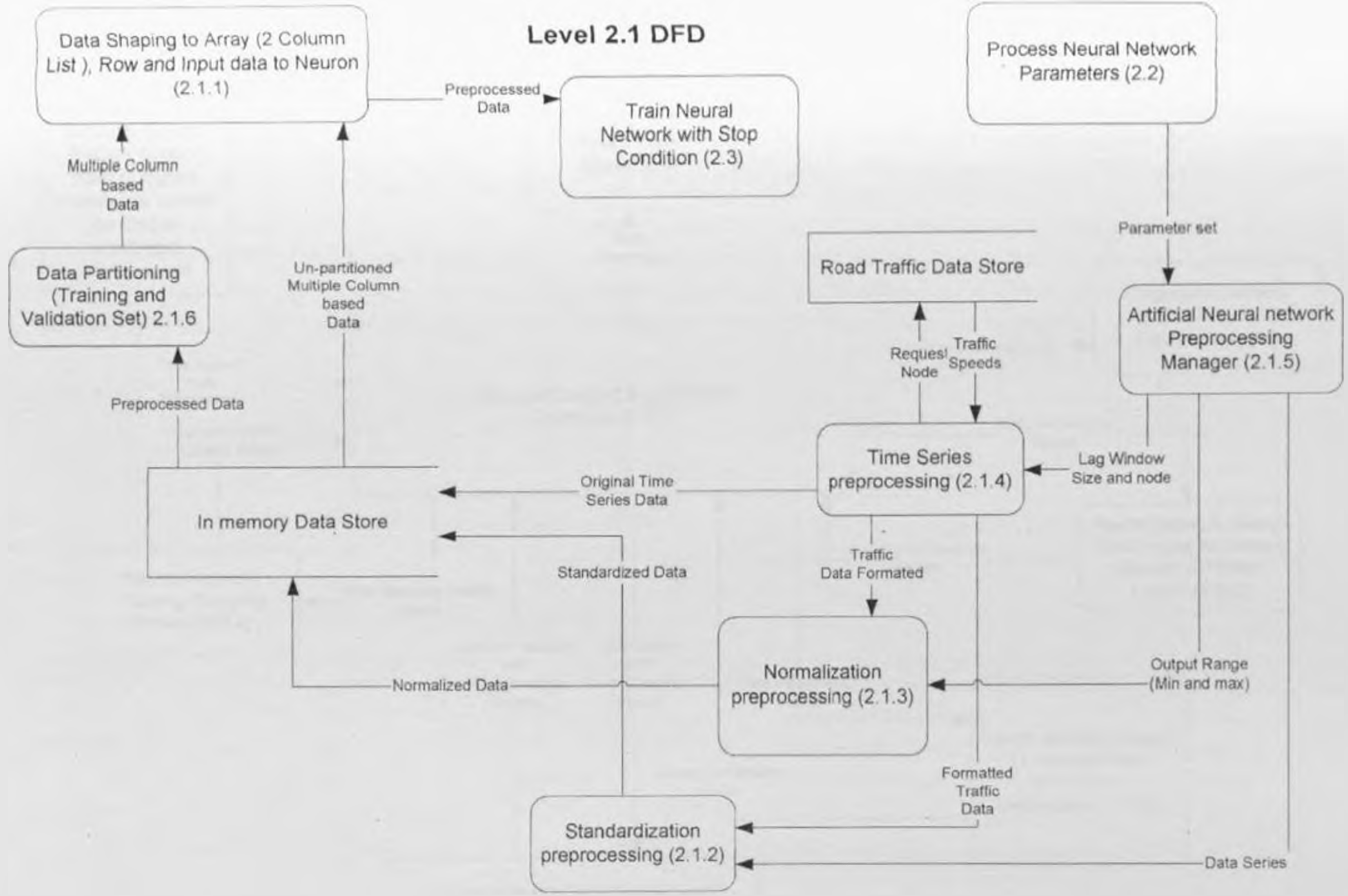


Figure 3-29 Level 2.1 Data Flow Diagram

Level 2.2 DFD

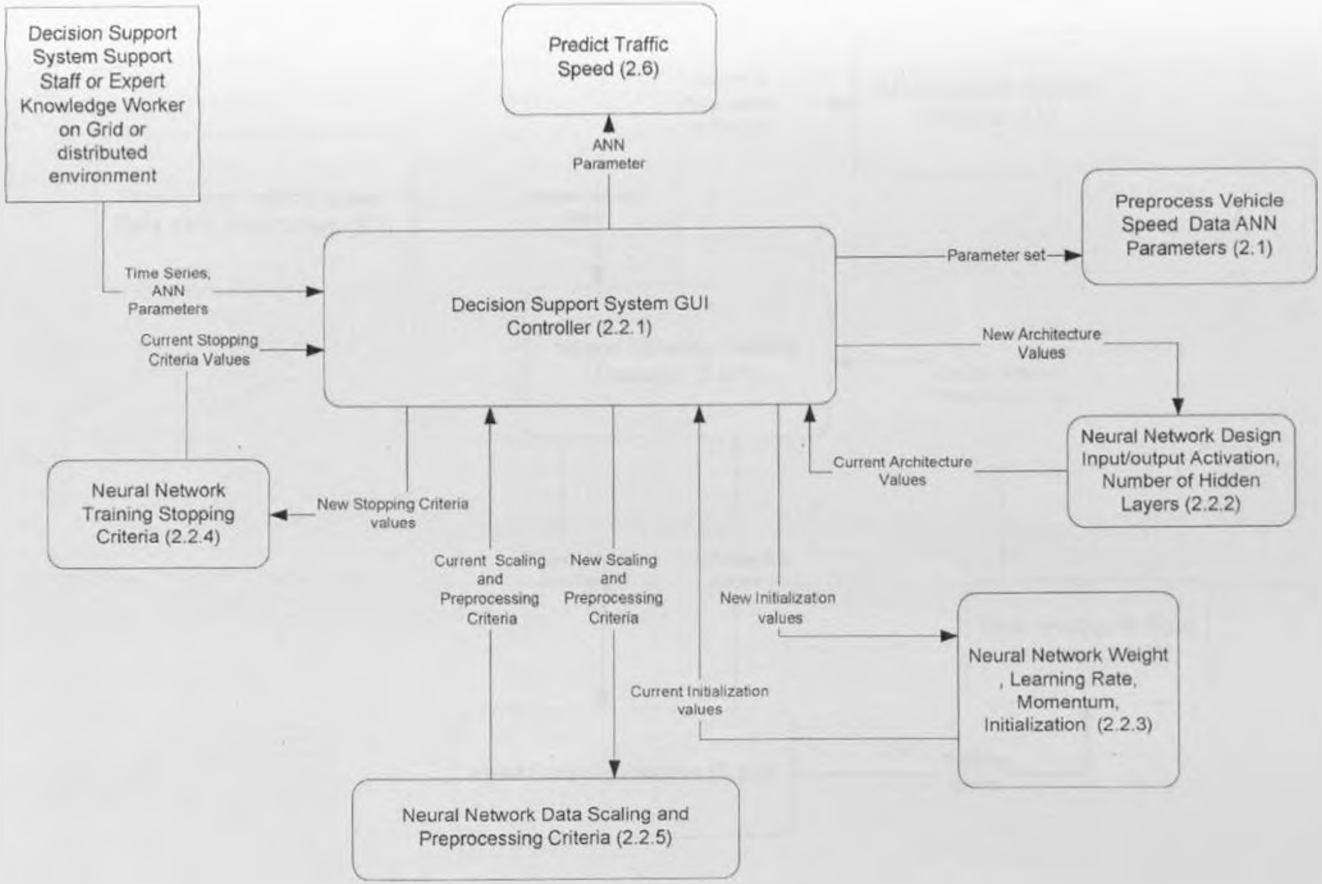


Figure 3-30 Level 2.2 Data Flow Diagram

Level 2.3 DFD

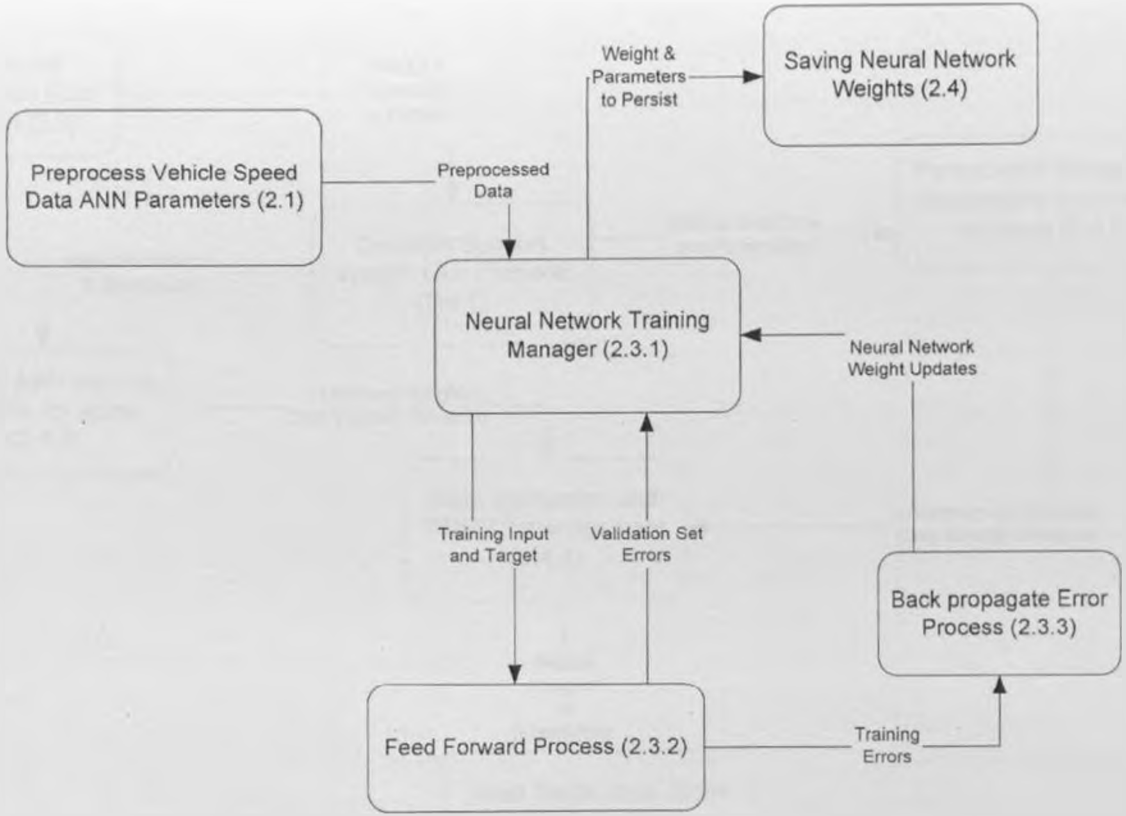


Figure 3-31 Level 2.3Data Flow Diagram

Level 2.4 DFD

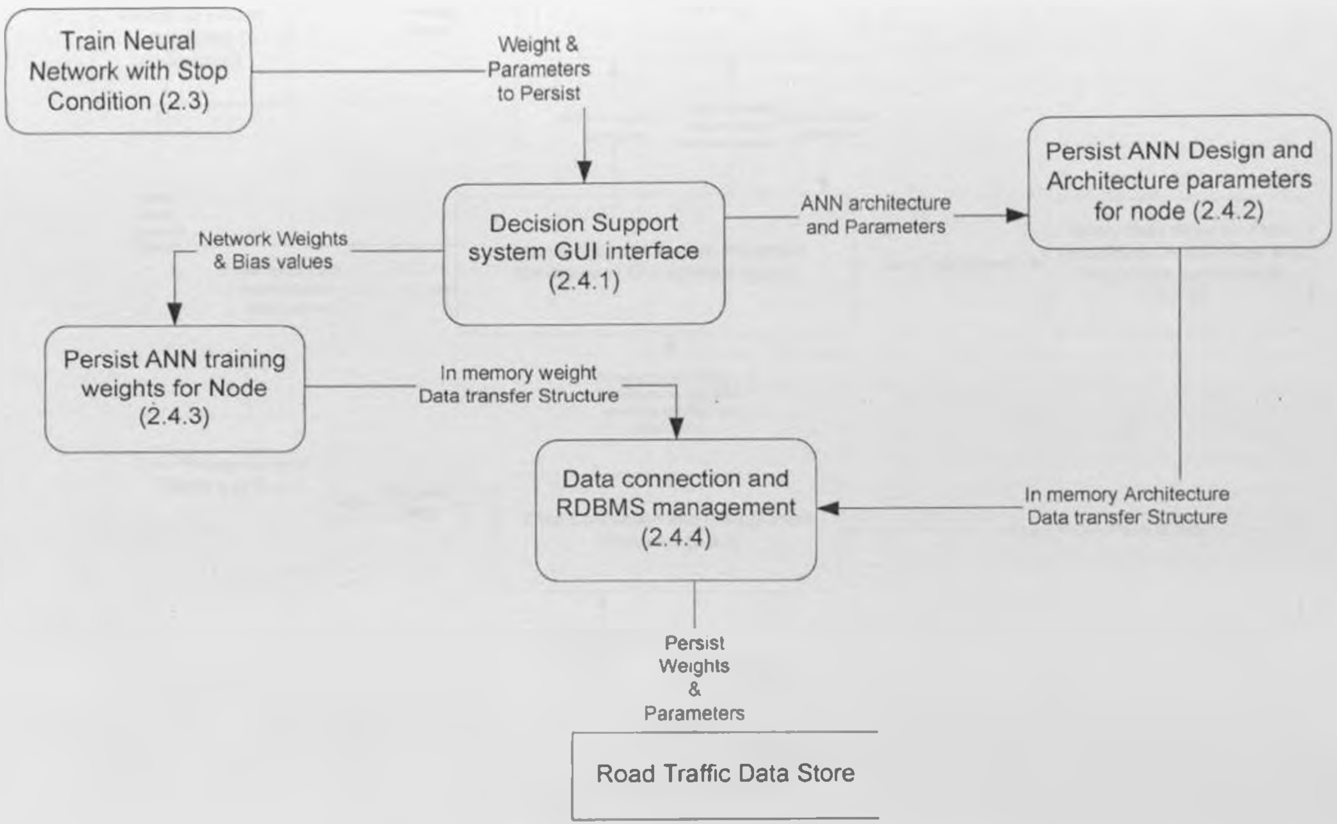


Figure 3-32 Level 2.4 Data Flow Diagram

Level 2.5 DFD

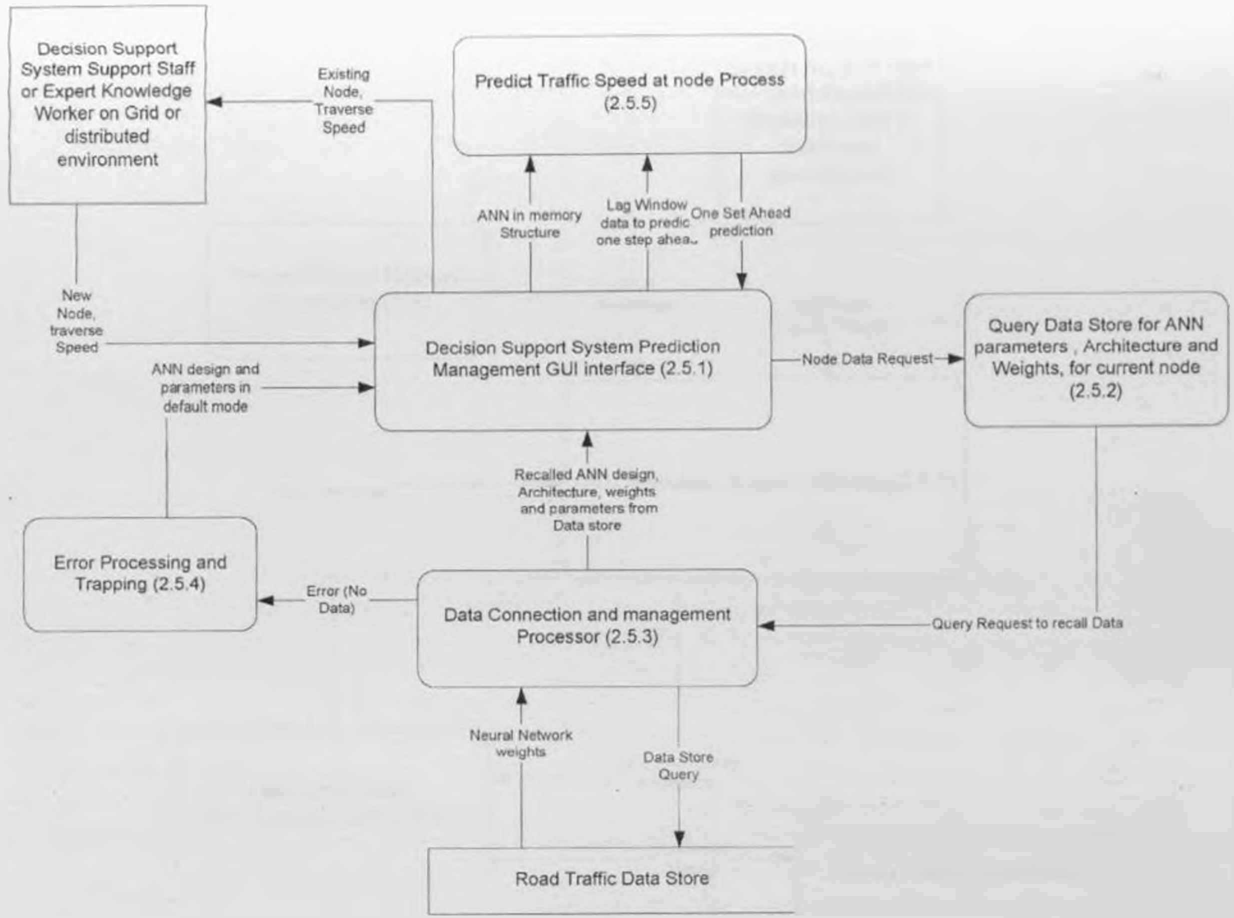


Figure 3-33 Level 2.5 Data Flow Diagram

Level 2.6 DFD

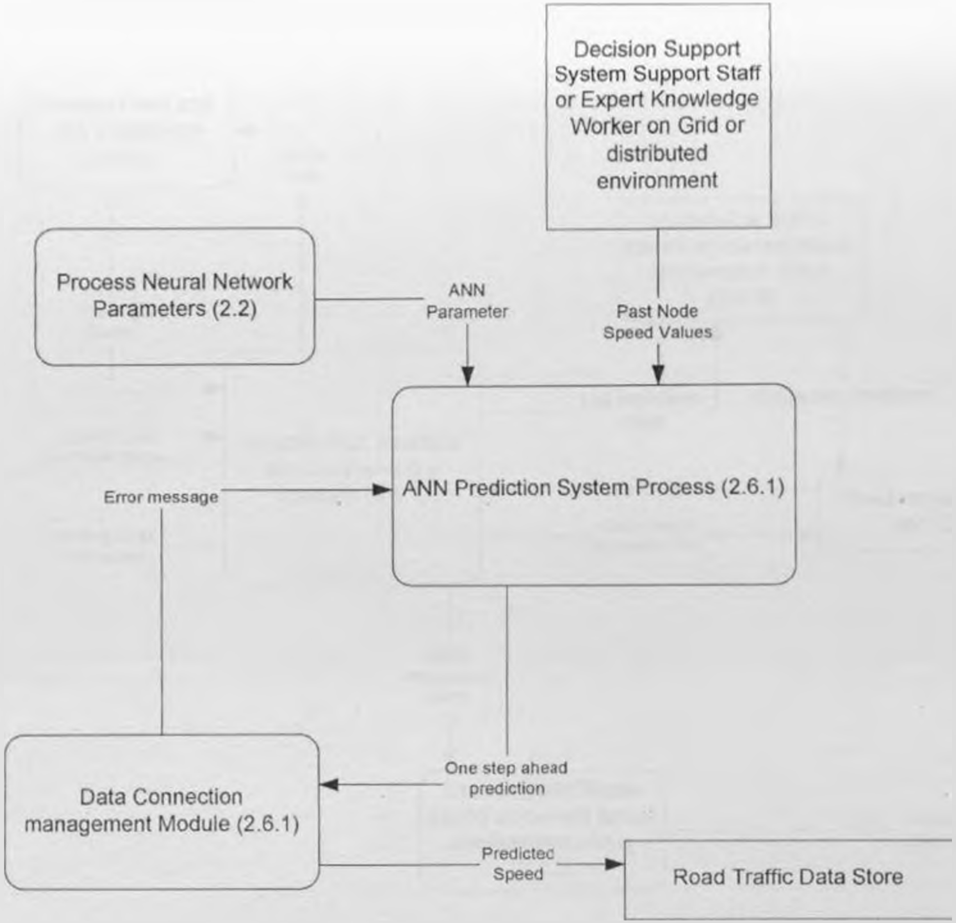


Figure 3-34 Level 2.6 Data Flow Diagram

Level 3.1 DFD

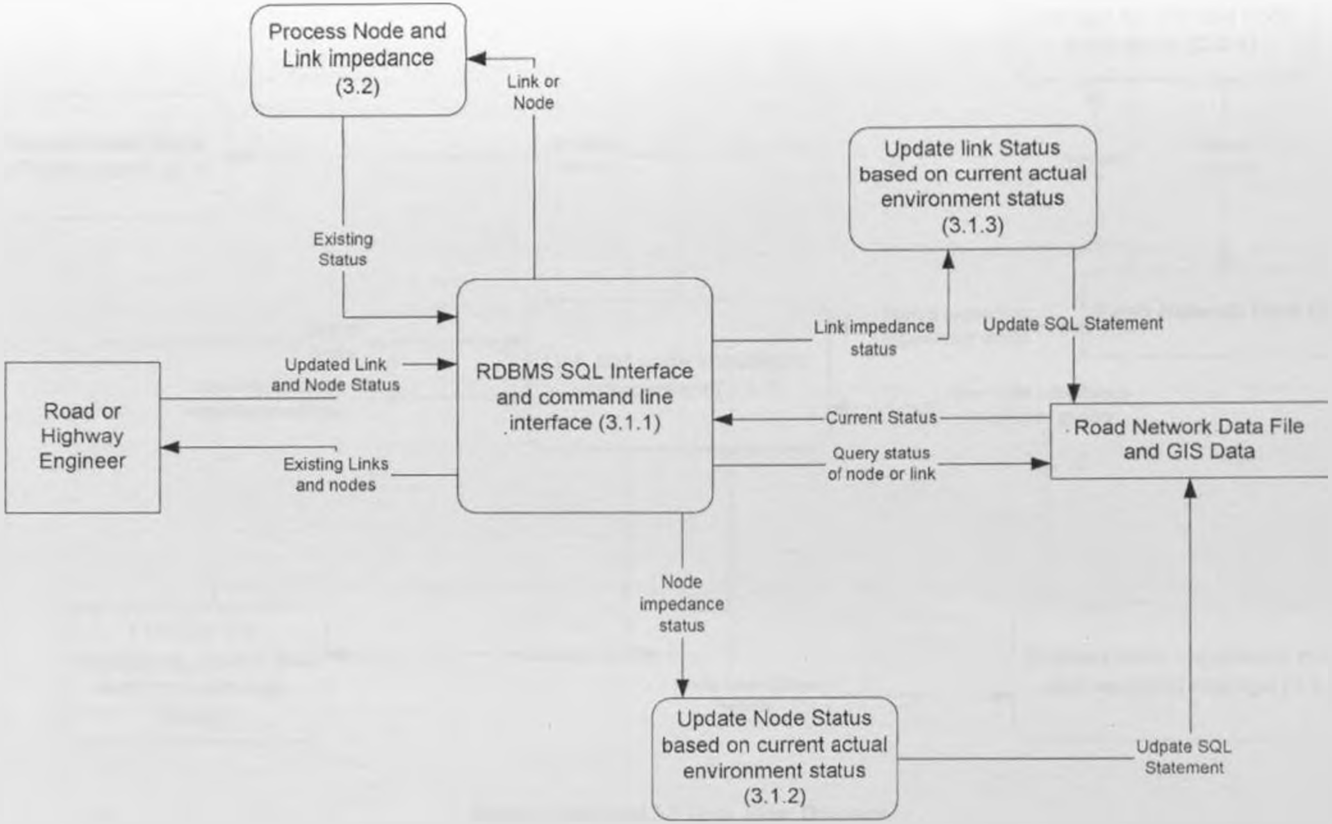


Figure 3-35 Level 3.1 Data Flow Diagram

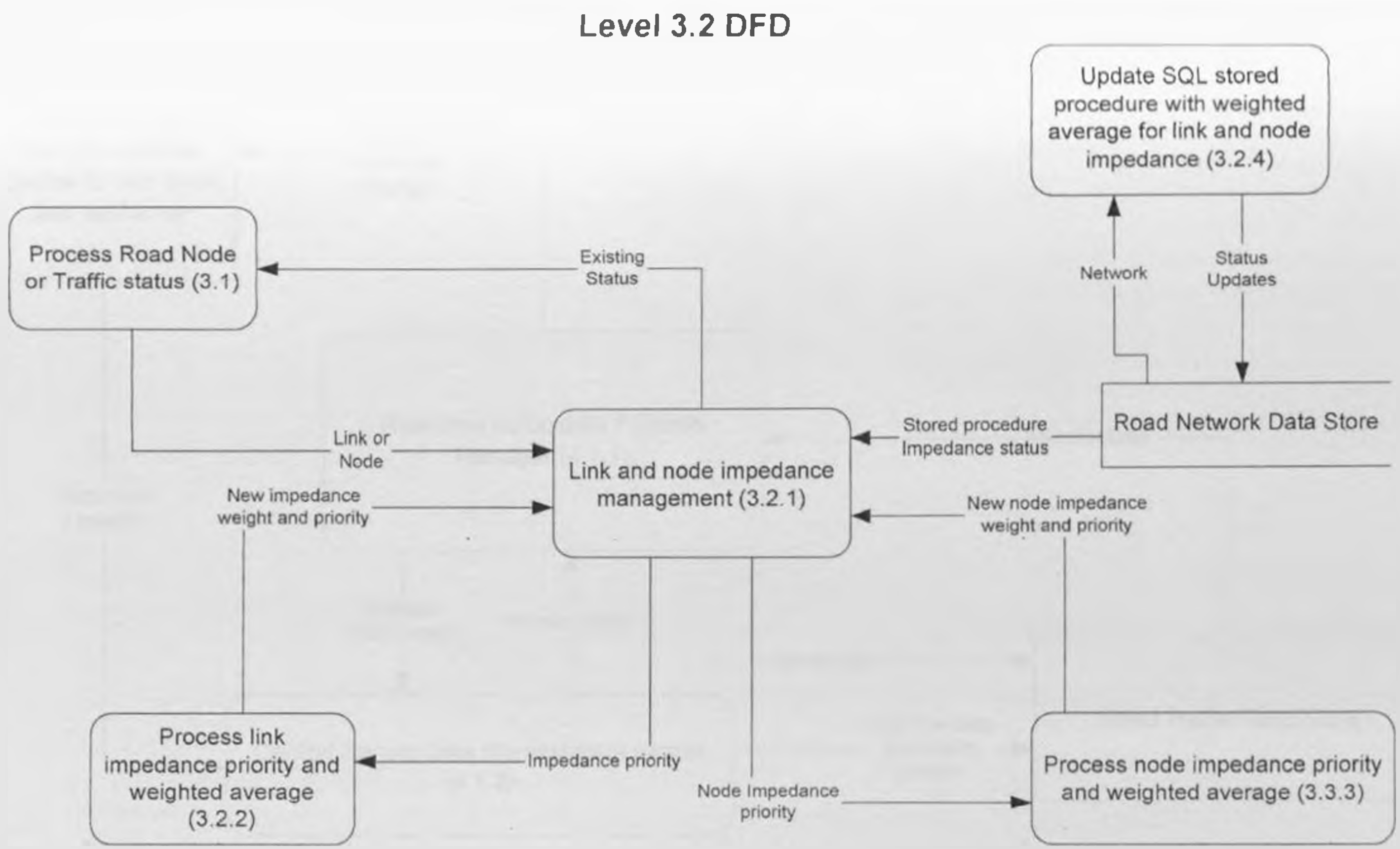


Figure 3-36 Level 3.2 Data Flow Diagram

Level 4.1 DFD

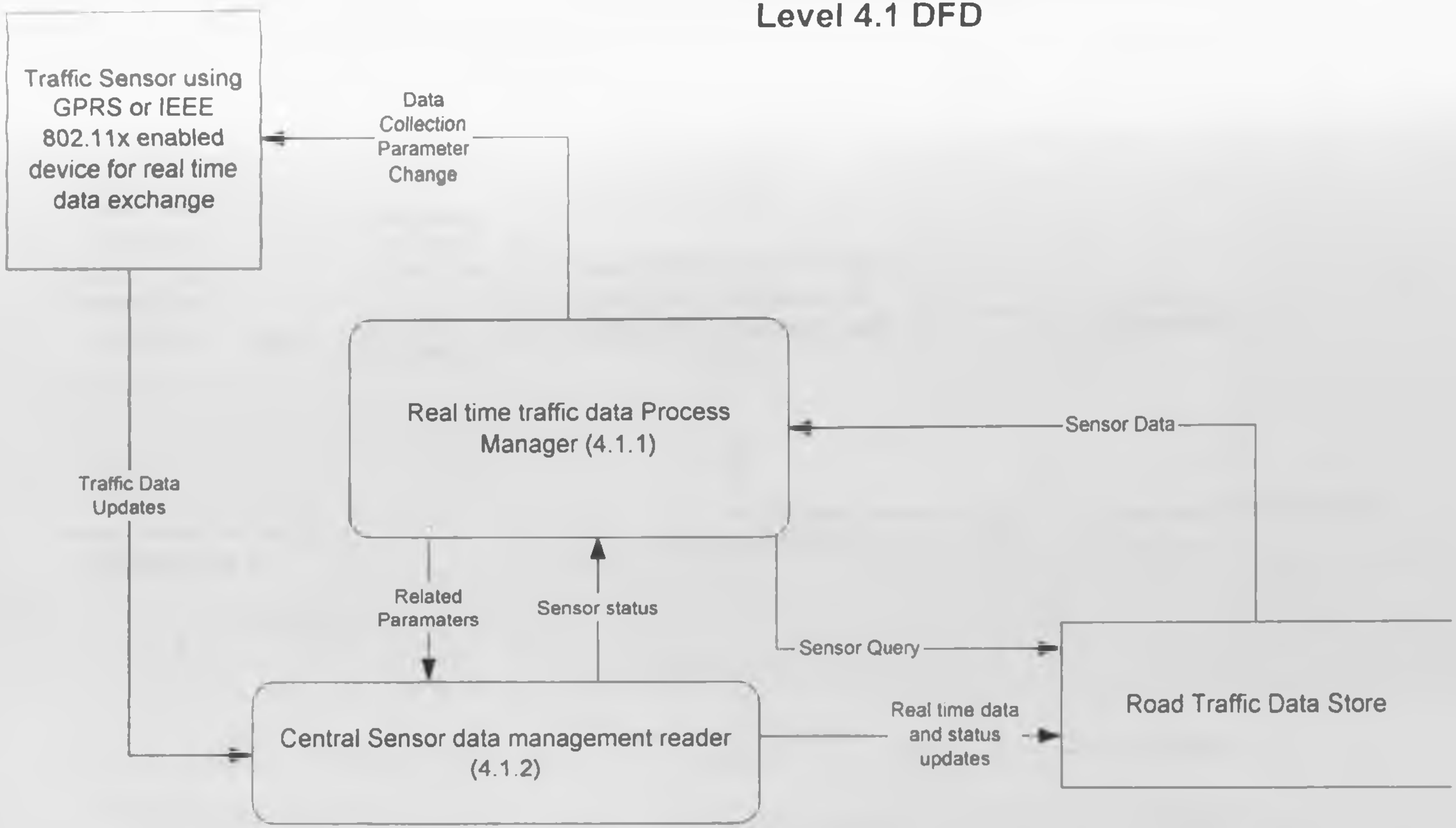


Figure 3-37 Level 4.1 Data Flow Diagram

Level 4.2 DFD

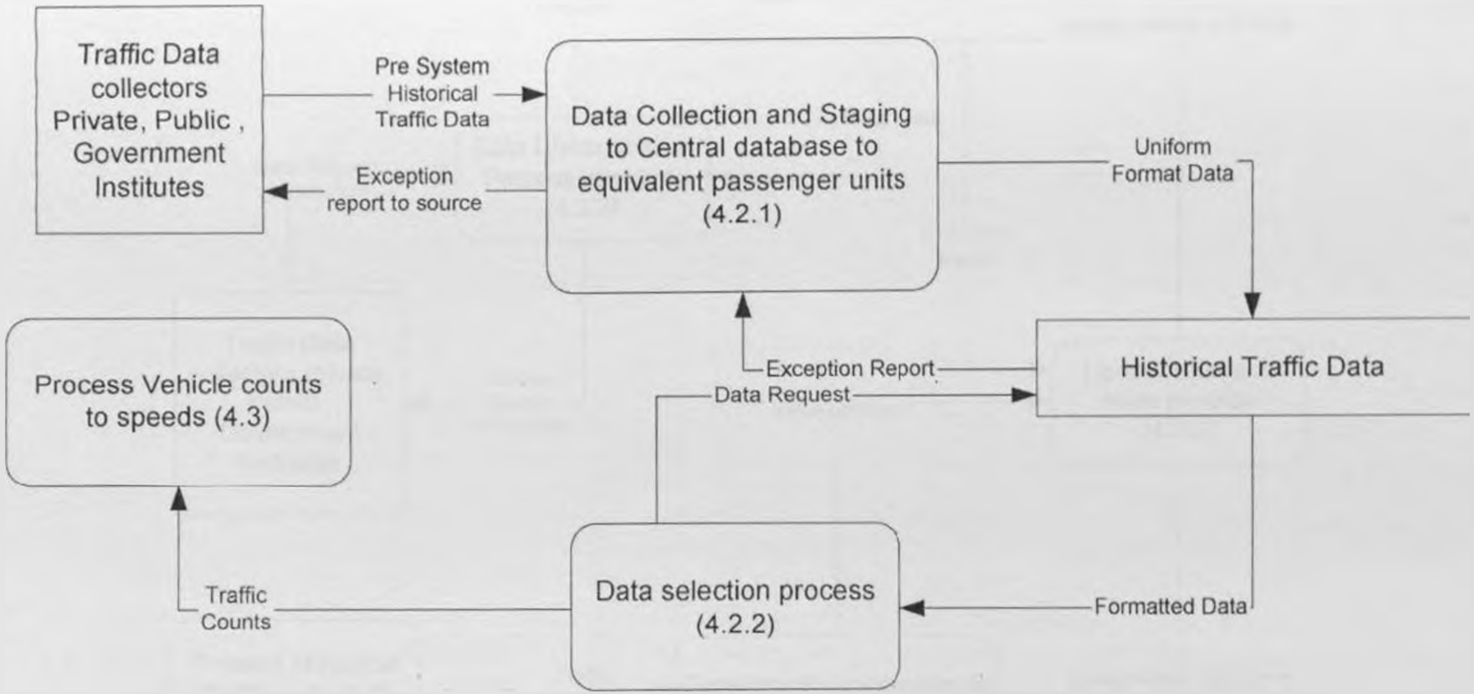


Figure 3-38 Level 4.2 Data Flow Diagram

Level 4.3 DFD

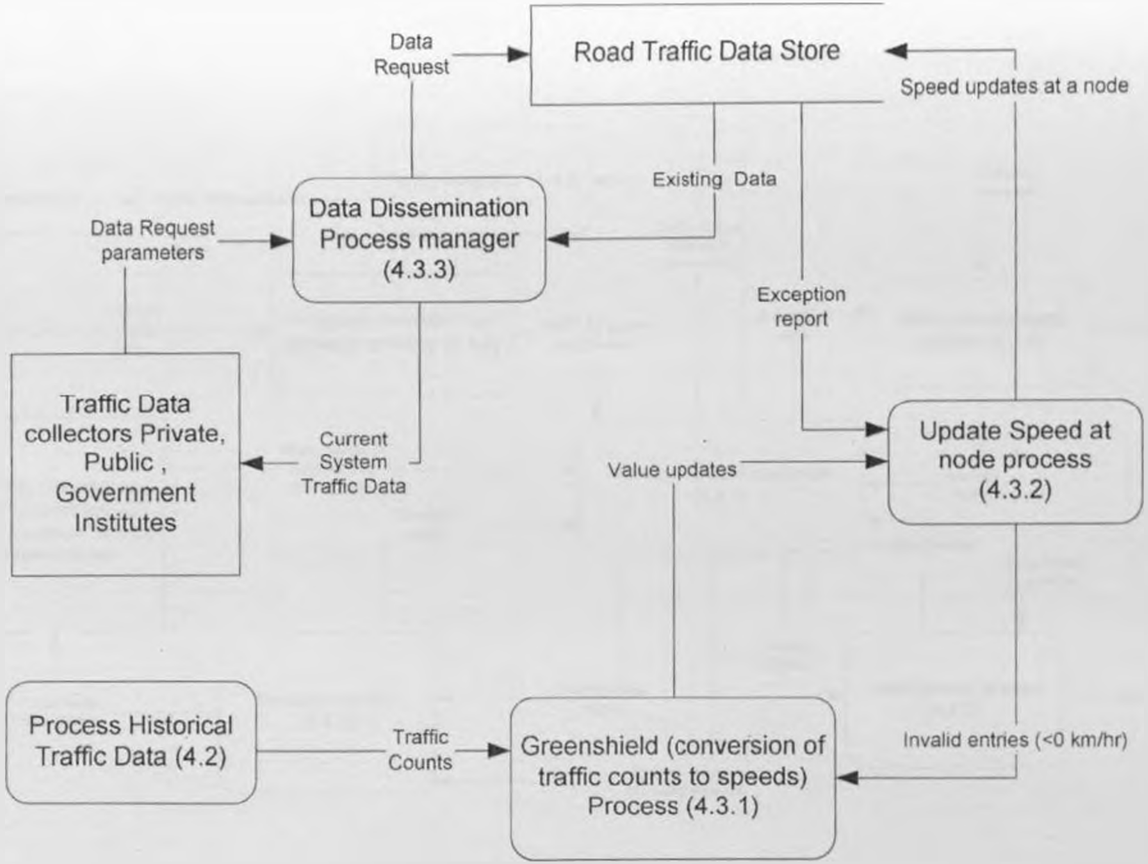


Figure 3-39 Level 4.3 Data Flow Diagram

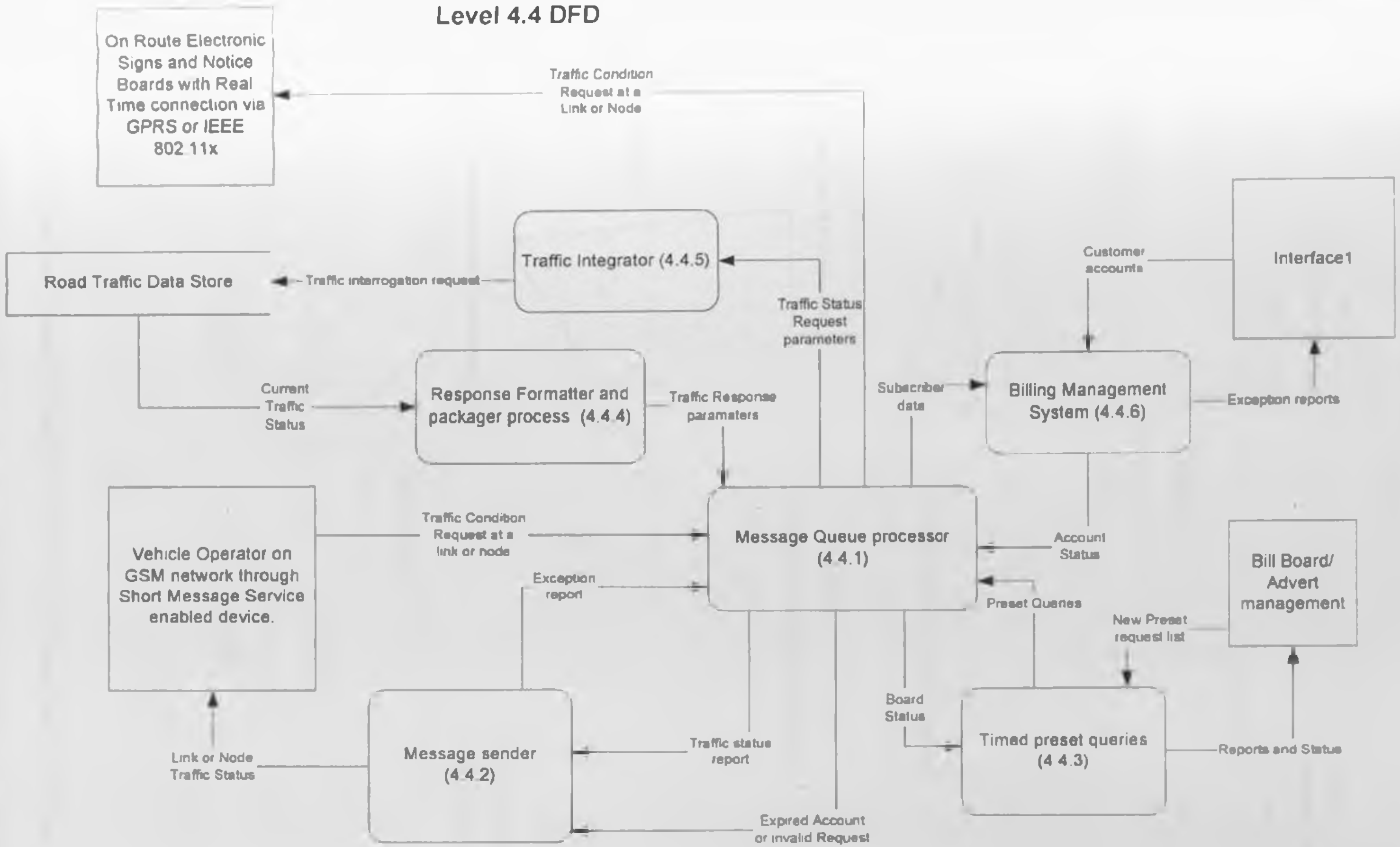


Figure 3-40 Level 4.4 Data Flow Diagram

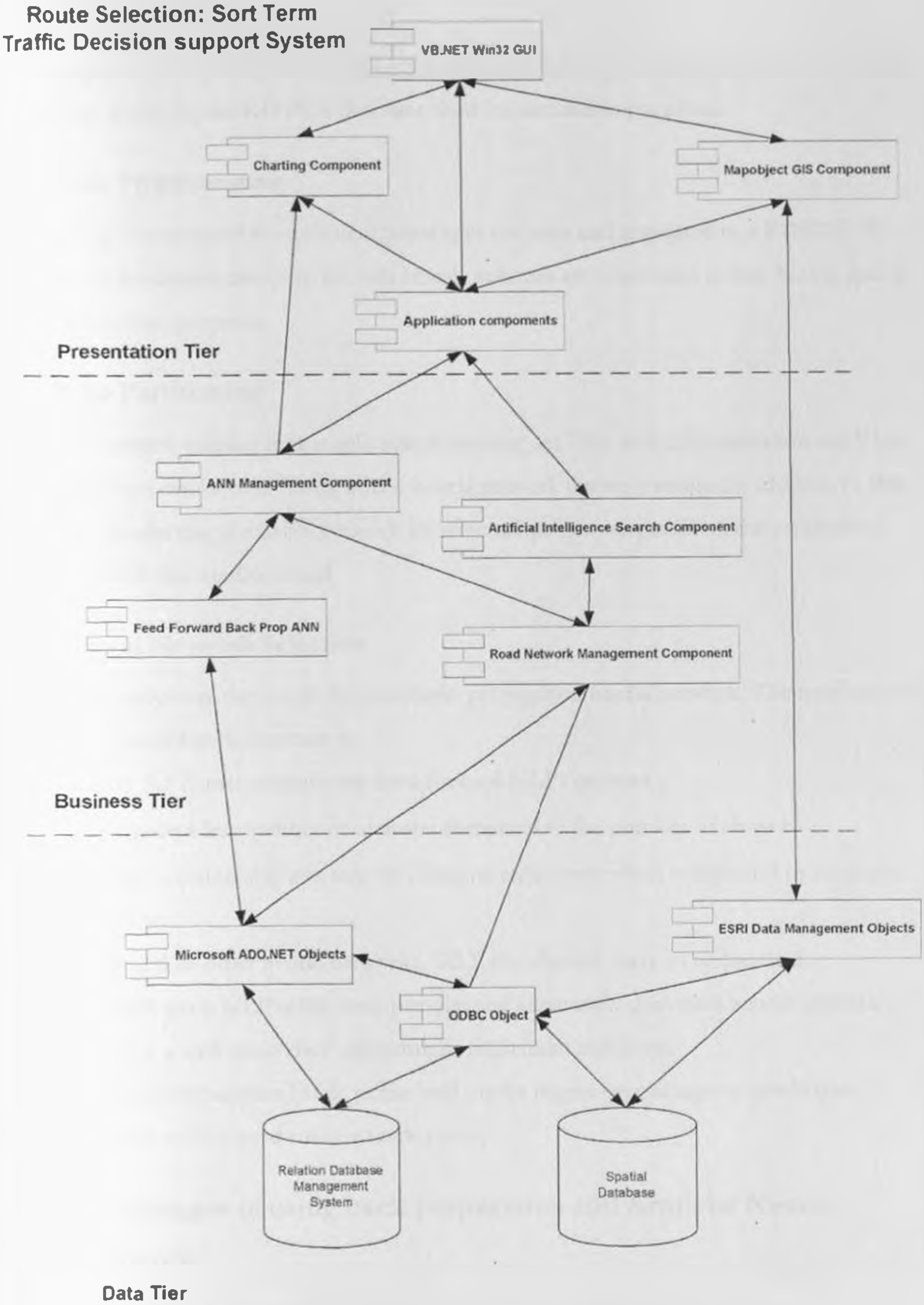


Figure 3-41 System interaction Diagram of the various systems and technologies

Figure 3-41 Illustrates the system interaction diagram as seen in terms of a 3 tier design approach framework.

3.10.1 Data Selection

All data availed by KIPPRA (63 data collection points) were used in the system. Since real time data was not available, the KIPPRA data was used for simulation purposes.

3.10.2 Data Preprocessing

KIPPRA data is converted to equivalent passenger car units and imported to a RDBMS. By employing Greenshields theorem, the half hourly volumes are converted to half hourly speeds used for prediction purposes.

3.10.3 Data Partitioning

The neural network training data is split into a training set 75% and 25% validation set. This split is critical to control over fitting of the neural network during training. In addition to this, the process of selecting a suitable network architecture primary depends on the measure of validation RMSE for models tested.

3.10.4 Neural Network Selection

This research proposes the use of the multilayer propagation neural network. The motivation for using this neural network structure is:

1. Looney [L97] recommends the feed forward MLPs because :
   - Require a less number of neurons compared to the number of classes
   - Take considerably less time to compute recursively when compound to recurrent ANN
2. Compared to other neural networks, MLP are relatively easy to understand.
3. The Back prop MLP is the most popular and extensively discussed neural network , making it a well researched algorithm to implement and learn,
4. The back propagation ML:P works well on the largest percentages of prediction problems with a good success track record.

3.10.4.1 Advantages of using back propagation and Artificial Neural Network

- Learning is independent of the order in which training data is presented.
- The architecture can be manipulated for better results
- Is able to run on parallel processors

3.10.4.2 Disadvantages of using back propagation and Artificial Neural Network

- Training may converge to a local minimum that is shallow so that learning is not robust.
- The learning rate cannot be predicted in advance. This results to iterative learning which can lead to oscillation if not small.

Figure 3-42 depicts the process of building a neural network.

The neural network is implemented as a class in the overall system whose input parameters are a dataset with input and output exemplars, neural network parameters and stopping criteria. Since the networks runs through a loop controlled by the number of epochs, a different thread is deployed to ensure the GUI remains interactive throughout the training process.

Training is an iterative process that involves heuristics in selecting suitable parameters to model with. The final trained neural network is experimentally build from a series of runs. The network with the best overall performance (least RMSE on validation set) is selected for purposes of predicting the speed at one step ahead.

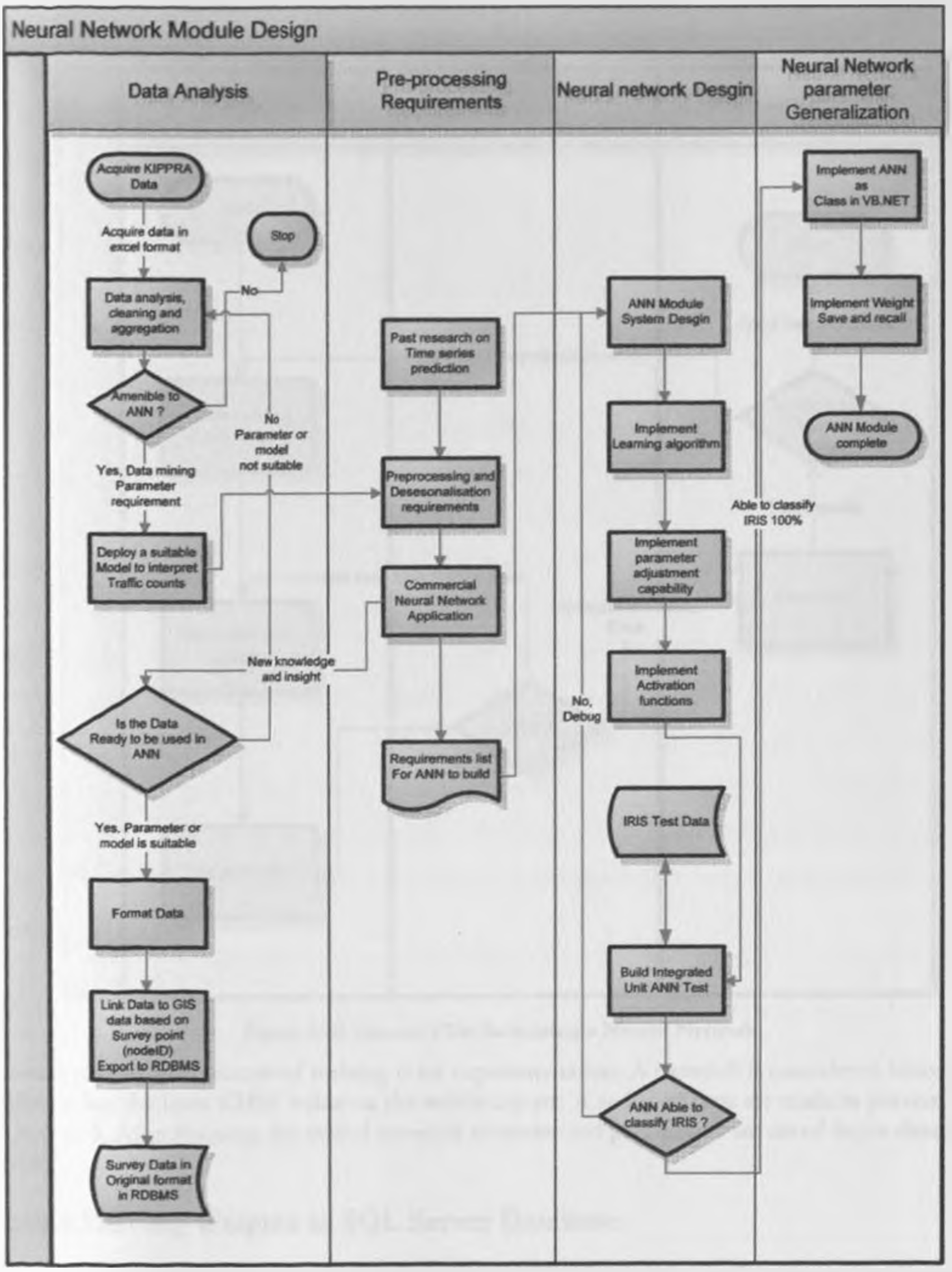


Figure 3-42 Neural Network Design

3.10.5 Neural Network Training

The purpose of training an MLP is to obtain an approximation to a function that maps input exemplar feature vector inputs into associated output target identifier vectors in a generalized (smoothed) fashion Looney [L97]. Figure 3-42 identifies pertinent steps in training a neural network.

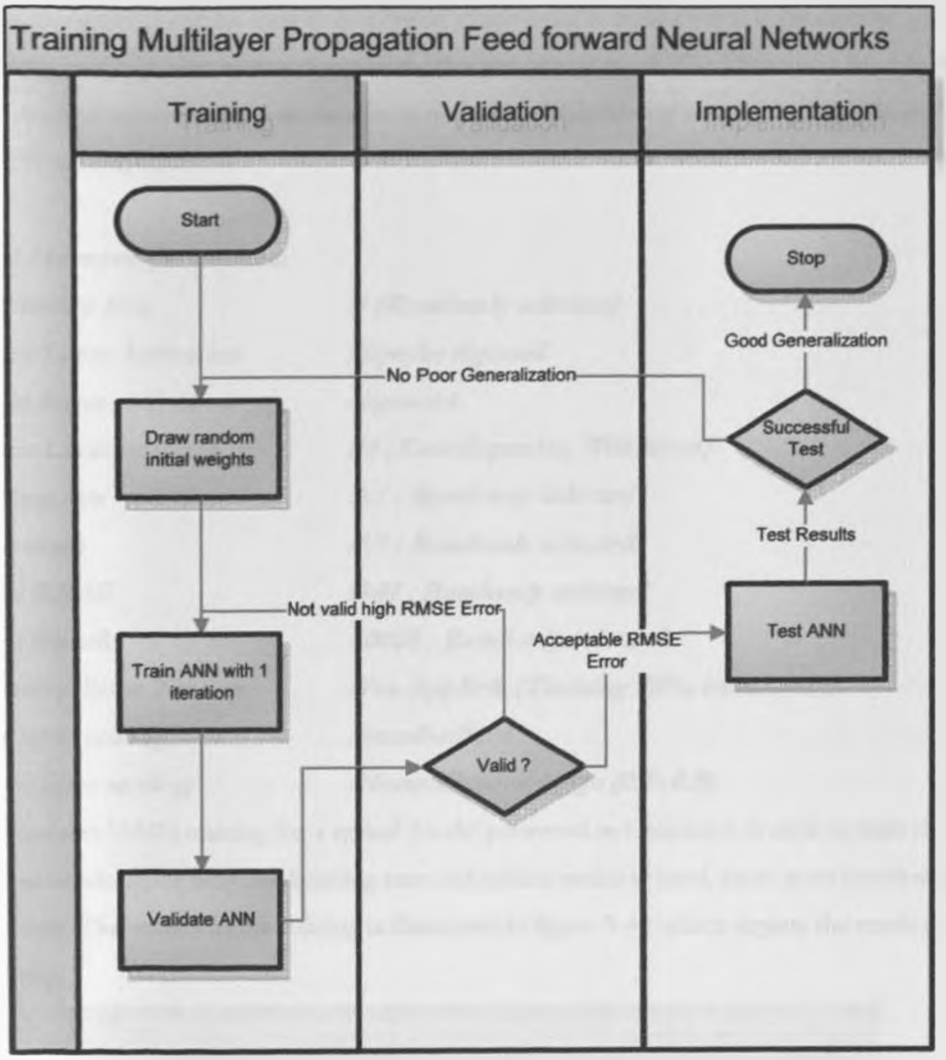


Figure 3-43 Process Flow in training a Neural Network.

In this project, the process of training is by experimentation. A network is considered trained when it has the least RMSE value on the validation set. A series of runs are made as presented in Chapter 5. After training, the neural network structure and parameters are saved into a database table.

3.10.5.1 Saving Weights to SQL Server Database

For purpose of making the system resistant to loss of training data, the last training run is saved on the database. However for purposes of simulation, KIPPRA data selected does not need to be subjected to weight saving as the system is not getting a continuous stream of traffic data. The process of saving the neural network weights for future prediction involves saving the following:

1. Neural network Architecture,
2. Neural Network weights,
3. Neural network Scaling and normalizing parameters.

Further, the weights can be used to determine which input is most critical in prediction (sensitivity analysis), due to time constraints, this was not perused. The illustration described next saves a trained network based on data from node 469 (Waiyaki way next to the Mall Shopping Center Westland)

