AN INTELLIGENT AUTOMATIC VEHICLE TRAFFIC FLOW MONITORING AND CONTROL SYSTEM

STUDENT NAME
Theko Emmanuel Marie
STUDENT NUMBER
9517464
CURRENT QUALIFICATION
BTech: Computer Systems Engineering

Dissertation submitted in fulfilment of the requirements for the degree of Magister Technologiae: Information Technology in the Department of Information and Communications Technology, Faculty of Applied and Computer Sciences, Vaal University of Technology

Supervisor: Prof. B.N. Gatsheni (Ph.D)

JANUARY, 2015

DECLARATION

I Theko Emmanuel Marie hereby declare that this work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed……………………………………….

Date…………………………………………..

ACKNOWLEDGEMENTS

I hereby wish to express my gratitude to the following individuals who enabled this document to be successfully and timeously completed:

- Supervisor - Prof B.N. Gatsheni. Your continual support and motivation drove me until completion.
- Ms Lombard (HOD ICT). Your support did not go unnoticed.
- VUT research office. I learnt a lot from students and supervisors’ training workshops which you organized for us.
- NRF for funding this project.
- Mr W. Tukisi. Unbelievable understanding and continual support during microcontroller experiments.
- Optinum Solutions for their free Matlab tutorials.
- My wife - Tshenolo Molebatsi Marie and my daughter Botle Bokamoso Marie. It was not easy for you guys when I sometimes woke up at 2:00 in the morning to continue with my research. You guys always understood and never complained. I love you guys.

ABSTRACT

Traffic congestion is a concern within the main arteries that link Johannesburg to Pretoria. In this study Matlab function randperm is used to generate random vehicle speeds on a simulated highway. Randperm is used to mimic vehicle speed sensors capturing vehicle traffic on the highway.

Java sockets are used to send vehicle speed to the Road Traffic Control Centre (RTCC)-database server through a wireless medium. The RTCC-database server uses MySQL to store vehicle speed data. The domain controller with active directory together with a certificate server is used to manage and provide security access control to network resources. The wireless link used by speed sensors to transmit vehicle speed data is protected using PEAP with EAP-TLS which employs the use of digital certificates during authentication.

A java database connectivity driver is used to retrieve data from MySQL and a multilayer perceptron (MLP) model is used to predict future traffic status on the highway being monitored i.e. next 5 minutes from previous 5 minutes captured data. A dataset of 402 instances was divided as follows: 66 percent training data was used to train the MLP model, 15 percent data used during validation and the remaining 19 percent was used to test the trained MLP model. An excel spreadsheet was used to introduce novel (19 percent data not used during training) data to the trained MLP model to predict. Assuming that the spreadsheet data represent captured highway vehicle data for the last 5 minutes, the model showed 100 percent accuracy in predicting the four classes: congested, out congested, into congested and normal traffic flow.

Predicted traffic status is displayed for the motorist on the highway to know. Ability of the proposed model to continuously capture the traffic pattern on the highway (monitor) helps in redirecting (controlling) the highway traffic during periods of congestion.

Implementation of this project will definitely decrease traffic congestion across main arteries of Johannesburg. Pollution normally experienced when cars idle for a long time during congestion will be reduced by free highway traffic flow. Frequent servicing of motor vehicles will no longer be required by the motorists. Furthermore the economy of Gauteng and South Africa as a whole will benefit due to increase in production. Consumers will also benefit in obtaining competitive prices from organizations that depend on haulage services.

Table of contents Page

Chapter 1 Introduction

Chapter 2 Literature review

Chapter 3 Candidate techniques

Chapter 4 Experiments

Chapter 5 Discussion of results

Chapter 6 Conclusion and future work

LIST OF FIGURES Page

LIST OF TABLES Page

LIST OF ANNEXURES

GLOSSARY OF TERMS

ANN	Artificial Neural Network
ARMA	Auto Regression Moving Average
BP	Back Propagation
CA	Certificate Authority
CAT	Category
CGI	Common Gateway Interface
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
DSRC	Dedicated Short Range Communication
EAP	Extensible Authentication Protocol
ES	Exponential Smoothing
GM	Gray Model
HTTP	Hyper Text Transfer Protocol
IAS	Internet Authentication Service
IDL	Interface Definition Language
IEEE	Institute of Electrical and Eletronics Engineers
IIOP	Internet Inter-ORB Protocol
IPC	Inter Process Communication
ISP	Internet Service Provider
ITS	Intelligent Transportation System
JDBC	Java DataBase Connectivity
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MSCHAPv2	MicroSoft Challenge Handshake Application Protocol version 2
MA	Moving Average
MAC	Media Access Control
MAPE	Mean Absolute Percentage Error
MIMO	Multiple Inputs Multiple Outputs

MLP	MultiLayer Perceptron
MSE	Mean Square Error
MSMQ	MicroSoft Message Queuing
ODBC	Open DataBase Connectivity
OMG	Object Management Group
ORB	Object Request Broker
PEAP	Protected Extensible Authentication Protocol
PHP	Hypertext Preprocessor
PIC	Programmable Input Controller
RADIUS	Remote Access Dial-In User Service
RBF	Radial Basis Function
RDMS	Relational Database Management System
RF	Radio Frequency
RMSE	Root Mean Square Error
RPC	Remote Procedure Call
RTCC	Road Traffic Control Centre
RTCCDOM	Road Traffic Control Centre Domain
RTS	Request To Send
SANRAL	South African Road Agency Limited
ScTP	Screened Twisted Pair
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SPICE	Simulation Program with Integrated Circuit Emphasis
SSID	Service Set Identifier
SSH	Secure Shell
SSL	Secure Socket Layer
STP	Shielded Twisted Pair
TCP	Transmission Control Protocol
TLS	Transport Level Security
UDP	User Datagram Protocol
URL	Universal Resource Locator
UTP	Unshielded Twisted Pair
VPN	Virtual Private Network

VSM	Virtual System Modelling
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
WAN	Wide Area Network
WAP	Wireless Access Point
WEKA	Waikato Environment for Knowledge Analysis
WEP	Wireless Encryption Protocol
WPA	Wireless Protected Access
XML	Extensible Mark-up Language

CHAPTER 1: INTRODUCTION

1.1 Background

South Africa is currently experiencing traffic jams of unprecedented levels especially in the big cities of Johannesburg and Pretoria and also along the Ben Schoeman highway which connects these cities. The development of alternative routes to link these urban metropolitan areas is hindered by the already established building infrastructure. Consequently, it takes one twice the free flow time to reach the intended destination.

With 3.8 million registered vehicles in Gauteng (Department of Transport 2011:8) there is continual annual traffic increase rate of 7 percent. The increase in traffic in-turn increases the average travel time to work with 50 minutes in Johannesburg. Most industries consequently experience economic uncertainty due to high traffic congestion in the main arteries of Johannesburg.

Congestion may lead to long routes being used to reach the intended destination, while trying to avoid known congested routes along the way. This is costly in terms of fuel consumption. Industry and commerce which depend on haulage services for transportation of goods thus suffer from transport costs. However, businesses tend to factor such costs into their products and services and thus making the prices/cost for South African goods and services globally uncompetitive.

Congestion also imposes costs on industry due to the time spent by employees travelling to work. This day-to day variation in time needed to drive from point A to point B make workers leave 1 to 2 hours earlier for work. On the way back home, they arrive home after 2 to 3 hours later than when there was free flow traffic. Less vehicle traffic congestion would make additional time available for productive work in the office, as well as more time at home with family.

During peak hours (7:00am – 9:00 am), delays due to accidents further increase the total time for the journey by 15 to 30 minutes. Current strategies to warn motorists of a traffic jam, include television and radio traffic broadcast. The main disadvantage of television broadcast is the fact that once someone has left home it is difficult to be up-to-date with information relating to traffic conditions. A radio broadcast to alert road users about the traffic condition on roads is not received by all motorists as not all have radios or will have tuned to that broadcasting station. When the motorist comes out of the traffic jam they try to make up for the lost time by driving fast, thus increasing the likelihood of being involved in an accident.

In traffic jams motor car engines are idling and thus waste fuel and pollutes the air with exhaust gases. Exhaust gases also contribute to global warming. There is also wear and tear hence frequent servicing of vehicles due to constant acceleration and braking which is costly to the motorists and also to the economy.

Currently the multi-billion rand rapid train link called Gautrain has been introduced. This is a stretch of over-ground and under-ground railway spanning 80 km that links the cities of Johannesburg and Pretoria and also links these cities with OR Tambo International Airport. Even though not enough, there is approximately 30% reduction of traffic congestion along Ben Schoeman highway (link between Johannesburg and Pretoria) as the result of the Gautrain project (Gautrain 2012:75). Vehicles can park at a cost of R12.00 per day and vehicle owners can make use of a train to travel to their intended destination (Gautrain 2012:75).

South African National Roads Agency Limited (SANRAL) has installed about 36 electronic tollgates along freeways (N1, N3, N4, N12) in the Gauteng province, whereby all motorists are required to pay a fee at these tollgates (SANRAL 2011:24). The anticipation is that the high cost at these tollgates might force some motorists to use public transport thus resulting in traffic decongestion (SANRAL 2011:4). There are questions about the introduction of tollgates on minor roads and streets to discourage motorists who want to evade tollgates traffic. In the north of Johannesburg, it is difficult to expand routes due to already available buildings near the roads.

Related work on short term vehicle traffic flow prediction models in Bejing China, Xu Ting, SUN-Xiaoduan, HE-Yulong and XIE-Changrong (2009:1006-1007) propose the use of a radial basis function (RBF) neural network and wavelet analysis in speed forecasting of the Beijing urban freeway. The gray model (GM) optimised using a sine function relation to make gray action into a dynamic time variable is proposed (Mao, Chen & Xiao , 2011:271-272) while Wang, Wang and Xia (2005:212-214) use a back-propagation neural network optimised using a genetic algorithm to predict the changing trend of the future. The proposed models successfully predicted dynamic traffic flow with less accuracy than the proposed model in the current study.

Complementing current efforts by SANRAL and Gautrain by building an intelligent system that can predict traffic congestion along the highway will be more beneficial to all road users, the economy of Gauteng and the economy of South Africa than any other strategy that has been employed before.

1.2 Research objective

- To design an intelligent decision making system which can monitor and help to control traffic on the highway thus allowing the motorists to make use of an alternative route in case of looming traffic congestion.
- To identify speed measurement (sensor(s)) methods to use to capture vehicle speeds.
- To identify a technology to use to transmit captured speed vehicle data to a central storage database server.
- Ensure that speed information from speed sensors to remote database management system (DBMS) travels along a secure communication channel thus preventing information from being accessed by hackers.

1.3 Problem statement

Traffic congestion negatively impacts on the lives of the Gauteng people, the economy of Gauteng and that of South Africa as a whole.

1.4 Limitation of the study

In this dissertation installed speed sensors will not be used to help to prosecute speeding drivers along the freeway. Speed sensors will only be used for continual speed monitoring so that the traffic status can be predicted based on the speed pattern captured on the highway.

1.5 Research Methodology

The qualitative research approach will be used and a literature study on the topic will be carried out. A literature survey will be done from books, conference proceedings and journal articles. Experiments on incorporating an intelligent system will be carried out. A database will be developed using a combination of MySQL and java. Experimentation on capturing vehicle speeds using sensors will be done. Wireless experiments and a means to secure information in transit will also be visited. Matlab, WEKA, Proteus and packet tracer toolboxes will be used for simulation and data analysis.

1.6 Layout of the study

Chapter 2 Literature review

Literature on intelligent systems and various intelligent systems that have been used to solve traffic related problems will be revisited.

Chapter 3 Candidate techniques

The tools to be used to solve the problem within the current study will be outlined. Also competing tools will be highlighted.

Chapter 4 Experiments

Experiments will be done using tools selected from chapter 3 and results and recipes will be documented.

Chapter 5 Discussions

Results obtained from experiments (chapter 4) will be discussed.

Chapter 6 Conclusions and Recommendations

Conclusions from the interpretation obtained from experiments done will be drawn. Recommendations for future studies based on the conclusions will be outlined.

1.7 Chapter summary

Congestion imposes an economic and logistical burden on most industries. Using Information Technology strategies to build an intelligent system that can predict looming traffic congestion along the motorway can be beneficial to all road users, the provincial and national economy. An intelligent system needs to eliminate the human element in the decision making process, as it simulates the human expert decision making or reasoning ability. Human beings sometimes make biased decisions.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter presents related work that has been done by other researchers in the field to date.

2.2 Vehicle Traffic Systems

Real time traffic information collected over 24 hours per day using inductive loop detectors is used by Xu *et al*. (2009:1006). A wavelet technology is applied to a signal derived from the loop data in order to remove noise from data. A RBF neural network is used to predict three traffic statuses which include free flow, transition status and congestion according to occupancy. RBF prediction models seem to confuse free flow and transition traffic statuses by predicting the same traffic pattern between 70 to 80 km/h for both.

Due to high uncertainty of traffic flow Mao *et al*. (2011:271) use a gray model theory to forecast short term traffic flow for the system containing incomplete information and uncertainty factors. Mao *et al.* use original traffic data to find the changing laws of the system to generate a strong regularity of the data series, thus predicting the changing trend of the future. The superiority of the used gray model is its ability to do prediction with some of the information known and partial information not known. However Mao *et al*. forecast future trends for the next one hour; using a time series of one hour contributes little towards traffic decongestion since traffic dynamics changes frequently within a period of one hour.

The past seven days' traffic is used to predict the following five days by Wan *et al.* (2005:212). The model is formed by training the artificial neural network with a back-propagation algorithm until a minimum training error is obtained and immediately upon reaching minimum training error, a genetic algorithm is used to search the weight and thresholds of the neural network thus forming an initial population and offspring population. The model shows a success rate of 91 percent. However, upon prediction of congestion on the freeway for the following 5 days Wan *et al*. (2005:212) introduce a human element by appointing a policeman to regulate traffic on the highway under inspection. Introduction of a human element can greatly re-introduce congestion.

In Hong Kong Vibha, Venkatesha, Prasanth, Suhas and Shenoy (2008:3-4) use a background registration technique to detect and classify vehicle types at the intersection. The count of the number of vehicles that enter and exit the intersection is also done. Knowing the number of vehicles that use the intersection is used to help to regulate traffic. A single closed circuit television camera mounted on a pole or other tall structure pointing to the road scene is used. First, the video information is segmented and turned into objects. Secondly, behaviour of these objects is monitored (tracked) for decision making purposes, i.e. vehicle identification and counting. In order to extract information such as vehicle type, speed of each vehicle and count the number of vehicles on the motorway, the video is segmented into foreground object of interest (vehicles) and the background object (road, trees). Even though there are high costs of buying video capturing cameras, the background registration technique used shows 100 percent vehicle detection accuracy and 94 percent accuracy of counting vehicles. Monitoring vehicle traffic with the aid of a video camera used is more suited in applications where there is a need to extract more features, than monitoring traffic such as in cases where vehicles need to be classified according to type (van, bus, truck) as well as tracking of vehicles which make use of a specific lane on the motorway, including tracking vehicles as they interchange lanes amongst others.

In the study conducted by Machine Intelligent Research Labs in Norway (Wang, Wang & Xia 2005:132-134) signals from different sensors are mixed and processed to extract features (axle distance, length of vehicle, height of chassis and occupancy time) helpful to vehicle class (van, truck) and associated speed. Features extracted are then fed into a fuzzy-neural-genetic hybrid controller as inputs which, after processing the information, generates two outputs (vehicle class and speed). A fuzzy logic controller is then used to simulate the output. A fuzzy-neural network approach is used during training. A genetic algorithm is used to search for lesser iterations and thus accurate vehicle classes and speed are identified. However, as the input dimension (characteristics) increases, the fuzzy rule base increases exponentially, which makes the computation cost, memory usage and training data requirements to increase as well.

The incorporation of an inductive loop detector in addition to the use of multiple sensors is done at the University of Science and Technology (Sroka 2004:2235) in Poland. Data fusion methods based on fuzzy logic are used for the vehicle classification process. Signals from two strip piezoelectric sensors and one inductive loop sensor placed on the surface of the pavement are used to evaluate traffic density, time interval between vehicles, voyage, retention times and vehicle velocity of the traffic. The fuzzy logic model with different types of membership functions (triangular and Gaussian) are then used for measurement of vehicles in motion parameters (traffic density, time interval between vehicles, voyage, retention times and vehicle velocity). Classification of two axle vehicles belonging to four classes (cars, delivery vans, lorries and buses) is done using five Triangular and Gaussian membership functions. Data Fusion is done using all five membership functions. With 50 profiles for each vehicle class the triangular membership function showed 92 percent efficiency and the Gaussian membership function provided a better efficiency of 94 percent in identifying whether a vehicle is a car or van or lorry or a bus while capturing different velocities.

In Cairo Egypt Ahmed , Bahaa and Mohamad (2010:29) proposed installation of four sensors (S0–S3) to direct traffic on the crossroads with traffic lights leading to the main road. The first sensor (S0) is installed on the traffic lights directly and the other 3 sensors (S1-S3) are distributed at suitable distances (distance between S1 to S3 is determined according to the studies done by the traffic administration department based on the incoming traffic) prior to the traffic light. Multiple sensors are used to capture the traffic pattern; based on this pattern the system will open for the traffic that is overcrowded and give it a longer time than the given time for other traffic with less density. This technique works well on motorways without exit ramps, since with exit ramps, the traffic density decreases as vehicles make use of available ramps after passing each of the installed sensors.

An emerging Dedicated Short Range Communication (DSRC) technology used for vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) data sharing among different Intelligent Transportation Systems (ITS) entities, is used by the School of Survey and Spartial Information Systems in Sydney Australia (Alam , Balaie & Dempster 2011:1) to determine vehicle speed and lane occupancy for passing vehicles. This is done by broadcasting packets in vehicle passage between two beacons with known coordinates and similar heights on each side of a two lane street opposite each other. Assuming the X axis is a line separating the lanes and the coordinates of the beacons being zero along the X axis, relative acceleration between the vehicles can be used to compute vehicle speeds and a lane which each vehicle is using. The model showed vehicle detection rate of 95 percent for different vehicle speeds. The lane used by each vehicle is always detected true for speeds lower than 40km/h and for speeds higher than 40km/h performance begins to decline.

Ying, Yang and Ying (2008:1080) designed a system which makes use of real time data instead of static data. Probe vehicles instrumented with electronic location and wireless communication equipment are used to collect real traffic data in a wider area. Then multi-source data fusions integrate data and build a data warehouse. The navigation terminal is then used to display a route map, route guidance and assists motorists in route planning. In Germany Rehborn and Palmer (2008:186-187) used information received from probes to generate an onboard traffic state detection which identifies traffic states along a vehicle's trajectory at any time. Kerner's Three Phase Traffic Theory is used to distinguish two different phases in congested traffic: synchronization flow and wide moving jam.