Neural Network parameters:

Lag Window Size	***:9 (Randomly selected)***
Hidden Layer Activation	***:Bipolar sigmoid***
Output Layer activation	***:Sigmoid***
Hidden Layer neurons	***:18 (Kanellopoulos, Wilkinsen)***
Learning rate	***:0.1 : Randomly selected***
Momentum	***:0.9 : Randomly selected***
Target RMSE	***:0.01 : Randomly selected***
Target Epoch	***:10000 : Randomly selected***
Validation Error Testing	***:Yes Applied, (Training 75%, validation 25%)***
Input layer scaling	***:Standardized***
Output layer scaling	***:Normalization range (0.1- 0.9)***

The processes 'ANN training for a typical Node' presented in Chapter 2, is used to train the neural network. Since only one learning rate and hidden nodes is used, there is no iteration of parameters. The results of the training is illustrated in figure 3-44 which depicts the result of overfitting.

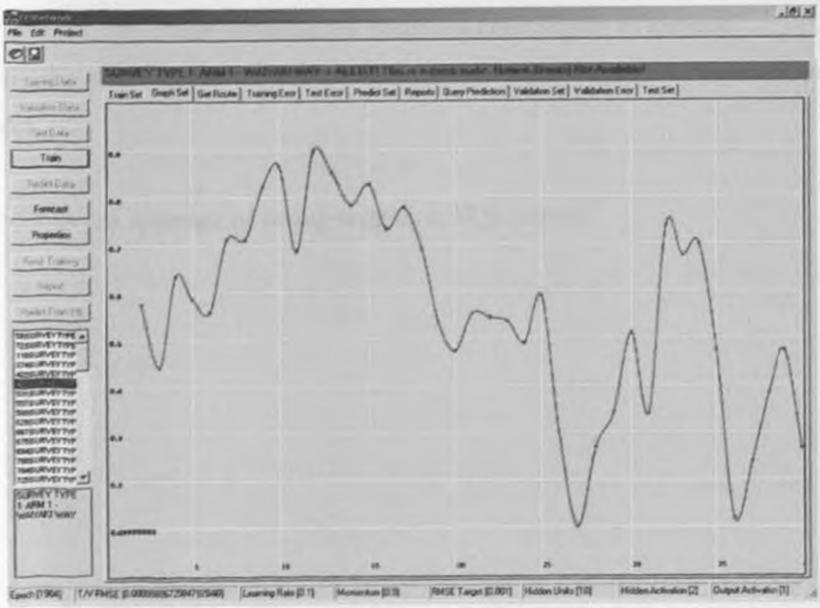


Figure 3-44 Training result for node Waiyaki way next to the Mall Shopping Center Westlands with overfitting allowed.

Figure 3-45 shows the results of training with respect to the RMSE error give the parameters used above.

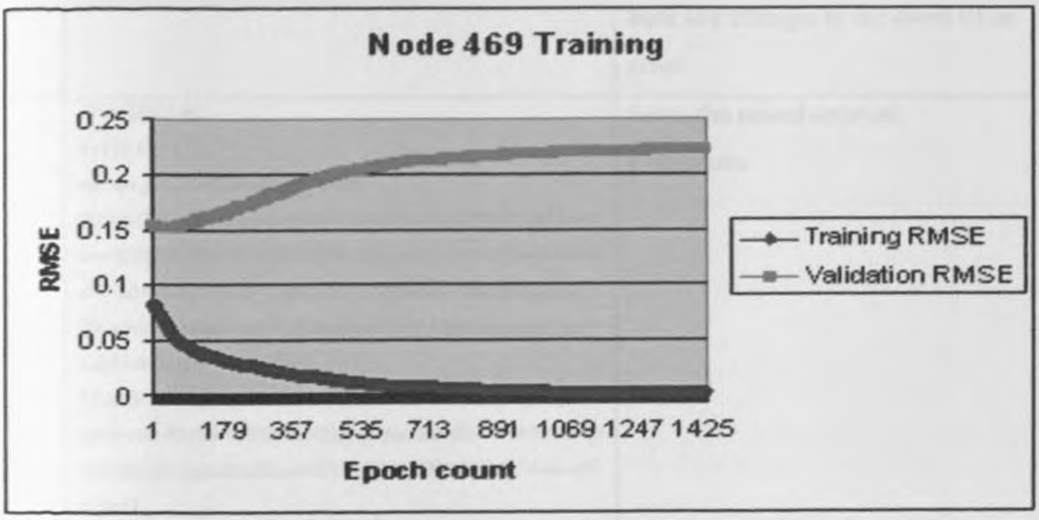


Figure 3-45 Training and validation RMSE

Overfitting results to poor generalization of the network on an out of training set. The figure 3-45 depicts the results of overfitting on a network. As training proceeds, the validation error increases and the training error decreases. In this example, the training should stop as less than 170 epochs if the other parameters are to be maintained. To control overfitting, a number of strategies can be deployed:

1) Introduce noise to the data,
2) Early stopping via cross validation or epoch count,
3) Early stopping the validation error begins to increase.

After training, the neural network parameters, architecture and weights are saved in the database for recall later.

The table 3-6 lists the sequence of saving weights to SQL Server.

Step	SQL Server Stored Procedure	Remark/Comment
1	*SET TRANSACTION ISOLATION LEVEL READ COMMITTED;BEGIN TRANSACTION*	Sets up a transaction space to roll back any changes in the event of an error
2	*declare @p16 int* *set @p16=14* *exec dbo.[proc_AddKnowledgeToNode]* *@NodeID=469,@LearningRate=0.1000000000000001,@Momentum=0.9000000000000002,@NumberInputUnits=9,@NumberHiddenUnits=18,@NumberOutputunits=1,@HiddenActivationType=2,* *@OutputActivationType=1,@InputScaleType=1,@OutputScaleType=0,@DateCreated="2006-06-18 17:00:02:697",@InputScaleMin=0.1000000000000001,@InputScaleMax=0.9000000000000002,@OutputScaleMin=0.1000000000000001,@OutputScaleMax=0.9000000000000002,@NetworkID=@p16* *output* *select @p16*	Saves the neural network parameters
3	*exec dbo.[proc_AddWeightsToKnowledge]* *@NetworkID=14,@FromLayer=0,@FromNode=0,@ToLayer=1,* *@ToNode=0,@WeightValue=-1.8391830107048914,@DateCreated="2006-06-18 17:00:02:737"*	This procedure saves the weights from the input to the hidden layer. In this case, it is called 162 times $(9 \times 18 = 162)$ one for each input node to hidden node.
4	*exec dbo.[proc_AddHBiasWeightsToKnowledge]* *@NetworkID=14,@ToNode=0,@WeightValue=-1.1126192489580798e-005,@DateCreated="2006-06-18 17:00:02:757"*	This procedure saves the bias weights from to the hidden layer units. It runs 18 times in this case.
5	*exec dbo.[proc_AddWeightsToKnowledge]* *@NetworkID=14,@FromLayer=1,@FromNode=17,@ToLayer=2,* *@ToNode=0,@WeightValue=1.2566606094877033,@DateCreated="2006-06-18 17:00:03:147"*	This procedure saves the weights from the hidden to the output layer. In this case, it is repeated 18 times $1 \times 18 = 18$.
6	*exec dbo.[proc_AddHOBiasWeightsToKnowledge]* *@NetworkID=14,@WeightValue=0.22293357297033528,@DateCreated="2006-06-18 17:00:03:147"*	This procedure saves the bias weights to the output layer unit. It runs once.
7	*exec dbo.[proc_AddInputScale]* *@NetWorkID=14,@InputColumn=0,@InputScaleMax=0,@InputScaleMin=0,@InputRange=0,@InputMean=17.296369230769237,* *@InputStdDev=7.0839644109686519,@DateCreated="2006-06-18* *17:00:03:147"*	This procedure saves the input scale parameters for each of the 9 input attributes for this case.
8	*exec dbo.[proc_AddOutputScale]*	This procedure saves the output

Step	SQL Server Stored Procedure	Remark/Comment
	@NetWorkID=14,@OutputColumn=0,@OutputScaleMax=30.638 40000000004,@OutputScaleMin=3.37440000000000085,@Output Range=27.2639999999999996,@OutputMean=0,@OutputStdDev=0 ,@DateCreated="2006-06-18 17:00:03:167"	scale parameters.
9	*COMMIT TRANSACTION*	Commits the transactions to the database saving all parameters and weights.

Table 3-6 Sequence in saving weights to SQL Server

The series of steps 1-9 presented are processed sequentially when saving a neural network. Together with training, this forms a process that is repeated for all nodes. The sequence established in table 3-6 can be depicted in terms of process as shown in figure 3-46. This process is repeated while changing key parameters identified. The winning structure has the least difference between the training and validation RMSE.

3.10.5.2 Recalling a neural network for prediction purposes from SQL Server Database

This is the reverse of the process described in section 3.6.5.2. Table 3-7 identifies significant steps in recalling a saved network structure in this case node 469:

1) Select the node to predict the traffic,
2) Complete the prediction table by entering the past traffic data for the window lag saved.
3) Click the load MLP button to recall the brain the update the predictions.

Table 3-7 identifies critical steps carried out at database level in the prediction process. Figure 3-47 illustrates the training set based on a 9 lag window.

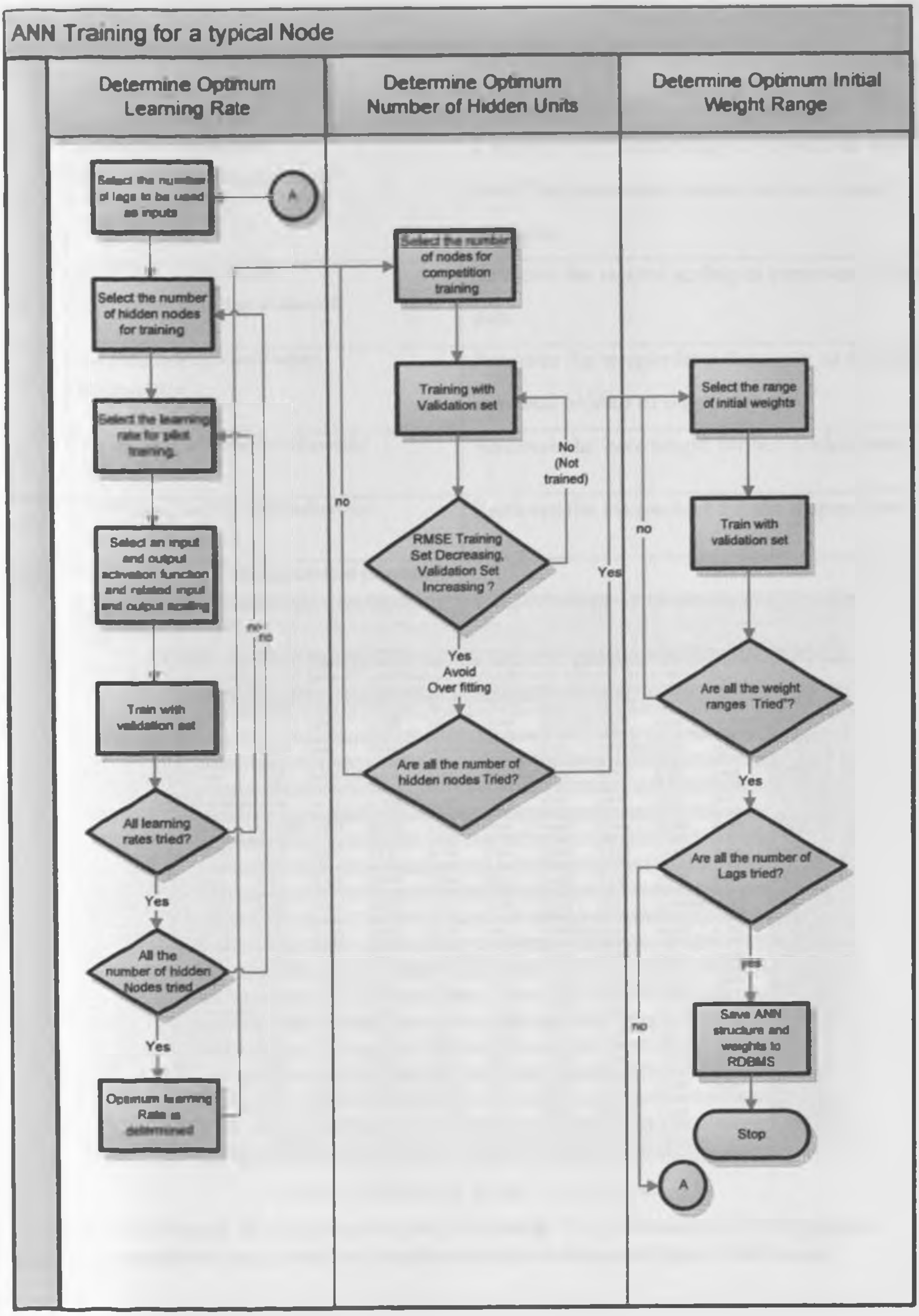


Figure 3-46 Training Process at a node

Step	SQL Server Stored Procedure	Remark/Comment
1	*exec dbo.[proc_GetKnowledgeFromNode] @NodeID=469*	Retrieves the latest neural network architecture to reconstruct the network for a specified node.
2	*exec dbo.[proc_GetInputScale] @NetworkID=14,@InputColumn=0*	Retrieves the input scaling to preprocess input data. This procedure is called for each input attribute.
3	*exec dbo.[proc_GetOutputScale] @NetworkID=14,@OutputColumn=0*	Retrieves the output scaling to preprocess output data.
4	*exec [proc_GetWeightFromKnowledge] @NetworkID=14*	Retrieves the weight from the input to the hidden layer and hidden to output layer.
5	*exec dbo.[proc_GetIHWeightFromKnowledge] @NetworkID=14*	Retrieves the bias weight for the hidden layer
6	*exec dbo.[proc_GetHOWeightFromKnowledge] @NetworkID=14*	Retrieves the bias weight for the output layer

Table 3-7 Series of steps in the prediction process

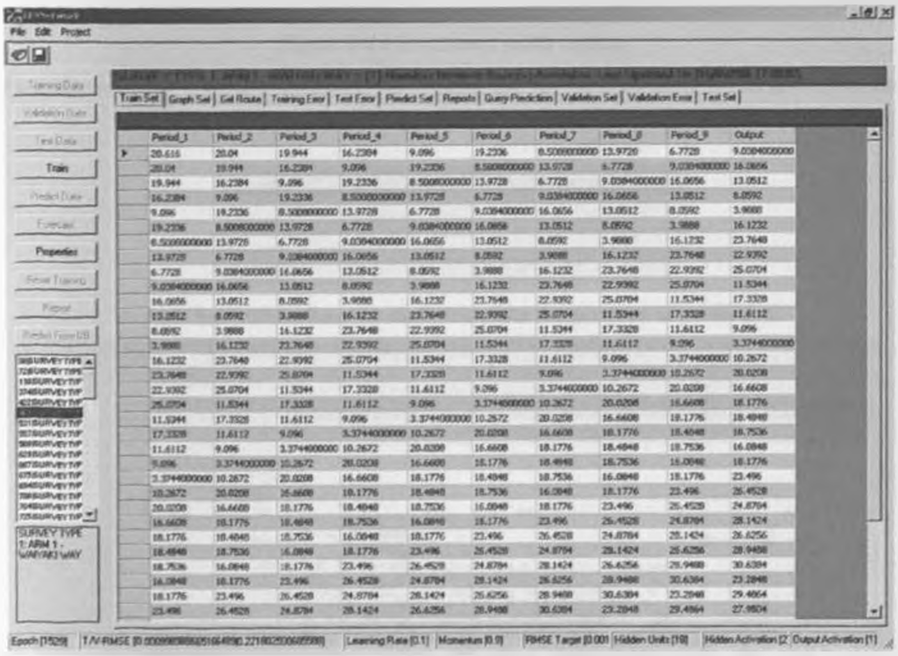


Figure 3-47 Node 469 Ready for prediction.

Node 469 is on Waiyaki Way next to the Mall , Westlands. The attributes to act as inputs for purposes of prediction are entered in a prediction table as shown in figure 3-48 shown

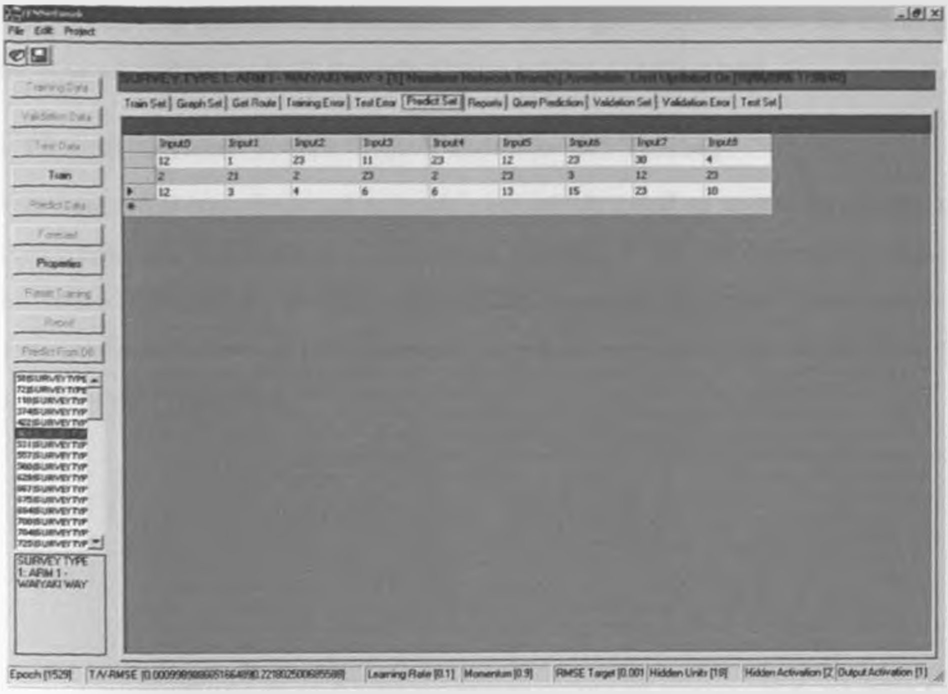


Figure 3-48 Prediction table entries

There after, the icon of a folder is clicked to predict the single one step ahead for all rows of data shown.

Figure 3-49 Folder Icon to click complete the process of prediction.

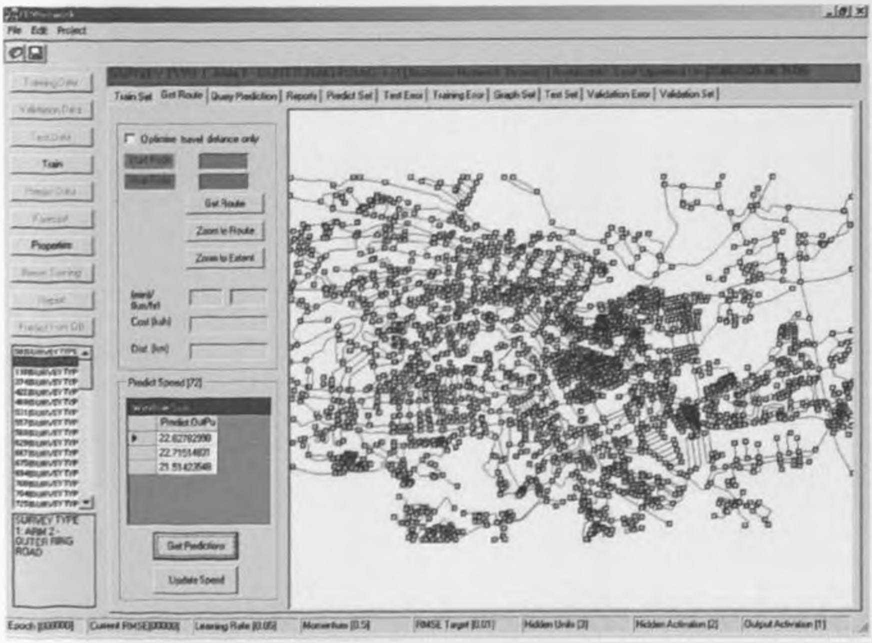


Figure 3-50 Prediction data ready to update the road network node

Finally, a prediction is selected to update the node value. Figure 3-50 illustrates the resulting window once a prediction is ready for saving hence process a shortest path.

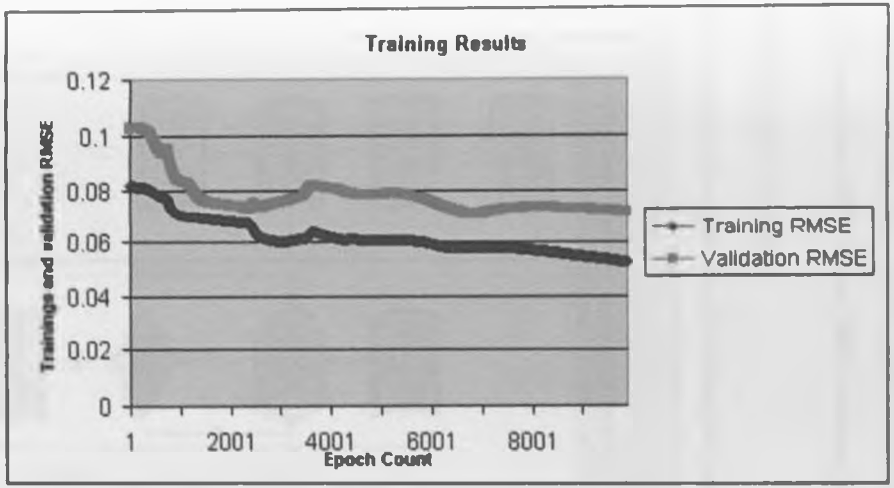
3.10.6 Neural Network Source Code Validation and Testing

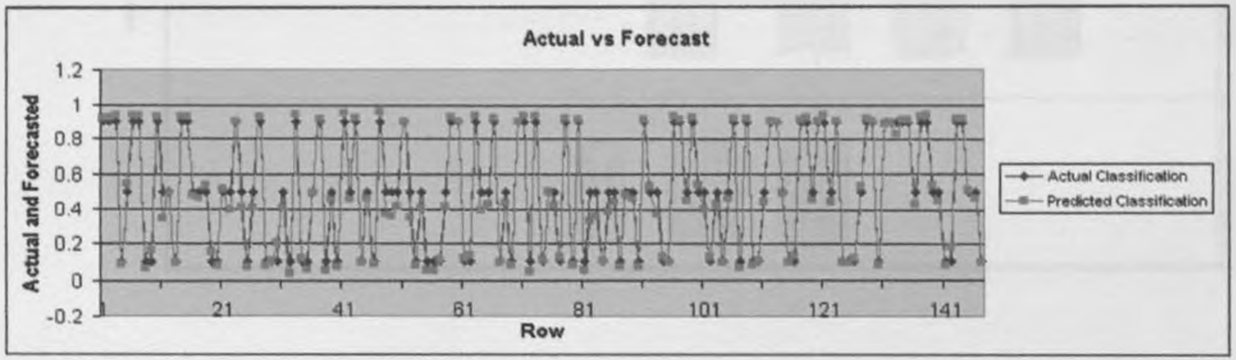
Before making actual runs, the neural network is extensively tested on sample data. In this case, IRIS data is used as a benchmark to verify that the algorithm works and the source code is correct. In the IRIS dataset, 3 different types of IRIS flowers are described by 4 attributes. Measurements in millimeters of 150 flowers, 50 of each species: sepal-length, sepal-width, petal-length, petal-width, class name.

3.10.7 Neural Network Training Result from IRIS data set.

Summary

	Parameters
# of training rows:	110
Number Inputs	4
Number Hidden layers	1
Number Hidden Neurodes	25
Number output Neurodes	1
Hidden Activation Function	Sigmoid
Output Activation Function	Bipolar Sigmoid
# of validation rows	37
Learning Rate	0.2
Momentum	0.9
Training Set RMSE	0.0521
Validation Set RMSE	0.0717





The results affirm that the neural network source code and algorithm is correct. It is able to correctly classify the IRIS dataset 100% (training set) and 70% validation set.

3.10.8 A* Search Design

Figure 3-51 depicts the process of building the road network earlier described.

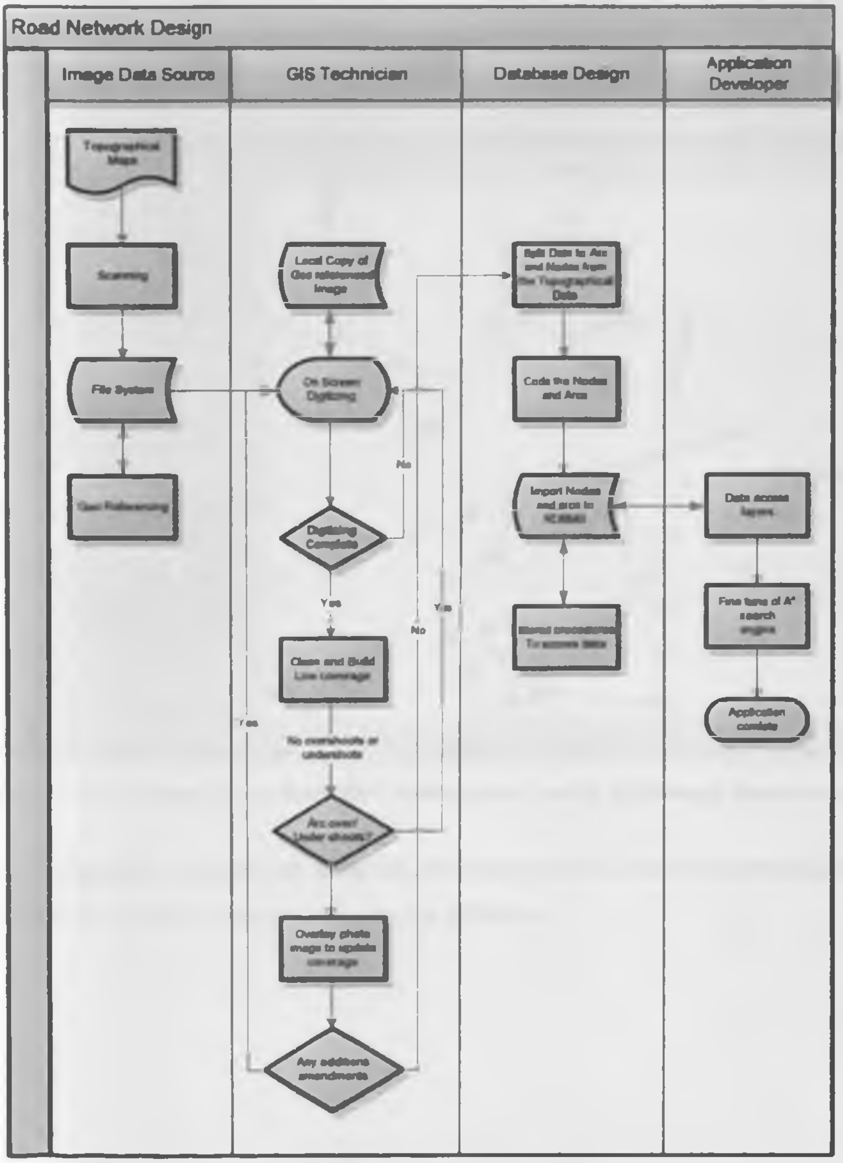


Figure 3-51 network Design Process

The process of developing the road network is not automated in the prototype. It combines some manual process and uses specialized software: ArcInfo/ArcView.

3.10.9 A* Search Testing

The A* search is tested on a graph network using dry runs to ensure that the heuristic function is indeed working. The figure 3-52 illustrates the result of a run from node 67 (KPC Staff quarters Thika road) to node 197 (Kariobangi, Kamunde Road).

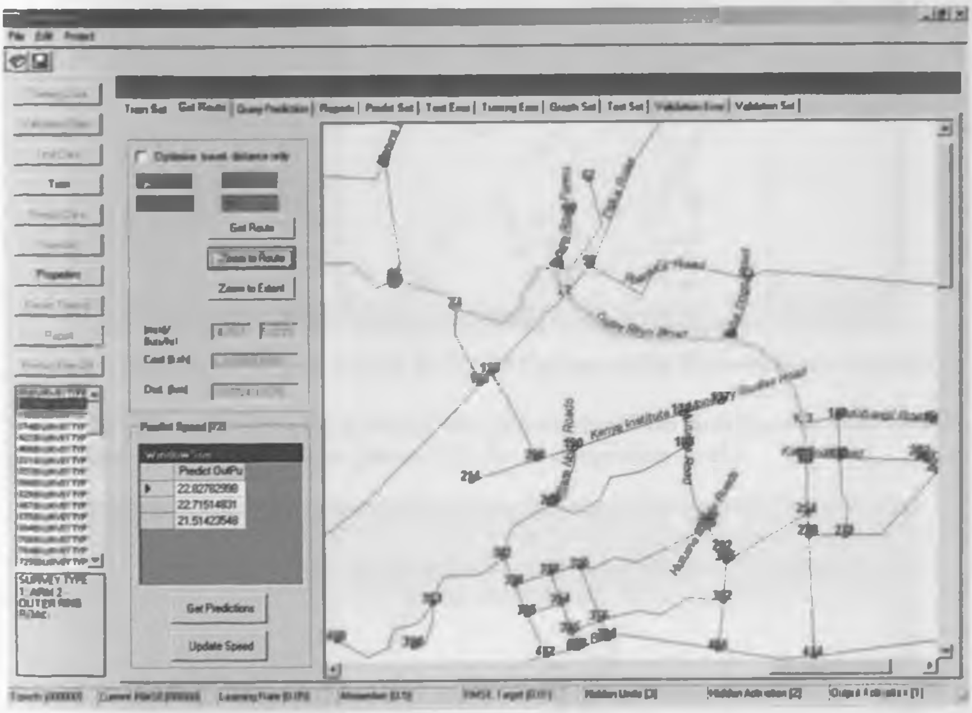


Figure 3-52 Run results from node 67 (KPC Staff Quarters) to 197 (Kariobangi, Kamunde road)

Figure 3-53 illustrates a second run from the same start point to a new destination (node 404 Huruma Road, Huruma). The path suggested is different.

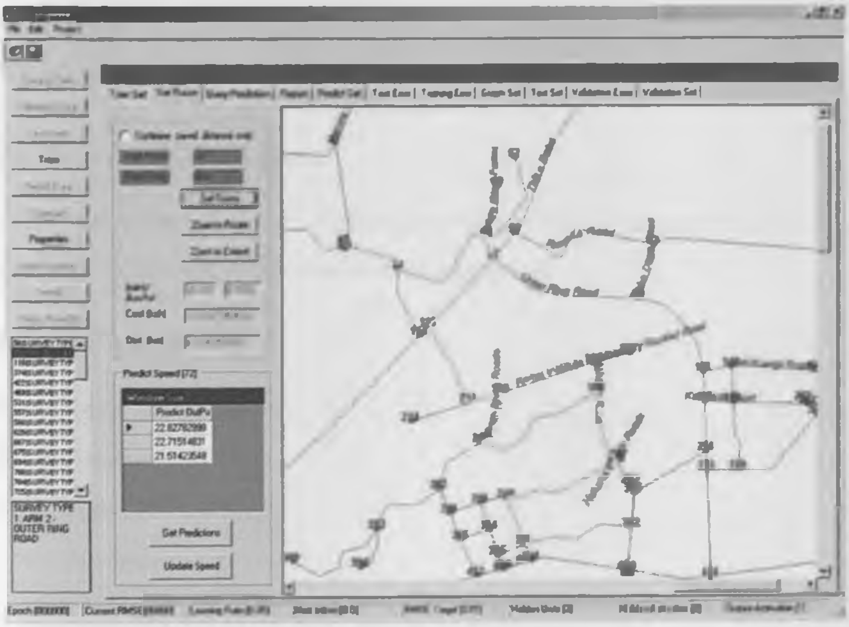


Figure 3-53 Run results from node 67 (KPA Staff Quarters) to 404 (Huruma Road, Huruma)

Figure 3-54 below illustrates the result of the same run based on path distance alone which gives a static result irrespective of time of the day or congestion level.

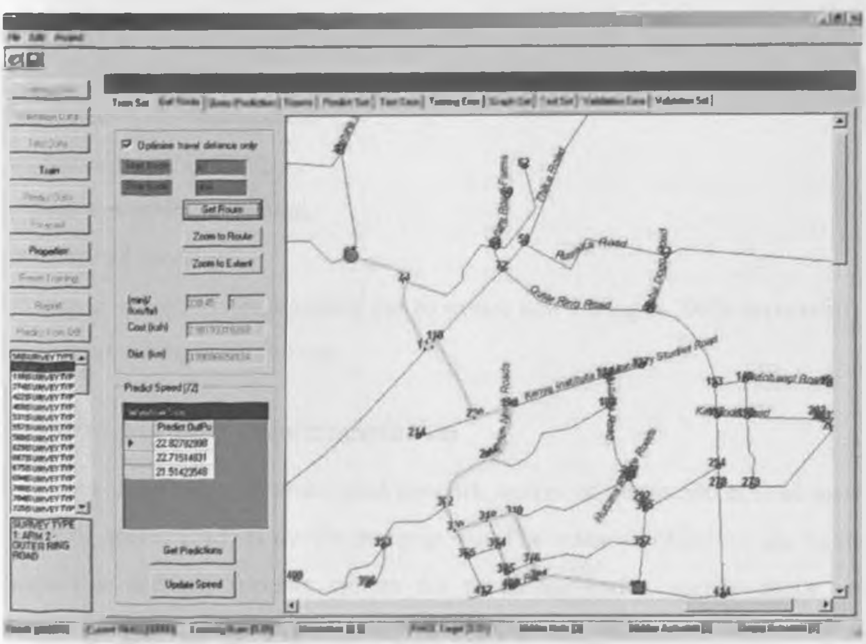


Figure 3-54 Shortest static path optimizing travel distance not time.

4 System Implementation

4.1 *Hardware Platform used*

- A Toshiba Laptop Celeron 1.7 GHz processor with 512 MB ram and 30 GB hard disk, with windows XP professional service pack II.

4.2 *Software used*

- Map Object 2.0,
- Visual Studio .NET 2003,
- SQL Server 2005 SPK1,
- .NET framework 1.2

4.3 *Implementation Process*

Using Visual Studio .NET 2003, the prototype is compiled and tested. The implementation process is as outlined:

1. Implement artificial neural network,
2. Implement A* search,
3. Implement GIS visualization,
4. Implement ANN saving,
5. Implement ANN prediction,
6. Integrate all modules.

After each step, a series of tests is carried out to ensure unit testing is 100% successful before integration testing is carried out.

4.4 *Decision Support Implementation*

The system covers the entire Nairobi road network system of about 500 Km of main road and 1500 km of access road, as shown in figure 4-1. The research objective was to develop and document a decision support system for predicting traffic speeds hence conduct minimum cost routing as described in section 4.5.2 late in this chapter.

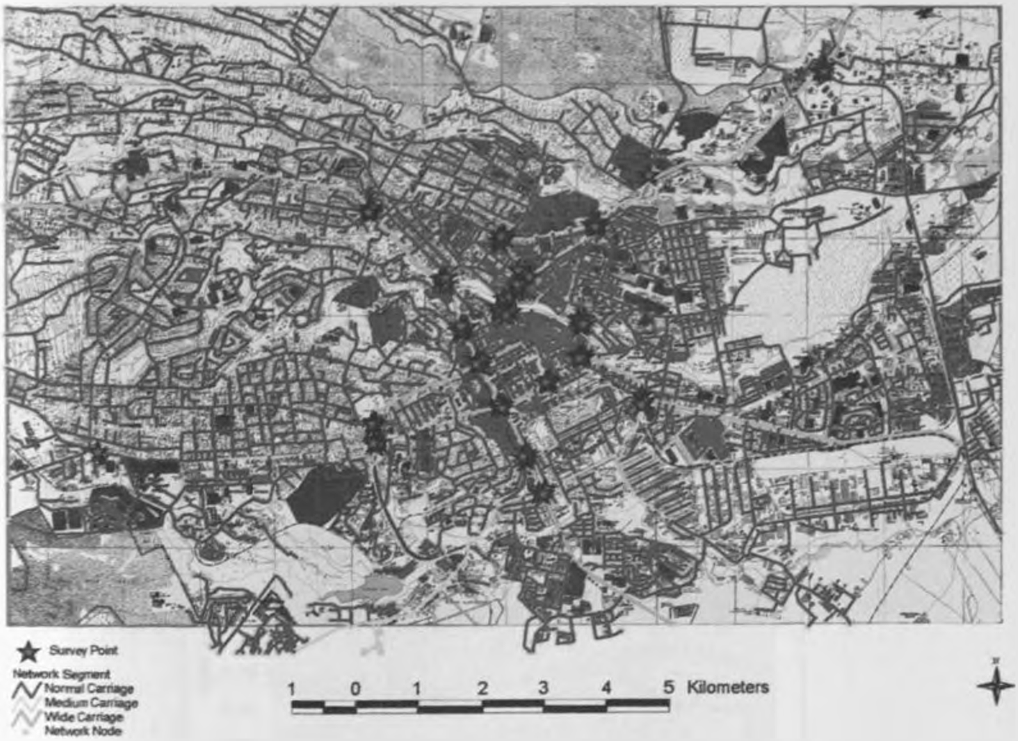


Figure 4-1 Network Coverage of the System: minor and major road network of Nairobi from 1:2000 source images.

The developed system has five features inbuilt differentiating it from exiting system found in other parts of the world Wang et al. [WTSW03]:

- Is based on the predicted information rather than the real-time (current) traffic information. Park et al. [PSHJ05] demonstrate that for certain future time periods travel time forecasting using only the historical profile without real-time profile is better that one using real-time profile or both.
- Each node on the road network is associated with a neural network hence greater flexibility and autonomy. This results to a more fine grained scale of analysis hence higher accuracy of prediction with results being less generalized spatially Innamaa [I01].
- Complete independence from proprietary complex GIS spatial road-network database storage, hence inexpensive to deploy and replicate Park et al. [PYR05].

- Designed to work with historical traffic count data converted to speed using Greenshield's theorem, with inbuilt flexibility to manage real-time data feed.
- Depends on traffic counts collected manually, with a strong linkage to road and highway design volume and specification making it compatible to many third world countries. In this context, traffic count from KIPPRA.

4.5 Testing

A number of sample runs to have a feel of the dataset is run to identify critical parameters to be tuned while developing an optimum network structure for nodes identified.

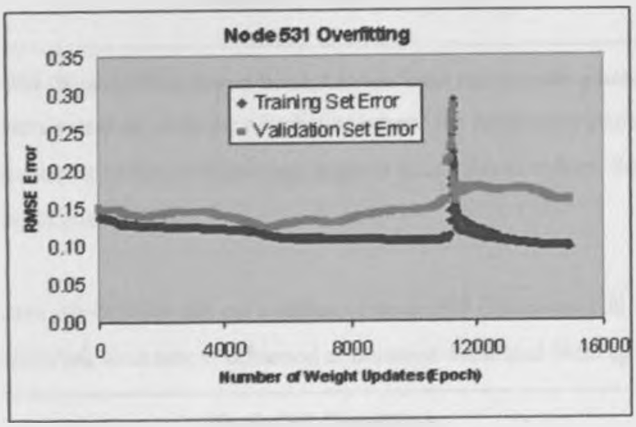


Figure 4-2 Node 531 (Former West View Hotel Westlands) Test Results

Figure 4-1 illustrates the results of oscillation and instability between 10000 and 12000 epochs. This means a local minimum was encountered. In this test, the optimum network achieved at between 4000 and 5000.

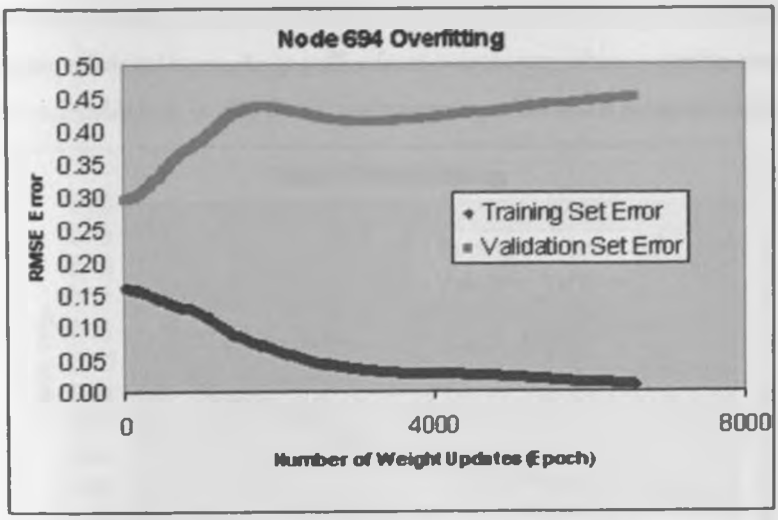


Figure 4-3 Node 694 (Waiyaki Way, Forest Road, Limuru Road Intersection parklands) Test Result

Figure 4-3 Illustrates a case of extreme overfitting where the network parameters need to be reduced from the current values and training begin at much lower values. Such a result depicts poor guess of parameters.

Figure 4-4 illustrates yet another run on a different node 958 (Museum Hill Center). The optimum neural network structure is achieved at between 4000 and 5000 epochs.

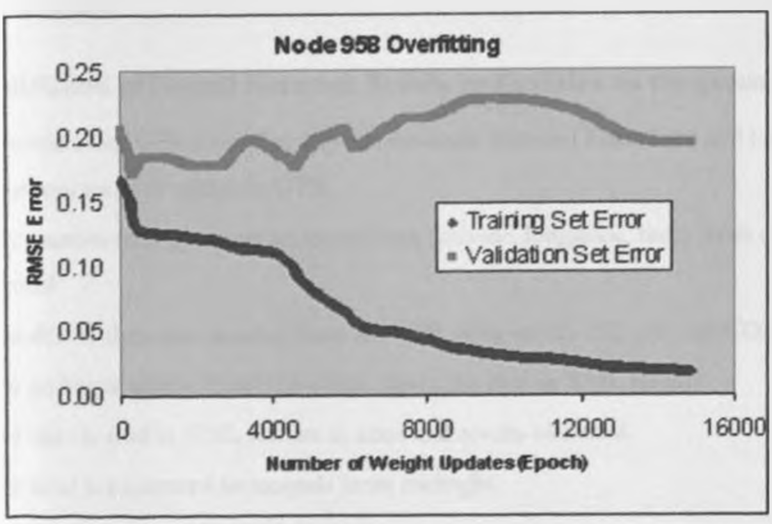


Figure 4-4 Node 958 (Museum Hill Center) Test Result

The validation line illustrates an increase in RMSE error as the training set error improves. It is characteristic of neural networks to suffer from overtraining when a suitable stopping function is not established. In this thesis, early stopping is enforced using the validation set.

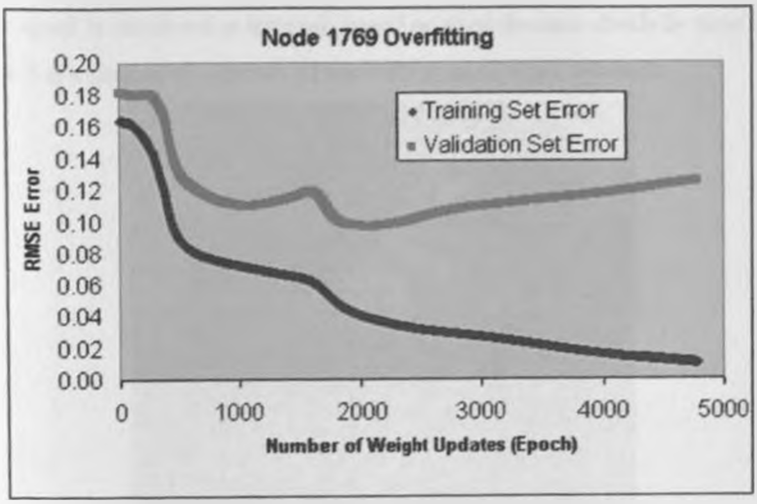


Figure 4-5 Node 1769 (Development House next to Railway Station) Test Result

In figure 4-5, the optimum structure is achieved between 1800 and 1200 epochs. Thereafter the validation error increases which results from over fitting. At about 1000 epochs, the network overcomes local minima. By using the delta rule, the neural network can converge to a global minimum.

4.5.1 Validation of Neural Network Results to Realities on the ground

Using a Garmin etrex GPS, a number of runs are made between Embakassi and Gigiri. Listed is outlines steps of using the GPS:

- The garmin etrex gps is set to record data (latitude, longitude, time) after one (1) second
- The data is then downloaded from the GPS using an RS-232 cable on COM1
- The software used is EasyGIS which stores the data in XML format.
- The data is read in XML format in excel and results obtained.
- The time is converted to seconds from midnight.

- The distance between successive points is calculated using Euclidean distance with a conversion to KM by multiplying by 110.592. (1 degree is approximately 110.592km at the equatorial region)
- The speed is calculated at intervals based on total distance divide by time interval

The figure 4-5 is a picture of a garmin eTrex GPS as used in the research.

Figure 4-6 Garmin eTrex Hand Held GPS

Table 4-1 is a ample of data obtained and processed to give speed of the probe car.