An aggregation approach to short term traffic flow prediction based on moving average (MA), exponential smoothing (ES), auto-regression MA (ARMA) and neural network models is proposed by Man, Wong, Xu, Guan and Zhang (2009:1). These models are used to provide prediction of three relevant time series (weekly, daily, hourly) from 24 hours time traffic volume collected over several years. MA, ES and ARMA models are used to provide prediction. Predictions from these three models are then fed into the neural network for aggregation purposes. An output of the neural network serves as the final prediction. Using the Mean Absolute Percentage Error (MAPE) to measure the accuracy of prediction, an aggregated neural network output shows prediction accuracy of 5.92, 8.44 and 12.21 MAPE for one hour ahead, two hours ahead and three hours respectively during test set 1. Using the second test set 6.84, 10.22 and 11.30 MAPE accuracy is obtained. ARMA results were not satisfactory since 12.56, 22.14 and 33.27 were obtained during the first test set, with 12.63, 22.14 and 33.27 MAPE obtained during the second testing set. These results show neural networks' flexibility when it comes to nonlinear modelling, which makes it one of the best predictive models while the ARMA model's inability to deal with nonlinear relationships was identified.

A kalman filtered-based Cam-shift vehicle tracking algorithm is proposed for video monitoring of highway traffic (Hu Li & Wei 2010: 271-273). The cam-shift algorithm is used to track size, location and speed of target vehicles in real time with high accuracy (track a 3D object moving at the rate of approximately 5cm/s). Knowing the speed of target vehicles allows a proposed algorithm to detect highway conditions accurately. However, the cam-shift algorithm does not work well when the scale of the target is changing or/and when the colour of the target vehicle is changing due to atmospheric conditions. In addition it does not work well also when the background colour becomes similar to the colour of the target vehicle. Furthermore, a cam-shift algorithm fares poorly when the size of the target vehicle changes or when the motor vehicle moves too fast.

2.3 Chapter summary

In this chapter different methods were used to mitigate traffic congestion, these methods include a background registration technique, data fusion, neural networks, fuzzy logic, inductive loop detectors, digital short range communication technology, probe vehicle instrumented devices, moving average, exponential smoothing, auto regression moving average and kalman filter. Some researchers used a combination of different methods in trying to monitor and control motor vehicle traffic status on the highway.

In chapter 3 tools to be used to create the entire network for monitoring and controlling motor vehicle traffic for this study will be identified.

CHAPTER 3: CANDIDATE TECHNIQUES

3.1 Introduction

In this chapter tools to be used in the design and implementation of the system required to monitor and control vehicle traffic on the highway will be identified.

3.2 Wired network technologies

Institutes of Electrical and Electronics Engineers' (IEEE) 802.3 standard for Ethernet technologies will be used to interconnect hosts and network devices within the Road Traffic Control Centre (RTCC). The 802.3 defines different cables with different characteristics which one can take advantage of when linking network devices to the hosts. These cables include fiber optic and the twisted pair.

- **Fiber Optic cable**

The fiber optic cable uses glass or plastic to transmit information using pulses of light. Fiber optic can be found in single-mode or multi-mode (Reid & Lorenz 2008:152-153). A single-mode optic fiber sends a signal using one path from source to destination while multi-mode creates multiple paths. The typical range for a single-mode fiber is 3 km while multi-mode fiber covers 2 km. Currently there are fiber optic cables whether single-mode or multi-mode, that can cover a distance of 10 to 70 km. Because of this longer coverage fiber optic is mostly used to link different Internet Service Providers (ISP) and thus creating the internet backbone. This is made possible due to the fact that fiber optic can run even under the sea. Fiber optic transmission rate ranges from 100 Mbps to 10,000 Mbps. High transmission rate and longer distances covered by fiber optic cable makes it to be twice the cost of Unshielded Twisted Pair (UTP) cable.

- **Twisted pair cable**

The twisted pair cables are copper wires used to transmit information. It makes use of electromagnetic pulses. A twisted pair cable consists of eight wires, grouped into four pairs of twisted copper wires and is used with RJ-45 plugs and sockets (Reid & Lorenz 2008:148). The maximum cable length of a twisted pair is 100 m. Data rates range from 100 Mbps to 10, 000 Mbps.

The three types of twisted-pair cable that exist include the unshielded twisted pair (UTP), shielded twisted pair (STP) and screened twisted pair (ScTP) (Reid & Lorenz 2008:150). Within the current study the UTP cable will be used to link network devices within the wired part of RTCC network. The UTP cable does not have a protective covering (shield) on top of its wires to get rid of interference (noise) compared to STP and ScTP. The UTP is the most used network media for home, office and business networking because it is cheap, flexible and easy to work with. Even though not protected from noise, there are installation guidelines that are followed when installing a UTP cable in order to eliminate chances of interference. Both STP and ScTP have protective covering on top of each wire pair. There is also protective covering on top of all the four pairs making STP and ScTP more suitable to be used in networks where there are more chances of interference due to high voltage sources (transformers, generators), which might be present. The fact that there is covering on top of each wire pair and on top of all the four pairs, makes STP and ScTP difficult to bend inside the conduit during installation. Also the protective covering available in STP and ScTP increases the cost of the two cables. As a result, in this project UTP will be used to connect computers, routers and switches within the RTCC.

The twisted pair cable is split (divided) into categories (CAT). The available category ranges include CAT 3, 5, 5e, 6 and 7. All these categories differ in transmission speeds as can be seen in Table 3.1.

Table 3.1: Twisted pairs cable categories

TWISTED PAIR CABLE CATEGORY	TRANSMISSION SPEEDS
3	16 Mbps
5	100 Mbps
5e	350 Mbps
6	1000 Mbps
7	Greater than 1000 Mbps

The UTP to be used in this project supports 350Mbps and falls within category 5e.

3.3 Wireless network technologies

Wireless technology is governed by IEEE 802.11 standard. The 802.11 standard comprises IEEE 802.11a, 802.11b, 802.11g and the 802.11n.

- **802.11a:**

Operates in the 5.15GHz to 5.35 GHz radio spectrums (Boggia , Camarda, Grieco & Zacheo 2008:1). Most wireless devices use 2.4GHz spectrum hence there are few wireless devices which are 802.11a compliant, i.e. there are few devices which use 5GHz spectrum. The fewer devices result in less traffic congestion in this spectrum thus less interference. The actual throughput of 802.11a is closer to 22Mbps and it restricts wireless devices to within signal coverage of 50 metres. Throughput refers to the volume of data that can flow through a network: 1 Mbps equals 1,048,576 bits. A single page contains 5380 bytes and as a result 1 Mbps equals 1048576 / (5380x8) = 25 A4 pages. 22 Mbps equals 550 A4 pages. The 802.11a standard is not compatible with 802.11b and as a result one cannot use 802.11a and 802.11b devices under the same network. Lastly the components needed to implement 802.11a network are twice as expensive as the price of other wireless devices.

- **802.11b:**

The 802.11b shares the same airspace with cell phones, Bluetooth and security radios amongst others (Boggia *et al*. 2008: 2-3). Many devices use this spectrum hence interference is a main concern. In order to try to get rid of network messages from two or more devices colliding due to many devices competing to use this spectrum, 802.11b uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The 802.11b supports a data rate speed of 11Mbps but due to the CSMA/CA protocol overhead, this speed is decreased to between 5.7 to 7.1 Mbps (178 A4 pages). In CSMA/CA (Boggia *et al*. 2008: 2-3), before any wireless device can send network messages, a wireless device needs to ask for permission to send from the wireless access point by sending a Request To Send (RTS) message. Upon receival of a RTS the wireless access point, because it is the controller of a wireless network, knows if there is any device that is sending at that instant in time. If there is no device sending, the wireless access point then sends a Clear To Send (CTS) message to signal to the wireless device that it can send network messages. RTS and CTS messages happen time and again within wireless networks and as a result increase traffic on already congested spectrum, thus resulting in decreased data rate speed.

- **802.11g:**

The 802.11g operates at a maximum raw rate of 54 Mbps (1350 A4 pages) with a throughput of 24.7

Mbps due to CSMA/CA overhead. The 802.11g shares the same spectrum as 802.11b hence it is backward compatible with 802.11b devices. The presence of an 802.11b device with 802.11g significantly reduces the speed of an 802.11g network. Furthermore there are more chances of collision of network messages from two or more wireless devices when trying to send at the same time (Boggia *et al*. 2008:5) since both the 802.11b and 802.11g use the same spectrum of 2.4 GHz.

- **802.11n:**

The 802.11n builds upon previous 802.11 standards by adding MIMO (multiple-input multiple-output) (Kolahi , Qu , Soorty & Chand 2009:1). MIMO uses multiple transmitter and receiver antennas to allow for increased data throughput through spatial multiplexing and increased range by exploiting the spatial diversity (through coding schemes like Alamouti coding).

The 802.11n devices operate in the 2.4 or 5GHz radio spectrum. They support a speed of up to 700Mbps over a distance of 450m (Kolahi *et al*. 2009:2). The 802.11n is backward compatible with 802.11a, 802.11b, and 802.11g i.e. all these wireless devices can reside in the same network and can communicate with one another.

Due to its compatibility with all wireless standards and higher transmission rates over long distances, the 802.11n will be used in this project to link up vehicle speed sensors so as to be able to transfer vehicle speed information from the highway to the database server located in the wired Local Area Network (LAN).

3.4 Speed Capturing Technologies

New vehicle detection and surveillance technologies now exist (Bhoraskar, Vankadhara, Raman & Kulkarni 2012:1) and they will provide vehicle speed monitoring, traffic counting, presence detection, headway measurement, vehicle classification, and weigh-in-motion data. Vehicle speed monitoring refers to the actual vehicle speed in km/h captured on the highway. Traffic counting is count of number of vehicles using the highway under inspection. Presence detection is the ability of sensors used to detect when vehicles are present on the highway. Headway measurement is a measurement of spaces in between motor vehicles. Vehicle classification is the ability of sensors to distinguish motor vehicles from other objects on the highway including identification of different types of motor vehicles. Weigh-in-motion is a measure of the weight of each vehicle.

Vehicle detection sensors can be classified into intrusive and non-intrusive (Bhoraskar *et al*. 2012:2-4). Intrusive sensors include inductive loops, magnetometers, micro loop probes, pneumatic road tubes, piezoelectric cables and other weigh-in-motion sensors. These devices are installed directly on the pavement surface, in saw-cuts or holes in the road surface, by tunneling under the surface, or by anchoring directly on the pavement surface as is the case with pneumatic road tubes.

Non-intrusive (aboveground) sensors can be mounted above the traffic lane being monitored or on the side of a roadway where they can view multiple lanes of traffic at angles perpendicular to or at an oblique angle to the vehicle flow direction. Currently these sensors include video image processing, microwave radar, laser radar, passive infrared, ultrasonic, passive acoustic array, and combinations of sensor technologies such as passive infrared and microwave Doppler or passive infrared and ultrasonic.

In this project simulation of magnetometer intrusive sensor capturing vehicle speeds on the highway will be done. Magnetometer is chosen in this study to provide basic speed parameters such as volume, presence, occupancy, speed and headway measurement. Unlike inductive loop detectors magnetometers wires can withstand high volumes of traffic and can transmit data over wireless Radio Frequency (RF) links.

3.5 Inter-Process Communication (IPC)

IPC is a set of techniques for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC techniques are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC) (Aung , New , Soe , Naing & Thein 2005:1).

Message passing is the paradigm of communication where messages are sent from a sender to one or more recipients. Forms of messages include (remote) method invocation, signals, and data packets. This kind of IPC technique can be beneficial for the proposed system in the current work during a process where the vehicle speeds captured by multiple vehicle speed sensors on the highway are sent to the storage database.

Process synchronisation or serialisation, strictly defined, is the application of particular mechanisms to ensure that two concurrently-executing threads or processes do not execute specific portions of a program at the same time. If one process has begun to execute a serialised portion of the program, any other process trying to execute this portion must wait until the first process finishes. In the proposed system multiple processes are going to run independently inside different speed capturing sensors and as a result synchronisation or serialisation of processes will not be required.

Shared memory refers to a large block of random access memory that can be accessed by several different central processing units (CPUs) in a multiple-processor computer system. In this project the subroutines will not be executed from a shared memory location but instead will execute inside each vehicle speed sensor placed inside a specific lane. Execution will depend on varying vehicle speed parameters within that specific lane e.g. speed (velocity) depends on time (how long it takes for a motor vehicle to reach point B from point A). Furthermore vehicle speeds on the highway are also influenced by accidents and availability of road workers among others. All these parameters will be considered.

3.5.1 Common Object Request Broker Architecture (CORBA)

Common Object Request Broker Architecture (Chen, Lai & Han 2006: 831) (CORBA) provide**s** a framework for the development and execution of distributed applications (applications residing within different computers). This may be due to the fact that the data, computation and users of such applications are distributed.

- **Distributed data**

Some applications must execute on multiple computers because the data that the application must access exist on multiple computers for administrative and ownership reasons. The owner may permit the data to be accessed remotely but not stored locally. Or perhaps the data cannot be co-located and must exist on multiple heterogeneous systems due to historical reasons.

- **Computation is distributed**

Some applications may execute on multiple computers in order to take advantage of some unique feature of a particular system, for example executing on multiple computers in order to take advantage of multiple processors computing in parallel to solve some problem. Distributed applications can take advantage of the scalability and heterogeneity of the distributed system.

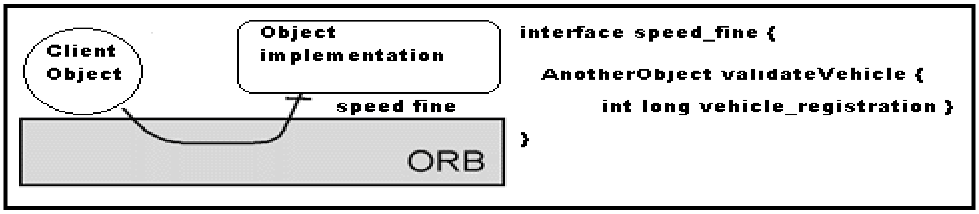
- **Users are distributed**

Some applications execute on multiple computers because users of the application are far apart and thus communicate and interact with each other via the application. Each user executes a piece of the distributed application on his or her computer, and shared objects typically execute on one or more servers.

3.5.1.1 CORBA paradigm

In situations whereby there is a need to connect to other databases such as the traffic department database in order to validate vehicle registration before any speed fine for over speeding vehicles is issued, one can make use of the CORBA paradigm to link different distributed databases. The services that an object provides are given by its *interface* defined in Object Management Groups (OMG) Interface Definition Language (IDL) (Chen *et al*. 2006: 831). Distributed objects are identified by object references, which are typed by IDL interfaces.

Figure 3.1 graphically depicts a request. A request is to issue out a speed fine for over-speeding vehicles. A client holds an object reference to a distributed object. A distributed object refers to a speed fine database and a traffic department database. The object reference is typed by an interface. In Figure 3.1 the object reference is typed by the `speed_fine` interface. The `speed_fine` interface line calls for speed fine database where fines are issued. The Object Request Broker, or ORB, delivers the request to the object and returns any results to the client. Before a fine is issued, in Figure 3.1, a `validateVehicle` line calls for the traffic department database to validate vehicle registration and returns an object reference typed by the `AnotherObject` interface. `AnotherObject` interface is an interface within speed_fine interface.



```
interface speed_fine {
  AnotherObject validateVehicle {
          int long vehicle_registration }
}
```

Figure 3.1: CORBA implementation

- **The Object Request Broker (ORB)**

Communication between a client-to-server or server-to-server is done using Internet Inter ORB Protocol (IIOP). The ORB is the distributed service that implements the request to the remote object (Chen *et al*. 2006:832). It locates the remote object on the network, communicates the request to the object, waits for the results and when available communicates those results back to the client. A remote object in Figure 3.1 is a remote traffic database called by `AnotherObject` interface i.e. an instance which needs to run from a remote location. ORB is being used in thousands of commercial products at many companies, as well as at various universities and research labs around the world, in many types of distributed, real-time, and embedded systems, particularly telecom, medical, aerospace, defence, and financial services.

The ORB implements location transparency. Location transparency means that the same request mechanism is used by the client and the CORBA object regardless of where the object is located. The client cannot tell the difference between local or remote object. The object might be in the same process with the client, down the hall or across the planet. The client issuing the request can be written in a different programming language from the implementation of the CORBA object. The ORB acts as a middleware, it does the necessary translation between programming languages (Chen *et al*. 2006:833).

With CORBA (Ingram , Ress & Norman 2006:5), to open up a hole in a firewall, one has to assign a port number to each server and then punch a hole into the firewall for that port (add each port used by each server to the firewall exceptions settings so as to allow communication to pass through the firewall). A firewall refers to a software or hardware based program used to filter incoming and outgoing network messages. Using CORBA for communications between organisations across the internet and through firewalls is today practically impossible or a nightmare because of the usual firewall configurations which only allow traffic on known ports (like Hyper Text Transfer Protocol (HTTP) 80, Simple Mail Transfer Protocol (SMTP) 25) but deny traffic which makes use of Transmission Control Protocol (TCP) on dynamic ports (ports between ranges 1024 to 49151. Ports ranging from 1024 to 49151 are mostly used by clients as source ports and by some server applications as destination ports to allow application access to clients. CORBA is mostly used in server-to-server communications or client-to-server communication inside the LAN because of its weakness of a need to open a firewall port to allow end to end communication across all routers especially when the clients are scattered across the internet.

3.5.2 Simple Object Access Protocol (SOAP)

SOAP is a simple XML-based protocol that allows applications to exchange information over HTTP (Tekli , Damiani , Chbeir & Gianini 2012:3). SOAP provides a middleware to communicate between applications running on different computers using different operating systems, with different technologies and programming languages. SOAP codifies the use of XML as an encoding scheme for request and response parameters using HTTP as a transport. A SOAP method is simply an HTTP request and response that complies with the SOAP encoding rules. A SOAP endpoint is simply an HTTP-based Universal Resource Locator (URL) that identifies a target for method invocation. Like CORBA, SOAP does not require that a specific object be tied to a given endpoint. An endpoint refers to a path which shows how to reach an intended object. Rather, it is up to the implementor to decide how to map the object endpoint identifier onto a server-side object.

With SOAP all data is converted to XML from human readable format. This conversion to XML and back to human readable format takes more processing power than with IIOP from CORBA. SOAP requires 10 to 15 times the bandwidth of IIOP and, if one use structures or unions extensively, using SOAP can easily end up consuming 50 to 100 times the network bandwidth (Tekli *et al*. 2012:8).

SOAP is designed to connect together programs running on different machines. Because no security is required in HTTP, XML, or SOAP, this may result in security of the computer system being compromised. In SOAP XML messages are delivered using HTTP which makes use of port 80. By putting everything through port 80, the entire load is placed on the single web server with another process that listens on that port. In this way SOAP cannot scale as much as other protocols that use separate ports because the single server process at port 80 eventually becomes the critical bottleneck.

3.5.3 Microsoft Message Queuing (MSMQ)

Microsoft Message Queuing (MSMQ) is mostly used in cases where there is a possibility that the other communicating process may not be available (Gupta, Pattander & De 2012:192-193). This can be due to network unavailability especially in wireless networks where due to network congestion the nearby communication circuits could interfere with the network signal required by a particular client resulting in the server process becoming unavailable. Figure 3.2 shows messages from different computers received by the MSMQ server. In Figure 3.2 when the other communicating process is not available, applications send messages to a message queue. The delivery of queued messages will occur when the process on the other end wakes up and receives the notification of the message's arrival.

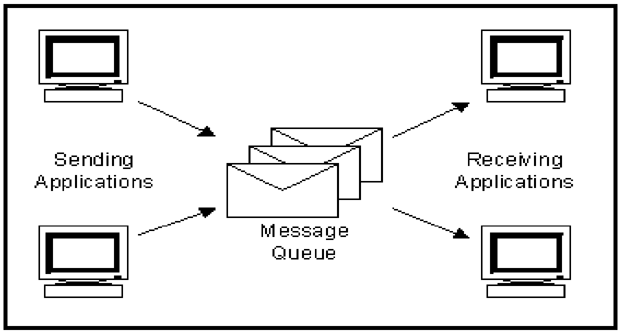


Figure 3.2: MSMQ message queuing

MSMQ guarantees message delivery and provides an efficient way of routing messages between processes. MSMQ also allows the use of priority in processing messages. In this project the use of MSMQ can be beneficiary in cases where weekly or monthly predictions are required since all processes will have enough time to read all queued messages. As for real-time predictions (prediction of the next 5 minutes based on historical 5 minutes data) MSMQ will not work well when other processes are not available. Unavailable processes could result in valuable traffic information being missed.

3.5.4 MAILSLOT

Mailslot is a RAM based file which resides inside a server (Wikipedia 2013:1). The mailslot in the server provides a server-client interface. Here servers create a mailslot and clients write to this mailslot. Many client programs can write records to a particular mailslot simultaneously. The messages are "queued" in the order that they arrive at the mailslot. These messages are stored in the server's RAM until such time as the server program read them. The server program can also simultaneously be reading records from the mailslot while the clients are writing to it. The records are retrieved in the same order that they are queued. In this study vehicle speed sensors installed on the highway can be tailored to send vehicle speed information to the mailslot. With mailslot as soon as the server and client programs "close down" the mailslot they are using, the mailslot file is automatically deleted from the memory of the server computer. In predicting traffic status for the next 5 minutes based on the last 5 minutes' historical data there is data that is reused from the previous 5 minutes which needs to be always available. Using temporary memory where deletion of records is done when disconnection is experienced will not work well in the current project. Sometimes network outages may cause disconnection of the link between a client and server programs while the two are exchanging the records, thus deleting the memory with records which would have been important when doing the next prediction.

Mailslots are generally a good choice when one process must broadcast a message to multiple processes (Wikipedia 2013:1). Even though easy to implement, mailslots do not operate over Wide Area Networks (WAN) as the internet. Also mailslots offer no confirmation that a message has been received unless it is programmed into an application.

3.5.5 Java sockets

A socket is a software endpoint that establishes bidirectional communication between a server program and one or more client programs (Minqiang 2012:610). A server is a computer which house network resources on behalf of other computers in a network. A client is a computer which requests access to network resources located inside a server computer. In this project a server will be an RTCC-database server where vehicle speed data will be stored. A client will be vehicle speed sensors which are used to capture vehicle speeds on the highway. The software endpoint refers to descriptors which need to be included within an application program which maps connection between two communicating hosts. The socket associates the server program with a specific port on the machine where it runs so that any client program anywhere in the network with a socket associated with that same port can communicate with the server program as shown in Figure 3.3. Figure 3.3 shows clients' server socket communication where a server program is associated with port 9999. When client programs want to communicate with the server they send their requests using client programs running in their computers to port 9999 of the server computer.

A server program typically provides network resources to client programs. An example of a resource provided by a server in the current traffic jam project will be database storage to be used by a client to store vehicle speed information. Client programs send requests to the server program, and the server program responds to the request by providing access to the resource requested. An example of a client request in this traffic jam project happens during sending of vehicle speed data by multiple vehicle speed sensors; a request to store these vehicle speeds to the remote database storage is made by vehicle speed sensors on the highway.

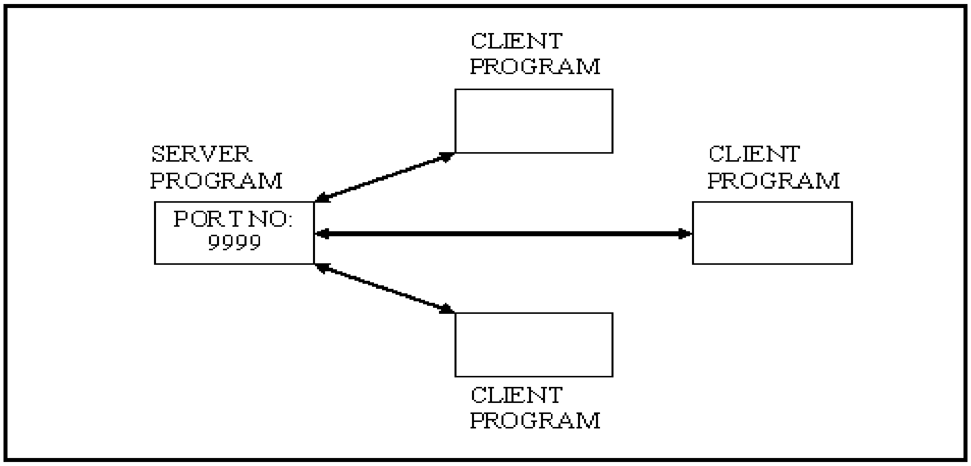


Figure 3.3: Client and a server java sockets interaction

One way to handle requests from more than one client is to make the server program multi-threaded. A multi-threaded server creates a thread for each communication it accepts from a client. A thread is a sequence of instructions that run independent of any other threads.

Unlike CORBA, messages sent using java sockets can pass through the router. Firewall allows java socket messages to pass without any problems i.e. there is no need to add a port used by the server inside the firewall exceptions. Java sockets also allow one to make use of any port above 1024 to 65536 as a destination port to be used by a server to listen and accept network messages (Minqiang 2012:611). In SOAP before any message can be sent it needs to be translated to XML format. Again before any message can be read it needs to be re-translated back to human readable format. This conversion and reconversion required by SOAP consumes a lot of time which can be used efficently to send or receive as many network messages as possible.

Furthermore using sockets reduces the network traffic. Unlike HTML forms and Common Gateway Interface (CGI) scripts that generate and transfer whole web pages for each new request, Java applets can send only necessary updated information and thus making network bandwidth available for other applications (Guillermo , Tourino , Doallo , Lin & Han 2009:332). As a result of simplicity in using java to program sockets and all mentioned advantages, in the current work java sockets will be used to send vehicle speed messages to the storage RTCC-database server.

Two communication protocols that can be used for socket programming are datagram communication and stream communication (Thanh & Urano 2010:971). The datagram communication protocol known as User Datagram Protocol (UDP) is a connectionless protocol. A connectionless protocol does not need a handshake and as a result you will never know if the message was delivered successfully. UDP needs a device to initiate the communication process required to send a local socket descriptor and the receiving socket address every time for communication to occur. UDP's competitor is Transmission Control Protocol (TCP) which is connection-oriented. In order to do communication over the TCP protocol, a connection must first be established between the two socket pairs (client and server). While one socket listens for connection request (server), the other asks for connection (client). Once the two sockets have been connected, they can be used to send data in both directions i.e. from client to a server or vice versa (TCP needs handshaking). Handshake introduces delays in a computer network because of a wait period needed to establish a connection before the two hosts can continue to communicate. But once a connection is set up network messages are guaranteed to be delivered to a destination.

Java sockets messages will be configured to use TCP during message delivery thus ensuring that all messages are successfully delivered to the storage RTCC-database server. The fact that for every message sent using TCP the sender obtains an acknowledgement from the receiver and in case where the sender sends a message but does not receive an acknowledgement from the receiver, the sender can re-send any undelivered message to the receiver. Using TCP guaranties that all the socket messages (vehicle traffic data) will successfully be delivered to the storage database server (RTCC-database server). Whereas in UDP undelivered (lost) messages are not re-transmitted to the receiver.

3.6 Database storage

Traffic data received from vehicle speed sensors will be stored inside a MySQL server at the RTCC. A MySQL database server provides the ultimate in scalability. MySQL has the ability of handling applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. MySQL runs on platforms that include Linux, UNIX, and also Windows. The open source nature of MySQL allows complete customization which allows for adding unique requirements to the database server. Other features of MySQL (Ngamsuriyaroj & Pornpattana 2010:1037-1038) includes, high performance, high availability, robust transactional support, web and data warehouse strengths, strong data protection, comprehensive application development, ease of management, open source freedom and 24 hours, 7 days in a week availability of technical support.

The unique storage-engine architecture allows configuration of the MySQL database server specifically for particular applications. Thus for a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet all these demanding performance expectations. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms (Kaitsa , Stavrakas , Kontogiannis , Daradimos , Panaousis & Triantis 2007:546-547), MySQL offers all the right ammunition for today's critical business systems. MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialised web functions like fast full text searches (Kaitsa *et al*. 2007:547-548). These same strengths also apply to data warehousing environments where MySQL scales up into the terabyte range for either single servers or scale-out architectures.

Other features (Yassin , Taib , Adnan , Khairul & Hamzah 2012:263-266) of MySQL like main memory tables, B-tree and hash indexes, and compressed archive tables that reduce storage requirements by up to 80% make MySQL a strong candidate for both web and business intelligence applications. Customers rely on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

Data is a valuable asset for corporations. MySQL offers exceptional security features that ensure absolute data protection (Zoratti 2006:188). In terms of database authentication, MySQL provides powerful mechanisms for ensuring that only authorised users have entry to the database server, with the ability to block users down to the client machine level being possible. Secure Shell (SSH) and Secure Socket Layer (SSL) support are also provided to ensure safe and secure connections. A granular object privilege framework (Zoratti 2006:192) is present so that users only see the data they are authorised to see and powerful data encryption and decryption functions ensure that sensitive data is protected from unauthorised viewing. Finally, backup and recovery utilities (Zoratti 2006:193) provided through MySQL and third party software vendors allow for complete logical and physical backup as well as full and point-in-time recovery.

Last but not least, MySQL provides connectors and drivers (Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), etc.) that allow all forms of applications to make use of MySQL as a preferred data management server (Lin , Kwee & Tsai 2009:1). It does not matter if it is Hypertext

Preprocessor (PHP), Perl, Java, Visual Basic, or .NET, as MySQL offers application developers everything they need to be successful in building database-driven information systems. MySQL (Lin , Kwee & Tsai 2009:2) also provides a complete suite of graphical management and migration tools that allow a DataBase Administrator (DBA) to manage, troubleshoot, and control the operation of many MySQL servers from a single workstation. Many third party software vendor tools are also available for MySQL that handle tasks ranging from data design to complete database administration, job management and performance monitoring.

There are other Relational Database Management Systems (RDMS) such as Oracle and Microsoft Access which could have been used in this work. The Oracle package is not cheap at all hence it is mostly used in large organisations since it can handle large amounts of data (petabytes). Oracle can be very complex and difficult to administer although it makes robust database systems (Shahamiri , Nasir & Ibrahim 2010:30). There is almost nothing that one cannot do in Oracle. But applications are not developed nearly as quickly.

The Microsoft access package is very cheap (it is part of Office Professional) and very easy to use. Most of the work in Access is done through wizards and GUI tools. One can quickly develop a small, single user database. But Microsoft Access is not very good for a multi-user application (Majstorovic, Siranovic & Kavran 2012:1290-1291). Furthermore it is not very robust and does not have good multi-user transactional control. In addition, Access cannot handle large amounts of data. All of the data in the database is stored in a single file which is a major limitation.

3.7 Predicting traffic status

There are different tools which other researchers have used to control or predict traffic congestion discussed in the literatrure review chapter (Chapter 2). It is evident that most of the tools do not have problems when it comes to forecasting but they all differ in the way they are used. In order to forecast traffic status within the current study, a supervised multilayer perceptron neural network which makes use of sequential back-propagation learning method will be used. A neural network (Haykin 1994:1) is a parallel distributed processor that stores experiential knowledge and makes this knowledge available for use in future when a novel instance that comes from the same distribution as data used to construct the Artificial Neural Network (ANN) is encountered.

Neural networks have a remarkable ability to derive meaning from complicated or imprecise data, extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations (novel data or instance) of interest. These situations must come from the same distribution as the data that was used to construct the neural network.

A neural network resembles the brain in three respects:

- Knowledge is acquired by the network through a learning process.
- Interconnection strengths known as synaptic weights (in Figure 3.4) are used to store the knowledge.
- It has generalisation ability.

Learning is a process by which the free parameters (i.e. synaptic weights and bias levels) of a neural network shown in Figure 3.4 are adapted through a continuing process of stimulation by the environment in which the network is embedded (Haykin 1994:45). The type of learning is determined by the manner in which the parameter changes take place. In general the learning process may be classified as follows:

- Supervised learning (learning with a teacher):

  In supervised training (Haykin 1994:57), both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs (actual outputs) against the desired outputs.

- Unsupervised learning (learning without a teacher):

  It is also referred to as self-organisation (Haykin 1994:65), in the sense that it self-organises data presented to the network and detects their emergent collective properties. It is based upon only local information.

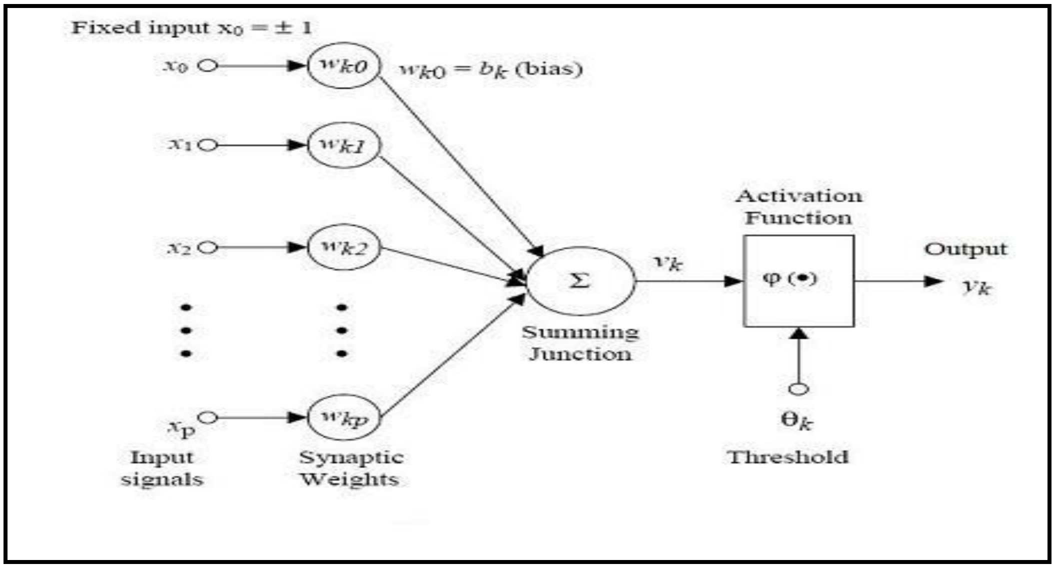


Figure 3.4: An artificial neural network model (adapted from ”Multilayer Perceptron”, by Haykin 1994)

3.7.1 Supervised learning

This form of learning assumes the availability of a labeled (i.e. desired outputs) set of training data made up of *N* input and *N* output examples shown in equation (3.1):

$$T = \{(x_p, d_p)\}_{p=1}^{N} , \qquad (3.1)$$

where x_p = input vector of the P^{th} example

d_p = desired (target) response of the P^{th} example

N = sample size

Given the training sample *T*, the requirement is to compute the free parameters of the neural network so that the actual output y_p of the neural network due to x_p is close enough to d_p for all *p* in a statistical analogy. The mean-square error (MSE) formula (3.2) is used as the index of performance to be measured.

$$MSE\ (n) = \tfrac{1}{N} \sum_{p=1}^{N} (d_p - y_p)^2 . \qquad (3.2)$$

The network is trained continuously (*n* number of times) until the MSE in equation (3.2) is minimal.

Even though most artificial neural network models use the MSE to measure the performance of the model, the square in MSE equation (3.2) puts a very high weight on large deviations (Saigal & Mehrotra 2012:61). In order to address appearance of these high weights during large deviations, in this work the average of the mean square errors (Root Mean Square Error (RMSE) in equation (3.3)) will be used to evaluate the performance of each of the prospective models during the process of selecting the best model.

$$RMSE = \sqrt{MSE} \tag{3.3}$$

In averaging the errors at individual instance, the RMSE provides a better way of measuring the performance.

3.7.1.1 Multilayer Peceptron (MLP)

The multilayer perceptron (Witten & Frank 2005:223-229) is a form of feed-forward neural network which makes use of back-propagation (errors are fed back into the system) during the learning phase. A MLP in Figure 3.5 has an input layer (P1, P2) where different input patterns are provided to the network of source nodes (x_1 - x_k) and an output layer (y_1 - y_p) of nodes (i.e. computation nodes).