Latitude	Longitude	Date	Time	Seconds	Distance	Speed
-1.233215	36.812696	2006-03-29	14:31:34	52294	0.007	28.70965292
-1.23328	36.812696	2006-03-29	14:31:35	52295	0.007	27.46943749
-1.233366	36.812696	2006-03-29	14:31:36	52296	0.010	27.36883642
-1.23343	36.812696	2006-03-29	14:31:37	52297	0.007	26.124376
-1.233559	36.812739	2006-03-29	14:31:38	52298	0.015	28.81852243
-1.233687	36.812782	2006-03-29	14:31:39	52299	0.015	32.48250898
-1.233795	36.812804	2006-03-29	14:31:40	52300	0.012	35.15866593
-1.233881	36.812825	2006-03-29	14:31:41	52301	0.010	36.97123116
-1.233988	36.812868	2006-03-29	14:31:42	52302	0.013	37.95888663
-1.234074	36.81289	2006-03-29	14:31:43	52303	0.010	38.06914366
-1.234159	36.812932	2006-03-29	14:31:44	52304	0.010	39.16213644
-1.234245	36.812954	2006-03-29	14:31:45	52305	0.010	40.10846899
-1.234331	36.812997	2006-03-29	14:31:46	52306	0.011	40.51260891

Table 4-1 Sample Data collected from a garmin GPS

The average speed is determined to be 23 km/hr. The neural network predicts speeds between 10 and 40 km/hr. The following charts and maps depict the speed at various times during the period of the survey.

Figure 4-7 Speed Survey Map of 22 March 2006 – Embakassi to Gigiri

A good section of the journey is made at speeds between 39 and 94 km/hr. However there are sections (Parklands, GPO, Moi Avenue) which seem to have speeds less than 12 km/hr.

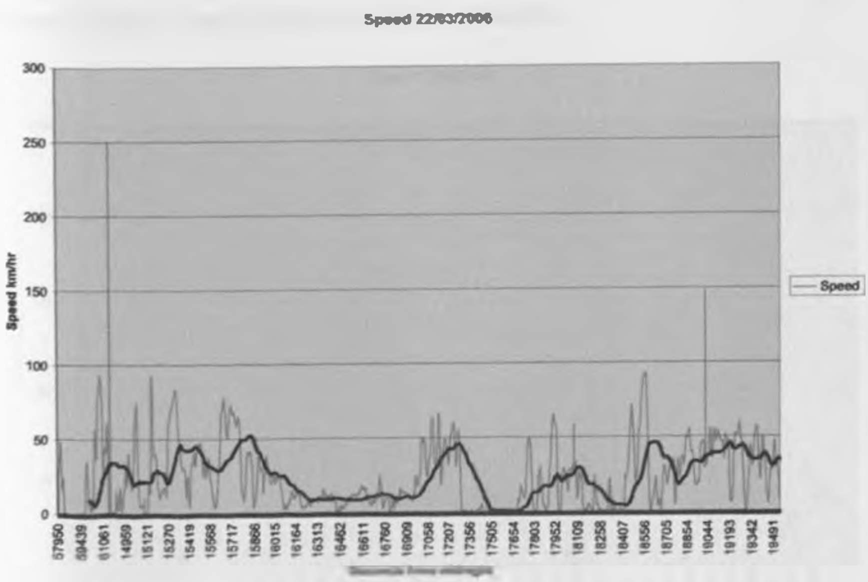


Figure 4-8 Speed Values and 3 Minute Moving for 22nd march 2006

Figure 4-8 illustrates a 3 minute moving average. A good part of the journey has a travel speed of less than 50km/hr.

Figure 4-9 Speed Survey Map of 23 March 2006 – Embakassi to Gigiri

The figure 4-9 illustrates a slightly different picture than 22nd. Embakassi Road is used and features a number of points where the speed reduces to less than 12 km/hr. There is an even distribution between 12 and 39 km/hr and 39 to 94 km/hr.

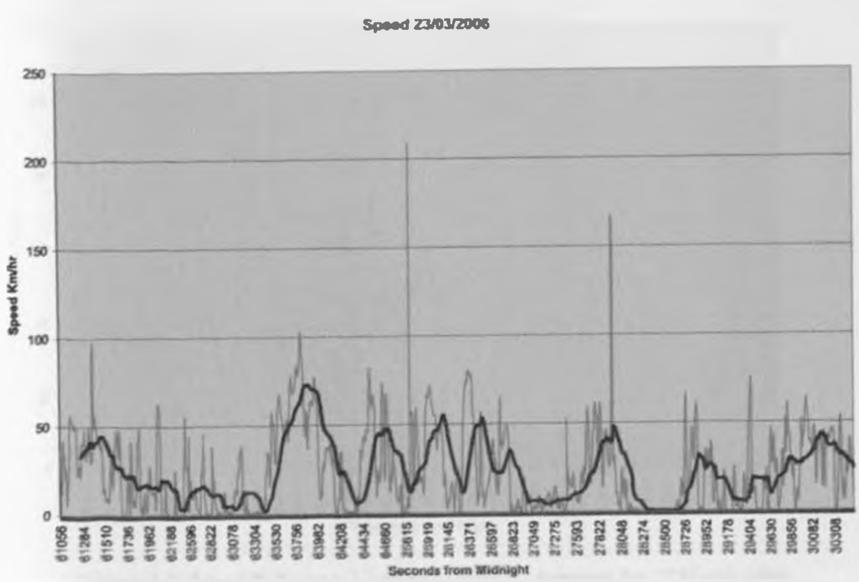


Figure 4-10 Speed Values of 23 March 2006 with 3 minute moving Average

From figure 4-10, the travel speed from using Enterprise road as a higher travel speed compared to Mombasa road but the travel distance is longer.

Figure 4-11 Speed Survey Map of 27 March 2006 – Gigiri to Embakassi

Figure 4-11 illustrates the journey back home using Mombasa Road on 27th. This is a relatively smooth journey with very few points of speeds less than 12 km/hr.

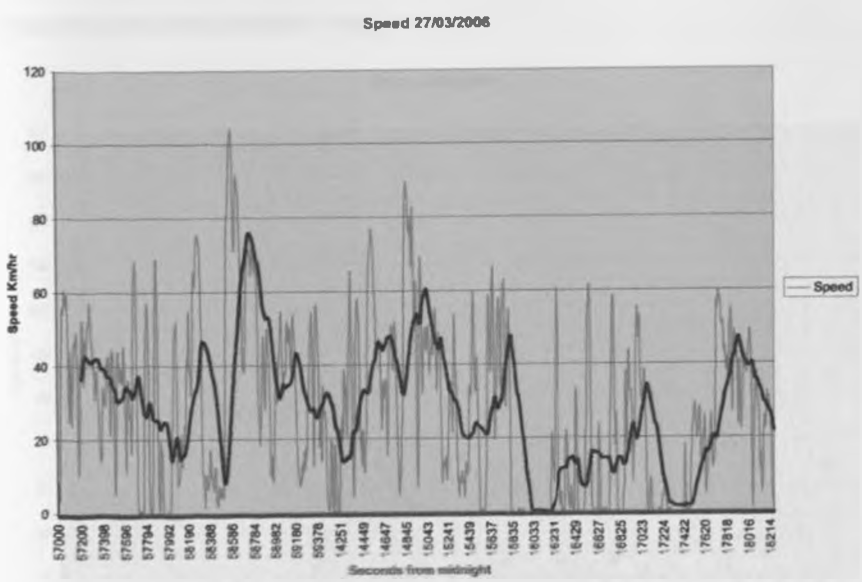


Figure 4-12 Speed Values and 3 Minute Moving Average for 27 March 2006

Figure 4-13 Speed Survey Map of 28 March 2006 –Gigiri to Embakassi

The journey home on the 28th has a very few points where the speed is less than 38 km/hr. The speed is reduced to less than 39 km/hr between Kenyatta Avenue and Hali Selssie. A similar situation after Bunyala Road later on.

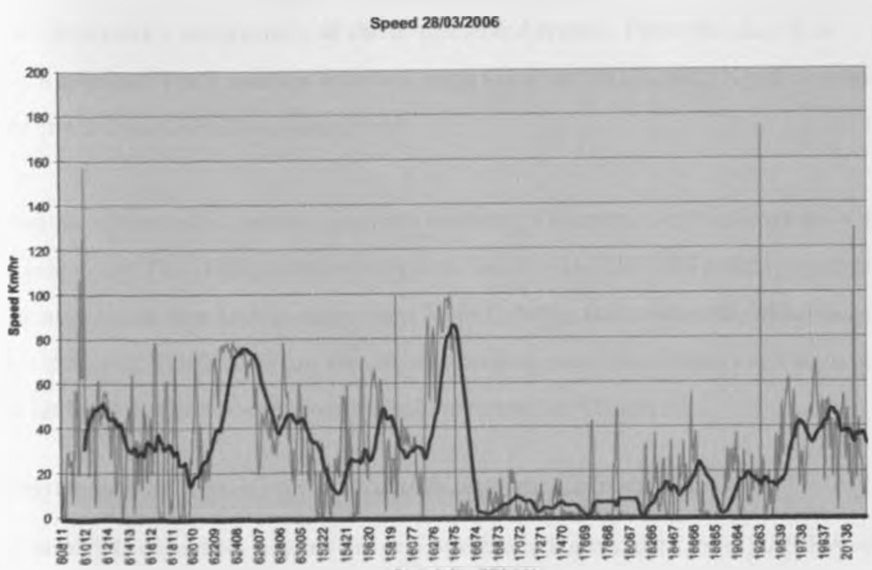


Figure 4-14 Speed values and 3 minute moving average for 28 march 2006

4.5.2 Motorists' use of the technology.

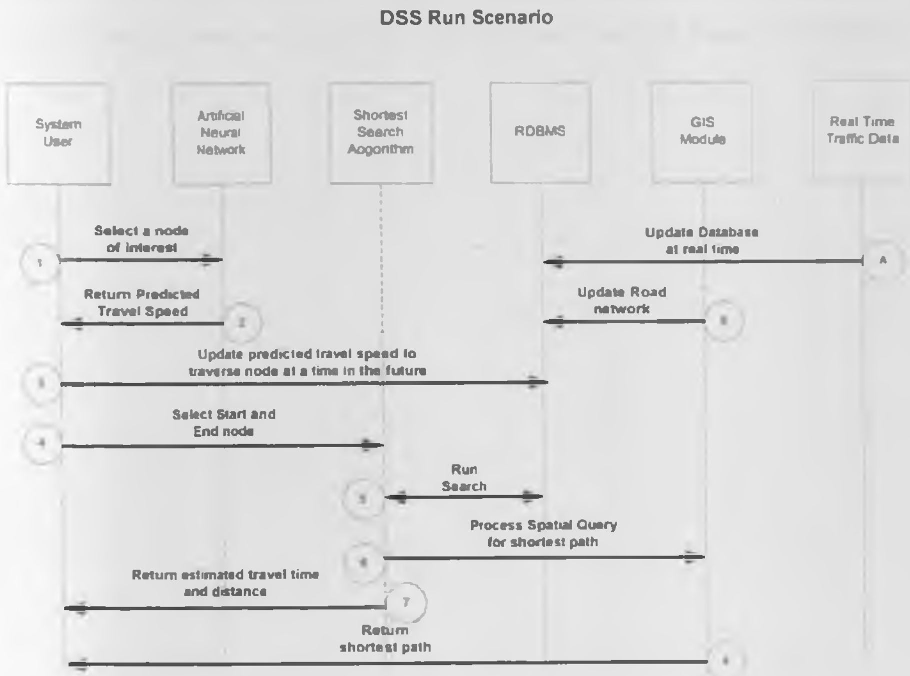


Figure 4-15 System Run Scenario

Figure 4-15 illustrates a run scenario of the implemented system. From the data flow diagram, it is assumed that a suitable interface using GPRS or IEEE 802.1X will be availed to support continuous seamless connectivity.

Using a remote system either before a journey or during a journey, a motorist selects a start node and end node. This is illustrated as step 4 on figure 4-14. The DSS returns a journey path, associated travel time and distance (step 7 and 8) to the motorists with which he/she can make a decision. Traffic data can also be displayed on road side displays and signs as illustrated on level 4..4 of the data flow diagram presented in Chapter 3.

4.6 Frequency of Training: Neural Network Calibration

Since real time data collection systems are not available in Kenya, data from KIPPRA was used for simulation using real data. If the system was connected to a real time data collection system, the training of the neural network would be after every 10 minutes and later on, once a day after years of data is available. Step 2 on figure 4-14 would be running automatically after any link and node status change or user intervention. As identified in the literature review, calibration of the network can take place even once a month Park et al [PSHJ05].

5 Results and Findings

5.1 Neural Network Analysis of the Result

Test	Lag Window 3			Lag Window 5			Lag Window 7			Lag Window 9		
	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference
1	0.1920	0.1570	0.0350	0.2142	0.0977	0.1165	0.1329	0.0571	0.0758	0.2494	0.0341	0.2153
2	0.1910	0.1570	0.0340	0.2152	0.0984	0.1168	0.1515	0.0657	0.0858	0.1628	0.0563	0.1066
3	0.2497	0.1958	0.0539	0.2937	0.2746	0.0191	0.2198	0.2661	0.0463	***0.3607***	***0.1378***	***0.2229***
4	0.2460	0.1190	0.1270	0.2423	0.0731	0.1693	***0.1760***	***0.0100***	***0.1660***	0.2203	0.0100	0.2103
5	0.2038	0.1119	0.0919	0.2169	0.0667	0.1502	0.0840	0.0100	0.0740	0.1740	0.0100	0.1640
6	0.2076	0.1033	0.1044	0.2386	0.0274	0.2112	0.0989	0.0100	0.0889	0.2209	0.0100	0.2109
7	0.2293	0.1062	0.1231	***0.2689***	***0.0258***	***0.2431***	0.0865	0.0100	0.0765	0.1575	0.0100	0.1475
8	***0.2777***	***0.0772***	***0.2005***	0.2190	0.0100	0.2090	0.1200	0.0100	0.1100	0.1715	0.0100	0.1615
9	0.2222	0.0841	0.1381	0.2365	0.0100	0.2265	0.1109	0.0100	0.1009	0.1774	0.0100	0.1674

* The optimum structure is Test 3 of lag Window 5

* The worst structure is Test 9 of lag window 5

The algorithm to determine the best and worst structure

1. Get the absolute minimum difference between the Validation RMSE and Training RMSE for each lag window
2. Get the minimum among the lag window minimum

Table 5-1 Node 1944 (Haile Selassie) Results

Test	Lag Window 3			Lag Window 5			Lag Window 7			Lag Window 9		
	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference
1	0.2397	0.1601	0.0796	0.2805	0.0977	0.1828	0.2326	0.0614	0.1712	0.1924	0.0417	0.1507
2	0.2472	0.1573	0.0899	0.2807	0.0984	0.1823	0.2226	0.0567	0.1659	0.2299	0.0366	0.1934
3	0.2535	0.1955	0.0580	0.3654	0.2637	0.1018	0.3755	0.3508	0.0247	0.2341	0.2112	0.0229
4	0.2224	0.1276	0.0948	0.3200	0.0748	0.2452	0.2074	0.0109	0.1965	***0.2398***	***0.0100***	***0.2298***
5	0.2262	0.1232	0.1030	0.3290	0.0485	0.2805	0.2183	0.0113	0.2070	0.2086	0.0100	0.1986
6	0.2024	0.0969	0.1055	***0.3526***	***0.0133***	***0.3393***	0.2012	0.0100	0.1912	0.1790	0.0100	0.1690
7	***0.2106***	***0.0919***	***0.1187***	0.3535	0.0190	0.3345	***0.2179***	***0.0100***	***0.2079***	0.1398	0.0100	0.1298
8	0.1839	0.0687	0.1152	0.2953	0.0100	0.2853	0.1881	0.0100	0.1781	0.1194	0.0100	0.1094
9	0.1729	0.0920	0.0809	0.2900	0.0100	0.2800	0.2042	0.0100	0.1942	0.1659	0.0100	0.1559

* The optimum structure is Test 3 of lag window 9

* The worst structure is Test 6 of lag window 5

The algorithm to determine the best and worst structure

1. Get the absolute minimum difference between the Validation RMSE and Training RMSE for each lag window

2. Get the minimum among the lag window minimum

Table 5-2 node 1769 (Hali Selassie Moi Avenue Intersection) Results

Test	Lag Window 3			Lag Window 5			Lag Window 7			Lag Window 9		
	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference
1	0.33455	0.07760	0.25695	0.28680	0.01735	0.26945	**0.21670**	**0.01000**	**0.20670**	0.38740	0.01000	0.37740
2	0.30800	0.08970	0.21830	0.28790	0.01620	0.27170	0.28370	0.01000	0.27370	0.43815	0.01000	0.42815
3	***0.36790***	***0.06825***	***0.29965***	***0.39300***	***0.01795***	***0.37505***	0.22220	0.01000	0.21220	0.49680	0.01000	0.48680
4	0.30395	0.05155	0.25240	0.28730	0.01000	0.27730	0.29760	0.01000	0.28760	***0.52030***	***0.01000***	***0.51030***
5	0.27440	0.05856	0.21584	0.33950	0.01000	0.32950	0.23650	0.01000	0.22650	0.47330	0.01000	0.46330
6	0.29655	0.04967	0.24689	0.32200	0.01000	0.31200	0.27750	0.01000	0.26750	0.48715	0.01000	0.47715
7	0.29810	0.05310	0.24500	0.29730	0.01000	0.28730	0.34190	0.01000	0.33190	0.38965	0.01000	0.37965
8	0.31705	0.04245	0.27460	0.32710	0.01000	0.31710	***0.37400***	***0.01000***	***0.36400***	0.42685	0.01000	0.41685
9	0.28505	0.05620	0.22885	0.28865	0.01000	0.27865	0.28695	0.01000	0.27695	0.48390	0.01000	0.47390

* The optimum structure is Test 1 of all lag window 7 is optimum
* The worst structure is Test 4 of lag window 9

The algorithm to determine the best and worst structure

1. Get the absolute minimum difference between the Validation RMSE and Training RMSE for each lag window
2. Get the minimum among the lag window minimum

Table 5-3 node 928 (Kipande Road Globe Cinema Round About)Results

Test	Lag Window 3			Lag Window 5			Lag Window 7			Lag Window 9		
	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference
1	0.2151	0.0900	0.1251	0.2180	0.0150	0.2030	0.3433	0.0100	0.3333	0.2953	0.0100	0.2853
2	**0.2147**	**0.0900**	**0.1247**	***0.2467***	***0.0114***	***0.2353***	0.3269	0.0100	0.3169	**0.2919**	**0.0100**	**0.2819**
3	0.2288	0.1022	0.1266	0.2093	0.0240	0.1853	***0.3882***	***0.0100***	***0.3782***	0.3206	0.0100	0.3106
4	0.3101	0.0569	0.2532	0.1891	0.0100	0.1791	0.3105	0.0100	0.3005	***0.3431***	***0.0100***	***0.3331***
5	0.3085	0.0541	0.2544	0.1607	0.0100	0.1507	**0.2960**	**0.0100**	**0.2860**	0.3385	0.0100	0.3285
6	0.3249	0.0374	0.2875	0.1532	0.0100	0.1432	0.3023	0.0100	0.2923	0.2999	0.0100	0.2899
7	0.3035	0.0495	0.2540	0.1701	0.0100	0.1601	0.3291	0.0100	0.3191	0.3189	0.0100	0.3089
8	***0.3468***	***0.0369***	***0.3100***	0.1522	0.0100	0.1422	0.3504	0.0100	0.3404	0.3293	0.0100	0.3193
9	0.3189	0.0465	0.2725	**0.1510**	**0.0100**	**0.1410**	0.3340	0.0100	0.3240	0.3232	0.0100	0.3132

* The optimum structure is Test 2 of all lag window 3 is optimum
* The worst structure is Test 3 of lag window 7

The algorithm to determine the best and worst structure

1. Get the absolute minimum difference between the Validation RMSE and Training RMSE for each lag window
2. Get the minimum among the lag window minimum

Table 5-4 Node 560 (Pangani Road Round About Intersection) Results

Test	Lad Window 3			Lag Window 5			Lag Window 7			Lag Window 9		
	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference	Validation RMSE	Training RMSE	Difference
1	0.22810	0.07405	0.15405	0.22855	0.02725	0.20130	0.36325	0.01000	0.35325	0.21885	0.01000	0.20885
2	0.22375	0.06807	0.15568	*0.27935*	*0.03850*	*0.24085*	0.32190	0.01000	0.31190	0.22060	0.01000	0.21060
3	**0.22895**	**0.07610**	**0.15285**	0.26555	0.05375	0.21180	0.29115	0.10510	0.18605	0.22335	0.01000	0.21335
4	0.22455	0.03500	0.18955	0.19410	0.01000	0.18410	*0.38270*	*0.01000*	*0.37270*	0.24115	0.01000	0.23115
5	0.23180	0.04260	0.18920	0.20680	0.01000	0.19680	0.36040	0.01000	0.35040	**0.20230**	**0.01000**	**0.19230**
6	0.23625	0.03375	0.20250	0.19490	0.01000	0.18490	0.33430	0.01000	0.32430	0.24115	0.01000	0.23115
7	0.22005	0.02920	0.19085	0.21180	0.01000	0.20180	0.34195	0.01000	0.33195	0.24465	0.01000	0.23465
8	*0.26685*	*0.01280*	*0.25405*	0.18750	0.01000	0.17750	0.37465	0.01000	0.36465	*0.31205*	*0.01000*	*0.30205*
9	0.23765	0.02120	0.21645	*0.18525*	*0.01000*	*0.17525*	0.36900	0.01000	0.35900	0.23145	0.01000	0.22145

* The optimum structure is Test 3 of lag Window 3
* The worst structure is Test 4 of lag window 7

The algorithm to determine the best and worst structure

1 Get the absolute minimum difference between the Validation RMSE and Training RMSE for each lag window

2 Get the minimum among the lag window minimum

Table 5-5 Node 72 (Outer Ring Road, Thika Road Round About) Results

5.2 A* and Dijkstra Search algorithm analysis of results.

A series of random points were selected to test the versatility of the A* and Dijkstra's search algorithm. Figure 5-1 illustrates the points. Various search runs were made from points in the same concentric circle or region to another.

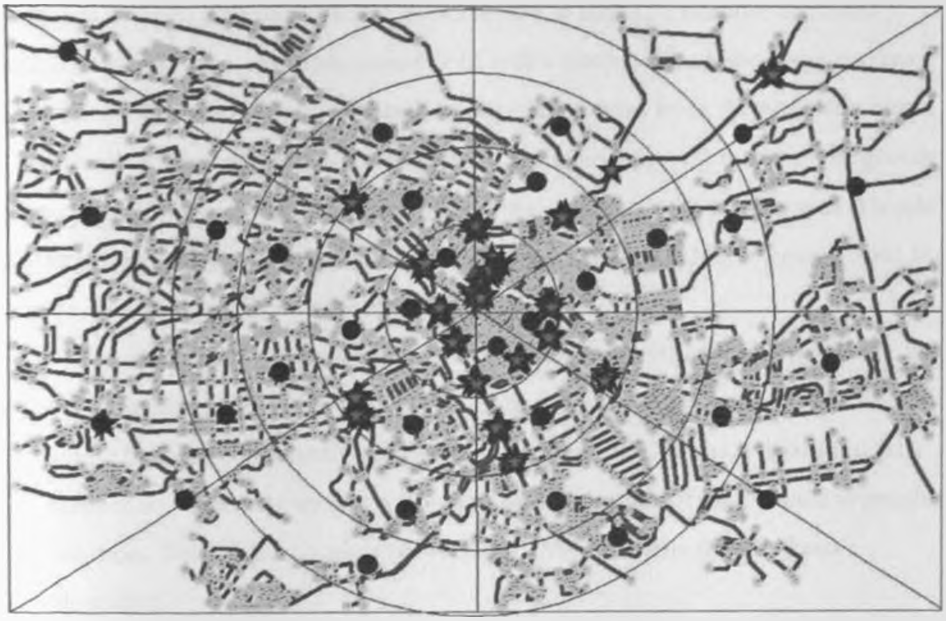


Figure 5-1 Random Points (black) selected to carry out shortest path analysis

The run results were tabulated as shown in table 5-6 below:

Search	Start Node	End Node	Path length in meters	Algorithm Loop count recorded	Algorithm Run Time (min)
A*	43	2578	16,877	11,283	62.44
Dijkstra	43	2578	17,448	9,013	49.42
A*	2578	43	16,877	10,195	37.47
Dijkstra	2578	43	16,640	8,283	38.47
A*	1328	1442	3,477	4,712	1.42
Dijkstra	1328	1442	3,504	3,521	1.26
A*	1442	1328	3,480	1,799	0.09
Dijkstra	1442	1328	3,447	1,550	0.11
A*	[illegible]	1844	2,884	1,844	0.09
Dijkstra	[illegible]	1844	2,211	1,319	0.04
A*	1844	[illegible]	2,884	2,286	0.17
Dijkstra	1844	898	2,886	1,853	0.11
A*	1844	1035	3,755	3,588	0.74
Dijkstra	1844	1035	3,918	2,733	0.40
A*	1035	1844	3,755	3,647	1.18
Dijkstra	1035	1844	4,158	3,271	1.31

Table 5-1 A* and Dijkstra search results

A complete list of run results is found in appendix B.

This thesis does not intend to compare nor contrast the performance of either the A* or Dijkstra's algorithm. A more objective test environment would be necessary for a comprehensive treatment.

Performance of both algorithms would be measured in terms of number of nodes opened, cost of comparisons made, number of cycles (loops). In addition, the current implementation of the A* search is based on impedance from both the node and link. The same code is altered to skip the summing of the heuristic component to the greedy component thus derive a primitive form of Dijkstra's algorithm. No correction is made to transfer the impedance associated on a node to adjacent links. Nevertheless, from the run results, the following observations can be made:

1. The number of loops evaluated by the Dijkstra's algorithm is approximately 20% less than the A* for most runs. This results in faster convergence.
2. The A* search consistently performs worse than the Dijkstra's algorithm with respect to processing time as a function of the number of loops made to reach a solution. The A* search takes on average 31% more time than Dijkstra's algorithm.
3. A bidirectional search would outperform a single direction search.
4. Dijkstra's algorithm doses not give consistent optimal routes when the start and end points are reversed.
5. A* search gives consistent results even on reversal of the start and end nodes.

The choice of using the A* search stems primarily from the fact that it will always give consistent results from either direction a search is carried out. This results in consistent replicable answers to shortest path questions..

The A* examines and expands nodes in an ellipsoid manner unlike the Dijkstra's which expands nodes in circular bands. A more thorough investigation is required to address performance of search algorithms with respect to real time decision support and the need for satisficing rather than 100% accuracy.

6 Discussion

6.1 Overview

We have developed a system that gives the shortest path between any two points in Nairobi using predicted traffic congestion values. A GIS based traffic transport system is developed for modeling real time traffic parameters. Two artificial intelligence models are deployed to make decision making a reality: A feed forward artificial neural network and an A* search algorithm.

This chapter is organized as follows: section 6.2 presents the conclusion from this research thesis in support of using computer techniques in managing traffic congestion as opposed to intensive capital investment on road infrastructure. Section 6.3 presents the limitations of this research with a view of improving on the same. Lastly section 6.4 presents suggestions for future development in the problem of time series prediction and shortest path analysis.

6.2 Conclusion

Artificial intelligence is capable of solving traffic management problems in Nairobi as an alternative to capital investment on road construction. A.I techniques can be deployed within the framework of GIS based decision support systems to fundamentally predict short term traffic congestion, simulate scenarios to enhance traffic management and help in creating policy for long term sustainability of infrastructure. From actual GPS runs, the average vehicle speed in Nairobi is 23km/hr/hr which is 46% lower than optimum speeds of 50Km/hr. Clearly Nairobi roads are nearing capacity hence traffic congestions will be at its worst within the next few years if no policy or directive is made with respect to management.
This thesis presented a methodology and framework suitable for building an A.I, GIS decision support system for road network and transportation analysis. It is recommended to by the positive research results that rapid application design with prototyping is most suitable for building DSS systems. Service oriented architecture using object oriented platform on a multithread OS is most essential for both building and supporting a DSS solution which are responsive.

Artificial neural networks are suitable in predicting road traffic congestion in Nairobi for short term purposes. Neural network are effective in short term prediction when there is massive historical data. Support data on weather, events, and season is also critical. Using a sliding window approach is most beneficial in traffic prediction however, care needs to be taken into account to avoid over fitting. Artificial intelligence is critical in decision support and knowledge management. Scenarios can be built to look at the impact of policy or capital development in a growing city like Nairobi. Of particular concern is traffic management where intelligent agents can be deployed to manage congestion and traffic in general.

In addition, the A* search is an efficient algorithm in automated traffic management and decision support when the system is converted to a static model using short time lags. The A* search is effective for small networks as seen in Nairobi however, care needs to be taken in developing the heuristic component. If it is small, the A* decomposes to a greedy search and performs similarly to the Dijkstra's algorithm. Other factors need to be considered as identified in this research in fine tuning the A* search in terms of road characteristics and traffic influence for instance surface condition, location, width and gradient.

One critical generic component of a DSS is a visualization system or graphical user interface. As demonstrated in this report, GIS is critical in traffic management as it helps one get a quick overview of what is happening on ground. By visualizing the results of the search module, one is able to asses the maturity of our road network and identify suitable routes to expand or build mechanisms to control traffic. The speed survey carried out identifies roundabouts as most critical bottle necks.

Lastly, the city of Nairobi needs to deploy a traffic and route management system as proposed by this research. This will cut down the response time of emergency services and also warn people on identified routes of oncoming emergency vehicles and personnel thus create space. It goes without saying that data is not ready accessible in Kenya as experienced by the researchers. It is important for the government and academic institutions to partner in research and surveys to ensure that data collected is readily available for future research and analysis.

6.3 Limitations of Prototype

The forward feed back propagation neural network is best suited for static data modeling. Due to the spatial temporal nature of traffic congestion, a fundamental assumption taken was to reduce the dynamic nature of traffic to static by taking time segments. Data collected from KIPPRA was based on 30 minute time intervals. A better prediction and mapping to real time dynamics would benefit from a shorter data collection interval. This can be achieved by deploying a GPS data collection mechanism as presented in this thesis or use of loop detectors. A more suitable approach is to use a time delay neural network, reinforcement learning or K-nearest neighborhood algorithm to model time dynamics. Traffic congestion as a probabilistic phenomenon is a good candidate for naïve Bayesian networks. Bayesian networks allow one to calculate the conditional probabilities of the nodes in the network given that the values of some of the nodes have been observed.

Recent developments in dynamic algorithms for routing have led to improvement of search algorithms. In dynamic transportation networks, weight changes can be classified as either deterministic or stochastic time dependent. A more practical approach to routing is to use time-dependent shortest path algorithms. The link and node weight as a time dependent random variable is modeled using probability density functions and time-dependency.

This prototype re-computes the optimal route after every change in network and node weights based on the neural network predictions. This is an intensive process. It is subject to improvement by localizing optimizations, use of bi-directional A* or improving the A* by using advanced refinements of the A* search algorithm. An improved A* search should use the results of previous search to speed up later searches. By using the principle of the ant algorithm, a definite improvement of the search can be realized.

6.4 *Suggestions for future work*

As echoed by Efraim [E95] artificial intelligence has a number of advantages over natural intelligence. This thesis has demonstrated the possibility of developing a traffic management system to aid in managing traffic on Nairobi roads. There is room for development and future research which can be addressed in the following broad titles:

1. Data collection methodology using real time sensors,
2. Transport network modeling,
3. Traffic congestion prediction,
4. Dynamic routing and search strategies,
5. Traffic simulation.

6.4.1 Data Collection methodology

Data presented in this research from KIPPRA was collected manually and prone to collection, recording and collation errors. The data collected based on arm counts on a round about and a junction varies between 97% and 99%. The Greenshield theorem presented and used to derive traffic speeds is a rather generalized approach to modeling traffic densities and flow from traffic counts. From a practical point of view, it is much easier to collect traffic counts than travel speeds. This has a drawback in analysis where speeds are required resulting to generalization. A better traffic count methodology is to use automatic data collection tools like loop detectors, sensors, video camera or GPS. As presented in this thesis, the GPS probe car used is able to obtain data at 1 second interval on all points traversed. Appendix H illustrates the data collection points and dates. For a comprehensive and accurate treatment of data, traffic counts should be collected all day through out the year.

The government of Kenya and academic institutions need to build up competency and systems to facilitate continuous collection of traffic data in all urban towns in Kenya. Listed are ways this can be done as potential future research areas:

1. Automated passive data collection (mobile phones, CCTV),
2. Point sensors (loop detectors),
3. Link sensors,
4. Vehicle probes (mobile phone, GPS, 911 requirements),
5. Video capture.

6.4.2 Transport network modeling

The road network developed in this research is based on topographical maps older than 15 years. The satellite image used to update the topographical maps is older than 7 years. An enhanced network based on current images and topographical sheets would result in better solutions. The digitizing of the data sources is most critical in developing a transport network. The transport network developed in this thesis has a number of errors ranging from dangles, missing links and incorrect attributes for roads.

6.4.3 Traffic congestion prediction

A feed forward neural network has been presented to predict traffic speeds. This approach is fraught with the major limitation of being suitable for static systems. An enhanced approach would be to use time-delayed neural networks, naïve Bayesian systems or K nearest neighborhood system. The use of probability densities and case based reasoning in prediction in the form of a hybrid system would greatly enhance the quality of results, accuracy of predictions and the step-ahead window from minutes to days or weeks with reasonable accuracy.

6.4.4 Dynamic routing and search strategies

A* search algorithm has been successfully been deployed to determine the shortest path between any two points. It is worthy to note the various variation of A* available to compare with in terms of performance. An enhancement to processing is to approach searching from parallel computing. This is achieved by use of bi-directional searches running on separate processing units. It will be most educative to compare A* runs with other search algorithms such Dijkstra's, floyd-Warshall, genetic algorithms and the ant.