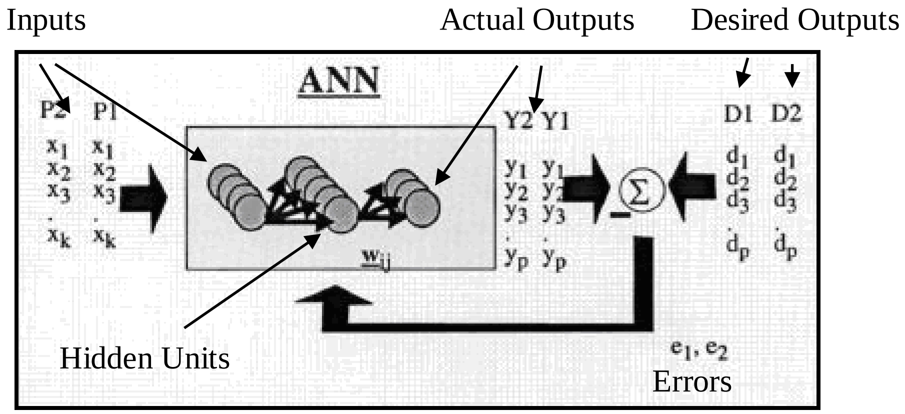


Figure 3.5: Multilayer perceptron model (adapted from "machine learning tools and techniques", by Witten & Frank 2005)

This network of two layers connects the network to the outside world. In addition to these two layers, the MLP has one or more layers of hidden neurons because these neurons are not directly accessible from the outside world. The hidden neurons extract important features contained in the input data and store these features.

The back propagation algorithm used by MLP during training involves two phases:

- *Forward phase*: During this phase the free parameters of the network are fixed, and the input signal is propagated through the network layer by layer. The forward phase finishes with the computation of an error signal. Equation (3.4) is used to determine an error signal.

  $$e_p = d_p - y_p , \tag{3.4}$$

  where d_p is the desired output

  and y_p is the actual output produced by the network in response to the input x_k in Figure 3.5.

- *Backward Phase.* During this phase, the error signal e_p is propagated through the network of Figure 3.5 in the backward direction, hence the name of the algorithm. It is during this phase that adjustments are applied to the free parameters of the network so as to minimize the error e_p in a statistical way.

Back-propagation learning may be implemented in one of two basic ways:

- *Sequential mode* (also referred to as the on-line mode or stochastic mode): In this mode of back-propagation (BP) learning, adjustments are made to the free parameters of the network on an example-by-example basis. The sequential mode is best suited for pattern classification (Witten & Frank 2005:230-232).

- *Batch mode*: In this mode adjustments are made to the free parameters of the network on an epoch-by-epoch basis, where each epoch consists of the entire set of training examples. The batch mode is best suited for nonlinear-regression (Witten & Frank 2005:230-232).

ANN tools contained in the Waikato Environment for Knowledge Analysis toolbox known as ‘WEKA’ will be used to carry out the traffic prediction experiment. The MLP from WEKA will be used as a classification method.

3.7.2 WEKA workbench

The WEKA workbench (Witten & Frank 2005:365) is a collection of machine learning algorithms and data preprocessing tools. It is designed so that users can quickly try out existing machine learning methods on new datasets in very flexible ways. It provides extensive support for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualising both the input data and the result of learning.

3.8 Displaying traffic status on the highway

As soon as the traffic status has been predicted, status information needs to be displayed to the motorists on the highway. In order to convey this traffic information, Proteus Virtual System modelling (VSM) toolbox will be used to design a display board to be installed on the simulated highway so that motorists can know about the traffic status of the route they are currently using.

3.8.1 Proteus Virtual System Modelling (VSM)

Proteus VSM (Su & Wang 2010:375-377) combines mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. It allows anyone to develop and test designs before a physical prototype is constructed. One can interact with the design using on screen indicators such as LED and LCD displays and actuators such as switches and buttons. The simulation takes place in real time. The Proteus VSM also provides extensive debugging facilities including breakpoints, single stepping and variable display for both assembly code and high level language sources.

An important feature of the Proteus VSM is its ability to simulate the interaction between software running on a microcontroller and any analog or digital electronics connected to it. The micro-controller model sits on the schematic along with the other elements of the product design. It simulates the execution of the object code (machine code), just like a real chip. If the program code writes to a port, the logic levels in the circuit change accordingly, and if the circuit changes the state of the processor's pins, this will be seen by the program code, just as in real life.

The VSM CPU models fully simulate I/O ports, interrupts, timers, Universal Serial ARTs and all other peripherals present on each supported processor. The VSM can even simulate designs containing multiple CPUs, since it is a simple enough matter to place two or more processors on a schematic and wire them together.

Using the Proteus VSM also contributes towards the monitoring and control part of the current project in that, during traffic monitoring the Proteus VSM will display the next 5 minutes predicted traffic status of the highway under inspection. Based on the predicted traffic status motorists will eventually make use of available alternative routes (control part of the project) so that they could avoid traffic congestion.

3.9 Securing RTCC network wireless links

In wireless networks network messages are transmitted through air, making network messages vulnerable to interception. Once messages have been intercepted different kinds of threats may arise including data loss, data manipulation, identity theft or denial of service attacks (Zhang 2012:59). Valuable information may be erased through data loss. Records may be edited through data manipulation. Someone may perform transactions as if those transactions were done by the original owner through identity theft. Access to network resources may be compromised through denial of service attack. In order to protect wireless networks from these threats, extremely strong authentication and encryption techniques need to be implemented.

IEEE 802.11 standard (Bachan & Singh 2010:792) identified different techniques which one could use to prove identity (authenticated) before access to a wireless network can be established. These techniques include use of a pre-shared key (secret word) between a wireless client and Wireless Access Point (WAP) whereby a wireless client needs to supply a key to a WAP before a client is allowed to connect. If a key supplied by a wireless client matches a key configured inside the WAP then a wireless client is allowed to connect. The main disadvantage of this technique is the fact that it performs a one-way authentication, that is, the wireless client authenticates to the WAP. In a pre-shared key environment the WAP is not authenticated if the user of the wireless client is not authenticated as well. A WAP might be spoofed and the user of a wireless computer might as well be a hacker.

IEEE 802.11 then defined Extensible Authentication Protocol (EAP) (Bachan & Singh 2010:793) as a more secure protocol to use to authenticate between a wireless client, WAP and the user of a wireless computer. EAP provides mutual or two-way authentication as well as user authentication. When EAP software is installed on the wireless client, the client communicates with a backend authentication server such as a Remote Authentication Dial-in User Service (RADIUS). This backend server functions separately from the WAP and maintains a database of valid users that can access the network. When using EAP, the user, not just the wireless computer, must provide a username and password which is checked against the RADIUS database for validity. If valid, the user is authenticated.

In this work the latest version of EAP, Protected Extensible Authentication Protocol (PEAP) will be used. PEAP supports all features of EAP and also creates a secure protected channel where authentication information is exchanged. PEAP works together with other protocols in order to encrypt a channel to be used.

PEAP uses Transport Level Security (TLS) to create an encrypted channel between an authenticating PEAP client, such as a wireless computer, and a PEAP authenticator, such as an Internet Authentication Service (IAS) or RADIUS server. PEAP does not specify an authentication method, but provides additional security for other EAP authentication protocols, such as EAP-Microsoft Challenge Hand shake Application Protocol version 2 (MSCHAPv2), that can operate through the TLS encrypted channel provided by PEAP. PEAP is used as an authentication method for 802.11 wireless client computers, but is not supported for virtual private network (VPN) or other remote access clients (Bachan & Singh 2010:794).

To enhance both the EAP protocols and network security, PEAP (Hoeper & Chen 2010:416-420) provides:

- Protection for the EAP method negotiation that occurs between client and server through a TLS channel. This helps prevent an attacker from injecting packets between the client and the IAS server to cause the negotiation of a less secure EAP method. The encrypted TLS channel also helps prevent denial of service attacks against the IAS server.
- Support for the fragmentation and reassembly of messages.
- Wireless clients with the ability to authenticate the IAS or RADIUS server. Because the server also authenticates the client, mutual authentication occurs.
- Protection against the deployment of an unauthorised wireless access point (WAP) when the EAP client authenticates the certificate provided by the IAS server. In addition, the TLS master secret

key created by the PEAP authenticator and client is not shared with the access point. Because of this, the access point cannot decrypt the messages protected by PEAP.

- PEAP fast reconnect reduces the delay in time between an authentication request by a client and the response by the IAS or RADIUS server, and allows wireless clients to move between access points without repeated requests for authentication. This reduces resource requirements for both client and server.

3.9.1 PEAP authentication process

There are two stages in the PEAP authentication process (Hoeper & Chen 2010:420-421) between the PEAP client and authenticator. The first stage sets up a secure channel between the PEAP client and the authenticating server. The second stage provides EAP authentication between the EAP client and authenticator.

3.9.1.1 Setting up a secure channel using TLS

When the wireless client associates with a wireless access point, the 802.11-based association is required. This association provides an Open System or Shared Key Authentication before a secure association is created between the client and access point. After the IEEE 802.11-based association is successfully established between the client and access point, the TLS session is negotiated with the access point. After authentication is successfully completed between the wireless client and the server (IAS), the TLS session is negotiated between the wireless client and IAS. The key that is derived during this negotiation is used to encrypt all subsequent communication.

3.9.1.2 EAP-authentication between EAP client and authenticator

Complete EAP communication, including EAP negotiation, occurs through the TLS channel. The IAS server then authenticates the user and client computer with the method that is determined by the EAP type and selected for use within PEAP (either EAP-TLS or EAP-MS-CHAPv2). The access point only forwards messages between the wireless client and the RADIUS server. The access point cannot decrypt these messages because it is not the TLS end point.

3.9.2 Deploying 802.11 wireless network using PEAP

PEAP can be tailored to use one of the two EAP types (Hoeper & Chen 2010:422-423): PEAP with EAP-MS-CHAPv2 or PEAP with EAP-TLS. EAP-MS-CHAPv2 uses credentials (user name and password) for user authentication, and a certificate in the IAS server computer for RADIUS server authentication. EAP-TLS uses both certificates installed in the client computer or a smart card for user and client computer authentication, and a certificate in the IAS server computer for RADIUS server authentication.

In this project the wireless link between vehicle speed sensors pointing on the highway capturing vehicle speeds (wireless client) and wired LAN where remote traffic control database (server which stores vehicle speeds received from speed sensors) will be secured using PEAP with EAP-TLS. EAP-TLS makes use of Public Key certificates which provide a much stronger authentication method than EAP-MS-CHAPv2 which uses password-based credentials. Furthermore, the Media Access Control (MAC) address filtering, disabling of the Service Set Identification (SSID) broadcast will be done inside the WAP to make it even more difficult for a hacker to gain access using the available wireless medium. In MAC address filtering MAC addresses of allowed wireless clients (vehicle speed sensors) which can connect to the wireless network will be added to the MAC address database inside the WAP. Disabling the SSID broadcast provides another layer of security since the name of the created wireless network which connects speed sensors to the database server will not be visible to others.

3.10 Chapter summary

Combination of wired and wireless technologies will be used to build the entire RTCC network. Randperm will simulate magnetometer vehicle speed sensors capturing vehicle speeds on the highway. Once vehicle speeds have been captured, java sockets will be used to send this speed information wirelessly to MySQL database server. Java database connectivity driver (JDBC) will be used to retrieve vehicle speed data stored inside MySQL. MLP will analyse the retrieved vehicle speed data and predict the future traffic status. Once future traffic status has been predicted, it will be sent wirelessly to an LCD display board designed for simulation purposes (not for the real design) using Proteus VSM for the motorists to read. PEAP with EAP-TLS will be used to secure wireless links between vehicle speed sensors and RTCC-database server (MySQL database server is located in the wired part of the RTCC LAN).

CHAPTER 4: EXPERIMENTS

4.1 Introduction

In this chapter experiments were carried out using tools selected from chapter 3. These tools include simulated magnetometer sensors, java sockets, MySQL, Multilayer perceptron (MLP) and a Programmable Input Controller (PIC) microcontroller. These tools were used to perform the following four points which are also captured by Figure 4.1.

1. Capture vehicle speeds on the highway.
2. Transfer captured vehicle speeds to a remote database management system for storage.
3. Manipulate data and predict traffic condition for the monitored highway.
4. Display predicted traffic status to the motorists

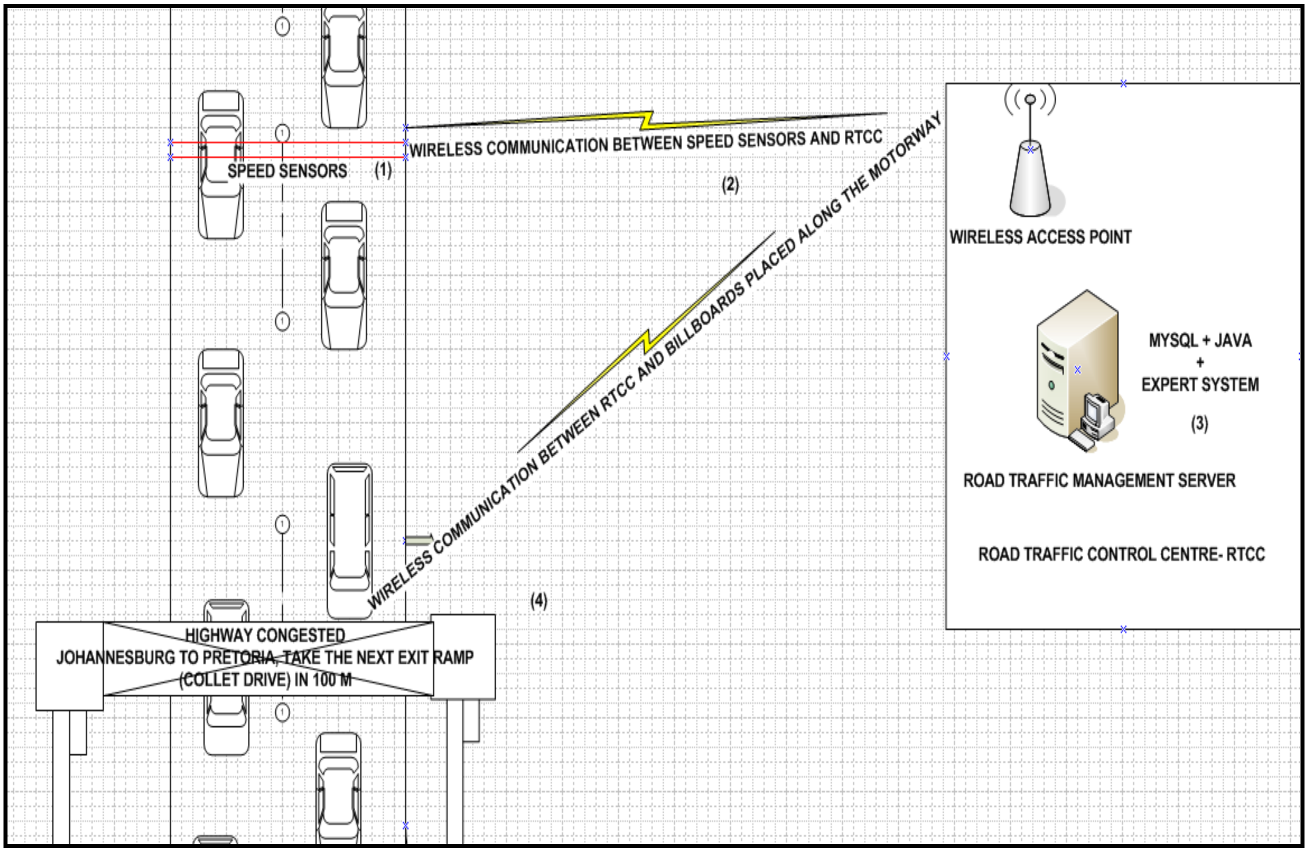

SPEED SENSORS
(1)
WIRELESS COMMUNICATION BETWEEN SPEED SENSORS AND RTCC
(2)
WIRELESS COMMUNICATION BETWEEN RTCC AND BILLBOARDS PLACED ALONG THE MOTORWAY
(4)
WIRELESS ACCESS POINT
MYSQL + JAVA
+
EXPERT SYSTEM
(3)
ROAD TRAFFIC MANAGEMENT SERVER
ROAD TRAFFIC CONTROL CENTRE- RTCC
HIGHWAY CONGESTED
JOHANNESBURG TO PRETORIA, TAKE THE NEXT EXIT RAMP
(COLLET DRIVE) IN 100 M

Figure 4.1: The vehicle traffic prediction model

4.2 Experiment 1: Building Road Traffic Control Centre (RTCC) network

The entire Road Traffic Control Centre (RTCC) network was constructed using both wired and wireless technologies as shown by a packet tracer physical topology diagram in Fig 4.2.

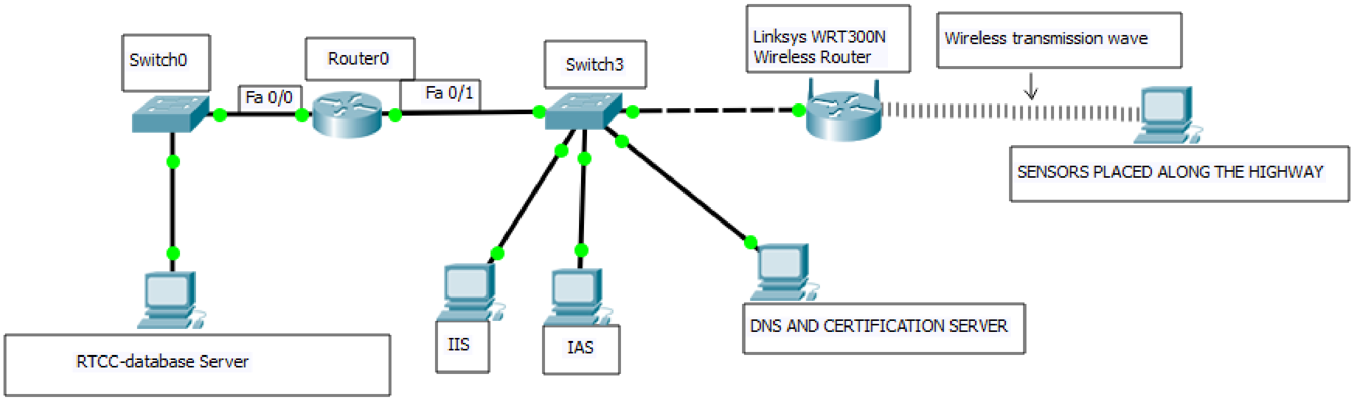
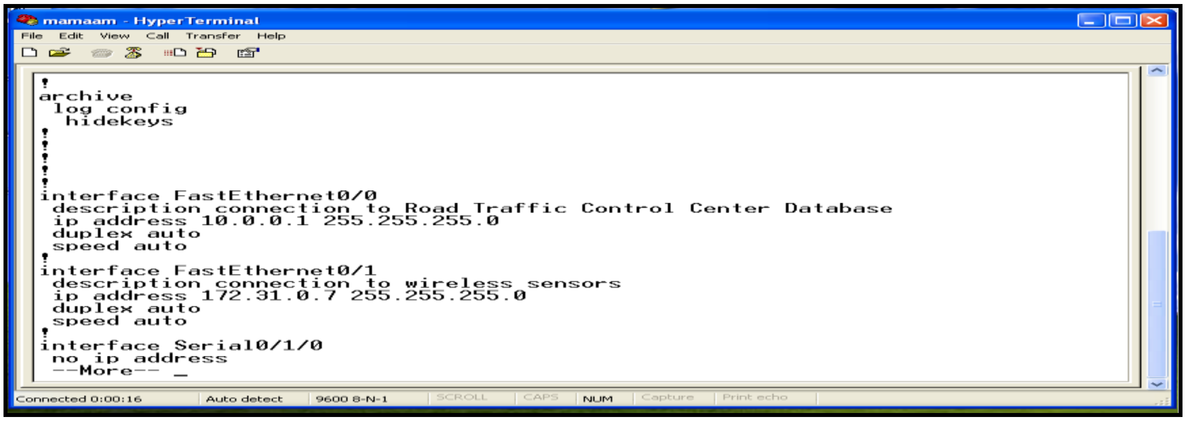


Figure 4.2: The proposed RTCC network

Servers that include the Domain Name System (DNS), Internet Authentication Service (IAS), Internet Information Service (IIS) and the certification servers used for authentication and authorization of wireless clients (vehicle speed sensors placed along the highway) are shown in the central part of the RTCC network in Figure 4.2. In Figure 4.2 RTCC-database Server is used to store motor vehicle speeds received from vehicle speed sensors (shown in Figure 4.2) placed along the Ben Schoeman highway. The RTCC-database server and the DNS server are linked by Fa 0/0 and Fa 0/1 which are Fast Ethernet ports of Router0.