6.4.5 Traffic simulation

The results of processing presented in this thesis are reported using a GIS. This would be greatly enhanced by use of a time variant continuous simulation engine, capable of using results of the prediction to animate agents on a scaled down model of the transport network. Such engines exist and use static mathematical models to run simulations of vehicle interaction on specific points of a road. This concept as used in computer games can be an invaluable tool in simulation and playback of predictions.

6.5 Government Maturity.

Schrader [S05] presents an evolutionary approach Governments pass though to maturity with respect to data collection and specifically transport network data. These phases illustrated as a S-curve (sigmoid) have the following listed critical zones:

1. No data collected due to lack of human resources or funds to support long term planning. What exist is not easily shared as characteristic in many third world countries where data collection is still very expensive and seen as proprietary with high resale value,
2. Data collected in quasi-real time and then is effective in developing ability to share information as characterized in many developed nations where value and service delivery is critical to effective and efficient running of the economy,
3. Data collected in real time and shared in real-time with the public (e-government).

Kenya is making a positive transition from phase 2 to phase 3. As illustrated in figure 6-1,

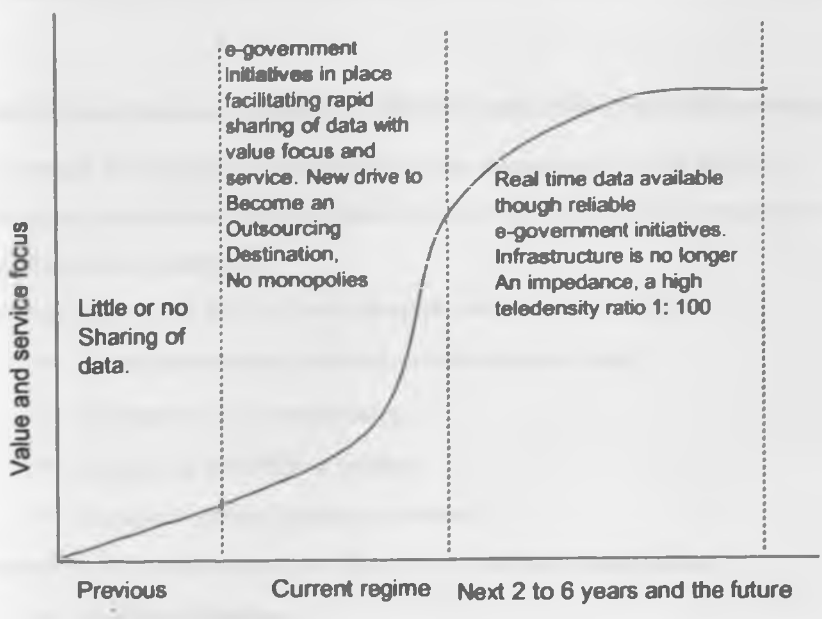


Figure 6-1 The progression of data sharing and e-government

This research would benefit immensely if road traffic data were available freely from a website, and would have at minimum:

1. Speed data,
2. Events on road (spatial-temporal),
3. Road closures,
4. Accidents and slow spots.

For major intersections in major cities or towns. The aforementioned data using the proposed system presented in this research can be used to update changeable message signs. These billboards can be used to share travel time, incident messages and abduction / carjack messages. As identified in Jan et al [JS05] car drivers on motorways network may be aided by information about traffic from electronic boards, radio broadcasts and mobile phone broadcasts as detailed in Diane [DM05]. Route control helps motorists select a route that optimizes travel time, travel cost or distance through a network. Provision of routing information or route direction is expected to result in an efficient use of the network capacity under all traffic conditions. Information may be lengths of traffic queues at particular locations, travel time estimates or predictions for one or several routes recommendation.

Wasike [W01] identifies that easing congestion, or improved roads could help reduce emissions of some pollutants. Tackling road congestion is the basis of both economic and environmental development. Both Wasike, Bennett [B06] suggest methods to improve congestion from making better use of existing road infrastructure, managing demand for travel by road, creating new infrastructure and doing nothing which results to the increasing congestion to influence road users

6.5.1 Social Investments as a result of effective and efficient traffic management

The benefits accrued from effective and efficient traffic management can be listed as :

1. An efficient transport system is essential for economic growth and to support an enjoyable and serve daily life.
2. Economic importance derived from transport research can result to:
    - just in time delivery thus reduce on inventory costs,
    - pre-trip and en-route planning,
    - emergency and military support,
    - economic service (premium advisory),
3. Environmental management as an objective in transport management:
    - pollution reduction,
    - health – road rage / anxiety

6.5.2 New markets available for route management services as proposed by this solution

A number of services can benefit from the proposed solution and prototype ranging from emergency services, to material and people movement. This opens up new avenues for business and knowledge sector growth in the Kenyan economy.

6.6 Closing remarks

The Government of Kenya through the ministry of information is in the process of presenting an ICT bill 2006 and e-government strategy which this research and future initiatives might benefit. There is provision for research and development which transport research initiatives will immensely benefit through:

1. Establishment of one or more cost / performance test beds for data collection approaches where all forms of data collection techniques can be easily tested and compared as suggested in this thesis.
2. Raise awareness of decision-makers on the potential value of artificial intelligence and strategic decision support systems as presented in this thesis and not limited to travel time, real-time and predictive systems.
3. Establishment of a transport research laboratory for focused research on time travel prediction in Kenya urban cities.
    a. Mainstream initiatives focusing on methods and frameworks to increase public and private investment into traffic management and congestion control research and implementation.
    b. Conduct research on the value of prediction versus-real-time information.
    c. Improve on this study to use other network architecture such as recurrent neural networks.
4. Invest in continuous travel speed data surveys.

Figure 6-2 illustrates a traffic prediction value chain to turn Kenya around with respect to traffic management and prediction services.

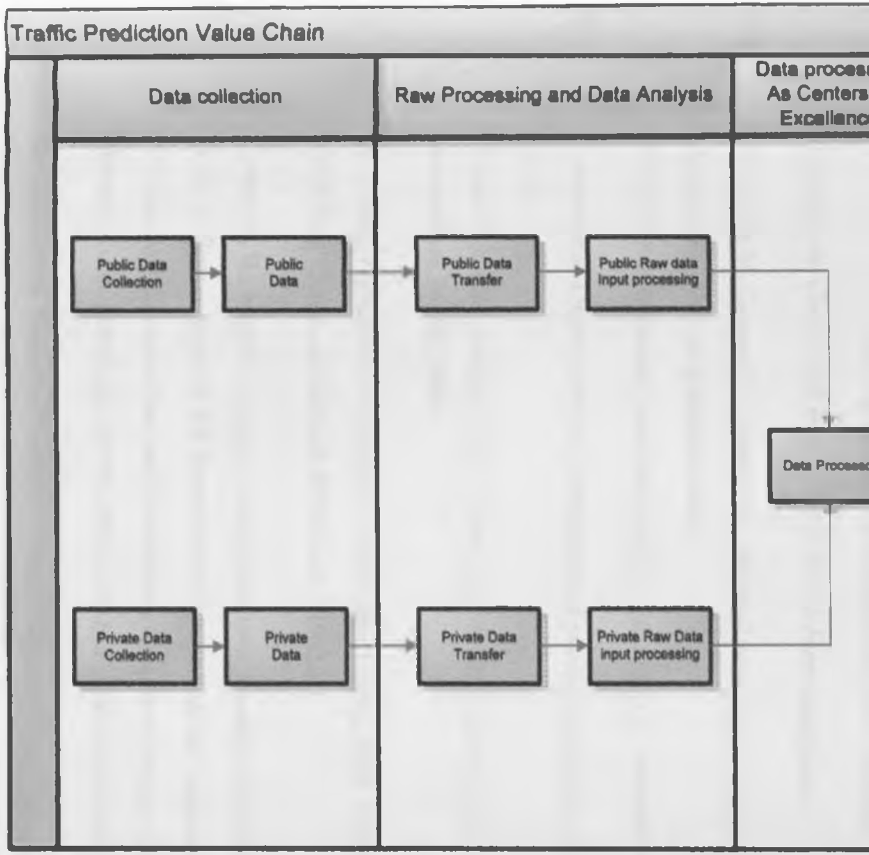


Figure 6-2 Traffic prediction value Ch

Figure 6-2 depicts a proposed value chain which involves both private and public sector pl
dissemination. The proposed system fits under the column of prediction and management

7 Reference and Bibliography

[AC98] M. Adya , F. Collopy *How effective are neural networks at forecasting and prediction, A review and evaluation.* 1998, Journal of forecasting 17, 481 - 495.

[AK05] Arroyo S. and Kornhauser L.A., 2005, *Modeling travel time distributions on a road network*,2005, Presentation at 2005 TRB Conference, Washington

[B06] Garvin Bennett *Traffic Jams: Why I accuse the planner*, May 21 2006., Lifestyle magazine, Motoring, Sunday Nation

[B95] C.M Bishop *Neural Networks for pattern Recognition.* Oxford University press, UK, 1995, ISBN: 0 19 853864 2

[C05] S.S. Crone, *Forecasting with Neural Networks*, 2005, Portal on forecasting with artificial neural networks, EVIC 2005 Tutorial Santiago de Chile, 2005 http://www.neural-forecasting.com/

[CCG03] Cecchelti M. Corani G, Guarsio G, *Artificial neural networks prediction of Pm 10 in Milan Area*, 2003 unpublished work. http://www.iemss.org/iems2004/pdf/ai/ceccorti.pdf accessed on 14/04/2006

[CD05] Connecticut Department of Environmental protection, *Roads and trails metadata, 2005* http://dep.state.ct.us/gis/dataguides/dep/layers/metadata/road_coverage.html accessed on 26/01/2006

[CH05] K. Chang, *Introduction to Geographical Information Systems, 2005*, Tata McGraw-Hill Publishing Company Limited ISDN-0-07-049552-1

[CMJ01] Cortez P., R., Miguel, N.,Jose. *Evolving time series forecasting Neural Networks Models*, 2001, In Proceedings of 3rd International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models (ISAS 2001), Havana, Cuba, March, pp. 84-91, 2001, ICIMAF, ISBN:959-7056-14-3.

[D94] Department of Trade and Industry , *Neural computing learning solutions*, 1994, Best practice guidelines for developing neural computing applications, Grown copyright 1994

[DES01] United Nations Department of Economic and Social Affairs, *Human Resources Development for Eco-partnership building in Local Governance*, 2001

[DM05] C. Diane, C.B.Mary, *The role of mobiles in Disasters and Emergencies*, 2005, http://www.gsmworld.com/documents/public-policy/disaster_relief_report-pod December 2005, accessed on 14/04/2006

[DMSS98] Demetsky M.J. McGhee C. C., Smith L.B, Sadek W.A., *Development of decision support systems for real-time freeway traffic routing: Volume II*, 1998, http://www.virginiadot.org/vtrc/main/online_reports/pdf/99-r10.pdf

[E95] E. Turban, *Decision Support and Expert Systems: Management Support Systems*, 1995, Edition 4, Prentice-Hall , ISBN 0-02-421701-8,

[EAL04] Turban E., J.E. Aronson, Ting-Peng Liong, *Decision Support Systems and Intelligent Systems* ,2004, Edition 7, Prentice Hall, ISBN 0-13-0461067

[F99] M. Florian *Untangling Traffic Congestion: Application of network equilibrium models in transport planning*, 1999, OR/MS,

http://www.lionhrtpub.com/orms/orms/orms-499/florian.html accessed on 14/04/2006

[FD01] Frank J.R., Davey N., Hunt S.P. *Time series prediction and neural networks* , 2001, Journal of Intelligent and Robotic Systems 31: 91 – 103,

[FJ61] Fu-Sheny Ho, Petros Joannou *Traffic flow modeling and control using Artificial Neural Networks*, 1961 IEEE control systems 0272-1708, http://www-rcf.usc.edu/~ioannou/2003update/d48pdf

[FL03] Fildes R, Liao K., *The accuracy of a procedural approach to specifying feed forward neural networks for forecasting*, 2003, Lancaster University Management School Working Paper 2003/098 http://www.lums.lancs.ac.uk/publications/viewpdf/000243/ accessed on 14/04/2006

[FS91] J.A. Freeman, D.M. Skapura , *Neural Networks: Algorithms, Applications, and Programming Techniques (Computation and Neural Systems Series)* 1991, Addison-Wesley ISBN-0201513765

[H05] J.Husdal *Fastest path problems in dynamic transportation networks*, 2000 http://www.husdal.com/mscgis/research.htm accessed on 29/9/2005

[HD04] Hossain, M.S., Davis, C.G. *A GIS to Reduce Flood Impact on Road Transportation Systems*, 2004, ESRI User Conference http://gis.esri.com/library/userconf/proc04/docs/pap1512.pdf accessed 20/08/2006

[HDR05] The United National Development Programme, *Human Development Report.*, 2005

[I01] Innamaa, S. *Short-Term Prediction of Highway Travel Time using MLP-Neural Networks*, 2001, 8th world Congress on Intelligent Transportation Systems.

[I04] ITS Orange Book , *Predictive Travel Time*, 2004 – Issue 1, Source:http://www.pbsj.com/itsorangebook/ accessed on 26/01/2006

[IW99] E. F. Ian H.Witten. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* 1999, The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, 1999.

[J03] C. Jacobs *Average speed prediction using Artificial Neural Networks* ,2003, Master of Science thesis Knowledge Based Systems group Delft University of Technology

[JS05] Jan H. V., Schuppen, *Dynamic Route Control for Motorway Networks*, 2000, ERCIM News No 40 – January 2000 http://www.ercim.org/publication/ercim_news/enw40/van_schuppenl.html accessed on 14/04/2006

[K98] N. K. Kasabov, 1998, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering (Computational Intelligences.)*, 2 Edition 1996 MIT ISBN 0262112124

[KB06] Alain K. Kornhauser, Megon Lynn Bernard *Traffic congestion: How predictable? Discovering volume trends across time and confirming fundamental speed-flow density relations.* – May 2006 http://www.ofe.princeton.edu/alaink/papers/Bernardtrafficcongestionpapers.doc accessed on 14/04/2006

[KHM05] Kajitani Y, Hipel. KW , McLead A.I *Forecasting non linear Time series with feed forward neural networks: A case study of condition Lynx Data*, 2005, Journal of forecasting 24, 105-117

[KR02] Kisgyergy L., Rilett R.L. *Travel time prediction by advanced neural network*, 2002, Periodica Polytechnica Ser. Civil Engineering Vol 46 No 1 Pp (15 – 32)

[L97] C.G. Looney *Pattern Recognition Using Neural Networks Theory and Algorithms for Engineers and Scientists.* 1997 Oxford University Press
ISBN 0-10-507920-5

[LS99] Lingras, P. and Sharma S. 1999. *Short-Term Traffic Volume Forecasts: Existing And Future Research*, Proceedings of the Annual Conference of Canadian Society of Civil Engineers, Regina, Saskatchewan, IV, pp. 429-438.

[M05] Muhomo Meshack Mbugua P15/7021/2001 , *Stock Price Prediction at the Nairobi Stock Exchange (NSE) a Neural Network Approach* , 2005 University of Nairobi; School of Computing and Informatics

[M05] Mwangi M. *Decision Support systems implementation strategy: Case Study REACT*, 2005, MSc. School of Computing and Technology, University of Sunderland United Kingdom.

[MH99] Miller H. J., Wu Y., Hung. M., *GIS based dynamic traffic congestion modeling to support time critical logistics*, 1999, Proceedings of the 32nd Hawaii International Conference on System Sciences – 1999

[MRS03] A., Michael, R.H Robert, P., Sumina , *GIS to minimize the impacts of hazardous waste shipments* ,
http://gis.esri.com/library/usercont/proc01/professional/papers/pap572/p572.htm accessed on 14/04/2006

[N03] S Niraj, *Application of GIS in Air Pollution Modeling*, 2003 Central Road Research Institute Delhi India Source ,
http://www.gisdevelopment.net/application/environment/air/index.htm accessed on 14/04/2006

[NHR99] Nelson M, Hill. T ,Remus W., *Time series forecasting using neural Neutrals: Should the data be de-easonalized first?*, 1999, Journal of forecasting 18(1999) 359—367

[O04] Okioga Teshamulwa F16/7950/00 *FCE 446: Highways Engineering Course Work: Traffic Flow Analysis and Junction Design*, 2004, Department of civil Engineering, University of Nairobi

[O05] N., H.Osoro, Msc Thesis, *Utilizing ICT to realize commuter needs in Nairobi transport sector*, 2005, University of Nairobi

[P00] Plummer Eric A, *Time series forecasting with feed forward neural networks: Guidelines and limitations*, 2000, MSc Computer Science, University of Wyoming 2000

[PA] Partha Chakroborty, Animesh Das *Principles of transportation Engineering*

Prentice Hall new Dheli – 110001 Page 66-67 (macroscopic traffic models)

[PB99] Post, Buckley International Inc. and Kaburu, Okello & Partners, *Nairobi Long Term Transport*, 1999, Study for the Ministry of Local Authorities.

[PDV03] K. Praween, R. Dhanunjaya, S. Vorum, *Intelligent Transport System using GIS*, 2003, MAP India Conference 2003

http://www.gisdevelopment.net/application/utility/transprot/mi03164.html

accessed on 10/11/2005

[PSHJ05] Park, D. Shin, H. Hong, S., Jung, C. *The use of Historical Data for Travel Time Forecasting in the ATIS*, 2005, Journal of the Eastern Asia Society for Transportation studies, Vol. 6, pp.2473-2486

[PYR05] Park, D. Yoo, K. Rhee, B. *Development of Location-based Dynamic Route guidance System of Korea Highway Cooperation*, 2005, Journal of the Eastern Asia Society for Transportation studies, Vol. 5, pp.1449-1463

[RN95] S.J. Russel and P. Norvig Artificial Intelligence A modern Approach. Prentice Hall, NJ, 1995, ISBN 0 13 103805 2

[S02] Warren Sarle *comp.ai.neural-nets FAQ, 2002*

ftp://ftp.sas.com/pub/neural/FAQ.html.zip or comp.ai.neural-nets accessed on 10/10/2005

[SPC03] I. Sherif, K. Prashanth, A. Ciprion, *Optimization of Dynamic Neural Networks Performance for short-term traffic prediction*, 2003, 82nd Transportation Research Board Meeting Revision, http://www.ltrc.lsu.edu/TRB_82/TRB2003-000349.pdf

[SSS00] Sahai, A K Soman M.K Satyan V *All Indian summer monsoon rainfall prediction using an artificial neural network*, 2000, Climate dynamic 16: 291- 302.

[TF93] Zaiyong Tang, Paul Fishwick *Feed-forward Neural Networks as models for time series forecasting*. - Technical Report TR91-008 Computer and Information Sciences, University of Florida, 1991, http://citeseer.ist.psu.edu/cache/papers/cs/1784/ftp:zSzzSzftp.cis.ufl.eduzSzcisz Sztech-reportszSztr91zSztr91-008.pdf/tang93feedforward.pdf accessed on 01/02/2006

[TLW05] Tsung-Hsien Tsai, Chi-Kong lee, Chiang-Hung WEI *Design of dynamic neural networks to forecast short-term railway passenger demand*, 2005 Journal of the Eastern Asia Society for Transport Studies Vol 6, pp 1651 – 1666, 2005

[TRL02] Transport Research Laboratory, *Scoping Study Urban mobility in three cities, Sub-Saharan Africa Transport Policy Programme*, 2002, The World Bank and Economic Commission for Africa. SSATP Working Paper No 70

[VT03] H.H. Versteegt, C. M. J.Tampere *Predict Time: State of the Art and functional Architecture*, 2003-07 , TNO Inro report , http://www.tno.nl accessed on 14/04/2006

[W01] Wasike S. K. W., *Road Infrastructure Policies in Kenya: Historical Trends and Current Challenges*, 2001, Infrastructure and Economics Service Division Kenya Institute of Public Policy Research and Analysis (KIPPRA) working paper no. 1 April 2001, http://www.kippra.or.ke accessed on 14/04/2006

[WTSW03] Wang, F.Y, Tang, S., Sui, Y. , Wang , X. *Toward intelligent Transport Systems for the 2008 Olympics*, 2003, IEEE Intelligent Systems, Nov-Dec 18(6), 2003, pp.8-11

[Y99] Yasdi R., *Prediction of Road Traffic using a Neutral Network approach, 1999,* Neutral computer Approach 8: 135 – 142

[YHYB04] F.Yong, S.Hongyu, T.Yong, R.Bin, 2004, *Temporal difference learning with recurrent neural network in multi-step ahead freeway speed prediction*, 2004, TRB annual Meeting # 04-4233

[YK00] You J., Kim TJ, *Development and evaluation of a hybrid travel time forecasting model, 2000,*. Transposition research, Vol 8C, No. 1-6, PP 231-256

[YK03] You J. Kim T.J., *Toward developing a travel time forecasting model for location-based services: A Review*" , 2003, ERSA conference papers ersa03p435, European Regional Science Association.

[YTPT98] Yao J., Teng N., Pon H., Ton C.L., *Forecasting and Analysis of Marketing data using Neural Networks*, 1998 Journal of Information Science and Engineering 14, 843-862, http://www.ersa.org/ersaconfs/ersa03/cdrom/papers/435.pdf accessed on 10/03/2006

[Z03] Zhang G-P, *Time series forecasting using a hybrid ARIMA and neural network model,* 2003, Neutral computing 50 pages (159-175)

[Z04] Zhang, G. P. *Business Forecasting with Artificial Neural Networks: An Overview,*" 2004, G. P. Zhang (ed.) Neural Networks in Business Forecasting. Hershey, PA: Idea Group Publishing, (2004): pp. 1-22.

https://www.idea-group.com/ downloads/excerpts/1591401763E.pdf

[ZCK03] L. Zhao, F. Collopy, M. Kennedy, *The problem of Neural Networks in Business Forecasting: An attempt to Reproduce the Hill, O'Connor and Remus Study*, 2003, Sprouts: Working Papers on Information Environments, Systems and Organizations Volume 3 Issue 4 Article 12 – 2003

[ZQ00] Zhang, P.G, Qi M *Seasonal and trend time series forecasting with neural networks*, 2000, Proceedings of the annual melting of the decision sciences institution Orlando, 963- 965

Bibliography

1 Dr. John A. Bullinaria *Introduction to Neural Networks 2nd Year UG, MSc in Computer Science*, http://www.cs.bham.ac.uk/~jxb/inn.html and http://www.cs.bham.ac.uk/~jxb accessed on 14/04/2006

2 Yu Zhanshon *Feed forward neural networks and their application in forecasting*, 2000, MSc Computer Science, University of Huston, 2000.

3 Wikipedia, the free encyclopedia, *Artifical Neural Networks* , http://en.wikipeida.org/wiki/artificial_nerual-network 29/9/2004 accessed on 14/04/2006

4 A* search algorithm, http://en.wikipedia.org.wiki/A%2a-search , accessed on 29/9/2005

5 Artificial Neural Networks – Wikipedia, the free encyclopedia, http://en.wikipeida.org/wiki/artificial_nerual-network accessed on 29/9/2005

6 Amit's Game Programming Site, 2003, Amit J. Patel, http://www-cs-students.stanford.edu/~amitp accessed on 29/9/2005

8 Prof. Leslie Smith, *An introduction to Neural Networks*, http://www.cs.stir.ac.uk/iss/NNintro/InvSlides.html accessed on 29/9/2005

9 Map representation, http://theory.stanford.edu/amitp/gameprogramming/maprepresentation.html accessed on 29/8/2005

10 Worren S. Sarle, SAS Institute , III – Conditioning in neural networks September 5, 1999

11 Google News Group, *Google Talk*, comp.ai.neural-net, http://groups.google.co.ke/group/comp/ai.neural-net/ accessed on 14/04/2006

12 Neural Networks, http://www.statsoft/cp./textbook accessed on 14/04/2006

13 D.R. Tveter, *Backpropagaters Review*, http://dontveter.com/opr/opr/html accessed on 29/9/2005

14 Hoang G. Vu, Lee J Hyun, *MLP – making MLP practical*, , http://ai.kaist.ac.kr/-jkim/cs570-2004/data/732 accessed on 29/9/2005

15 *Choice of ANN model*, http://www.thep.lu.se/public_html/jetnet_30manual/node14.html accessed on

15/12/2005

16 F. M. Christopher *Give your . NET App Brains and brown with the intelligence of Neural Networks*, 2005, http//msdn.microsoft.com/msdnmag/issues/os/os/neuralnetworks/default.aspx

17 Lo C P Yeung Albert K.W., *Concepts and techniques of Geographic Information Systems*, 2002, Prentice Hall, ISBN-81-203-2230-4

1 Appendix A: User Manual

Start up Screen

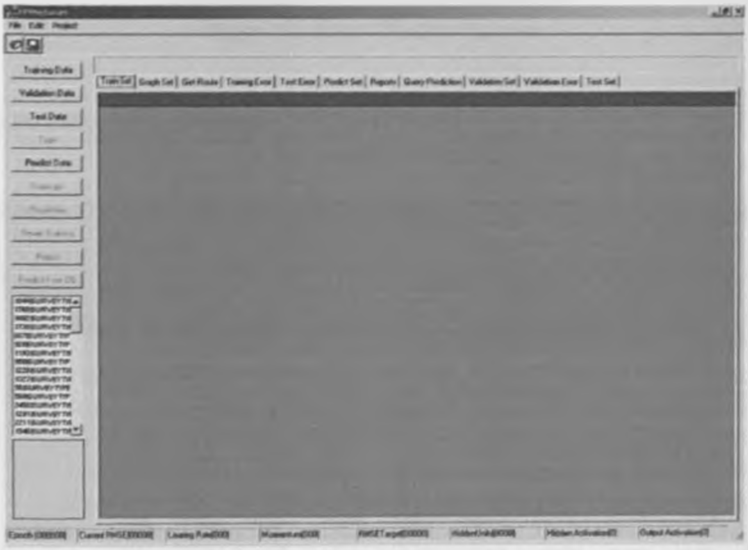


Figure 1-1 Start up screen of DSS

Loading Traffic Data

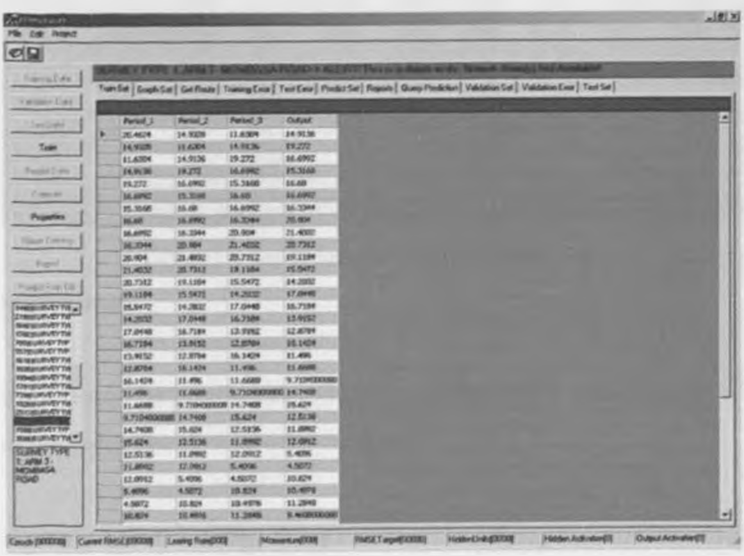


Figure 1-2 Data Formatted in time Series format ready fro training

Click any of the nodes listed to build create a neural network

Setup parameters

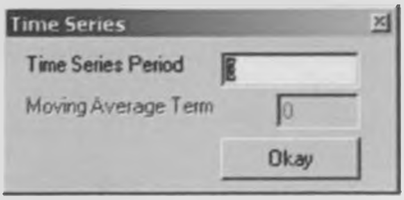


Figure 1-3 Select Window lag size

Select a suitable time series window

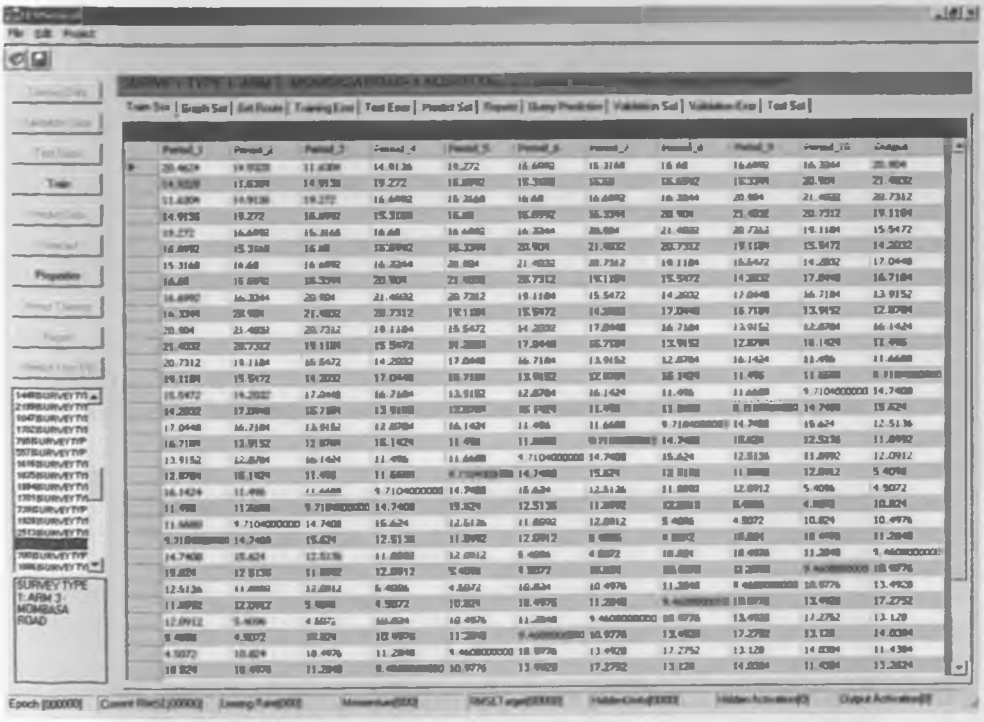


Figure 1-4 Input data based on varied lag size

Network parameters

Systematically identify the parameters to be changed iteratively while training with a view of getting an optimum neural network structure.

Step 1 is to create a network design. This involves identifying the number of hidden units and both hidden and output activation functions.

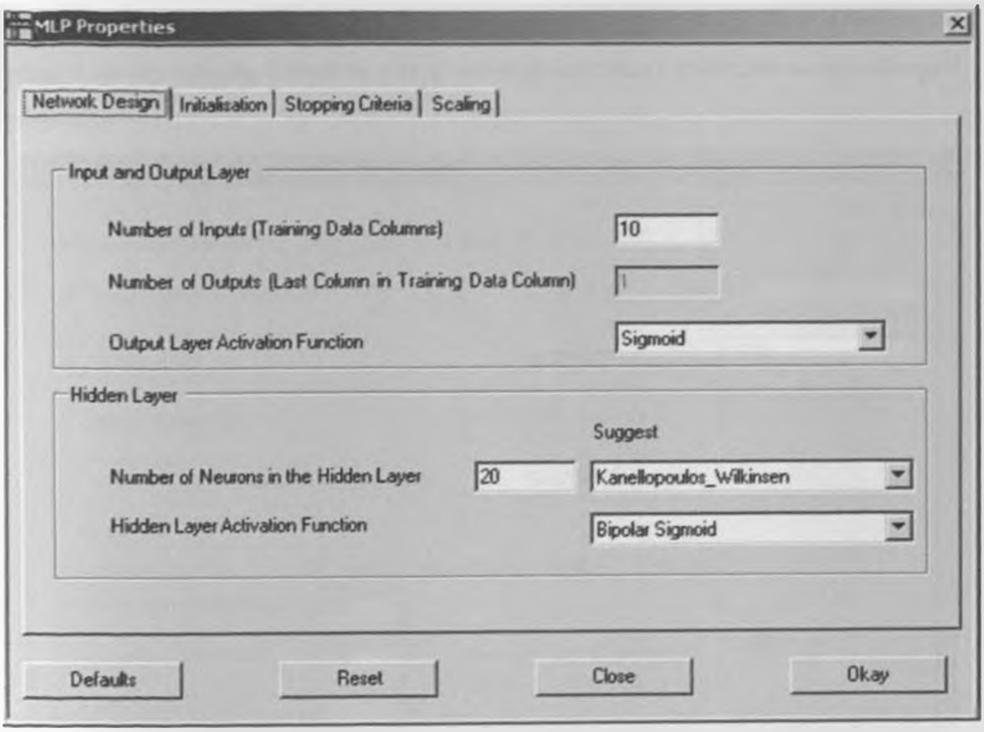


Figure 1-5 network Design Parameters

Step 2-update the learning rate and momentum iteratively as learning proceeds.

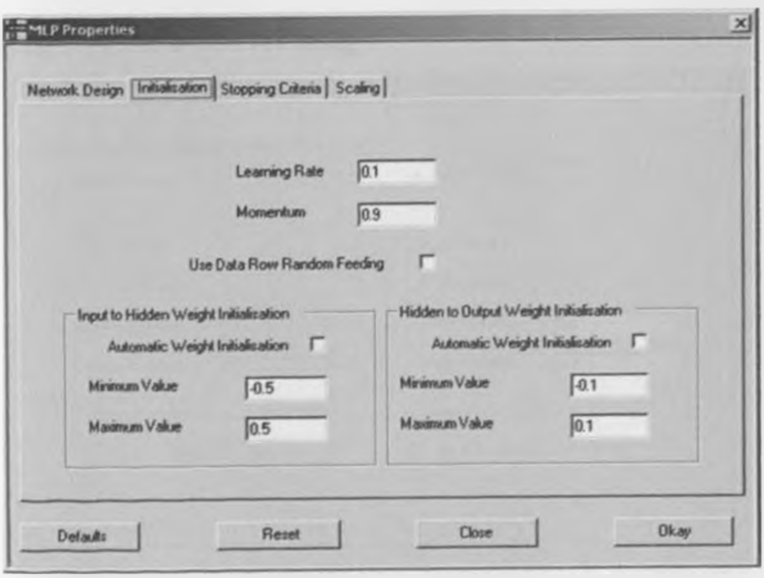


Figure 1-6 Network Initialization Parameters

The weight initialization parameters for both input and output weight do not need to be changed during training. However where convergence takes a while, the weight changed iteratively.

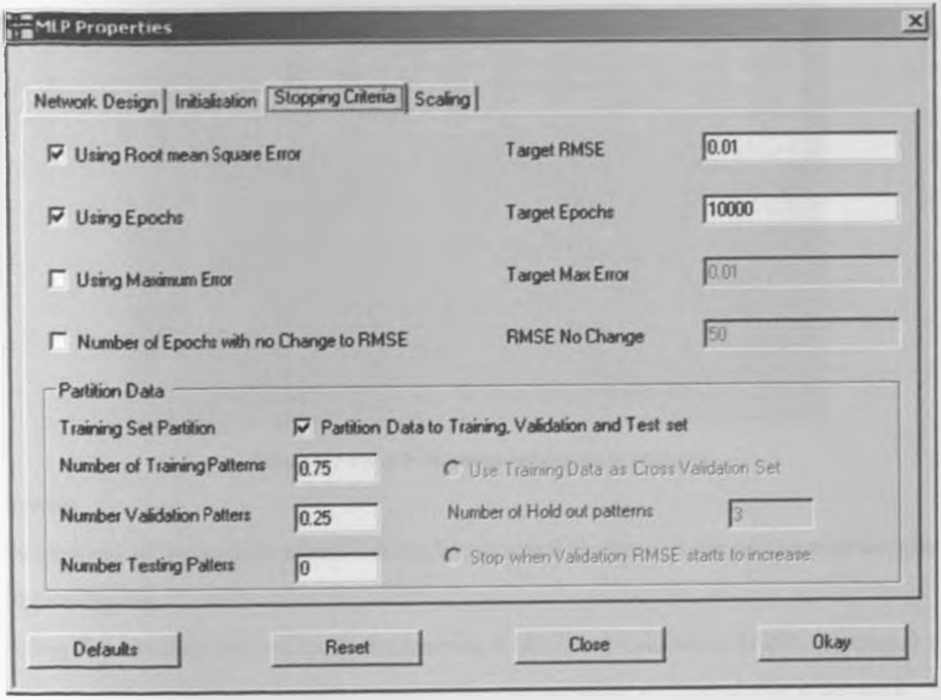


Figure 1-7 Neural Network stopping Criteria

Early stopping is used to avoid over fitting

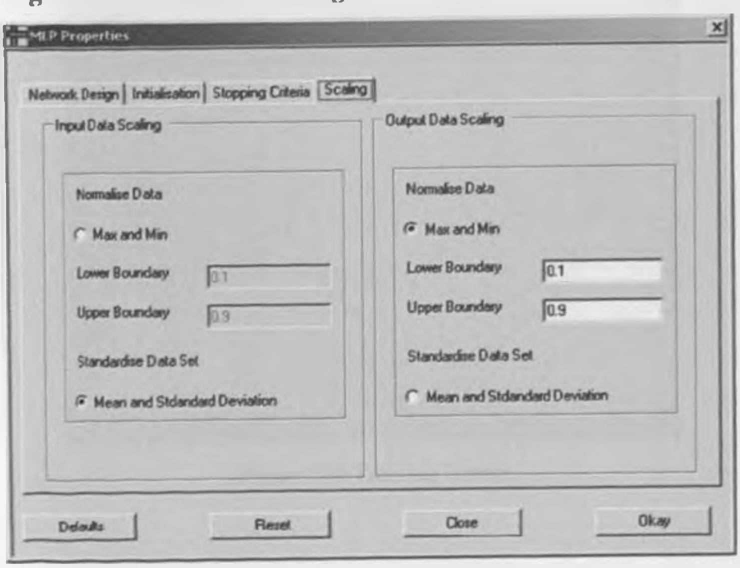


Figure 1-8 Input and Output Scaling Parameter

Validation Set

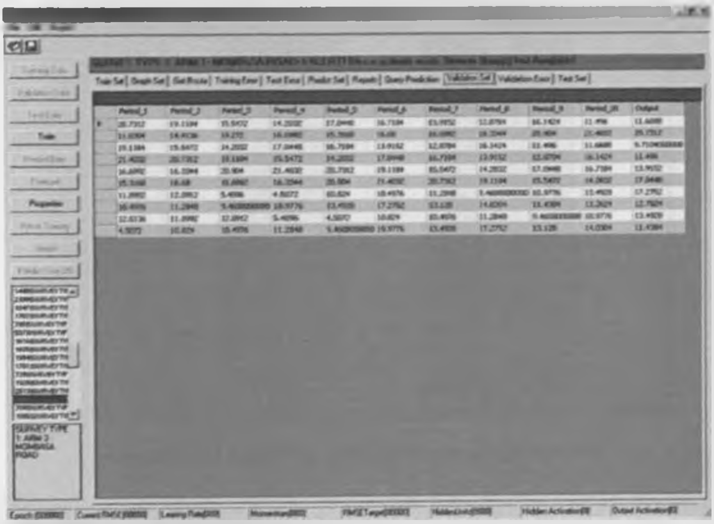


Figure 1-9 Post Validation setting data removal

Training

The process of training involves making incremental changes in the connection weights between layers.

During the training process, both the training RMSE and validation RMSE (optional) will be generated. A graph of the target versus the output value is dynamically generated giving the user a change to make a visual inspection of the progression of learning.

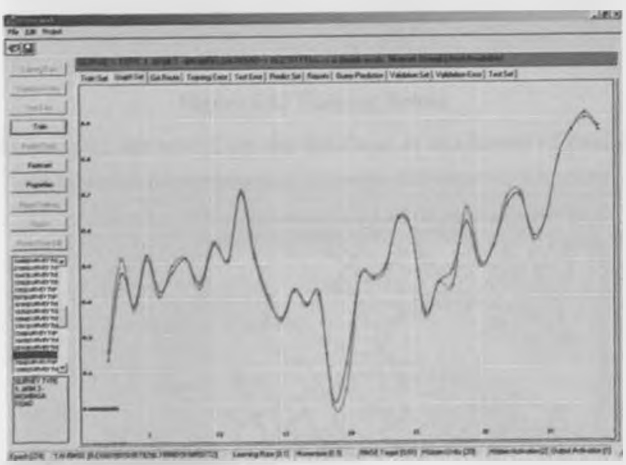


Figure 1-10 Training Result

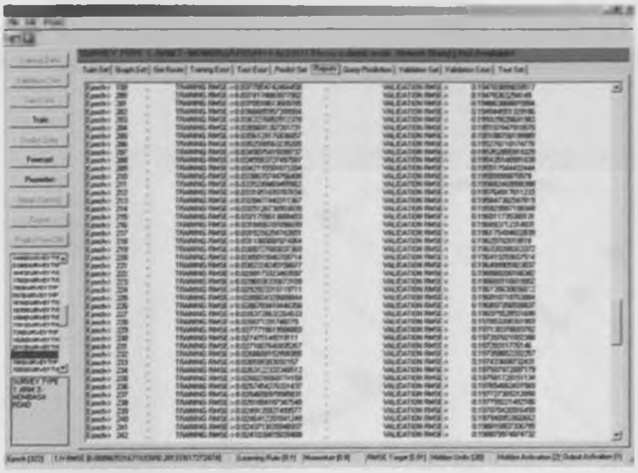


Figure 1-11 Training and validation RMSE results

Training Error

Figure 1-12 Training Errors

After training, the weights are saved on the database in readiness of prediction

Figure 1-13 Route engine Ready to Star Search

Route Search is used to determine the shortest path between any two points.

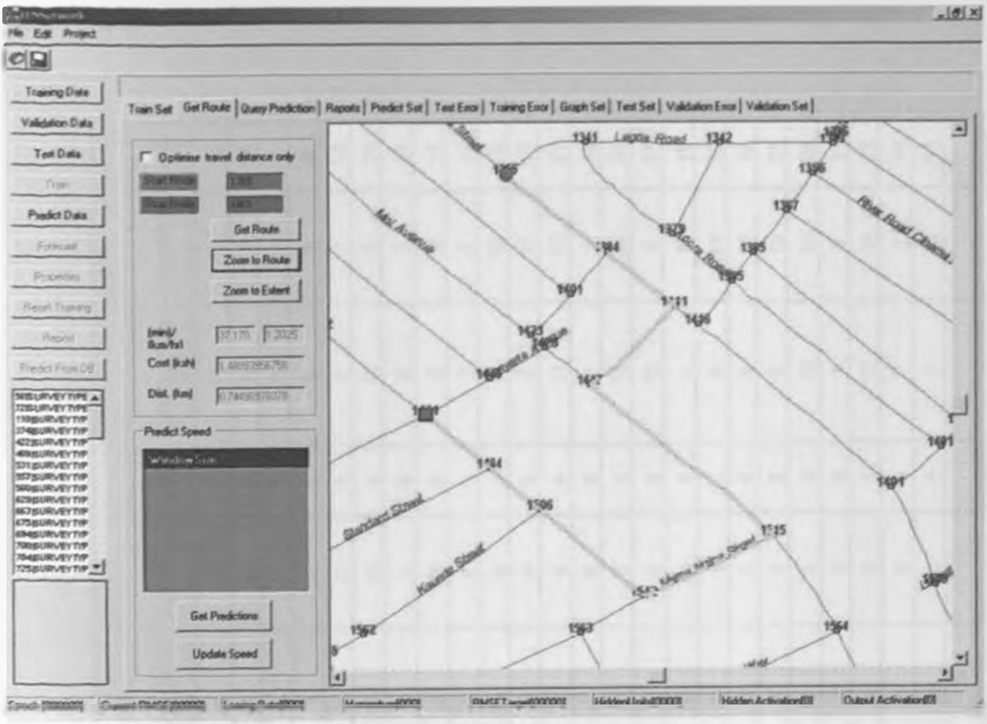


Figure 1-14 Route Selected after search

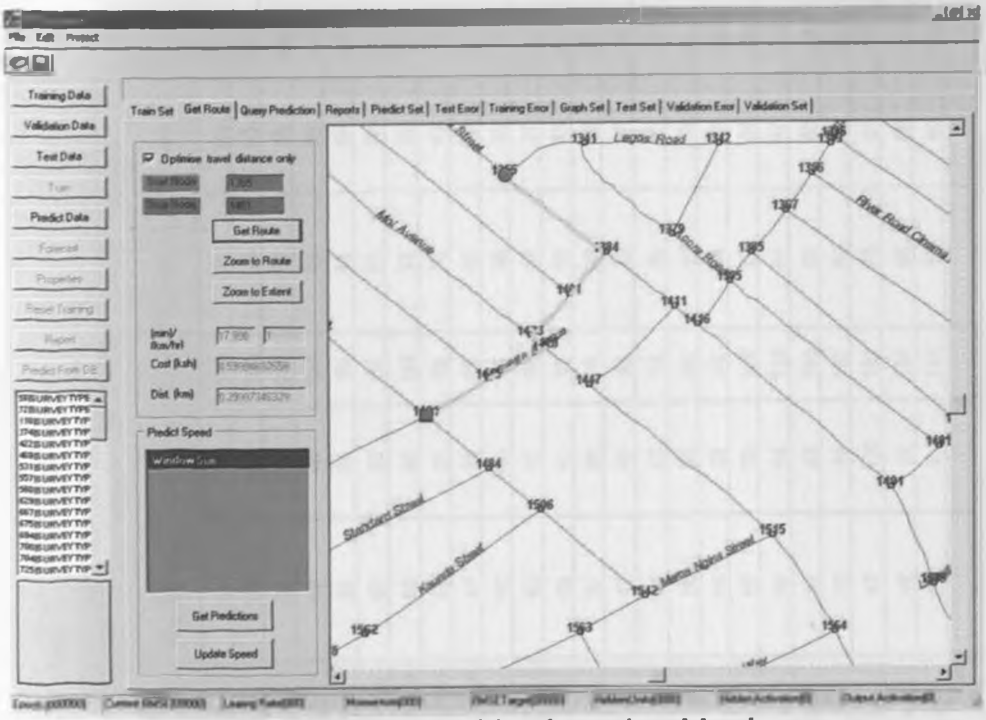


Figure 1-15 the same path based on static path length

APPENDIX B : Sample Data

STATION 1: RUAKA

Tuesday May 11, 2004

STATION 1: SURVEY TYPE 1: LIMURU ROAD

Period	Cars			Matatus			Buses			Lorries		
	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total
7:00 - 7:30AM	126	20	146	91	56	147	4	0	4	11	2	13
7:30 - 8:00AM	109	34	143	73	63	136	0	0	0	8	2	10
8:00 - 8:30 AM	97	34	131	57	46	103	2	0	2	3	4	7
8:30 - 9:00 AM	63	40	103	65	66	131	3	0	3	5	1	6
9:00 - 9:30 AM	55	34	89	65	84	149	1	0	1	6	7	13
9:30 - 10:00 AM	62	33	95	68	73	141	0	0	0	12	6	18
10:00 - 10:30 AM	60	65	125	52	58	110	0	0	0	8	7	15
10:30 - 11:00 AM	55	31	86	50	43	93	0	0	0	9	2	11
11:00 - 11:30 AM	47	36	83	40	78	118	1	0	1	7	7	14
11:30 - 12:00 AM	42	36	78	38	38	76	0	0	0	9	13	22
12:00 - 12:30 PM	48	46	94	47	29	76	0	0	0	5	12	17
12:30 - 1:00 PM	46	41	87	48	33	81	0	0	0	11	10	21
1:00 - 1:30 PM	37	58	95	28	29	57	0	0	0	6	9	15
1:30 - 2:00 PM	47	46	93	29	37	66	0	0	0	11	14	25
2:00 - 2:30 PM	44	50	94	46	41	87	0	0	0	12	5	17
2:30 - 3:00 PM	38	55	93	46	43	89	0	0	0	4	15	19
3:00 - 3:30 PM	46	39	85	48	35	83	0	1	1	9	14	23
3:30 - 4:00 PM	65	55	120	39	39	78	0	0	0	3	13	16
4:00 - 4:30 PM	48	64	112	44	47	91	0	2	2	5	12	17
4:30 - 5:00 PM	41	62	103	52	42	94	0	2	2	12	14	26
5:00 - 5:30 PM	43	94	137	60	63	123	0	0	0	7	4	11
5:30 - 6:00 PM	47	120	167	75	81	156	1	0	1	11	11	22
6:00 - 6:30 PM	42	92	134	68	59	127	0	0	0	7	7	14
6:30 - 7:00 PM	27	97	124	58	60	118	0	0	0	7	11	18
Day Total	1,335	1,282	2,617	1,287	1,243	2,530	12	5	17	188	202	390

STATION 2: KIAMBU

Wednesday May 12, 2004
SURVEY TYPE 1: KIAMBU ROAD

Period	Cars			Matatus			Buses			Lorries		
	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total
7:00 - 7:30AM	206	23	229	93	71	164	0	0	0	6	6	12
7:30 - 8:00AM	142	30	172	83	45	128	0	0	0	6	0	6
8:00 - 8:30 AM	106	66	172	74	77	151	0	0	0	4	4	8
8:30 - 9:00 AM	100	68	168	67	75	142	0	0	0	9	4	13
9:00 - 9:30 AM	92	69	161	66	65	131	0	0	0	5	8	13
9:30 - 10:00 AM	84	42	126	53	44	97	0	0	0	6	1	7
10:00 - 10:30 AM	83	50	133	52	52	104	0	0	0	8	10	18
10:30 - 11:00 AM	64	63	127	45	45	90	0	0	0	5	11	16
11:00 - 11:30 AM	76	54	130	45	47	92	0	0	0	9	8	17
11:30 - 12:00 AM	73	65	138	44	53	97	0	0	0	10	16	26
12:00 - 12:30 PM	68	82	150	55	57	112	0	0	0	4	9	13
12:30 - 1:00 PM	63	92	155	44	44	88	1	0	1	13	10	23
1:00 - 1:30 PM	87	77	164	55	66	121	0	1	1	15	17	32
1:30 - 2:00 PM	63	83	146	43	73	116	0	1	1	19	13	32
2:00 - 2:30 PM	98	68	166	62	62	124	2	0	2	7	9	16
2:30 - 3:00 PM	72	50	122	50	57	107	1	0	1	13	7	20
3:00 - 3:30 PM	64	61	125	70	56	126	0	0	0	11	9	20
3:30 - 4:00 PM	76	54	130	52	49	101	0	0	0	17	8	25
4:00 - 4:30 PM	50	64	114	53	50	103	0	0	0	9	14	23
4:30 - 5:00 PM	49	79	128	66	66	132	0	0	0	7	5	12
5:00 - 5:30 PM	52	148	200	56	74	130	0	0	0	7	8	15
5:30 - 6:00 PM	41	125	166	72	74	146	0	0	0	9	12	21
6:00 - 6:30 PM	31	130	161	82	75	157	0	1	1	7	7	14
6:30 - 7:00 PM	46	140	186	62	80	142	0	0	0	2	5	7
Day Total	**1,886**	**1,783**	**3,669**	**1,444**	**1,457**	**2,901**	**4**	**3**	**7**	**208**	**201**	[illegible]

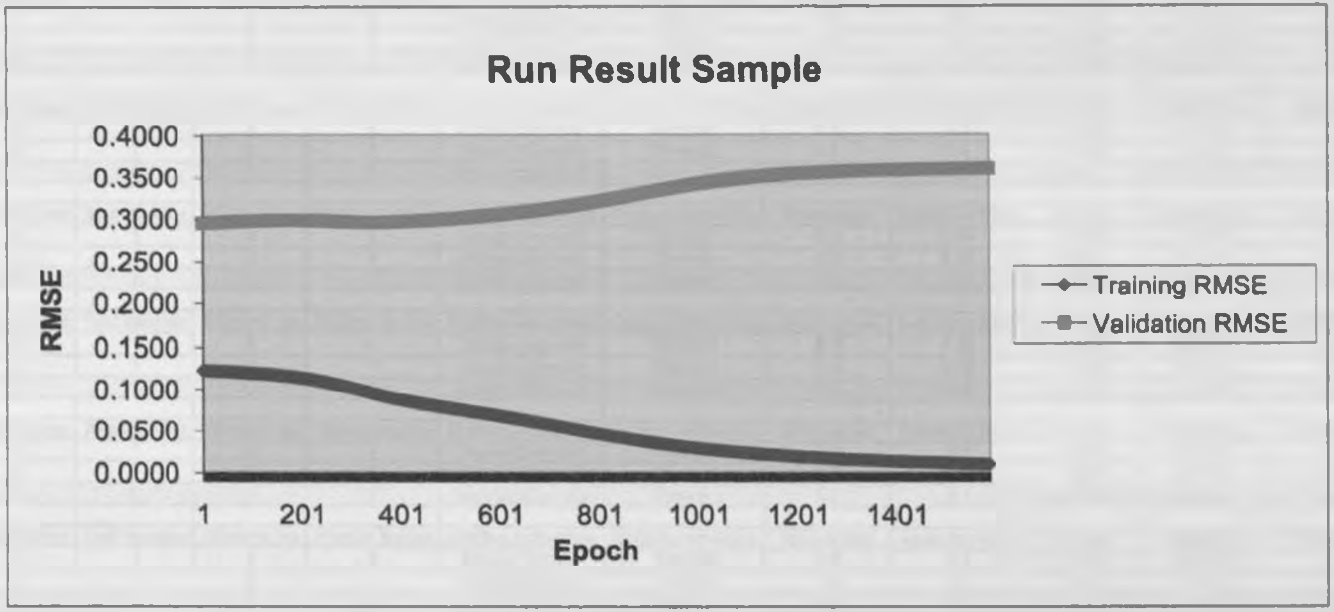
STATION 9: KOMMAROCK-OUTER RING

Wednesday May 12, 2004

SURVEY TYPE 1: JUJA ROAD

Period	Cars			Matatus			Buses			Lorries		
	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total	Approaching	Departing	Total
7:00 - 7:30AM	55	75	130	65	11	76	0	4	4	17	9	26
7:30 - 8:00AM	44	130	174	64	101	165	4	5	9	7	14	21
8:00 - 8:30 AM	30	76	106	47	80	127	1	11	12	5	11	16
8:30 - 9:00 AM	36	63	99	93	51	144	3	2	5	8	5	13
9:00 - 9:30 AM	57	55	112	109	56	165	2	8	10	10	12	22
9:30 - 10:00 AM	47	62	109	110	57	167	4	5	9	11	13	24
10:00 - 10:30 AM	41	46	87	83	52	135	5	2	7	12	8	20
10:30 - 11:00 AM	60	53	113	86	55	141	3	4	7	9	3	12
11:00 - 11:30 AM	64	55	119	79	56	135	3	6	9	13	6	19
11:30 - 12:00 AM	67	47	114	79	53	132	4	4	8	12	4	16
12:00 - 12:30 PM	65	55	120	80	53	133	2	5	7	17	13	30
12:30 - 1:00 PM	44	45	89	42	70	112	3	3	6	10	16	26
1:00 - 1:30 PM	67	70	137	73	31	104	4	2	6	21	4	25
1:30 - 2:00 PM	65	67	132	77	40	117	3	3	6	15	9	24
2:00 - 2:30 PM	63	52	115	65	45	110	2	4	6	17	4	21
2:30 - 3:00 PM	62	60	122	73	44	117	4	3	7	19	2	21
3:00 - 3:30 PM	68	43	111	87	53	140	4	4	8	17	6	23
3:30 - 4:00 PM	56	43	99	74	52	126	3	6	9	8	4	12
4:00 - 4:30 PM	58	50	108	73	42	115	3	1	4	10	3	13
4:30 - 5:00 PM	79	42	121	117	41	158	4	3	7	8	4	12
5:00 - 5:30 PM	114	36	150	116	62	178	11	2	13	12	6	18
5:30 - 6:00 PM	119	42	161	117	81	198	6	10	16	13	6	19
6:00 - 6:30 PM	113	36	149	124	82	206	8	9	17	14	15	29
6:30 - 7:00 PM	69	35	104	123	87	210	6	6	12	5	14	19
Day Total	1,543	1,338	2,881	2,056	1,355	3,411	92	112	204	290	191	481

Structure	
Lag	5
Epoch	10000
Momentum	0.9
Learning Rate	0.08
Hidden	7
Hidden Activation	Bipolar sigmoid
Output Activation	Sigmoid



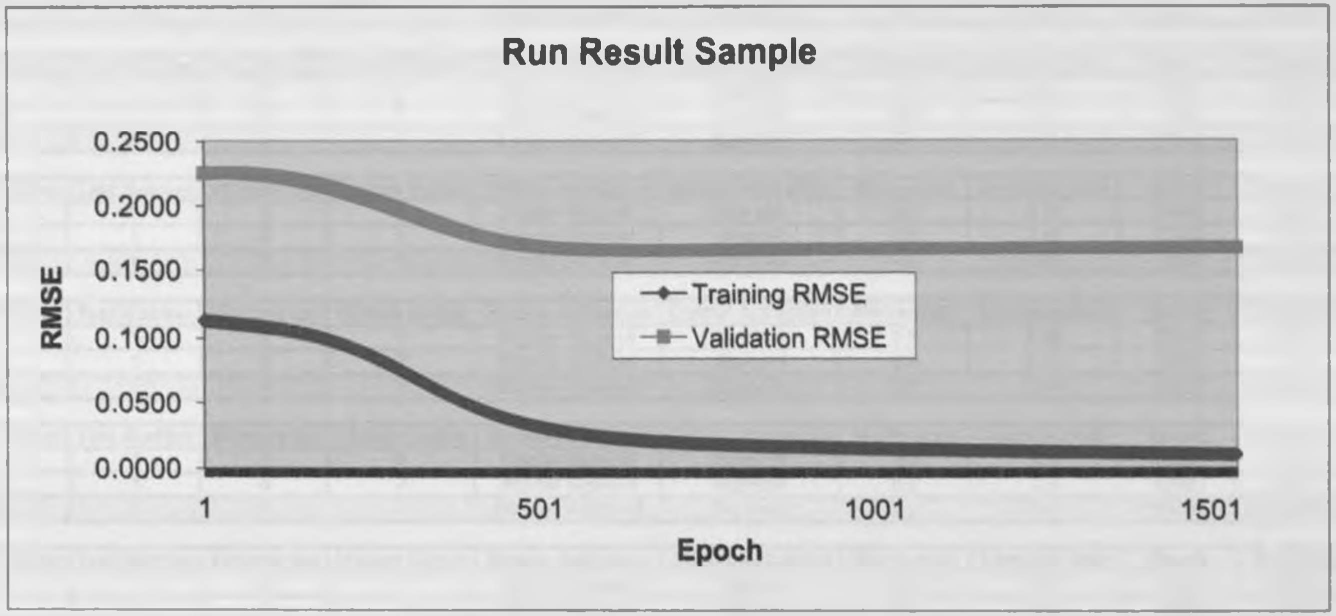
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	(1..10)		(3,5,7,9)	(3,5,7,9)	(Bipolar Sigmoid, Sigmoid, Tanh)	(Bipolar Sigmoid, Sigmoid, Tanh)	(0.9, 0.7, 0.5, 0.3, 0.1)	(0.08, 0.1, 0.5)	(Min 1000 Max 10000)	(Stop Epoch)	(0.01)
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2280	0.0812
1944	2	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2282	0.0669
1944	***2***	***1***	***3***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2281***	***0.07405***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2275	0.0668
1944	4	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2200	0.0693
1944	***2***	***1***	***3***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.22375***	***0.06807***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2473	0.0795
1944	6	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2106	0.0727
1944	***2***	***1***	***3***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.22895***	***0.0761***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	9	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2244	0.0350
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2247	0.0350
1944	***2***	***1***	***3***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.22455***	***0.035***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2304	0.0423
1944	11	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2332	0.0429
1944	***2***	***1***	***3***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2318***	***0.0426***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2243	0.0334
1944	13	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2482	0.0341
1944	***2***	***1***	***3***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.23625***	***0.03375***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2285	0.0211
1944	15	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2116	0.0373
1944	***2***	***1***	***3***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.22005***	***0.0292***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2687	0.0104
1944	17	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2650	0.0152
1944	***2***	***1***	***3***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.26685***	***0.0128***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2601	0.0181
1944	19	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1952	0.0243
1944	***2***	***1***	***3***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.23765***	***0.0212***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	1	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2335	0.0255
1944	2	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2236	0.0290
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.22855***	***0.02725***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	3	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3454	0.0500
1944	4	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2133	0.0270
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.27935***	***0.0385***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	5	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2110	0.0477
1944	6	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3201	0.0598
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.26555***	***0.05375***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	7	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1437	0.0100
1944	8	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2445	0.0100
1944	***2***	***1***	***5***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.1941***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	10	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2099	0.0100
1944	11	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2037	0.0100
1944	***2***	***1***	***5***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2068***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	12	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1859	0.0100
1944	13	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2039	0.0100
1944	***2***	***1***	***5***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.1949***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	14	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2000	0.0100
1944	15	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2236	0.0100
1944	***2***	***1***	***5***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2118***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	16	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1755	0.0100
1944	17	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1995	0.0100
1944	***2***	***1***	***5***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.1875***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	18	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2260	0.0100
1944	19	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1445	0.0100
1944	***2***	***1***	***5***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.18525***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3305	0.0100
1944	2	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3960	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.36325***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3193	0.0100
1944	4	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3245	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3219***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2949	0.2002
1944	6	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2874	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.29115***	***0.1051***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3736	0.0100
1944	8	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3918	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3827***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3848	0.0100
1944	11	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3360	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3604***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3455	0.0100
1944	13	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3231	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3343***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3566	0.0100
1944	15	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3273	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.34195***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3803	0.0100
1944	17	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3690	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.37465***	***0.01***
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3539	0.0100
1944	19	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3841	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.369***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolar Sigmoid, Sigmoid, Tanh}	{Bipolar Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2210	0.0100
1944	2	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2167	0.0100
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.21885*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2273	0.0100
1944	4	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2139	0.0100
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2206*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2246	0.0100
1944	6	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2221	0.0100
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.22335*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2212	0.0100
1944	8	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2611	0.0100
1944	*2*	*1*	*9*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.24115*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2016	0.0100
1944	11	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2030	0.0100
1944	*2*	*1*	*9*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2023*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2338	0.0100
1944	13	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2485	0.0100
1944	*2*	*1*	*9*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.24115*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2720	0.0100
1944	15	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2173	0.0100
1944	*2*	*1*	*9*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.24465*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3001	0.0100
1944	17	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3240	0.0100
1944	*2*	*1*	*9*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.31205*	*0.01*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2385	0.0100
1944	19	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2244	0.0100
1944	*2*	*1*	*9*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.23145*	*0.01*

Structure	
Lag	5
Epoch	10000
Momentum	0.9
Learning Rate	0.1
Hidden	5
Hidden Activation	Bipolar sigmoid
Output Activation	Sigmoid



NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	(1..10)		(3,5,7,9)	(3,5,7,9)	(Bipolar Sigmoid, Sigmoid, Tanh)	(Bipolar Sigmoid, Sigmoid, Tanh)	(0.9, 0.7, 0.5, 0.3, 0.1)	(0.08, 0.1, 0.5)	(Min 1000 Max 10000)	(Stop Epoch)	(0.01)

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2150	0.0900
1944	2	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2151	0.0900
1944	*2*	*1*	*3*	*3*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.08*	*10000*	*0.21505*	*0.09*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2147	0.0900
1944	4	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2147	0.0900
1944	*2*	*1*	*3*	*3*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.1*	*10000*	*0.2147*	*0.09*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2100	0.0992
1944	6	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2476	0.1052
1944	*2*	*1*	*3*	*3*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.5*	*10000*	*0.2288*	*0.1022*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	9	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3101	0.0568
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3100	0.0569
1944	*2*	*1*	*3*	*5*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.1*	*10000*	*0.31005*	*0.05685*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3064	0.0502
1944	11	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3106	0.0580
1944	*2*	*1*	*3*	*5*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.08*	*10000*	*0.3085*	*0.0541*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3297	0.0332
1944	13	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3200	0.0416
1944	*2*	*1*	*3*	*7*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.08*	*10000*	*0.32485*	*0.0374*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3095	0.0558
1944	15	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2975	0.0432
1944	*2*	*1*	*3*	*7*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.1*	*10000*	*0.3035*	*0.0495*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3451	0.0369
1944	17	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3485	0.0368
1944	*2*	*1*	*3*	*9*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.1*	*10000*	*0.3468*	*0.03685*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3160	0.0567
1944	19	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3218	0.0362
1944	*2*	*1*	*3*	*9*	***Bipolar Sigmoid***	***Sigmoid***	*0.9*	*0.08*	*10000*	*0.3189*	*0.04645*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2243	0.0115
1944	2	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2116	0.0184
1944	**2**	***1***	**5**	**3**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.08**	**10000**	**0.21795**	**0.01495**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3049	0.0127
1944	4	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1885	0.0101
1944	**2**	***1***	**5**	**3**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.1**	**10000**	**0.2467**	**0.0114**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2248	0.0150
1944	6	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.1937	0.0329
1944	**2**	***1***	**5**	**3**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.5**	**10000**	**0.20925**	**0.02395**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2108	0.0100
1944	8	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1674	0.0100
1944	**2**	***1***	**5**	**5**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.1**	**10000**	**0.1891**	**0.01**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1564	0.0100
1944	11	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1649	0.0100
1944	**2**	***1***	**5**	**5**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.08**	**10000**	**0.16065**	**0.01**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1469	0.0100
1944	13	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1595	0.0100
1944	**2**	***1***	**5**	**7**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.08**	**10000**	**0.1532**	**0.01**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1686	0.0100
1944	15	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1715	0.0100
1944	**2**	***1***	**5**	**7**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.1**	**10000**	**0.17005**	**0.01**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1500	0.0100
1944	17	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1544	0.0100
1944	**2**	***1***	**5**	**9**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.1**	**10000**	**0.1522**	**0.01**
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1442	0.0100
1944	19	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1578	0.0100
1944	**2**	***1***	**5**	**9**	***Bipolar Sigmoid***	***Sigmoid***	**0.9**	**0.08**	**10000**	**0.151**	**0.01**

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	(1. 10)		(3,5,7,9)	(3,5,7,9)	(Bipolar Sigmoid, Sigmoid, Tanh)	(Bipolar Sigmoid, Sigmoid, Tanh)	(0.9, 0.7, 0.5, 0.3, 0.1)	(0.08, 0.1, 0.5)	(Min 1000 Max 10000)	(Stop Epoch)	(0.01)
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	1	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3978	0.0100
1944	2	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2888	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3433***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	3	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3838	0.0100
1944	4	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2699	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.32685***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	5	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3942	0.0100
1944	6	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3821	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.38815***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	7	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2796	0.0100
1944	8	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3413	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.31045***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	10	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3020	0.0100
1944	11	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2900	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.296***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	12	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2921	0.0100
1944	13	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3125	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3023***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	14	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3587	0.0100
1944	15	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2995	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3291***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	16	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3341	0.0100
1944	17	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3667	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3504***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	18	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3130	0.0100
1944	19	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3550	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.334***	***0.01***

Node Result 560

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2929	0.0100
1944	2	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2976	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.29525***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2505	0.0100
1944	4	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3333	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2919***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3234	0.0100
1944	6	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3177	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.32055***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3408	0.0100
1944	8	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3454	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3431***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3075	0.0100
1944	11	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3695	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3385***	***0.01***