The straight-through cable represented by a solid line and a crossover cable represented by a dashed line were used to interconnect different devices in Figure 4.2. A wave in Figure 4.2 represents wireless links. Vehicle speed sensors were linked wirelessly using a wireless router (Linksys WRT300N). A wireless router was connected to switch3 (2950T) which is linked to authentication and authorisation servers and is also linked to Router0 in Figure 4.2. In this experiment Router0 was used to connect two different subnets using fast Ethernet (Fa) 0/1 (subnet 172.31.0.0 /24) and Fa 0/0 (subnet 10.0.0.0 /24) as can be seen by Router0 running configuration file in Figure 4.3. All computers in Figure 4.2 which are connected to Router0 using Fa 0/1 thus belong to subnet address 172.31.0.0. These computers are IIS, IAS, DNS, certification server and vehicle speed sensors. RTCC-database server in-turn connects to Router0 using Fa 0/0 and as a result belongs to subnet address 10.0.0.0

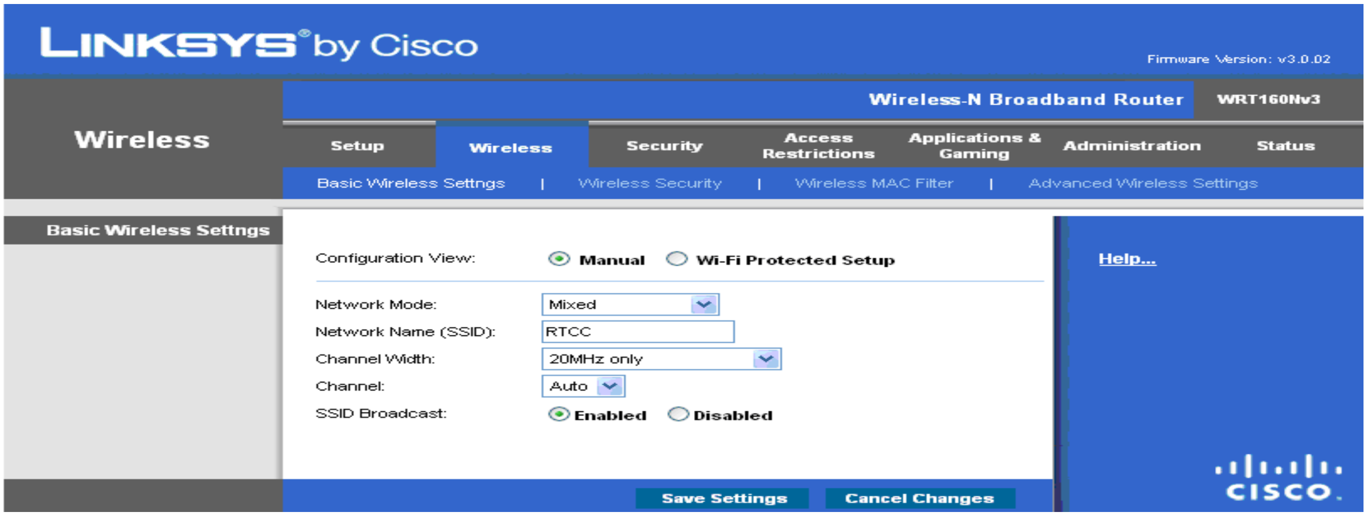

mamaam - HyperTerminal
File Edit View Call Transfer Help
Connected 0:00:16 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

```
!
archive
 log config
  hidekeys
!
!
!
!
interface FastEthernet0/0
 description connection to Road Traffic Control Center Database
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description connection to wireless sensors
 ip address 172.31.0.7 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/1/0
 no ip address
 --More-- _
```

Figure 4.3: Router0 subnets shown by the running configuration file

The purpose of using two subnets in a network is to allow validation of vehicle speed sensors to occur within a subnet address 172.31.0.0 where authentication and authorisation servers reside before the vehicle speed data is sent. Once validation is successful and the vehicle speed sensors are connected to the RTCC network, only then is vehicle speed data transmitted to the RTCC-database server located in subnet address 10.0.0.0. The subnet addresses (refer to annexure A for detailed information) used in this experiment were taken from a private range of network addresses reserved to be used internally within Local Area Networks (LAN) as indicated by Request For Comment (RFC) 1918 (Rekhter & Li 1993:4).

Linksys wireless router 1 shown in a network diagram in Figure 4.2 was configured to beam a wireless signal (with a network name: RTCC) shown in Figure 4.4. Vehicle speed sensors use the "RTCC" wireless network name to gain access to the entire RTCC network.



Configuration View: ● Manual ○ Wi-Fi Protected Setup

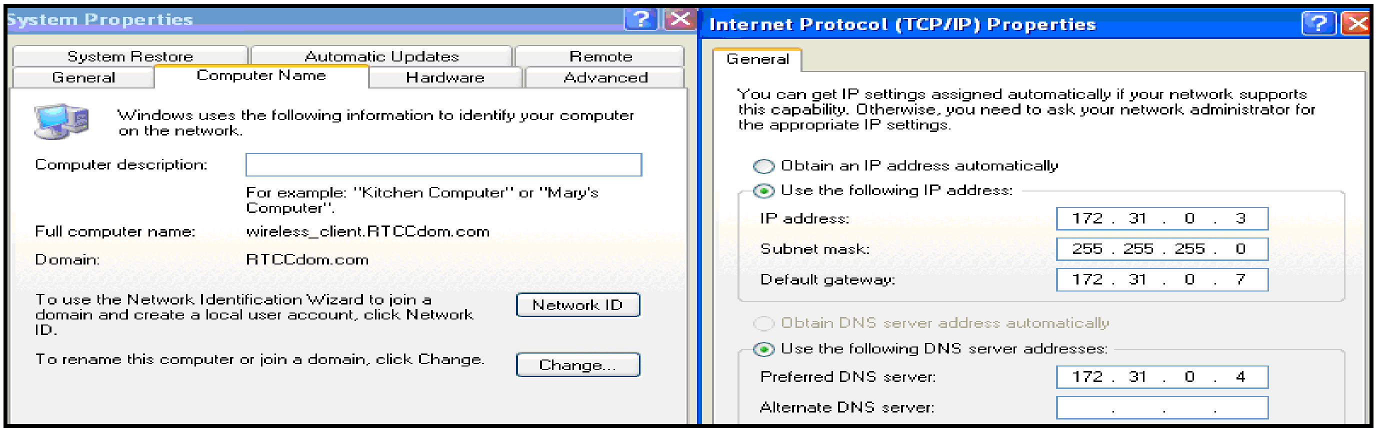
Network Mode: Mixed
Network Name (SSID): RTCC
Channel Width: 20MHz only
Channel: Auto
SSID Broadcast: ● Enabled ○ Disabled

Figure 4.4: Configuring wireless network with network name: RTCC

In this experiment the servers (IIS, IAS, DNS and Certification server) at the RTCC were used to authenticate wireless speed sensors before any speed information could be transmitted to the storage database server (RTCC-database server).

After physical connections were made between different network components, in order for these components to be able to send network messages between one another IP addresses were assigned. “My network connections” inside the windows control panel was double clicked. In the overall scheme shown in Figure 4.2, the wireless speed sensor (wireless client) resides on the wireless part of the RTCC network. Double clicking network connections present a window where all network adapters (local area connections and wireless connections) installed inside the computer could be seen. In the case of a wireless client it was accessed using a mouse by right clicking on wireless network connection, then selecting properties to assign an IP address to a wireless client (vehicle speed sensor) resulting in Figure 4.5.

A static IP address shown in Figure 4.5 was assigned to the TCP/IP properties of vehicle speed sensor (wireless client) installed on the highway.



Windows uses the following information to identify your computer on the network.

Computer description:

For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: wireless_client.RTCCdom.com

Domain: RTCCdom.com

To use the Network Identification Wizard to join a domain and create a local user account, click Network ID.

To rename this computer or join a domain, click Change.

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 31 . 0 . 3

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 172 . 31 . 0 . 7

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

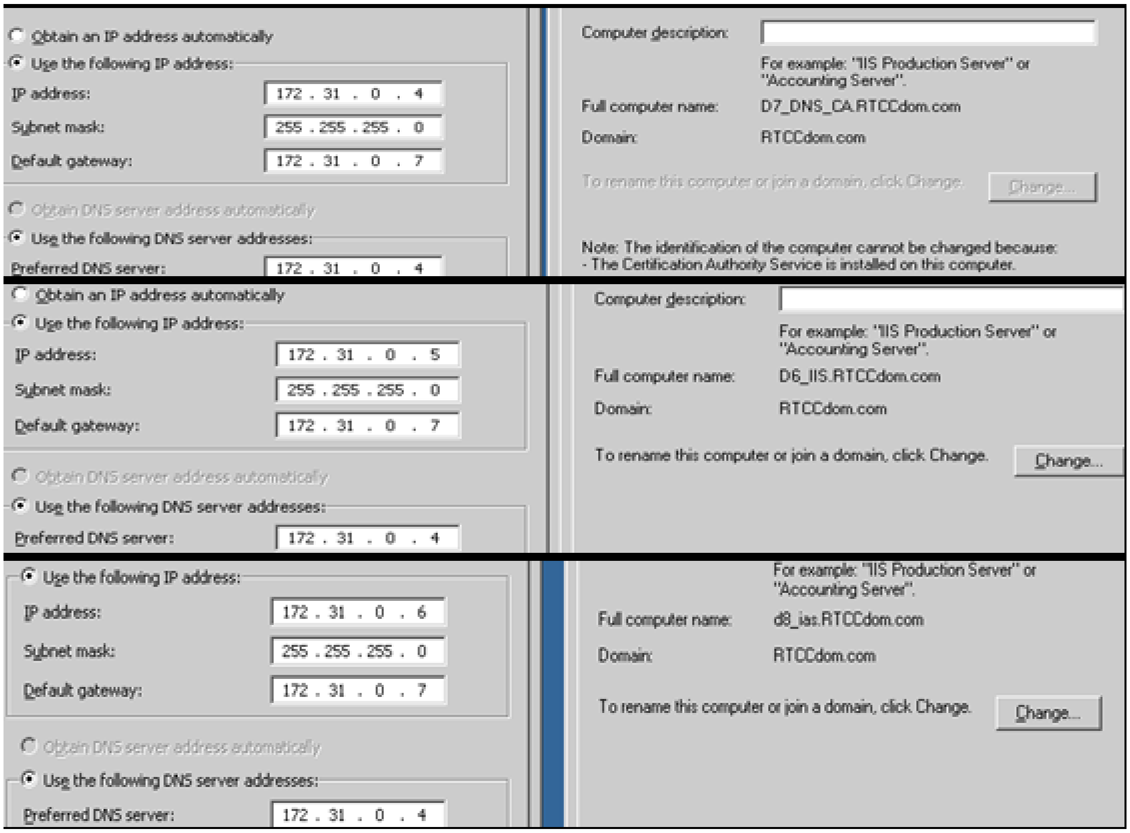
Preferred DNS server: 172 . 31 . 0 . 4

Alternate DNS server: . . .

Figure 4.5: IP address assigned to a vehicle speed sensor (wireless client)

The same procedure in Figure 4.5 was repeated in order to assign IP address settings shown in Figure 4.6 to the DNS and certification server, IIS and IAS respectively (authentication and authorization servers located on the centre of RTCC network shown in Figure 4.2). However the local area connection was used instead of a wireless network connection because the DNS, certification server, IIS and IAS were connected to the network using physical UTP network cables on the wired part of the RTCC network.

The IP addresses were assigned in a sequence from the lowest to highest. The first IP address 172.31.0.1 of network 172.31.0.0/24 was assigned to the wireless router. The second IP address (172.31.0.2) was assigned to switch3 Virtual Local Area Network (VLAN) 1 interface. Assigning an IP address to switch3 VLAN 1 interface will ease management and maintenance of this switch by allowing any remote computer within the RTCC network to be used to access this switch. The IP addresses were assigned in that sequence incrementing the host value by 1 from the preceding host as can be seen in Figure 4.6. Thus the wireless client (vehicle speed sensor) was allocated the next available IP address 172.31.0.3 since the first two IP addresses (172.31.0.1 and 172.31.0.2) were already used by the wireless router and switch3 VLAN 1 interface respectively. In Figure 4.6 the DNS and certification server, IIS and the IAS server were all assigned IP addresses following the same incremental sequence after the wireless client IP address.



○ Obtain an IP address automatically
● Use the following IP address:
IP address: 172 . 31 . 0 . 4
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 172 . 31 . 0 . 7
○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 172 . 31 . 0 . 4

Computer description:
For example: "IIS Production Server" or "Accounting Server".
Full computer name: D7_DNS_CA.RTCCdom.com
Domain: RTCCdom.com
To rename this computer or join a domain, click Change. Change...
Note: The identification of the computer cannot be changed because:
- The Certification Authority Service is installed on this computer.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 172 . 31 . 0 . 5
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 172 . 31 . 0 . 7
○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 172 . 31 . 0 . 4

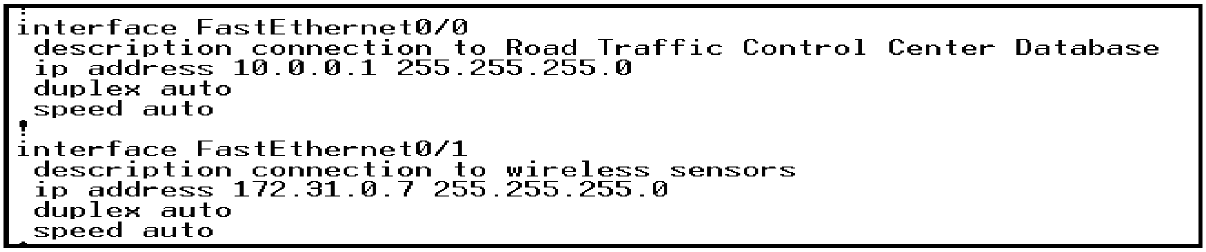
Computer description:
For example: "IIS Production Server" or "Accounting Server".
Full computer name: D6_IIS.RTCCdom.com
Domain: RTCCdom.com
To rename this computer or join a domain, click Change. Change...

● Use the following IP address:
IP address: 172 . 31 . 0 . 6
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 172 . 31 . 0 . 7
○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 172 . 31 . 0 . 4

For example: "IIS Production Server" or "Accounting Server".
Full computer name: d8_ias.RTCCdom.com
Domain: RTCCdom.com
To rename this computer or join a domain, click Change. Change...

Figure 4.6: IP addresses assigned to the DNS, IIS and IAS

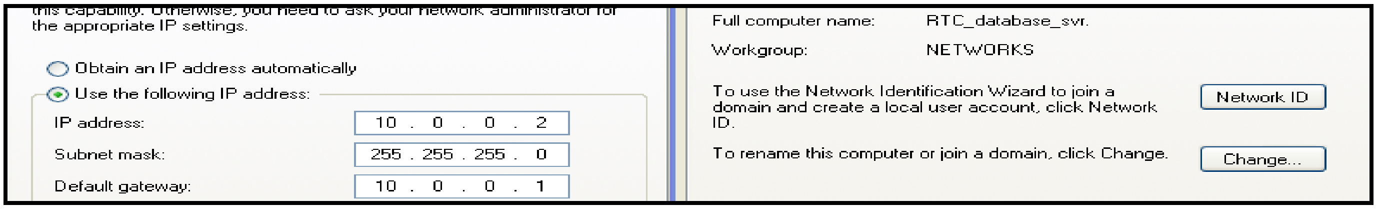
The same procedure used to determine the IP address ranges for subnet 172.31.0.0/24 outlined in annexure A was used to identify IP address ranges to be used for subnet 10.0.0.0/24 where RTCC-database server resides. The IP address ranges from 10.0.0.1 to 10.0.0.254 were obtained. These first two IP addresses (10.0.0.1 and 10.0.0.2) were used by Router0 Fa 0/0 and the RTCC-database server respectively. Router0 has two Fast Ethernet interfaces which provide links between subnets 10.0.0.0 connected to Fa 0/0 and subnet 172.31.0.0 connected to Fa 0/1, in Figure 4.3, these interfaces were configured with IP addresses 10.0.0.1 and 172.31.0.7 respectively. Router0 Fa 0/1 was allocated the seventh IP address of subnet 172.31.0.0 (172.31.0.1 to 172.31.0.254) which was the next available IP address after allocating the IAS server in Figure 4.6 with 172.31.0.6. Router 1 global configuration mode was used to assign these IP address settings with a detailed description indicating the purpose of each of these two interfaces. Assigned IP addresses are shown inside the running configuration file of Router0 in Figure 4.7. The two interfaces shown in Figure 4.7 (Fa0/0 and Fa 0/1) are used by a Router0 when it delivers messages from vehicle speed sensors from the Ben Schoeman highway to the RTCC-database server. Router0 uses the Fa 0/1 to receive the messages and deliver such messages to the RTCC-database server using Fa 0/0.



```
!
interface FastEthernet0/0
 description connection to Road Traffic Control Center Database
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description connection to wireless sensors
 ip address 172.31.0.7 255.255.255.0
 duplex auto
 speed auto
!
```

Figure 4.7: IP addresses assigned to Router0 Fast Ethernet interfaces

The RTCC-database server located on the Left Hand Side in Figure 4.2 was allocated the next available IP address 10.0.0.2 from subnet address 10.0.0.0/24 since 10.0.0.1 was already assigned to Fa 0/0 as shown in Figure 4.7. The Router0 Fa 0/0 IP address became a default gateway for the RTCC-database server when receiving network messages from 172.31.0.0/24 network, which is where wireless speed sensors resides. This whole IP address assignment including the default gateway to the RTCC-database server is shown in Figure 4.8.



this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 0 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 10 . 0 . 0 . 1

Full computer name: RTC_database_svr.

Workgroup: NETWORKS

To use the Network Identification Wizard to join a domain and create a local user account, click Network ID.

To rename this computer or join a domain, click Change.

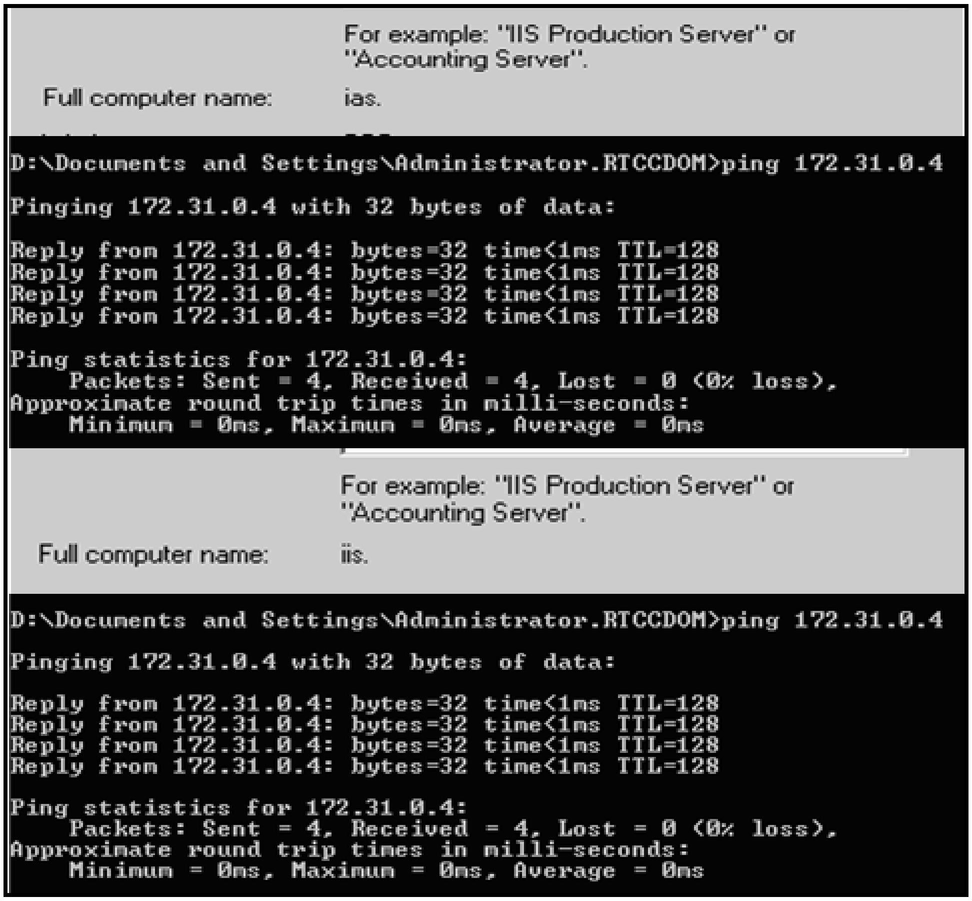
Figure 4.8: RTCC-database server IP address

4.2.1 Results

At the RTCC network the DNS server was used to manage user accounts, computer accounts and permissions allocation. Thus in experiment 1 connectivity tests were done once all connections were completed in order to test if all computers can reach the DNS so they could join the DNS domain thus becoming members of the DNS server controlled database.

4.2.1.1 IAS server, IIS server, RTCC-database server and wireless client connectivity tests

The ping utility was used to test reachability to the DNS server by vehicle speed sensors (wireless client), IIS server, RTCC-database server and IAS server as shown in Figure 4.9 and Figure 4.10. In Figure 4.9 and Figure 4.10 the IP address 172.31.0.4 is the IP address of the DNS server as shown in Figure 4.6. The ping command was executed from the command prompt of all the 4 computers to test connection to the DNS server (that is why the IP address of all ping screens is 172.31.0.4).



For example: "IIS Production Server" or "Accounting Server".

Full computer name: ias.

```
D:\Documents and Settings\Administrator.RTCCDOM>ping 172.31.0.4

Pinging 172.31.0.4 with 32 bytes of data:

Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 172.31.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

For example: "IIS Production Server" or "Accounting Server".

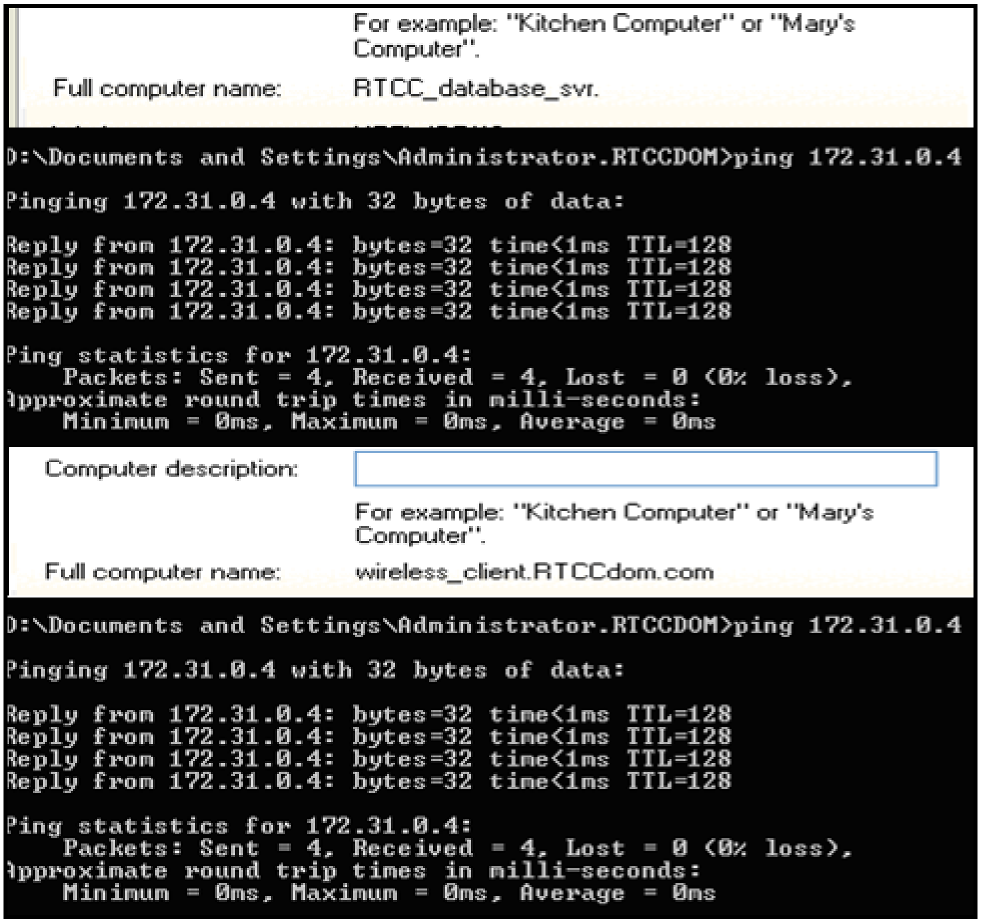
Full computer name: iis.

```
D:\Documents and Settings\Administrator.RTCCDOM>ping 172.31.0.4

Pinging 172.31.0.4 with 32 bytes of data:

Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 172.31.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 4.9: The IAS and IIS servers' connectivity tests results

The Internet Control Message Protocol (ICMP) is a protocol used in computer networks to send error and control network messages between communicating network devices. The Ping utility used in this experiment uses ICMP to deliver messages between communicating devices. The results shown in Figure

4.9 and Figure 4.10 showed that there was end to end connectivity between the DNS server, IAS server, IIS server, wireless client and RTCC-database server. This is indicated by the replies being received from the DNS server as shown in Figure 4.9 and Figure 4.10. The 4 ICMP request messages sent from IAS server, IIS server, RTCC-database server and wireless client (vehicle speed sensors) respectively were successfully delivered to a DNS server (DNS server using IP Address 172.31.0.4) which in-turn replied with 4 ICMP echo replies to each computer as shown by the number of messages sent and received (Sent = 4, received = 4, lost = 0) by each computer.



For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: RTCC_database_svr.

```
D:\Documents and Settings\Administrator.RTCCDOM>ping 172.31.0.4

Pinging 172.31.0.4 with 32 bytes of data:

Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 172.31.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
pproximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Computer description:

For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: wireless_client.RTCCdom.com

```
D:\Documents and Settings\Administrator.RTCCDOM>ping 172.31.0.4

Pinging 172.31.0.4 with 32 bytes of data:

Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 172.31.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
pproximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 4.10: The RTCC-database server and wireless client connectivity test results

The results in Figure 4.9 and 4.10 shows that the computers in a wired and wireless part of the RTCC network are able to communicate with one another. The appropriate cables were used to connect different network components. Router0 was correctly configured to direct messages from one network segment to the other. Correct IP address settings were provided to all devices that form the RTCC network. All computers can now be able to join the domain controller and become members of the RTCC domain controller database.

4.3 Experiment 2: vehicle speed capturing

In this experiment a Matlab M-File was used to simulate capturing of vehicle speeds in each lane of incoming traffic. The Equation 4.1 was used to calculate vehicle speeds.

$$v = \frac{\Delta s}{\Delta t}, \qquad (4.1)$$

Where v is the velocity of cars in respect to change in displacement (s) over change in time (t).

A simulated magnetometer loop detector was used to capture motor vehicle speeds. A magnetometer loop detector uses two cables to capture vehicle speeds along the simulated highway. These cables were separated by distance s_0 to s_1 = 5m, as shown in Figure 4.11.

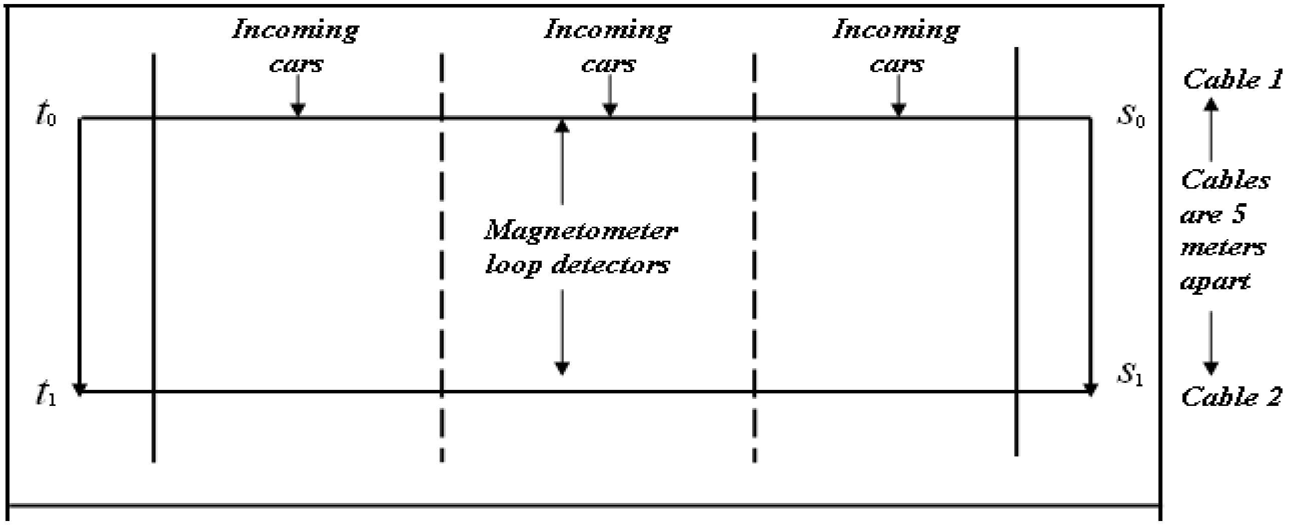
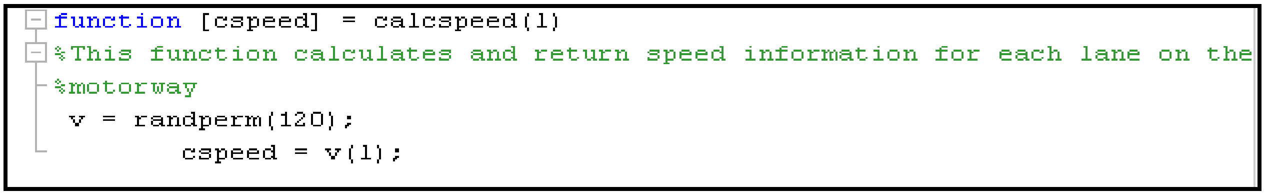


Figure 4.11: Simulated magnetometer loop detectors on the highway

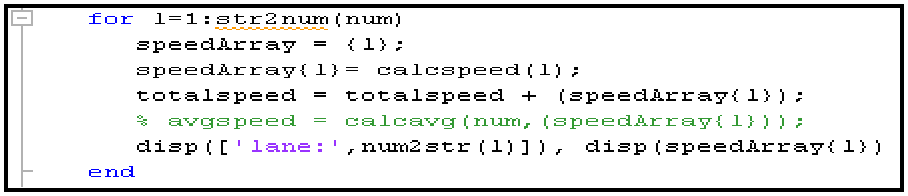
With a known distance between two cables shown by s_0 and s_1 in Figure 4.11, the time it takes for a vehicle to reach s_1 from s_0 could be determined. In this experiment due to unavailability of real world data a random function was used to determine the time t_1 (which is the time it takes to traverse from s_0 to s_1) and hence the velocity of a car as it exits s_1. With known distance between two cables and the time t_1 it takes to exit, the velocity of each car could be computed. Since both vehicle velocity (v) and time it takes to reach cable 2 (t_1) are random numbers, a random function randperm, which is included as one of the functions within Matlab, was used to provide different speeds ranging between 0 to 120.4 km/h. The reason for this is the fact that the speed of cars on the highway takes a random form e.g. without notice the speed of vehicles increases and without notice suddenly decreases or even get to a halt. This random function could mimic the vehicle speed being captured on the highway using random generated numbers.



```
function [cspeed] = calcspeed(l)
%This function calculates and return speed information for each lane on the
%motorway
 v = randperm(120);
        cspeed = v(l);
```

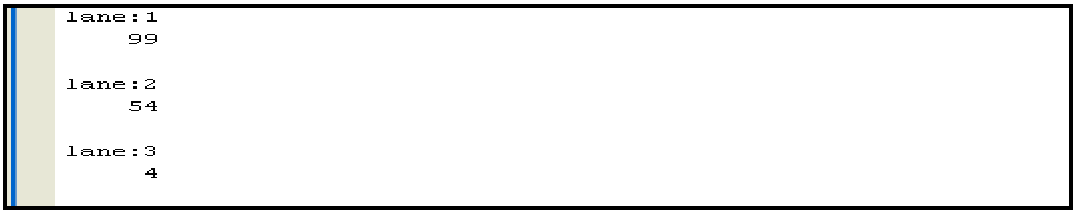
Figure 4.12: Random speed generator function

Randperm in Figure 4.12 was called within another function viz. calcspeed. Calcspeed is a function which when called inside the main function (wireless speed sensor) returns generated random vehicle speed using a variable csspeed as shown in Figure 4.13.



```
    for l=1:str2num(num)
        speedArray = {l};
        speedArray{l}= calcspeed(l);
        totalspeed = totalspeed + (speedArray{l});
        % avgspeed = calcavg(num,(speedArray{l}));
        disp(['lane:',num2str(l)]), disp(speedArray{l})
    end
```

Figure 4.13: The vehicle speed sensor calling calcspeed function from the main function



```
lane:1
    99

lane:2
    54

lane:3
     4
```

Figure 4.14: Vehicle speeds generated using randperm matlab function

The results of different vehicle speeds shown in Figure 4.14 show the vehicle traffic pattern as it happens along the simulated Ben Schoeman highway in Gauteng.

4.4 Experiment 3: Java sockets

Now that the link has been set up between vehicle speed sensors and RTCC network servers, message passing was used to transfer vehicle traffic data from vehicle speed sensors on the highway wirelessly to the database server located in the wired part of the RTCC network.

The vehicle speeds generated in experiment 2 were forwarded to the database control server using java sockets. On the proposed model in Figure 4.2 message passing using java sockets was configured between a wireless client (sensors placed on the highway) and RTCC-database server as shown in Figure 4.15.

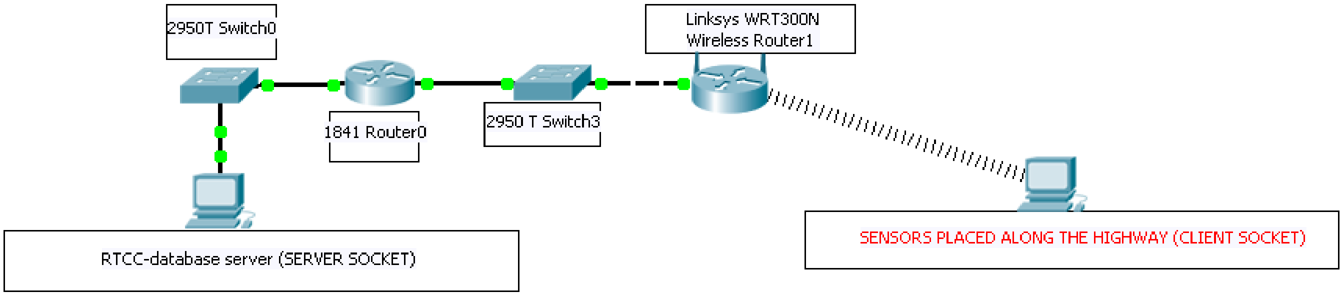

2950T Switch0
Linksys WRT300N
Wireless Router1
1841 Router0
2950 T Switch3
RTCC-database server (SERVER SOCKET)
SENSORS PLACED ALONG THE HIGHWAY (CLIENT SOCKET)

Figure 4.15: Computers within the RTCC network where java sockets were programmed

Equipment and software used

This experiment was carried out using the following equipment and software:

- 2 Lenovo Pentium 4 dual core E8400 with 3 GHz CPU and 2 GB of RAM
- 802.11n Cisco Linksys wireless router
- 2800 Cisco router
- 2 x 2950 Cisco switches

Operating Systems installed inside the computers

- Windows 2003 enterprise server with service pack 1 (RTCC-database server)
- Windows XP version 5.1.2600 service pack 2 build 2600 (wireless client)

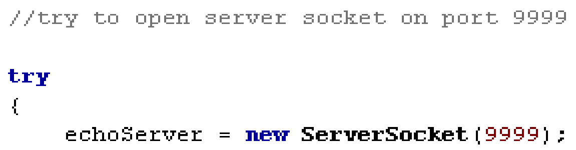
In a client-server architecture it is necessary for a server to know which service is being requested by a client. A server in this experiment is the RTCC-database server. A client is a wireless client (vehicle speed sensors). The RTCC-database server must know when a wireless client (vehicle speed sensors) requests access to it. In order for the RTCC-database server to know that vehicle speed sensors want to send vehicle speed data, the RTCC-database server uses a destination port number. This destination number is included in vehicle speed data sent from vehicle speed sensors. Internet Corporation for Assigned Names and Numbers (ICANN) manages port numbers used over computer networks in order to distinguish a service being requested or provided. ICANN has broken down ports into well-known ports, registered ports and private ports (Reid & Lorenz 2008:208).