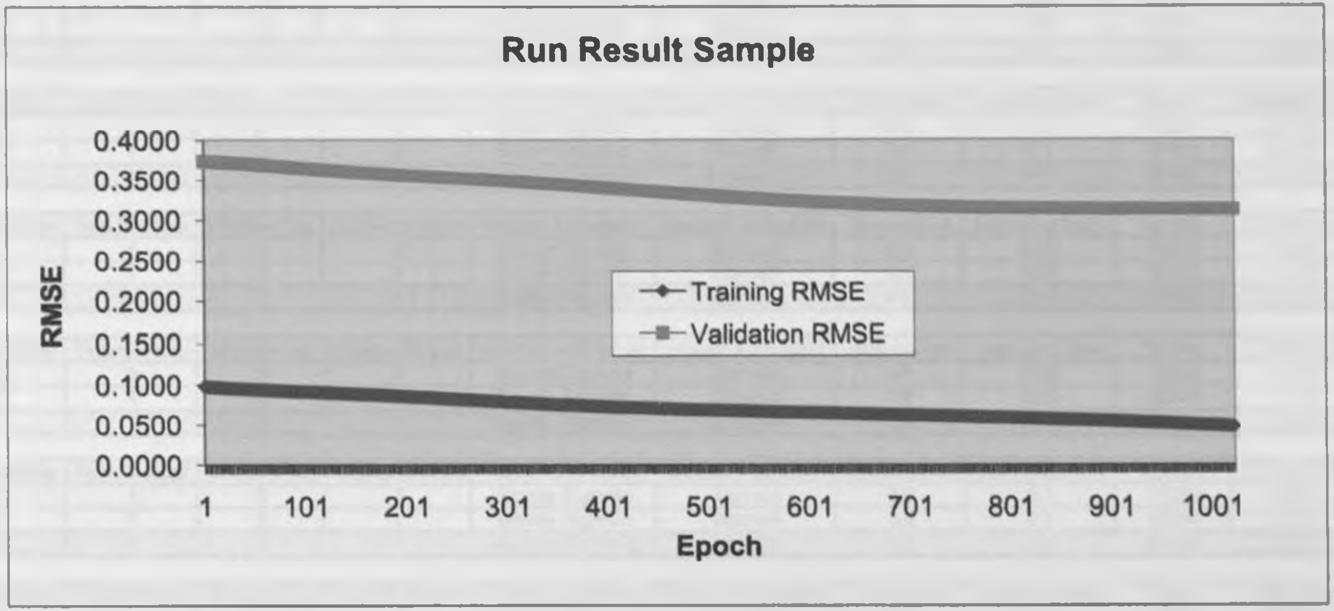
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3044	0.0100
1944	13	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2954	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2999***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3327	0.0100
1944	15	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3050	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.31885***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3533	0.0100
1944	17	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3053	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3293***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2902	0.0100
1944	19	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3561	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.32315***	***0.01***

Structure	
Lag	5
Epoch	10000
Momentum	0.9
Learning Rate	0.08
Hidden	9
Hidden Activation	Bipolar sigmoid
Output Activation	Sigmoid



NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network }	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2989	0.0891
1944	2	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3702	0.0661
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.33455*	*0.0776*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2996	0.0893
1944	4	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3164	0.0901
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.308*	*0.0897*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3737	0.0681
1944	6	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3621	0.0684
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.3679*	*0.06825*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	9	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3148	0.0466
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2931	0.0565
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.30395*	*0.05155*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2809	0.0558
1944	11	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2679	0.0613
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2744*	*0.05856*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2733	0.0578
1944	13	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3198	0.0415
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.29655*	*0.049665*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2915	0.0545
1944	15	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3047	0.0517
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2981*	*0.0531*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3315	0.0354
1944	17	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3026	0.0495
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.31705*	*0.04245*
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2741	0.0587
1944	19	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2960	0.0537
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.28505*	*0.0562*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	1	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2871	0.0169
1944	2	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2865	0.0178
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2868***	***0.01735***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	3	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2880	0.0161
1944	4	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2878	0.0163
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2879***	***0.0162***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	5	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3500	0.0224
1944	6	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.4360	0.0135
1944	***2***	***1***	***5***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.393***	***0.01795***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	7	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2888	0.0100
1944	8	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2858	0.0100
1944	***2***	***1***	***5***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2873***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	10	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3529	0.0100
1944	11	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3261	0.0100
1944	***2***	***1***	***5***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3395***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	12	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2866	0.0100
1944	13	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3574	0.0100
1944	***2***	***1***	***5***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.322***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	14	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3007	0.0100
1944	15	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2939	0.0100
1944	***2***	***1***	***5***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2973***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	16	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3025	0.0100
1944	17	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3517	0.0100
1944	***2***	***1***	***5***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3271***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	18	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2616	0.0100
1944	19	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3157	0.0100
1944	***2***	***1***	***5***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.28865***	***0.01***

HIDDEN-MOMENTUM-LR-7

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2618	0.0100
1944	2	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1716	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2167***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2587	0.0100
1944	4	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3087	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2837***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2682	0.0100
1944	6	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.1762	0.0100
1944	***2***	***1***	***7***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.2222***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3625	0.0100
1944	8	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2327	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2976***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2107	0.0100
1944	11	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2623	0.0100
1944	***2***	***1***	***7***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2365***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3135	0.0100
1944	13	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2415	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2775***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3421	0.0100
1944	15	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3417	0.0100
1944	***2***	***1***	***7***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.3419***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3616	0.0100
1944	17	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3864	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.374***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2674	0.0100
1944	19	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3065	0.0100
1944	***2***	***1***	***7***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.28695***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4208	0.0100
1944	2	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3540	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.3874***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3600	0.0100
1944	4	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.5163	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.43815***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.4965	0.0100
1944	6	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.4971	0.0100
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.4968***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.5220	0.0100
1944	8	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.5186	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.5203***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4703	0.0100
1944	11	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4763	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.4733***	***0.01***

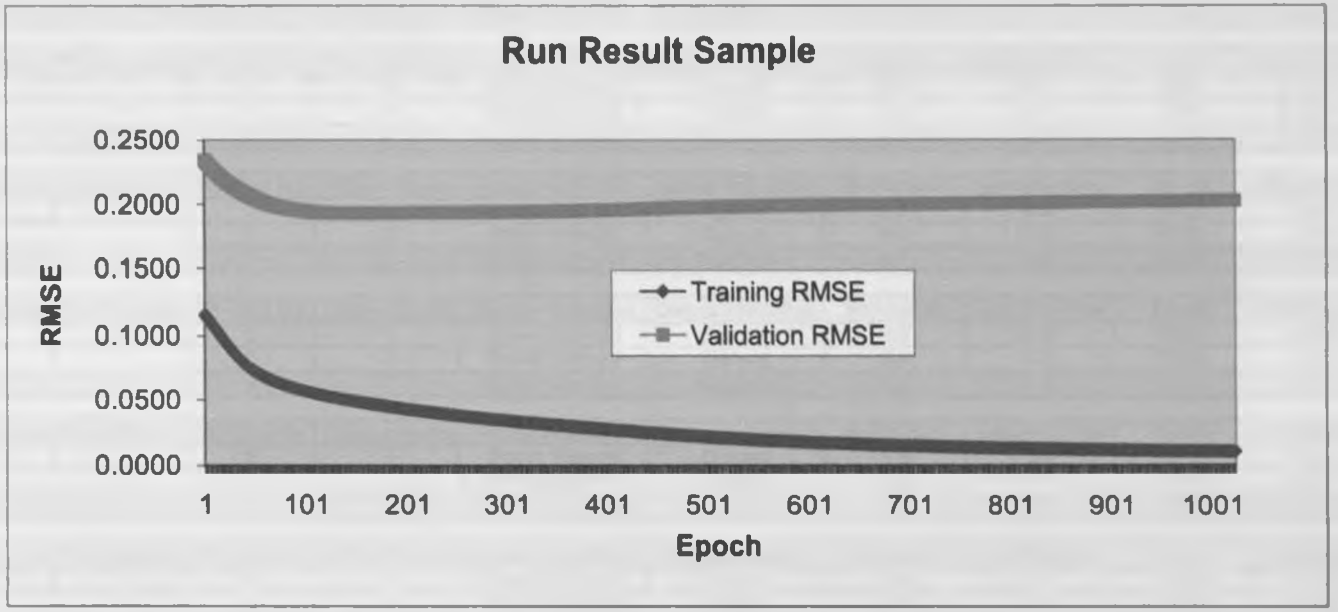
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4756	0.0100
1944	13	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4987	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.48715***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3862	0.0100
1944	15	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3931	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.38965***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3830	0.0100
1944	17	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.4707	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.42685***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4930	0.0100
1944	19	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.4748	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.4839***	***0.01***

Structure	
Lag	7
Epoch	10000
Momentum	0.9
Learning Rate	0.08
Hidden	9
Hidden Activation	Bipolar sigmoid
Output Activation	Sigmoid



NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	(1..10)		{3,5,7,9}	{3,5,7,9}	{Bipolar Sigmoid, Sigmoid, Tanh}	{Bipolar Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2351	0.1636
1944	2	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2442	0.1565
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.23965*	*0.16005*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2472	0.1573
1944	4	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2471	0.1573
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.24715*	*0.1573*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2535	0.1955
1944	6	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2535	0.1955
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	[illegible]	*0.1955*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	9	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2122	0.1204
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2326	0.1348
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2224*	[illegible]

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2207	0.1236
1944	11	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2316	0.1227
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.22615*	*0.12315*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2095	0.0971
1944	13	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1953	0.0967
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	[illegible]	*0.0969*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2044	0.0875
1944	15	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2167	0.0982
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	[illegible]	[illegible]

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1708	0.0602
1944	17	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1969	0.0771
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.18385*	*0.06865*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1640	0.0904
1944	19	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1818	0.0936
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.1729*	*0.092*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2805	0.0977
1944	2	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2805	0.0977
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2805*	*0.0977*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2807	0.0984
1944	4	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2807	0.0984
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2807*	*0.0984*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3277	0.3069
1944	6	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.4031	0.2204
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.3654*	*0.26365*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3677	0.0726
1944	8	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2723	0.0770
1944	*2*	*1*	*5*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.32*	*0.0748*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3190	0.0550
1944	11	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3389	0.0419
1944	*2*	*1*	*5*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.32895*	*0.04845*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3208	0.0101
1944	13	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3843	0.0164
1944	*2*	*1*	*5*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.35255*	*0.01325*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3311	0.0270
1944	15	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.3758	0.0110
1944	*2*	*1*	*5*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.35345*	*0.019*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2921	0.0100
1944	17	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2984	0.0100
1944	*2*	*1*	*5*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.29525*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2700	0.0100
1944	19	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.3099	0.0100
1944	*2*	*1*	*5*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.28995*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolar Sigmoid, Sigmoid, Tanh}	{Bipolar Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2508	0.0690
1944	2	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2143	0.0538
1944	*2*	*1*	*7*	*3*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.23255***	***0.0614***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2301	0.0597
1944	4	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2151	0.0537
1944	*2*	*1*	*7*	*3*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2226***	***0.0567***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3642	0.3499
1944	6	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3868	0.3517
1944	*2*	*1*	*7*	*3*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.3755***	***0.3508***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2331	0.0100
1944	8	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1817	0.0118
1944	*2*	*1*	*7*	*5*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2074***	***0.0109***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2137	0.0126
1944	11	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2229	0.0100
1944	*2*	*1*	*7*	*5*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2183***	***0.0113***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2074	0.0100
1944	13	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1950	0.0100
1944	*2*	*1*	*7*	*7*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2012***	***0.01***

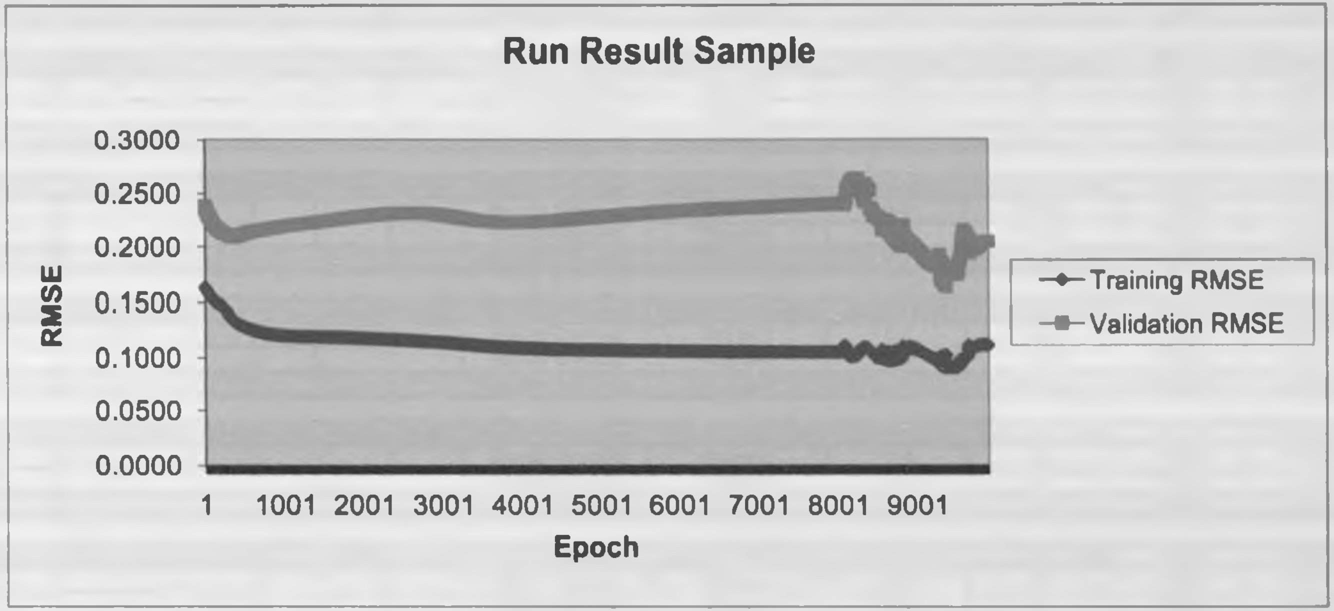
NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2258	0.0100
1944	15	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2100	0.0100
1944	*2*	*1*	*7*	*7*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2179***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1998	0.0100
1944	17	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1763	0.0100
1944	*2*	*1*	*7*	*9*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.18805***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2062	0.0100
1944	19	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2021	0.0100
1944	*2*	*1*	*7*	*9*	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.20415***	***0.01***

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	1	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2083	0.0341
1944	2	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1764	0.0493
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.19235***	***0.0417***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	3	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2036	0.0380
1944	4	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2562	0.0351
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.2299***	***0.03655***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	5	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3360	0.3523
1944	6	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.1321	0.0700
1944	***2***	***1***	***9***	***3***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.5***	***10000***	***0.23405***	***0.21115***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	7	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2319	0.0100
1944	8	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2476	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.23975***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	10	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1754	0.0100
1944	11	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2418	0.0100
1944	***2***	***1***	***9***	***5***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.2086***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	12	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1976	0.0100
1944	13	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1604	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.179***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	14	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2009	0.0100
1944	15	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.0787	0.0100
1944	***2***	***1***	***9***	***7***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.1398***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	16	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1160	0.0100
1944	17	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1228	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.1***	***10000***	***0.1194***	***0.01***
NodeID	**Test Number**	**Test Number**	**Window lag**	**Hidden Nodes**	**Hidden Activation**	**Output Activation**	**Momentum**	**Learning Rate**	**Epoch**	**Validation**	**RMSE Value**
1944	18	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1747	0.0100
1944	19	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1571	0.0100
1944	***2***	***1***	***9***	***9***	***Bipolar Sigmoid***	***Sigmoid***	***0.9***	***0.08***	***10000***	***0.1659***	***0.01***

Structure	
Lag	3
Epoch	10000
Momentum	0.9
Learning Rate	0.1
Hidden	7
Hidden Activation	Bipolar sigmoid
Output Activation	Sigmoid



NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1920	0.1570
1944	2	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1920	0.1570
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.192*	*0.157*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1910	0.1570
1944	4	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1910	0.1570
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.191*	*0.157*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2630	0.1960
1944	6	1	3	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2364	0.1956
1944	*2*	*1*	*3*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.2497*	*0.1958*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	9	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2439	0.1221
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2481	0.1159
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.246*	*0.119*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1886	0.1159
1944	11	1	3	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2189	0.1078
1944	*2*	*1*	*3*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.20375*	*0.11185*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2074	0.1032
1944	13	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2078	0.1033
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2076*	*0.10325*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2539	0.1014
1944	15	1	3	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2046	0.1109
1944	*2*	*1*	*3*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.22925*	*0.10615*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2771	0.0772
1944	17	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2783	0.0772
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2777*	*0.0772*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2681	0.0808
1944	19	1	3	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1762	0.0874
1944	*2*	*1*	*3*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.22215*	*0.0841*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
{Network}	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2142	0.0977
1944	2	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2142	0.0977
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2142*	*0.0977*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2152	0.0984
1944	4	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2152	0.0984
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2152*	*0.0984*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2885	0.3326
1944	6	1	5	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2988	0.2165
1944	*2*	*1*	*5*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.29365*	*0.27455*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2678	0.0737
1944	8	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2168	0.0724
1944	*2*	*1*	*5*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2423*	*0.07305*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2153	0.0731
1944	11	1	5	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2185	0.0603
1944	*2*	*1*	*5*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2169*	*0.0667*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2455	0.0275
1944	13	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2316	0.0272
1944	*2*	*1*	*5*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.23855*	*0.02735*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2500	0.0266
1944	15	1	5	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2878	0.0250
1944	*2*	*1*	*5*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2689*	*0.0258*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2253	0.0100
1944	17	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2126	0.0100
1944	*2*	*1*	*5*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.21895*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1959	0.0100
1944	19	1	5	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2770	0.0100
1944	*2*	*1*	*5*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.23645*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	(1..10)		(3,5,7,9)	(3,5,7,9)	(Bipolor Sigmoid, Sigmoid, Tanh)	(Bipolor Sigmoid, Sigmoid, Tanh)	(0.9, 0.7, 0.5, 0.3, 0.1)	(0.08, 0.1, 0.5)	(Min 1000 Max 10000)	(Stop Epoch)	(0.01)

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1215	0.0538
1944	2	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1442	0.0603
1944	*2*	*1*	*7*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	[illegible]	*0.13285*	*0.05705*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2017	0.0643
1944	4	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1012	0.0670
1944	*2*	*1*	*7*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.15145*	*0.06565*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.1962	0.2081
1944	6	1	7	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.2434	0.3240
1944	*2*	*1*	*7*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.21*[illegible]	*0.26605*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1664	0.0100
1944	8	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1855	0.0100
1944	*2*	*1*	*7*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.17595*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.0834	0.0100
1944	11	1	7	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.0845	0.0100
1944	*2*	*1*	*7*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.08395*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.0768	0.0100
1944	13	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1210	0.0100
1944	*2*	*1*	*7*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.0989*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.0917	0.0100
1944	15	1	7	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.0812	0.0100
1944	*2*	*1*	*7*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.08645*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.0966	0.0100
1944	17	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1434	0.0100
1944	*2*	*1*	*7*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.12*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.0987	0.0100
1944	19	1	7	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1230	0.0100
1944	*2*	*1*	*7*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.11*[illegible]	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
(Network)	{1..10}		{3,5,7,9}	{3,5,7,9}	{Bipolor Sigmoid, Sigmoid, Tanh}	{Bipolor Sigmoid, Sigmoid, Tanh}	{0.9, 0.7, 0.5, 0.3, 0.1}	{0.08, 0.1, 0.5}	{Min 1000 Max 10000}	{Stop Epoch}	{0.01}

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	1	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2692	0.0311
1944	2	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2296	0.0371
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2494*	*0.0341*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	3	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1017	0.0639
1944	4	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2239	0.0486
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.1628*	*0.05625*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	5	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.4151	0.2000
1944	6	1	9	3	Bipolar Sigmoid	Sigmoid	0.9	0.5	10000	0.3062	0.0756
1944	*2*	*1*	*9*	*3*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.5*	*10000*	*0.36065*	*0.1378*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	7	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2789	0.0100
1944	8	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1617	0.0100
1944	*2*	*1*	*9*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.2203*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	10	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1681	0.0100
1944	11	1	9	5	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1799	0.0100
1944	*2*	*1*	*9*	*5*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.174*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	12	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.2738	0.0100
1944	13	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1680	0.0100
1944	*2*	*1*	*9*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.2209*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	14	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1759	0.0100
1944	15	1	9	7	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1391	0.0100
1944	*2*	*1*	*9*	*7*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.1575*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	16	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.1356	0.0100
1944	17	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.1	10000	0.2073	0.0100
1944	*2*	*1*	*9*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.1*	*10000*	*0.17145*	*0.01*

NodeID	Test Number	Test Number	Window lag	Hidden Nodes	Hidden Activation	Output Activation	Momentum	Learning Rate	Epoch	Validation	RMSE Value
1944	18	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1889	0.0100
1944	19	1	9	9	Bipolar Sigmoid	Sigmoid	0.9	0.08	10000	0.1659	0.0100
1944	*2*	*1*	*9*	*9*	*Bipolar Sigmoid*	*Sigmoid*	*0.9*	*0.08*	*10000*	*0.1774*	*0.01*

A* and Dijikstra Search Algorithm Results

Run Number	Search	Start Node	End Node	Path length in meters	Win	Loops	Win	Run Time (min)	Win	All or nothing ranking: Path length weight for being best (3/6)	Loop count for being best (2/6)	Processing time for being best (1/6)	Weight	Overall best
1	A*	43	2576	16,877		11,283		62.44		-	-	-	-	Dijkstra algorithm
2	Dijkstra	43	2576	17,446		9,013		49.42		-	-	-	-	
3	A*	2576	43	16,877		10,195		37.47	✓	-	-	1	0.17	
4	Dijkstra	2576	43	16,640	✓	8,283	✓	38.47		1	1	-	0.83	
5	A*	1328	1442	3,477		4,712		1.42		-	-	-	-	Dijkstra algorithm
6	Dijkstra	1328	1442	3,504		3,521		1.26		-	-	-	-	
7	A*	1442	1328	3,480		1,799		0.09	✓	-	-	1	0.17	
8	Dijkstra	1442	1328	3,447	✓	1,550	✓	0.11		1	1	-	0.83	
9	A*	898	1644	2,884		1,844		0.09		-	-	-	-	Dijkstra algorithm
10	Dijkstra	898	1644	2,211	✓	1,319	✓	0.04	✓	1	1	1	1.00	
11	A*	1644	898	2,884		2,286		0.17		-	-	-	-	
12	Dijkstra	1644	898	2,886		1,953		0.11		-	-	-	-	
13	A*	1844	1035	3,755	✓	3,588		0.74		1	-	-	0.50	Either Dijkstra or A* search
14	Dijkstra	1844	1035	3,918		2,733	✓	0.40	✓	-	1	1	0.50	
15	A*	1035	1844	3,755		3,647		1.18		-	-	-	-	
16	Dijkstra	1035	1844	4,158		3,271		1.31		-	-	-	-	
17	A*	494	2333	5,333	✓	5,277		3.62		1	-	-	0.50	Dijkstra algorithm
18	Dijkstra	494	2333	5,333	✓	4,456	✓	2.00		1	1	-	0.83	
19	A*	[illegible]	494	5,333	✓	4,787		1.68		1	-	-	0.50	
20	Dijkstra	[illegible]	494	5,626		3,560		0.78	✓	-	-	1	0.17	
21	A*	1417	1145	5,388	✓	6,336		7.64		1	-	-	0.50	A* Search Algorithm
22	Dijkstra	1417	1145	5,458		5,007		6.06		-	-	-	-	
23	A*	1145	1417	5,388		4,480	✓	2.60	✓	-	1	1	0.50	
24	Dijkstra	1145	1417	6,252		4,748		3.31		-	-	-	-	
25	A*	425	2419	7,374	✓	8,398		23.63		1	-	-	0.50	Either Dijkstra or A* search
26	Dijkstra	425	2419	7,589		5,882		7.61		-	-	-	-	
27	A*	2419	425	7,374		6,379		7.48		-	-	-	-	
28	Dijkstra	2419	425	7,379		4,983	✓	4.89	✓	-	1	1	0.50	
29	A*	2291	214	15,617		10,614		327.68		-	-	-	-	Dijkstra algorithm
30	Dijkstra	2291	214	15,474	✓	8,366	✓	33.29		1	1	-	0.83	
31	A*	214	2291	15,617		11,154		50.78		-	-	-	-	
32	Dijkstra	214	2291	15,617		8,691		30.47	✓	-	-	1	0.17	
33	A*	2636	414	14,333		10,162		40.92		-	-	-	-	Dijkstra algorithm
34	Dijkstra	2636	414	14,313	✓	8,070	✓	28.61	✓	1	1	1	1.00	
35	A*	414	2636	14,333		11,272		49.76		-	-	-	-	
36	Dijkstra	414	2636	14,462		8,872		48.79		-	-	-	-	
37	A*	2183	178	11,311		9,243		24.36		-	-	-	-	Dijkstra algorithm
38	Dijkstra	2183	178	11,172	✓	7,292	✓	14.57	✓	1	1	1	1.00	
39	A*	178	2183	11,295		10,151		35.30		-	-	-	-	
40	Dijkstra	178	2183	11,311		8,086		23.57		-	-	-	-	
41	A*	668	2886	11,465		9,209		28.84		-	-	-	-	Either Dijkstra or A* search
42	Dijkstra	668	2886	11,445		7,436	✓	20.12	✓	-	1	1	0.50	
43	A*	2886	668	11,465		10,181		42.27		-	-	-	-	
44	Dijkstra	2886	668	11,257	✓	7,876		30.60		1	-	-	0.50	
45	A*	84	623	9,730	✓	9,605		30.14		1	-	-	0.50	Either Dijkstra or A* search
46	Dijkstra	84	623	10,070		7,698		32.82		-	-	-	-	
47	A*	623	84	9,732		9,390		29.27		-	-	-	-	
48	Dijkstra	623	84	9,742		7,321	✓	11.94	✓	-	1	1	0.50	
49	A*	2803	1604	7,170		6,478		6.42		-	-	-	-	Either Dijkstra or A* search
50	Dijkstra	2803	1604	7,338		4,999	✓	2.60	✓	-	1	1	0.50	
51	A*	1604	2803	7,170	✓	8,263		3.97		1	-	-	0.50	
52	Dijkstra	1604	2803	7,526		5,332		2.71		-	-	-	-	
53	A*	1301	1158	2,961		2,956		0.33		-	-	-	-	Dijkstra algorithm
54	Dijkstra	1301	1158	3,296		2,236	✓	0.16	✓	-	1	1	0.50	
55	A*	1158	1301	2,961		2,864		0.36		-	-	-	-	
56	Dijkstra	1158	1301	2,928	✓	2,766		0.32		1	-	-	0.50	
57	A*	1954	739	3,329	✓	3,705		1.13		1	-	-	0.50	Either Dijkstra or A* search
58	Dijkstra	1954	739	4,315		3,041	✓	0.62	✓	-	1	1	0.50	
59	A*	739	1954	3,381		4,683		1.63		-	-	-	-	
60	Dijkstra	739	1954	3,338		3,895		1.07		-	-	-	-	
61	A*	1359	1417	3,170	✓	906	✓	0.02	✓	1	1	1	1.00	A* Search Algorithm
62	Dijkstra	1359	1417	3,355		962		0.02		-	-	-	-	
63	A*	1417	1359	3,170		3,428		0.59		-	-	-	-	

A* and Dijikstra Search Algorithm Results

Run Number	Search	Start Node	End Node	Path length in meters	Win	Loops	Win	Run Time (min)	Win	All or nothing ranking: Path length weight for being best (3/6)	All or nothing ranking: Loop count for being best (2/6)	All or nothing ranking: Processing time for being best (1/6)	Weight	Overall best
64	Dijkstra	1417	1359	3,366		3,019		0.78		-	-	-	-	
65	A*	1359	425	[illegible]		[illegible]		[illegible]		-	-	-	-	Dijkstra algorithm
66	Dijkstra	1359	425	[illegible]	✓	1,113	✓	0.83	✓	1	1	1	1.00	
67	A*	425	1359	3,548		6,118		[illegible]		-	-	-	-	
68	Dijkstra	425	1359	[illegible]		3,355		1.19		-	-	-	-	
69	A*	1359	494	2,877		1,309		0.13		-	-	-	-	Dijkstra algorithm
70	Dijkstra	1359	494	[illegible]	✓	1,262	✓	[illegible]	✓	1	1	1	1.00	
71	A*	494	1359	2,877		2,740		0.52		-	-	-	-	
72	Dijkstra	494	1359	2,819		2,513		0.34		-	-	-	-	
73	A*	831	1579	[illegible]		8,783		[illegible]		-	-	-	-	A* Search Algorithm
74	Dijkstra	831	1579	12,576		7,743	✓	[illegible]		-	1	-	0.33	
75	A*	1579	831	11,371	✓	8,932		[illegible]		1	-	-	0.50	
76	Dijkstra	1579	831	[illegible]		8,812		[illegible]	✓	-	-	1	0.17	

Appendix C: Sample Database Creation Scripts

tblCount

```
USE [RouteManer]
GO
/****** Object:  Table [dbo].[tblCount]    Script Date: 06/04/2006 20:11:56 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tblCount](
	[CountID] [int] IDENTITY(1,1) NOT NULL,
	[NodeID] [int] NOT NULL,
	[TimeID] [int] NOT NULL,
	[DateCollected] [datetime] NOT NULL CONSTRAINT [DF_tblCount_DateCollected]  DEFAULT (getdate()),
	[Approaching] [int] NOT NULL,
	[Departing] [int] NOT NULL,
	[VehicleID] [int] NOT NULL,
	[SurveyPoint] [varchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
 CONSTRAINT [PK_tblCount] PRIMARY KEY CLUSTERED
(
	[CountID] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
USE [RouteManer]
GO
ALTER TABLE [dbo].[tblCount]  WITH CHECK ADD  CONSTRAINT [FK_tblCount_tblNode] FOREIGN KEY([NodeID])
REFERENCES [dbo].[tblNode] ([NodeID])
GO
ALTER TABLE [dbo].[tblCount]  WITH CHECK ADD  CONSTRAINT [FK_tblCount_tblTime] FOREIGN KEY([TimeID])
REFERENCES [dbo].[tblTime] ([timeID])
GO
ALTER TABLE [dbo].[tblCount]  WITH CHECK ADD  CONSTRAINT [FK_tblCount_tblVehicle] FOREIGN KEY([VehicleID])
REFERENCES [dbo].[tblVehicle] ([vehicleid])
```

[proc_AddKnowledgeToNode

```
USE [RouteManer]
GO
/****** Object:  StoredProcedure [dbo].[proc_AddKnowledgeToNode]    Script Date: 06/04/2006 20:26:58 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO




-- =============================================
-- Author:		<Edgar Okioga>
-- Create date: <29/04/2006>
-- Description:	<Adds new network parameters to the associated node from the learning Model>
-- =============================================
CREATE PROCEDURE [dbo].[proc_AddKnowledgeToNode]
	@NodeID int,
	@LearningRate float,
	@Momentum float,
	@NumberInputUnits int,
	@NumberHiddenUnits int,
	@NumberOutputunits int,
	@HiddenActivationType int,
	@OutputActivationType int,
	@InputScaleType int,
	@OutputScaleType int,
```

```
[proc_ AddKnowledgeToNode
            @DateCreated datetime,
            @InputScaleMin float,
            @InputScaleMax float,
            @OutputScaleMin float,
            @OutputScaleMax float,
            @NetworkID int Output
AS
BEGIN
            -- SET NOCOUNT ON added to prevent extra result sets from
            -- interfering with SELECT statements.
            SET NOCOUNT ON;

   -- Insert statements for procedure here
            --insert data into the underling table and return the networkID
            INSERT INTO tblNeuralNetwork
                        (NodeID, LearningRate, Momentum, NumberInputUnits, NumberHiddenunits, NumberOutputUnits,
HiddenActivationType, OutputactivationType, inputScaletype, outputscaletype, datecreated, InputscaleMin, InputscaleMax,
outputscalemin, outputscalemax)
            values
                        (@NodeID, @LearningRate, @Momentum, @NumberInputUnits, @NumberHiddenunits,
@NumberOutputUnits, @HiddenActivationType, @OutputactivationType, @inputScaletype, @outputscaletype,
@datecreated,@InputscaleMin, @InputscaleMax, @outputscalemin, @outputscalemax)

            SET @NetworkID = Scope_Identity()
END
```

```
[proc_ GetBrainDate
USE [RouteMiner]
GO
/****** Object:  StoredProcedure [dbo].[proc_GetBrainDate]    Script Date: 06/04/2006 20:30:01 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:                  <Edgar Okioga>
-- Create date: <31/05/2006>
-- Description:             <Tells us if there is a brain or not>
-- =============================================
CREATE PROCEDURE [dbo].[proc_GetBrainDate]
            -- Add the parameters for the stored procedure here
            @NodeID as int
AS
BEGIN
Declare @DateSave as Datetime
Declare @Number as int
Declare @Inputs as int
            -- SET NOCOUNT ON added to prevent extra result sets from
            -- interfering with SELECT statements
            SET NOCOUNT ON;

   -- Insert statements for procedure here
            SELECT     TOP (1) @DateSave= dbo.tblNeuralNetwork.DateCreated, @Inputs=dbo.tblNeuralNetwork.NumberInputunits
FROM        dbo.tblNeuralNetwork INNER JOIN
              dbo.tblNode ON dbo.tblNeuralNetwork.NodeID = dbo.tblNode.NodeID
WHERE     (dbo.tblNode.NodeID = @nodeID)
ORDER BY dbo.tblNeuralNetwork.DateCreated DESC
--net get the number of brains avilable
select @Number=Count(nodeID) from tblNeuralnetwork where nodeID = @nodeID
--Return the Number and Last Save Date

SELECT @Number, @Datesave,@Inputs
END
```