Well-known ports are port numbers which can only be used as destination ports by client computers. These ports (in the range of 1 to 1023) are associated with common network applications such as internet service and e-mail services among others.

Registered ports are ports in the range of 1024 to 49151. These ports can be used by different organizations to register other services provided by servers. For example services such as Instant Messaging and socket messages.

Private ports are ports within the range of 49152 to 65536 and are used as source ports by client computers.

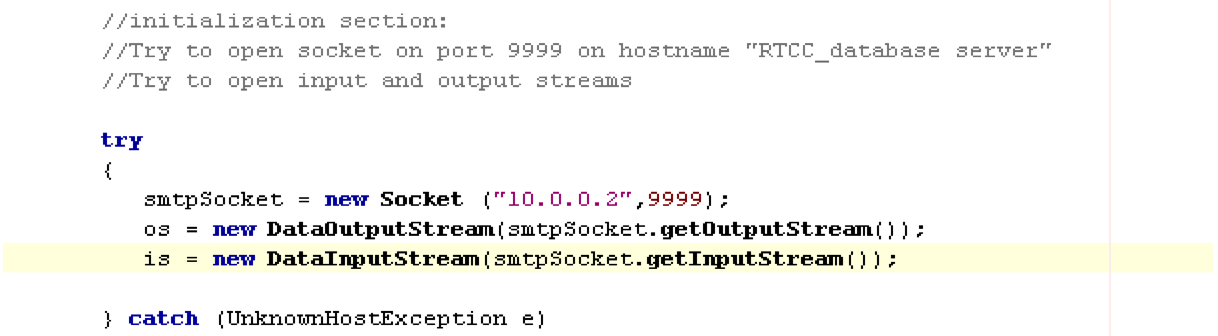
In this experiment port number 9999 was used, which falls within a range of registered ports as a destination port to be used by wireless speed sensors when sending vehicle speed data using socket messages. In Figure 4.16 port 9999 was opened inside the RTCC-database server shown in Figure 4.15 and put in a listening state waiting for any messages from vehicle speed sensors.



```
//try to open server socket on port 9999

try
{
    echoServer = new ServerSocket(9999);
```

Figure 4.16: Creating a server socket object

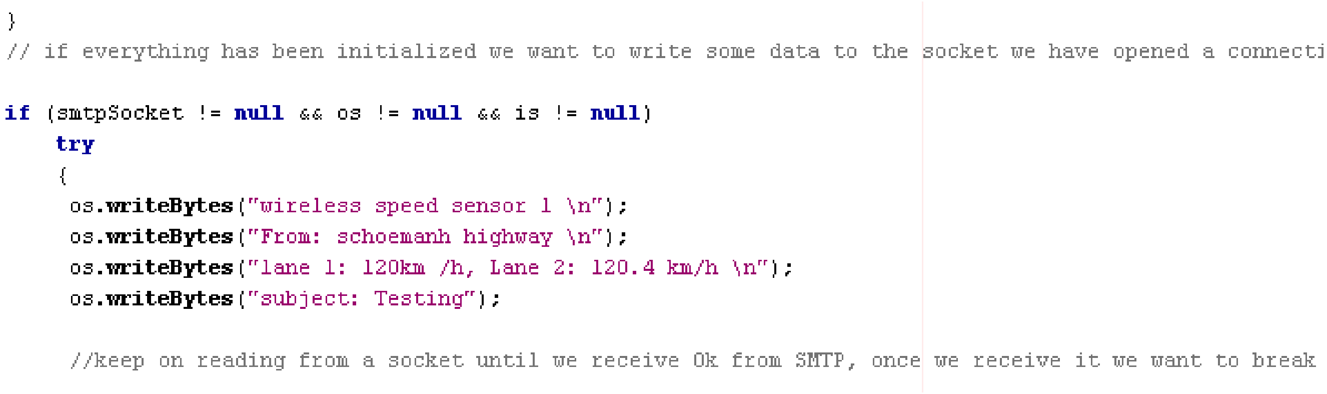
On the client side in Figure 4.17 a wireless client (vehicle speed sensor) was configured to connect to port 9999 of the RTCC-database server (identified by the IP address in Figure 4.8) already opened in Figure 4.16. At the same time the input-output streams were initialized so that a wireless client could start to send vehicle speeds.



```
//initialization section:
//Try to open socket on port 9999 on hostname "RTCC_database server"
//Try to open input and output streams

try
{
   smtpSocket = new Socket ("10.0.0.2",9999);
   os = new DataOutputStream(smtpSocket.getOutputStream());
   is = new DataInputStream(smtpSocket.getInputStream());

} catch (UnknownHostException e)
```

Figure 4.17: Creating a client socket object

Once a connection was established in Figure 4.18 a wireless client (vehicle speed sensor) was used to send the vehicle speed data to the traffic control database server (RTCC-database server). The result of this implementation is shown in Figure 4.19.

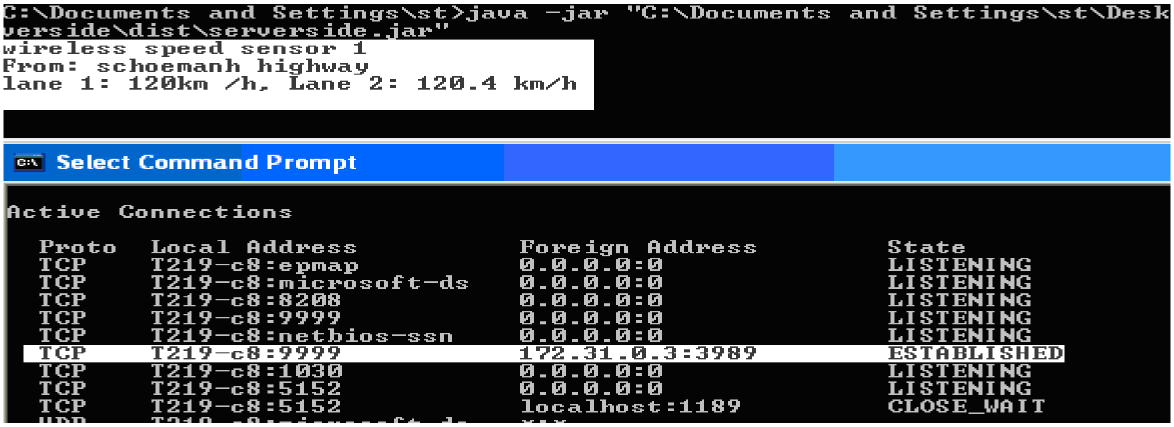


```
}
// if everything has been initialized we want to write some data to the socket we have opened a connecti

if (smtpSocket != null && os != null && is != null)
    try
    {
     os.writeBytes("wireless speed sensor 1 \n");
     os.writeBytes("From: schoemanh highway \n");
     os.writeBytes("lane 1: 120km /h, Lane 2: 120.4 km/h \n");
     os.writeBytes("subject: Testing");

     //keep on reading from a socket until we receive Ok from SMTP, once we receive it we want to break
```

Figure 4.18: Vehicle speed data being sent to the RTCC-database server

4.4.1 Results

Figure 4.19 shows a successful socket connection between a wireless client and the RTCC-database server. The three-way handshake was done using Transmission Control Protocol (TCP) to establish a connection between 172.31.0.3: 3989 client socket (wireless client) to T219-c8 (10.0.0.2): 9999 server socket (RTCC-database server) as shown by the highlighted “protocol”, “local address” and “foreign address” fields in Figure 4.19. The connection was successfully created between the wireless client and the traffic control database server as shown by the ESTABLISHED state field in Figure 4.19. Figure 4.19 also shows vehicle speeds (lane 1 and lane 2) as they are delivered from vehicle speed sensor 1 installed on Ben Schoeman highway to the RTCC-database server.



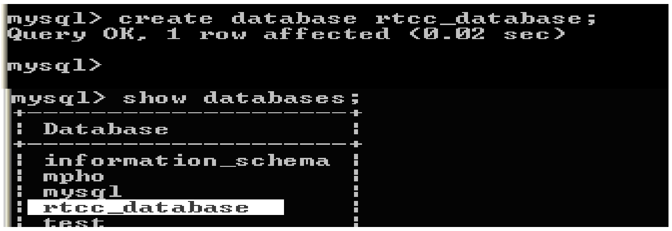
```
C:\Documents and Settings\st>java -jar "C:\Documents and Settings\st\Desk
verside\dist\serverside.jar"
wireless speed sensor 1
From: schoemanh highway
lane 1: 120km /h, Lane 2: 120.4 km/h
```

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    T219-c8:epmap          0.0.0.0:0              LISTENING
  TCP    T219-c8:microsoft-ds   0.0.0.0:0              LISTENING
  TCP    T219-c8:8208           0.0.0.0:0              LISTENING
  TCP    T219-c8:9999           0.0.0.0:0              LISTENING
  TCP    T219-c8:netbios-ssn    0.0.0.0:0              LISTENING
  TCP    T219-c8:9999           172.31.0.3:3989        ESTABLISHED
  TCP    T219-c8:1030           0.0.0.0:0              LISTENING
  TCP    T219-c8:5152           0.0.0.0:0              LISTENING
  TCP    T219-c8:5152           localhost:1189         CLOSE_WAIT
```

Figure 4.19: Connected client and server socket pairs

4.5 Experiment 4: Creating the Road Traffic Control Database using MySQL server

MySQL was used to create the “Road Traffic Control Centre (RTCC)-database” shown in Figure 4.20 to provide storage of vehicle speeds delivered from vehicle speed sensors using java sockets in experiment 4. MySQL database in Figure 4.20 was built based on the Entity Relationship Diagram (ERD) in Figure 4.21. This “RTCC-database” was created inside the Traffic Control Database Server in Figure 4.20.



```
mysql> create database rtcc_database;
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> show databases;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| mpho                 |
| mysql                |
| rtcc_database        |
| test                 |
```

Figure 4.20: The RTCC-database

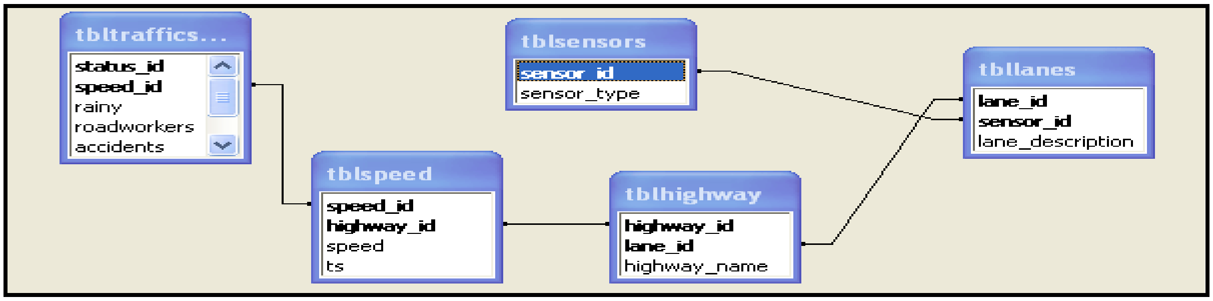


Figure 4.21: The RTCC-database entity relationship diagram (ERD)

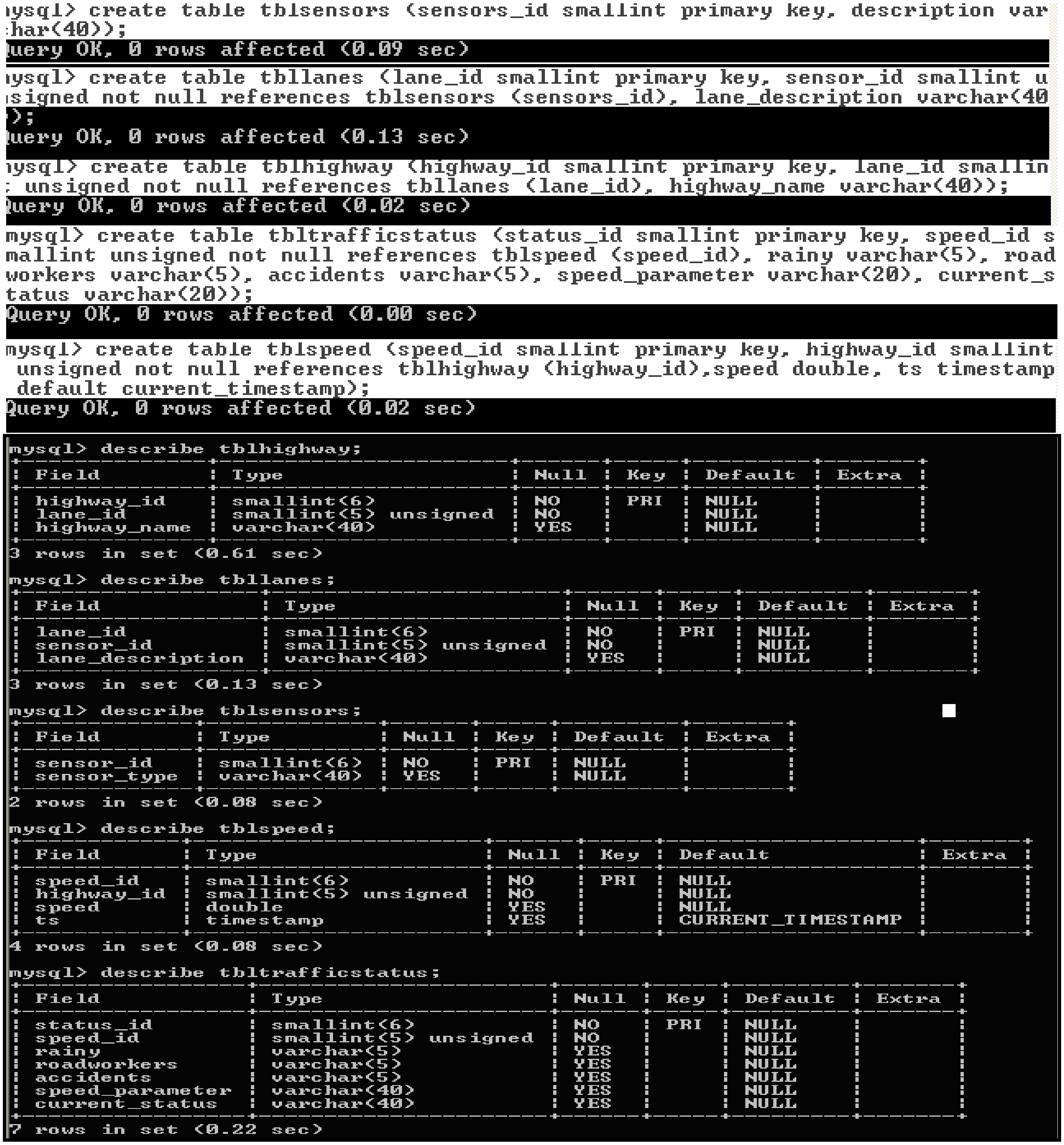
Once the “RTCC-database” was created, “use rtcc_database” command was executed from the command line interface (CLI). Executing the “use” command allows for creation of tables shown in Figure 4.21 inside the RTCC-database. These 5 tables that include tblhighway, tbllanes, tblsensors, tblspeed and tbltrafficstatus with different column properties were created using “create table” command as shown in Figure 4.22. The vehicle traffic data from vehicle speed sensors will be stored inside these tables.

In Figure 4.21 table tblhighway is used to identify a highway name where vehicle traffic status is being monitored. Table tbllanes identifies the number of lanes available inside the highway. Table tblsensors identifies the types of sensors installed inside the lanes on the highway. These sensors are used to capture parameters which influence the traffic status on the highway being monitored. Table tblspeed is used to store different vehicle speeds captured on the highway. In table tbltrafficstatus each vehicle speed upon arrival is discretised into “congestion” if the velocity is $v \le$ 45.4km/h**,** “into congestion” if the velocity is

45.5km/h $\leq v \leq$ 60.4km/h, “out of congestion” if the velocity is 60.5km/h $\leq v \leq$ 80.4km/h “normal” if the velocity is 80.5km/h $\leq v \leq$ 120.4km/h. Other factors that can influence the vehicle speed on the highway such as rainy weather condition, road works and accidents are also stored inside Tbltrafficstatus. Furthermore, Figure 4.21 shows that a highway could have multiple lanes as shown by the relationship identified by lane_id columns in both tblhighway and tbllanes respectively in Figure 4.21 and 4.22. Each lane could have one or more sensors installed to capture vehicle speeds (sensor_id column in both tbllanes and tblsensors). The vehicle speed to be captured would be for a specific highway at specific times in a day (speed_id, highway_id and timestamp in tblspeed). Based on all parameters (all columns of tbltrafficstatus in Figure 4.22) the current state of traffic could be determined.

Figure 4.22 shows all the tables inside the RTCC-database. A field value identifies a column to be used to store data. A type value identifies properties of each column. A null value with a NO indicates that a column cannot be left empty, a YES indicate that a column may be left empty. A key value with PRI shows the primary key columns which uniquely identifies the occurrence of a new record. By default all columns of tables in Figure 4.22 except a timestamp column are created with a NULL (an empty) value inside.

Vehicle speeds generated by a randperm Matlab function in experiment 2 as shown in Figure 4.14 and other factors which influence traffic behaviour on the highway were stored inside different tables inside the RTCC-database as shown in Figure 4.23 and Figure 4.24.

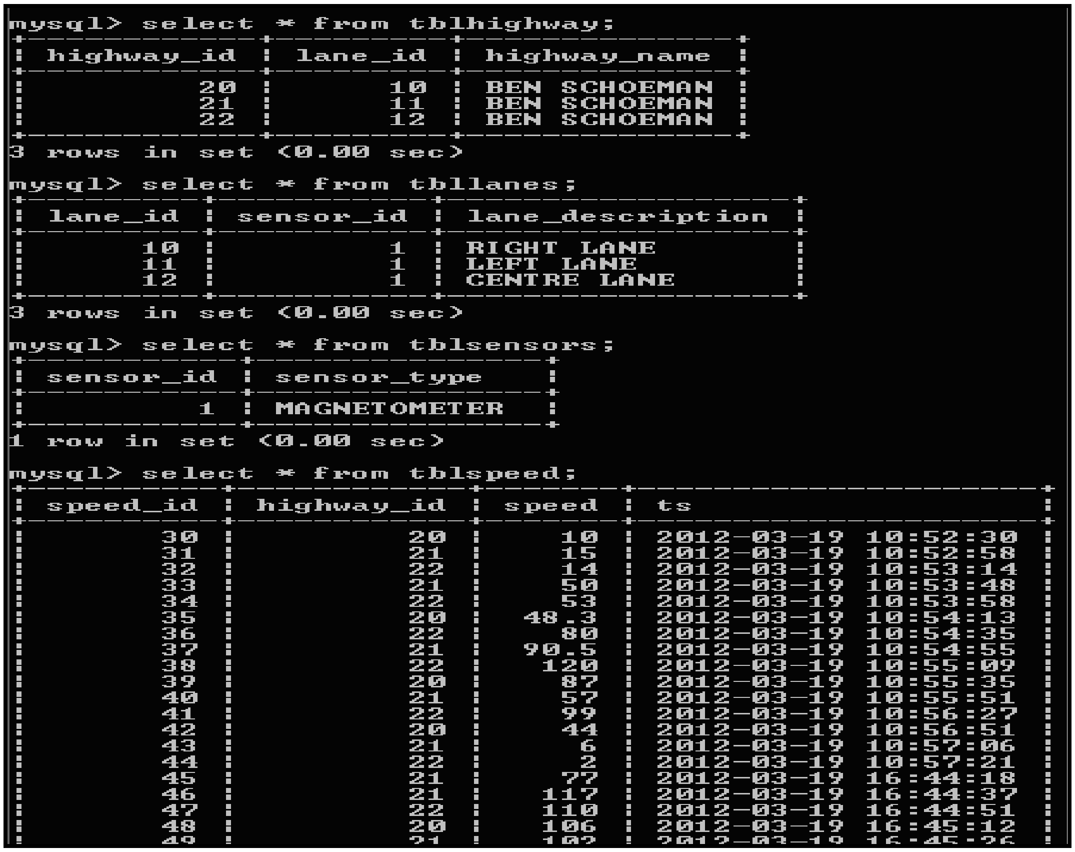