APPENDIX D – Sample Neural Network Source Code

Neural network Class

```
'Train the Neural Network now()
Public Sub TrainNetwork()
   'First Initialise the weights and bias
   Dim CurrentEpoch As Integer = 0
   Dim CurrentRMSEThreshold As Double
   Dim OldRmseThreshold As Double
   Const ConstRMSENoChange As Double = 0.0000001
   Dim CurrentPattern As Integer                    'Each Training pattern
   Dim PatterIndex As Integer
   Dim RMSENoChangeCount As Integer
   'For the purpose of getting the RMSE
   Dim rmseRows As Integer
   Dim StopLearning As Boolean = False
   Dim CurrentInputs() As Double
   Dim OldValidationError As Double
   Dim CurrentValidationError As Double
   Dim RmseCheck As Integer

   'Delegate to show epoch to main window
   Dim dlgEpochShow As ShowCurrentEpoch
   dlgEpochShow = New ShowCurrentEpoch(AddressOf UpdateEpoch)
   Dim Arg(1) As Object                 'Parameter to pass to the main form
   'Train the network
   ReDim Preserve CurrentInputs(mNumberInputUnits - 1)
   'train the network
   'Initialise weights
   InitialiseWeights()
   While StopLearning = False
        'Process for each case presented to the neural network
        For CurrentPattern = 0 To mNumberTrainingPatterns - 1
         For PatterIndex = 0 To mNumberInputUnits - 1
              CurrentInputs(PatterIndex) = mInputs(CurrentPattern, PatterIndex)
         Next
         'Feed Forward
         HiddenInput(mNumberHiddenUnits, mNumberInputUnits, CurrentInputs)
         HiddenTransfer(mNumberHiddenUnits, mHiddenlayerActivation)
         OutputInput(mNumberHiddenUnits)
         OutputTransfer(mOutputlayerActivation)
         'Back propagate the error values
         UpdateOutputWeights(CurrentPattern, mNumberHiddenUnits, mOutputlayerActivation)
         UpdateHiddenWeights(mNumberInputUnits,      mNumberHiddenUnits,      mHiddenlayerActivation,
CurrentInputs)
        Next CurrentPattern
        'Add the Current Epoch
        CurrentEpoch = CurrentEpoch + 1
        'Poke to update the epoch
        'Carry oput post epoch calculations
        For rmseRows = 0 To mNumberTrainingPatterns - 1
         For PatterIndex = 0 To mNumberInputUnits - 1
              CurrentInputs(PatterIndex) = mInputs(rmseRows, PatterIndex)
         Next
         HiddenInput(mNumberHiddenUnits, mNumberInputUnits, CurrentInputs)
         HiddenTransfer(mNumberHiddenUnits, mHiddenlayerActivation)
         OutputInput(mNumberHiddenUnits)
         OutputTransfer(mOutputlayerActivation)
         'Update the Output Array
         mOutput(rmseRows) = mOutputActivation(0)
        Next
        'Calculate the Current RMSEError
        OldRmseThreshold = CurrentRMSEThreshold
        CurrentRMSEThreshold = ReturnRMSEThreshold(mOutput, mTarget)
        'Update the main form

        Arg(0) = CurrentEpoch
        Arg(1) = CurrentRMSEThreshold & "|" & CurrentValidationError
        frmMainForm.Invoke(dlgEpochShow, Arg)
```

Neural network Class

```
            'Process the stopping criteria once again
            'Base of Epoch
            If (CInt(2 ^ 0) And CInt(mStopTrainingMode)) <> 0 Then
             StopLearning = CBool(IIf(CurrentEpoch >= mEpoch, True, False))
             If StopLearning = True Then
                 Exit While
             End If
            End If
            'RMSEThreshold
            If (CInt(2 ^ 1) And CInt(mStopTrainingMode)) <> 0 And CurrentEpoch > 100 Then
             StopLearning = CBool(IIf(CurrentRMSEThreshold <= mRMSEThreshold, True, False))
             If StopLearning = True Then
                 Exit While
             End If
            End If
            'RMSEThreshold no Change Count
            If (CInt(2 ^ 5) And CInt(mStopTrainingMode)) <> 0 And CurrentEpoch > 100 Then
             If Math.Abs(OldRmseThreshold - CurrentRMSEThreshold) <= ConstRMSENoChange Then
                 RMSENoChangeCount += 1
             End If
             StopLearning = CBool(IIf(RMSENoChangeCount >= mRMSECount, True, False))
             If StopLearning = True Then
                 Exit While
             End If
             'stop if the erro is increaing
             If OldRmseThreshold - CurrentRMSEThreshold < 0 Then
                 Exit While
             End If
            End If
            'Use validation error to ensure we are okay
            If (CInt(2 ^ 2) And CInt(mStopTrainingMode)) <> 0 And CurrentEpoch > 100 Then
             CurrentValidationError = GetValidationError()
             If CurrentValidationError > OldValidationError Then
                 RmseCheck += 1
             End If
             If RmseCheck > 50 Then
                 Exit While
             End If
             OldValidationError = CurrentValidationError
            End If
            'Update the staus bar
            mTrainingResults = TellError()
        End While
        'Process the output form the neural netork
        Dim myTimer As TurnOffTimer
        myTimer = New TurnOffTimer(AddressOf TimerOff)
        frmMainForm.Invoke(myTimer)
    End Sub

    Sub UpdateEpoch(ByVal CurrentEpoch As Integer, ByVal CurrentRMSE As String)
        frmMainForm.stsBar.Panels(0).Text = "Epoch [" & CurrentEpoch.ToString & "]"
        frmMainForm.stsBar.Panels(1).Text = "T/V-RMSE [" & String.Format("{0:0.######}",
CurrentRMSE.ToString) & "]"
        frmMainForm.txtreport.Text = frmMainForm.txtreport.Text & "Epoch->" & vbTab &
CurrentEpoch.ToString & vbTab & "," & vbTab & "TRAINING RMSE->" & vbTab &
String.Format("{0:0.0000}", CurrentRMSE.Split("|"c)(0)) & vbTab & "," & vbTab & vbTab & "VALIDATION
RMSE->" & vbTab & String.Format("{0:0.0000}", CurrentRMSE.Split("|"c)(1)) & vbCrLf
        If frmMainForm.tmrDraw.Enabled = False Then
            frmMainForm.tmrDraw.Enabled = True
        End If
        Application.DoEvents()
    End Sub
```

APPENDIX E: Sample A* search Source Code

Route Class

```
While (open.QueueCount > 0)
    'Get the lowest node
    Mynode = Nothing
    Mynode = open.QueuePop
    'Check if the goal node has been found
    If Mynode.NodeID = Stopnode.NodeID Then
        'Exit now, found
        While Not Mynode Is Nothing
            Solution.Insert(0, Mynode)
            Mynode = Mynode.Parent
        End While
        Exit While
    End If
    'Expand the node
    cmdGetArcFromNode.Parameters("@StartNode").Value = Mynode.NodeID
    cmdGetArcFromNode.Parameters("@StopNode").Value = Stopnode.NodeID
    drRoute = cmdGetArcFromNode.ExecuteReader()
    Dim Found As Boolean = False
    Dim Added As Boolean = False
    'end of expansion now loop thru the expanded list
    'For each node in the expansion
    While drRoute.Read = True
        temp = New _Path
        'Populate the node
        temp.NodeID = CType(drRoute.Item(0), Integer)
        temp.NodeCost = CType(drRoute.Item(1), Decimal)
        temp.PathCost = CType(drRoute.Item(2), Decimal)
        temp.TotalCost = CType(drRoute.Item(3), Decimal) + Mynode.TotalPathCost
        temp.TotalPathCost = temp.PathCost + Mynode.TotalPathCost
        temp.ArcID = CType(drRoute.Item(4), Integer)
        'Set the parent of this node
        temp.Parent = Mynode
        'end of population
        'Is the tempnode found in the open list
        Dim temp1 As _Path
        Dim Discard1 As Boolean = False
        Dim Discard2 As Boolean = False
        If open.QueueFind(temp) = True Then
            temp1 = open.QueuePeek(temp.NodeID)
            If temp1.TotalCost < temp.TotalCost Then
                Discard1 = True
            End If
        End If
        If closed.QueueFind(temp) = True Then
            temp1 = closed.QueuePeek(temp.NodeID)
            If temp1.TotalCost < temp.TotalCost Then
                Discard2 = True
            End If
        End If
        If Discard1 = True Then
            If Discard2 = True Then
                open.QueueRemoveNode(temp)
                closed.QueueRemoveNode(temp)
            End If
        Else
            If Discard2 = False Then
                open.QueueAdd(temp)
            End If
        End If
        LoopCount += 1
    End While
    drRoute.Close()
    'Add node to closes
    closed.QueueAdd(Mynode)
End While
```

APPENDIX F: Samples GIS Dangle detection code

```
ArcView Script to Identify dangles
'on 03/06/2006
'To identify dangle links in the network
'************************************
'Get the current View
theview = av.GetActiveDoc

'Get the active theme
thetheme = theview.GetActiveThemes.Get(0)

'Get the Feature table
theFtab = theTheme.getFtab

'Return the number of records
thetotal = theftab.GetNumRecords
theCurrent = theftab.GetNumSelRecords

'Get the feature Bitmap
thebitmap = thefTab.GetSelection

'Create an array list
theShapes={}

'Iterate through all links beginning with the current one selecting all other links the
'Intersect with it and add them to the array.
for each item in 1..thetotal
        for each rec in thebitmap
            theshape = theftab.ReturnValue(theFtab.findfield("Shape"),rec)
            'Add the current link to the array
            theshapes.add(theshape)
    end

        'Select all links the intersect with the current selected links
        theftab.SelectByShapes (theshapes, #VTAB_SELTYPE_NEW)

        'Update the statistics
        thebitmap = thefTab.GetSelection
end

'Refresh the view
theView.invalidate
theCurrent = theftab.GetNumSelRecords

msgbox.info((thetotal-thecurrent)..AsString,"Links not connected")
```

APPENDIX G: Probe Car Sample

Date 09_03_2006

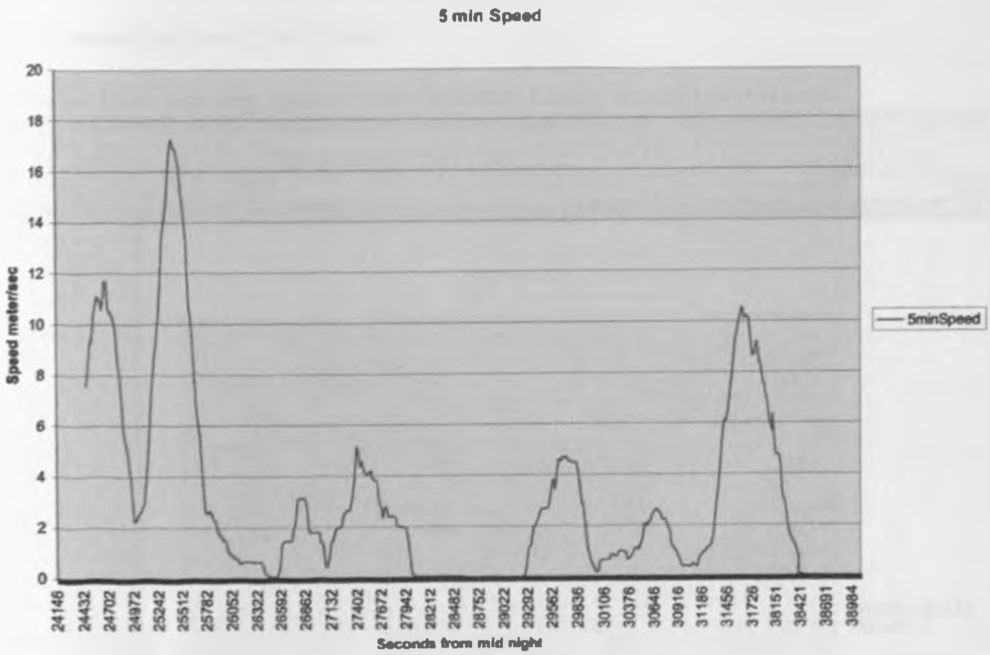
Start point: Nyayo Estate
End point: Unep gigiri

Tools

1) Garmin eTrex GPS
2) Software
    a. Arciew 3.2
    b. EasyGIS
    c. Excel

Methodology

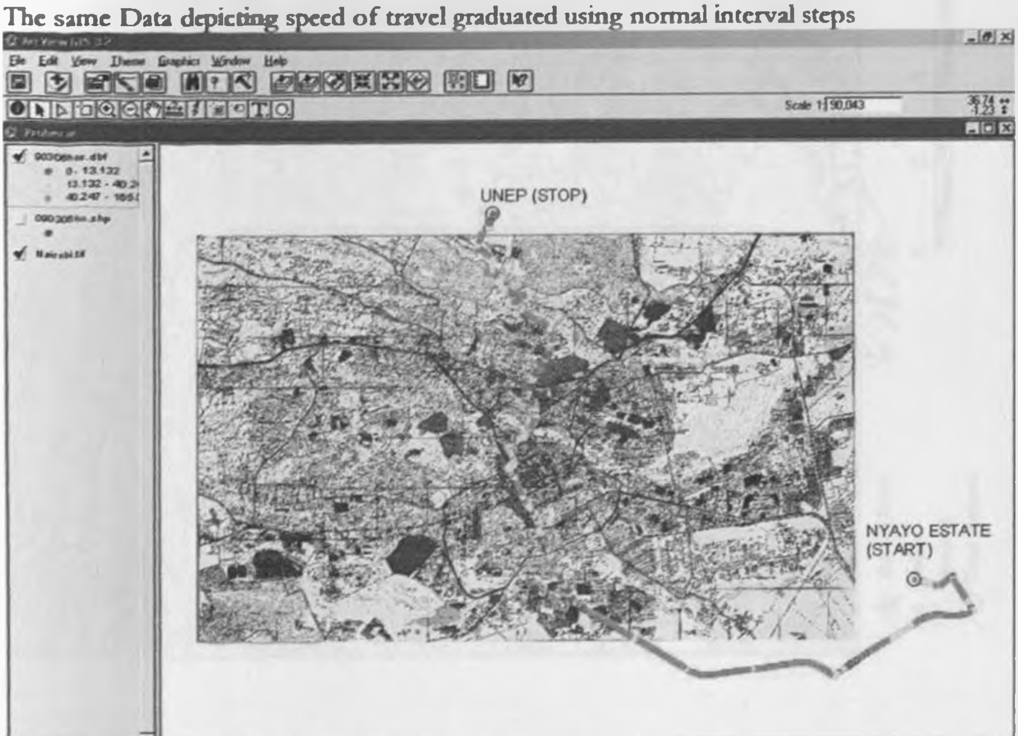
- The garmin etrex gps is set to record data after one (1) second
- The data is then downloaded from the GPS using an RS-232 cable on COM1
- The software used is EasyGIS which stores the data in XML format.
- The data is read in XML format in excel and results obtained.
- The time is converted to seconds from midnight.
- The distance between successive points is calculated using Euclidean distance with a conversion to KM by multiplying by 110.592. (1 degree is approximately 110.592km at the equatorial region)
- The speed is calculated at intervals based on total distance divide by time interval



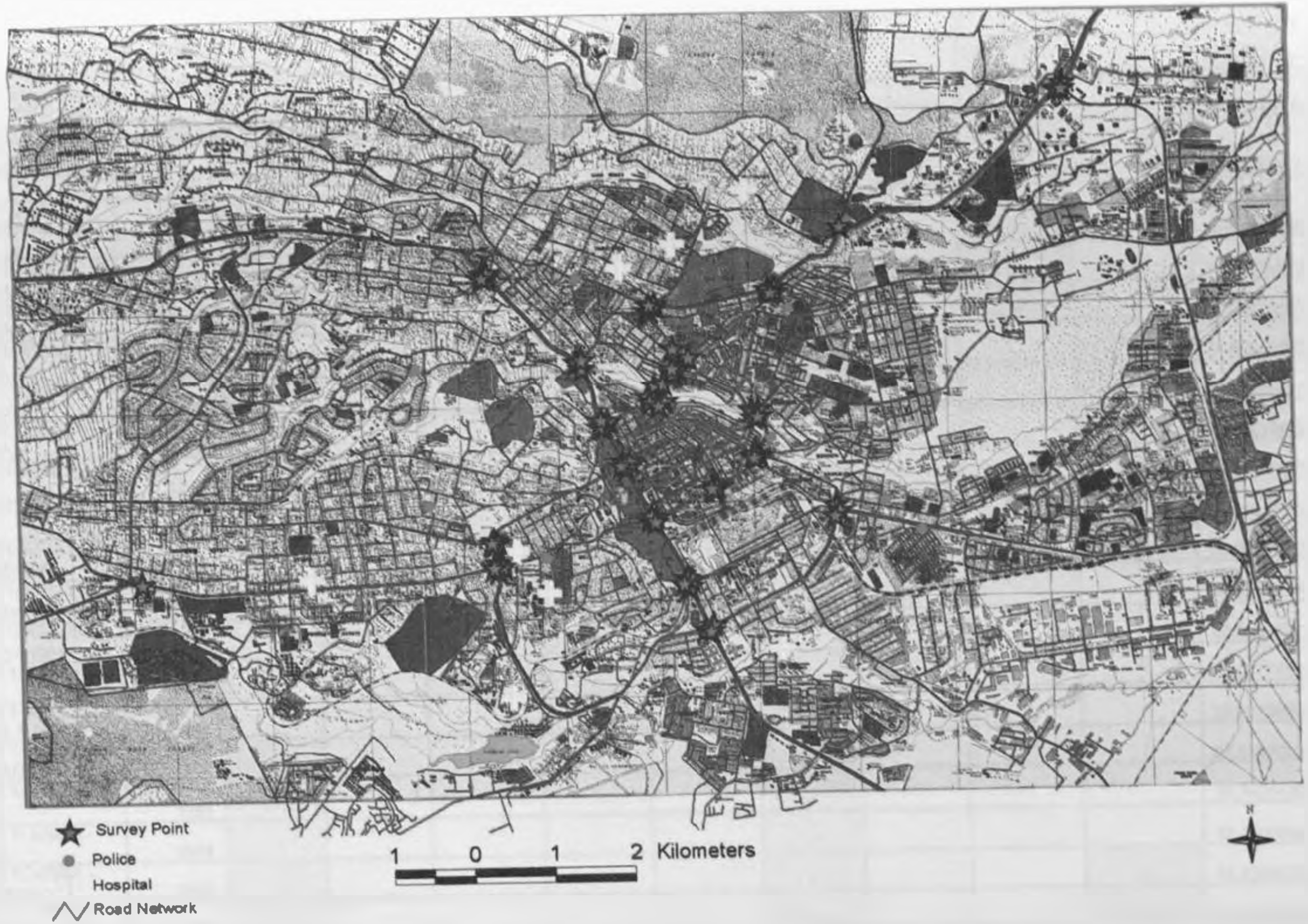
This is the average speed at 5 minute intervals from start to finish.

The data collected from Start to stop.

The same Data depicting speed of travel graduated using normal interval steps



APPENDIX H: Traffic Survey locations.



SURVEY LOCATION NAME	NODE ID	TUE JAN	TUE MAY	TUE JUNE	WED JAN	WED MAY	THU JAN	THU MAY	THU JUNE	WED JUNE	LONGITUDE	LATITUDE
SURVEY TYPE 1: ARM 1 - UNIVERSITY WAY	1291	✓				✓	✓		✓		36.81654358	-1.28119206
SURVEY TYPE 1: ARM 2 - UHURU HIGHWAY S	1449	✓				✓	✓		✓		36.81608200	-1.28372300
SURVEY TYPE 1: ARM 3 - STATE HOUSE ROAD	1340	✓				✓	✓		✓		36.81438446	-1.28206706
SURVEY TYPE 1: ARM 4 - UHURU HIGHWAY NORTH	1283	✓				✓	✓		✓		36.83436966	-1.27934492
SURVEY TYPE 1: ARM 1 - HAILE SELASSIE AVENUE NE	1944	✓				✓	✓		✓		36.82144547	-1.29243147
SURVEY TYPE 1: ARM 3 - HAILE SELASSIE AVENUE W	1994	✓				✓	✓		✓		36.81680298	-1.29349661
SURVEY TYPE 1: ARM 4 - UHURU HIGHWAY NW	1887	✓				✓	✓		✓		36.81980515	-1.29129326
SURVEY TYPE 1: ARM 2 - UHURU HIGHWAY SE	2199	✓				✓	✓		✓		36.82342148	-1.29763091
SURVEY TYPE 1: ARM 6 - KIPANDE ROAD	1066				✓					✓	36.81892395	-1.27554369
SURVEY TYPE 1: ARM 1 - MURANGA ROAD	1193				✓					✓	36.82196426	-1.27730918
SURVEY TYPE 1: ARM 2 - KIRINYAGA ROAD	1212				✓					✓	36.82210922	-1.27905655
SURVEY TYPE 1: ARM 5 - KIJABE ROAD	1146				✓					✓	36.81716156	-1.27725983
SURVEY TYPE 1: ARM 4 - SLIP ROAD	1258				✓					✓	36.81977463	-1.28038108
SURVEY TYPE 1: ARM 3 - TOM MBOYA STREET	1230				✓					✓	36.82140350	-1.27964115
SURVEY TYPE 1: ARM 2 - LANDHIES/PUMWANI ROAD	1601				✓						36.83413696	-1.28612518
SURVEY TYPE 1: ARM 1 - RING ROAD PUMWANI	1327				✓						36.83332062	-1.28183830
SURVEY TYPE 1: ARM 4 - RIVER ROAD	1565				✓						36.83174515	-1.28538179
SURVEY TYPE 1: ARM 3 - HAILE SELASIE AVENUE	1616				✓						36.83196640	-1.28643930
SURVEY TYPE 1: ARM 1 - LANDHIES ROAD	1738	✓					✓				36.83950806	-1.28860188
SURVEY TYPE 1: ARM 2 - JOGOO ROAD	1938	✓					✓				36.84356308	-1.29236519
SURVEY TYPE 1: ARM 3 - LUSAKA ROAD	1928	✓					✓				36.84096527	-1.29220736
SURVEY TYPE 1: ARM 4 - AERODROME ROAD	2485	✓					✓			✓	36.82036209	-1.30640972
SURVEY TYPE 1: ARM 1 - UHURU HIGHWAY NORTH	2450	✓					✓			✓	36.82665253	-1.30496871

SURVEY LOCATION NAME	NODE ID	TUE_JAN	TUE_MAY	TUE_JUNE	WED_JAN	WED_MAY	THU_JAN	THU_MAY	THU_JUNE	WED_JUNE	LONGITUDE	LATITUDE
SURVEY TYPE 1: ARM 3 - UHURU HIGHWAY S	2320	✓					✓			✓	36.82427597	-1.29991293
SURVEY TYPE 1: ARM 2 - BUNYALA ROAD E	2303	✓					✓			✓	36.82533264	-1.29954314
SURVEY TYPE 1: ARM 5 - BUNYALA ROAD W	2314	✓					✓			✓	36.82273102	-1.29976690
SURVEY TYPE 1: ARM 4 - LANGATA ROAD	2465	✓					✓			✓	36.82370758	-1.30570745
SURVEY TYPE 1: ARM 4 - LANGATA ROAD	2445	✓					✓			✓	36.82762527	-1.30441809
SURVEY TYPE 1: ARM 3 - MOMBASA ROAD	2608	✓					✓			✓	36.82894516	-1.30990696
SURVEY TYPE 1: ARM 1 - VALLEY ROAD NORTH	2211	✓					✓				36.80291748	-1.29789639
SURVEY TYPE 1: ARM 3 - MBAGATHI ROAD SOUTH	2513	✓					✓				36.80381393	-1.30723608
SURVEY TYPE 1: ARM 2 - NGONG ROAD EAST	2189	✓					✓				36.80507660	-1.29741907
SURVEY TYPE 1: ARM 4 - NGONG ROAD WEST	2254	✓					✓				36.79870987	-1.29864299
SURVEY TYPE 1: ARM 1 - VALLEY ROAD NORTH	1940	✓					✓				36.80515289	-1.29238355
SURVEY TYPE 1: ARM 2 - ARGWINGS KODHEK EAST	2055	✓					✓				36.80647278	-1.29514205
SURVEY TYPE 1: ARM 3 - VALLEY ROAD SOUTH	2060	✓					✓				36.80243301	-1.29521990
SURVEY TYPE 1: ARM 4 - ARGWINGS KODHEK WEST	2057	✓					✓				36.80044556	-1.29515493
SURVEY TYPE 1: ARM 4 - MOI AVENUE NW	1692	✓									36.82693481	-1.28779161
SURVEY TYPE 1: ARM 2 - MOI AVENUE SE	1789	✓									36.79620361	-1.28924775
SURVEY TYPE 1: ARM 1 - HAILE SELASSIE AVENUE NE	1769	✓									36.82833481	-1.28889573
SURVEY TYPE 1: ARM 3 - HAILE SELASSIE AVENUE SW	1835	✓									36.82584381	-1.29023111
SURVEY TYPE 1: ARM 3 - CHIROMO ROAD	795				✓	✓			✓		36.80783081	-1.27120376
SURVEY TYPE 1: ARM 2 - UHURU HIGHWAY SE	1047				✓	✓			✓		36.81265259	-1.27508628
SURVEY TYPE 1: ARM 1 - MUSEUM HILL ROAD	958				✓	✓			✓		36.81309509	-1.27383900
SURVEY TYPE 1: ARM 1 - WAIYAKI WAY	469	✓				✓	✓				36.80176544	-1.26425540
SURVEY TYPE 1: ARM 3 - CHIROMO ROAD SOUTH	557	✓				✓	✓				36.80316162	-1.26596081

SURVEY LOCATION NAME	NODE ID	TUE JAN	TUE MAY	TUE JUNE	WED JAN	WED MAY	THU JAN	THU MAY	THU JUNE	WED JUNE	LONGITUDE	LATITUDE
SURVEY TYPE 1: ARM 2 - RING ROAD WESTALNDS EAST	422	✓				✓	✓				36.80318832	-1.26297390
SURVEY TYPE 1: ARM 4 - RING ROAD WESTLANDS WEST /RHAPTA	531	✓				✓	✓				36.80103683	-1.26555657
SURVEY TYPE 1: ARM 4 - UHURU HIGHWAY NW	1569	✓			✓	✓			✓		36.81690979	-1.28541434
SURVEY TYPE 1: ARM 2 - UHURU HIGHWAY SE	1702	✓			✓	✓			✓		36.81816101	-1.28797388
SURVEY TYPE 1: ARM 1 - KENYATTA AVENUE NE	1602					✓			✓		36.81838989	-1.28612983
SURVEY TYPE 1: ARM 3 - KENYATTA AVENUE SE	1701	✓			✓	✓			✓		36.81473541	-1.28792572
SURVEY TYPE 1: ARM 3 - THIKA ROAD SW	110						✓				36.86303329	-1.24871302
SURVEY TYPE 1: ARM 1 - THIKA ROAD NE	58						✓				36.86796951	-1.24347496
SURVEY TYPE 1: ARM 2 - OUTER RING ROAD	72						✓				36.86677170	-1.24489605
SURVEY TYPE 1: ARM 4 - FOREST ROAD W	725	✓									36.81975937	-1.26917064
SURVEY TYPE 1: ARM 3 - LIMURU ROAD S	739	✓									36.82059479	-1.26960254
SURVEY TYPE 1: ARM 2 - FOREST ROAD E	704	✓									36.82141876	-1.26868391
SURVEY TYPE 1: ARM 1 - LIMURU ROAD NE	667	✓									36.82145691	-1.26806915
SURVEY TYPE 1: ARM 5 - WAIYAKI WAY	694	✓									36.82072067	-1.26851153
SURVEY TYPE 1: ARM 2 - RING ROAD NGARA	675						✓				36.83552933	-1.26822567
SURVEY TYPE 1: ARM 1 - THIKA ROAD NE	560						✓				36.83536911	-1.26604676
SURVEY TYPE 1: ARM 4 - FOREST ROAD W	629						✓				36.83289337	-1.26716566
SURVEY TYPE 1: ARM 3 - MURANGA ROAD SW	700						✓				36.83322906	-1.26864862
SURVEY TYPE 1: NGONG ROAD KSTC ARM	2261		✓								36.76322937	-1.29882741
SURVEY TYPE 1: KIAMBU ROAD	374					✓					36.84178543	-1.26078546
SURVEY TYPE 1: ARM 4 - PARK ROAD	1275	✓									36.83239746	-1.28072643
SURVEY TYPE 1: ARM 1 - RACECOURSE ROAD NE	1228	✓									36.83350372	-1.27959728
SURVEY TYPE 1: ARM 3 - RACECOURSE ROAD SE	1324	✓									36.83214188	-1.28160206

SURVEY LOCATION NAME	NODE ID	TUE_JAN	TUE_MAY	TUE_JUNE	WED_JAN	WED_MAY	THU_JAN	THU_MAY	THU_JUNE	WED_JUNE	LONGITUDE	LATITUDE
SURVEY TYPE 1: ARM 2 - RING ROAD PUMWANI	1311	✓									36.83320618	-1.28166735
SURVEY TYPE 1: ARM 1 - LIMURU ROAD NW	928				✓						36.82317734	-1.27341366
SURVEY TYPE 1: ARM 2 - MURANGA ROAD NE	983				✓						36.82489395	-1.27433729
SURVEY TYPE 1: ARM 3 - NGARA ROAD E	1086				✓						36.82321548	-1.27584565

APPENDIX I: TRAFFIC IMPEDANCE QUESTIONNAIRE

Dynamic Route Selection: Short Term Traffic Decision Support in Nairobi

Traffic jams are a daily occurrence in Nairobi and its environs. About 30,000 extra vehicles per year have been added to our roads over the last decade. Gavin Bennett, a regular columnist with the Sunday Nation Motoring column suggests 3 solutions. First option is reducing the number of vehicles (ensuring they are more expensive to buy or enforcing minimum standards). The second option is to improve the carrying capacity of the roads by improving junction design, removing bottlenecks and adding more ring-roads routes, lanes or links and lastly, enforce traffic rules and regulations resulting to a shift in driver attitude, conduct and responsibility.

This survey is to identify the various causes of traffic snarls in Nairobi. The results will be used to identify the least to the most significant causes of traffic snarls on Nairobi roads.

Definitions

The following words have their meanings as indicated here;

i. **Car:** A vehicle of less than 2500 cc used for private transport that carries a maximum of 5 passengers.
ii. **Matatu:** A vehicle of less than 5000 cc used for public transport with a carrying capacity of between 14 and 28 passengers.
iii. **Bus:** A vehicle of more than 5500 cc used for public transport with a carrying capacity of more than 28 passengers.
iv. **Lorry:** A vehicle that carries load over and above 14 tones for commercial use.
v. **Bottleneck:** A factor that restricts or retards free flow of vehicles from a current location to the next during normal travel.
vi. **Balancing:** The ability to maintain the steady idle of a vehicle engine when on a hill by means of using the clutch and accelerator pedals only.
vii. **Normal traffic flow:** A situations where you are driving at a comfortable speed of about 50 km/hr, being able to change lanes easily as your progress.

Instructions

For each question, unless otherwise stated, **tick only one value per factor and no repeated values per question as a means of indicating a rank relative to importance to you as a driver on Nairobi Roads.**

Section A:

(Please tick the relevant choice)

1) Your gender:

Male ☐ Female ☐

2) How long have you held a valid driving license? :

Less than 1 year ☐ between 2 to 5 years ☐

More than 6 years ☐

3) What vehicle do you most frequently drive?

Car ☐ Bus ☐

Lorry ☐ Matatu ☐

Section B

(Please assign a value to the relevant importance of each of the items presented)

4) In your opinion, at a roundabout or traffic intersection point, what is the relative importance of the following items in obstructing normal traffic flow?

Bottleneck: (Rank: 1 least, 5 Most relevant) [Per row , select a unique value with no repetition in the complete set]

Your observation of traffic ahead of you	:①	②	③	④	⑤
If the section is a climbing lane requiring balancing[1] of the vehicle	:①	②	③	④	⑤
The presence or absence of police/traffic lights	:①	②	③	④	⑤
The level of security of the location	:①	②	③	④	⑤
The weather condition of the day	:①	②	③	④	⑤

[1] A factor that restricts or retards free flow of vehicles from a current location to the next during normal travel.

APPENDIX I: TRAFFIC IMPEDANCE QUESTIONNAIRE

5) In your opinion, on an unrestricted section of a road, what is the relative importance of the following items in obstructing normal traffic flow?

Bottleneck: (Rank: 1 least, 6 Most relevant) [Per row , select a unique value with no repetition in the set]

Your observation of traffic ahead of you :① ② ③ ④ ⑤ ⑥
The number of lanes on the road :① ② ③ ④ ⑤ ⑥
The presence or absence of speed bumps or potholes :① ② ③ ④ ⑤ ⑥
The level of security of the location :① ② ③ ④ ⑤ ⑥
The weather condition of the day :① ② ③ ④ ⑤ ⑥
The general drainage of the road section :① ② ③ ④ ⑤ ⑥

6) The natural bottleneck areas in Nairobi have been thoroughly studied; there is a master plan which shows where even new link roads should be built to rationalise the traffic flow. There are also a few dozen junctions or roundabouts which could take much higher traffic flow with just a slip-lane of two. But a major part of the solution is reducing the number of obstructions to flow – a bad pothole, speed bump or irresponsible driver stopping at a non stopping zone or doing something to slow down traffic as characterised by matatus. In your opinion, what is the most obvious solution to take to achieve reduced traffic jams?

Solution: (Rank: 1 least, 6 Most relevant) [Per row , select a unique value with no repetition in the set]

Build the new link roads :① ② ③ ④ ⑤ ⑥

Add slip lanes to junctions and roundabouts :① ② ③ ④ ⑤ ⑥

Enforce the traffic act with effective policing :① ② ③ ④ ⑤ ⑥

Resurface all roads to eliminate potholes :① ② ③ ④ ⑤ ⑥

Remove speed bumps unless in a school zone :① ② ③ ④ ⑤ ⑥

Introduce night work and shopping thought out a 24 hour period :① ② ③ ④ ⑤ ⑥

Other:

Thank you:

Edgar Nyamweya Okioga,
P56/P/8066/03,
MSc. Information Systems,
University of Nairobi,
edgar.okioga@wfp.org
0727-676594