```
ysql> create table tblsensors (sensors_id smallint primary key, description var
har(40));
uery OK, 0 rows affected (0.09 sec)

ysql> create table tbllanes (lane_id smallint primary key, sensor_id smallint u
signed not null references tblsensors (sensors_id), lane_description varchar(40
);
uery OK, 0 rows affected (0.13 sec)

ysql> create table tblhighway (highway_id smallint primary key, lane_id smallin
 unsigned not null references tbllanes (lane_id), highway_name varchar(40));
uery OK, 0 rows affected (0.02 sec)

mysql> create table tbltrafficstatus (status_id smallint primary key, speed_id s
mallint unsigned not null references tblspeed (speed_id), rainy varchar(5), road
workers varchar(5), accidents varchar(5), speed_parameter varchar(20), current_s
tatus varchar(20));
Query OK, 0 rows affected (0.00 sec)

mysql> create table tblspeed (speed_id smallint primary key, highway_id smallint
 unsigned not null references tblhighway (highway_id),speed double, ts timestamp
 default current_timestamp);
Query OK, 0 rows affected (0.02 sec)

mysql> describe tblhighway;
+--------------+----------------------+------+-----+---------+-------+
| Field        | Type                 | Null | Key | Default | Extra |
+--------------+----------------------+------+-----+---------+-------+
| highway_id   | smallint(6)          | NO   | PRI | NULL    |       |
| lane_id      | smallint(5) unsigned | NO   |     | NULL    |       |
| highway_name | varchar(40)          | YES  |     | NULL    |       |
+--------------+----------------------+------+-----+---------+-------+
3 rows in set (0.61 sec)

mysql> describe tbllanes;
+------------------+----------------------+------+-----+---------+-------+
| Field            | Type                 | Null | Key | Default | Extra |
+------------------+----------------------+------+-----+---------+-------+
| lane_id          | smallint(6)          | NO   | PRI | NULL    |       |
| sensor_id        | smallint(5) unsigned | NO   |     | NULL    |       |
| lane_description | varchar(40)          | YES  |     | NULL    |       |
+------------------+----------------------+------+-----+---------+-------+
3 rows in set (0.13 sec)

mysql> describe tblsensors;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| sensor_id   | smallint(6) | NO   | PRI | NULL    |       |
| sensor_type | varchar(40) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.08 sec)

mysql> describe tblspeed;
+------------+----------------------+------+-----+-------------------+-------+
| Field      | Type                 | Null | Key | Default           | Extra |
+------------+----------------------+------+-----+-------------------+-------+
| speed_id   | smallint(6)          | NO   | PRI | NULL              |       |
| highway_id | smallint(5) unsigned | NO   |     | NULL              |       |
| speed      | double               | YES  |     | NULL              |       |
| ts         | timestamp            | YES  |     | CURRENT_TIMESTAMP |       |
+------------+----------------------+------+-----+-------------------+-------+
4 rows in set (0.08 sec)

mysql> describe tbltrafficstatus;
+-----------------+----------------------+------+-----+---------+-------+
| Field           | Type                 | Null | Key | Default | Extra |
+-----------------+----------------------+------+-----+---------+-------+
| status_id       | smallint(6)          | NO   | PRI | NULL    |       |
| speed_id        | smallint(5) unsigned | NO   |     | NULL    |       |
| rainy           | varchar(5)           | YES  |     | NULL    |       |
| roadworkers     | varchar(5)           | YES  |     | NULL    |       |
| accidents       | varchar(5)           | YES  |     | NULL    |       |
| speed_parameter | varchar(40)          | YES  |     | NULL    |       |
| current_status  | varchar(40)          | YES  |     | NULL    |       |
+-----------------+----------------------+------+-----+---------+-------+
7 rows in set (0.22 sec)
```

Figure 4.22: MySQL tables created inside the RTCC-database server

4.5.1 Results

Vehicle speed parameters were received from the Ben Schoeman highway as shown in Figure 4.23 tblhighway. Simulated magnetometer sensors (tblsensors) were used to send vehicle speed parameters captured from right, centre and left lanes (tbllanes) of Ben Schoeman highway. The actual parameters received include the actual vehicle speeds, the date and time through which these parameters were captured (tblspeed).

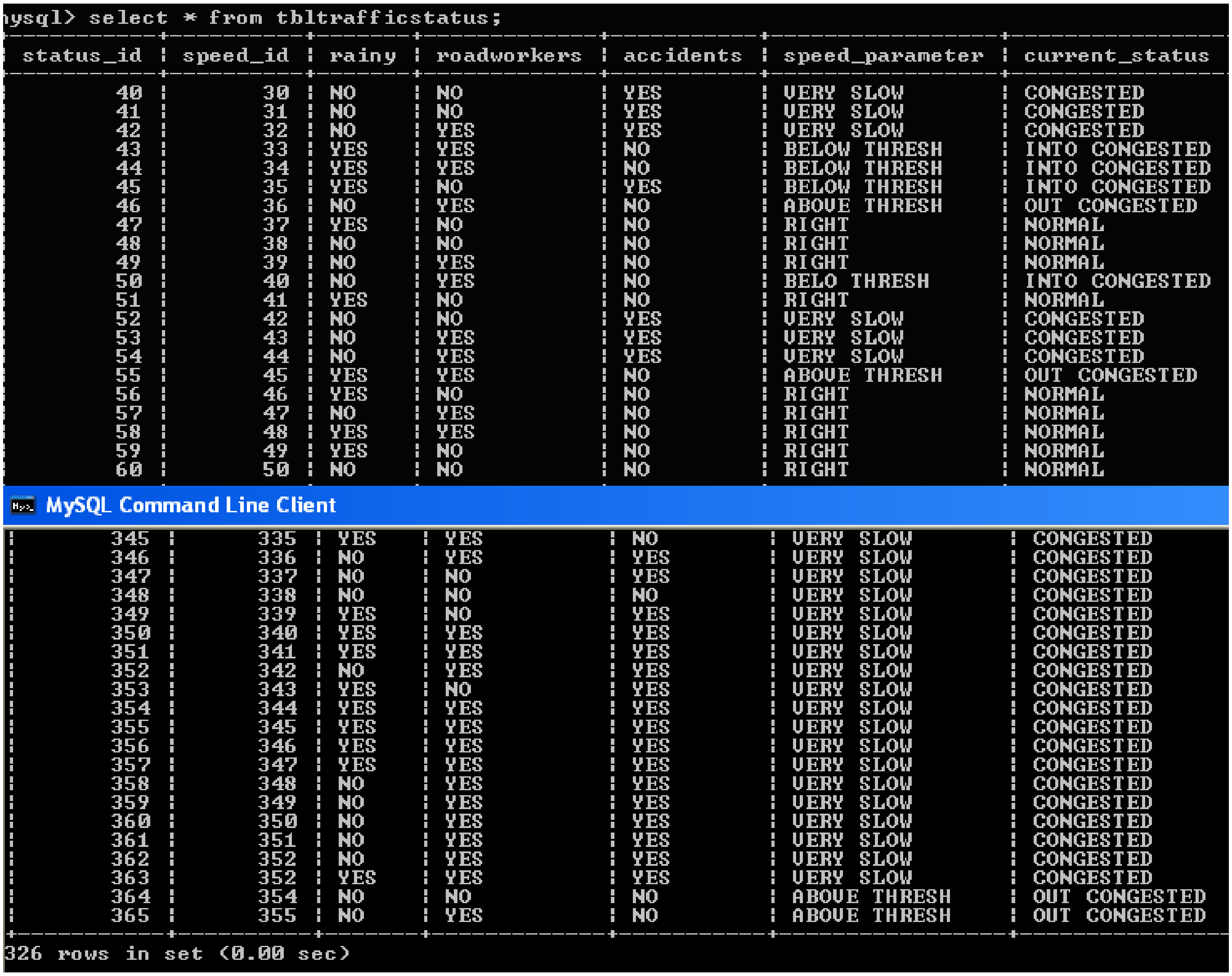


```
mysql> select * from tblhighway;
+------------+---------+---------------+
| highway_id | lane_id | highway_name  |
+------------+---------+---------------+
|         20 |      10 | BEN SCHOEMAN  |
|         21 |      11 | BEN SCHOEMAN  |
|         22 |      12 | BEN SCHOEMAN  |
+------------+---------+---------------+
3 rows in set (0.00 sec)

mysql> select * from tbllanes;
+---------+-----------+-------------------+
| lane_id | sensor_id | lane_description  |
+---------+-----------+-------------------+
|      10 |         1 | RIGHT LANE        |
|      11 |         1 | LEFT LANE         |
|      12 |         1 | CENTRE LANE       |
+---------+-----------+-------------------+
3 rows in set (0.00 sec)

mysql> select * from tblsensors;
+-----------+---------------+
| sensor_id | sensor_type   |
+-----------+---------------+
|         1 | MAGNETOMETER  |
+-----------+---------------+
1 row in set (0.00 sec)

mysql> select * from tblspeed;
+----------+------------+-------+---------------------+
| speed_id | highway_id | speed | ts                  |
+----------+------------+-------+---------------------+
|       30 |         20 |    10 | 2012-03-19 10:52:30 |
|       31 |         21 |    15 | 2012-03-19 10:52:58 |
|       32 |         22 |    14 | 2012-03-19 10:53:14 |
|       33 |         21 |    50 | 2012-03-19 10:53:48 |
|       34 |         22 |    53 | 2012-03-19 10:53:58 |
|       35 |         20 |  48.3 | 2012-03-19 10:54:13 |
|       36 |         22 |    80 | 2012-03-19 10:54:35 |
|       37 |         21 |  90.5 | 2012-03-19 10:54:55 |
|       38 |         22 |   120 | 2012-03-19 10:55:09 |
|       39 |         20 |    87 | 2012-03-19 10:55:35 |
|       40 |         21 |    57 | 2012-03-19 10:55:51 |
|       41 |         22 |    99 | 2012-03-19 10:56:27 |
|       42 |         20 |    44 | 2012-03-19 10:56:51 |
|       43 |         21 |     6 | 2012-03-19 10:57:06 |
|       44 |         22 |     2 | 2012-03-19 10:57:21 |
|       45 |         21 |    77 | 2012-03-19 16:44:18 |
|       46 |         21 |   117 | 2012-03-19 16:44:37 |
|       47 |         22 |   110 | 2012-03-19 16:44:51 |
|       48 |         20 |   106 | 2012-03-19 16:45:12 |
|       49 |         21 |   102 | 2012-03-19 16:45:26 |
```

Figure 4.23: Speed parameters received by different tables inside the RTCC-database

Tblhighway in Figure 4.23 shows columns *highway_id*, *lane_id* and *highway_name* used to store data. Inside the column *highway_id* there are numbers 20, 21 and 22. These numbers are primary keys which identify occurrence of 3 records inside tblhighway. *Lane_id* 10, 11 and 12 identifies right, left and centre lanes inside the Ben Schoeman highway where magnetometer sensors identified by *sensor_id* 1 were installed as shown by *sensor_id* in tbllanes and tblsensors of Figure 4.23. The *Speed_id* column with numbers from 30 to 49 within tblspeed uniquely identifies a record of each vehicle speed in the column *speed* as it is received from the speed sensors on the Ben Schoeman highway. For each vehicle speed record a *ts* column identifies the time at which each vehicle speed was captured.

Other parameters which influence the traffic congestion that include rainy weather conditions, the road works and accidents on the Ben Schoeman highway were received by tbltrafficstatus in Figure 4.24. Discretisation of each vehicle speed received occurs inside the *speed_parameter* column and a target concept based on all captured vehicle speed parameters is outlined inside the *current_status* column shown in Figure 4.24.



```
ysql> select * from tbltrafficstatus;
+-----------+----------+-------+-------------+-----------+-----------------+-----------------+
| status_id | speed_id | rainy | roadworkers | accidents | speed_parameter | current_status  |
+-----------+----------+-------+-------------+-----------+-----------------+-----------------+
|        40 |       30 | NO    | NO          | YES       | VERY SLOW       | CONGESTED       |
|        41 |       31 | NO    | NO          | YES       | VERY SLOW       | CONGESTED       |
|        42 |       32 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|        43 |       33 | YES   | YES         | NO        | BELOW THRESH    | INTO CONGESTED  |
|        44 |       34 | YES   | YES         | NO        | BELOW THRESH    | INTO CONGESTED  |
|        45 |       35 | YES   | NO          | YES       | BELOW THRESH    | INTO CONGESTED  |
|        46 |       36 | NO    | YES         | NO        | ABOVE THRESH    | OUT CONGESTED   |
|        47 |       37 | YES   | NO          | NO        | RIGHT           | NORMAL          |
|        48 |       38 | NO    | NO          | NO        | RIGHT           | NORMAL          |
|        49 |       39 | NO    | YES         | NO        | RIGHT           | NORMAL          |
|        50 |       40 | NO    | YES         | NO        | BELO THRESH     | INTO CONGESTED  |
|        51 |       41 | YES   | NO          | NO        | RIGHT           | NORMAL          |
|        52 |       42 | NO    | NO          | YES       | VERY SLOW       | CONGESTED       |
|        53 |       43 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|        54 |       44 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|        55 |       45 | YES   | YES         | NO        | ABOVE THRESH    | OUT CONGESTED   |
|        56 |       46 | YES   | NO          | NO        | RIGHT           | NORMAL          |
|        57 |       47 | NO    | YES         | NO        | RIGHT           | NORMAL          |
|        58 |       48 | YES   | YES         | NO        | RIGHT           | NORMAL          |
|        59 |       49 | YES   | NO          | NO        | RIGHT           | NORMAL          |
|        60 |       50 | NO    | NO          | NO        | RIGHT           | NORMAL          |
```

```
|       345 |      335 | YES   | YES         | NO        | VERY SLOW       | CONGESTED       |
|       346 |      336 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       347 |      337 | NO    | NO          | YES       | VERY SLOW       | CONGESTED       |
|       348 |      338 | NO    | NO          | NO        | VERY SLOW       | CONGESTED       |
|       349 |      339 | YES   | NO          | YES       | VERY SLOW       | CONGESTED       |
|       350 |      340 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       351 |      341 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       352 |      342 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       353 |      343 | YES   | NO          | YES       | VERY SLOW       | CONGESTED       |
|       354 |      344 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       355 |      345 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       356 |      346 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       357 |      347 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       358 |      348 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       359 |      349 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       360 |      350 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       361 |      351 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       362 |      352 | NO    | YES         | YES       | VERY SLOW       | CONGESTED       |
|       363 |      352 | YES   | YES         | YES       | VERY SLOW       | CONGESTED       |
|       364 |      354 | NO    | NO          | NO        | ABOVE THRESH    | OUT CONGESTED   |
|       365 |      355 | NO    | YES         | NO        | ABOVE THRESH    | OUT CONGESTED   |
+-----------+----------+-------+-------------+-----------+-----------------+-----------------+
326 rows in set (0.00 sec)
```

Figure 4.24: Tbltrafficstatus with attributes that influence traffic flow shown as headings of each column

The traffic pattern stored in tables in Figure 4.23 and Figure 4.24 is used in experiment 5 to predict future traffic condition.

4.6 Experiment 5: Traffic prediction using multilayer perceptron (MLP)

Equipment and software used

This experiment was carried out using the following equipment and software:

- Dell OptiPlex GX280 computer with a 3 GHz CPU and 2 GB of RAM

Operating System

- Windows XP professional version 2002 service pack 2

Application software

- MySQL server version 5.0.22
- MLP from WEKA version 3.6.6

4.6.1 Data Pre-processing

The vehicle traffic flow along the Ben Schoeman freeway was observed. Then based on this observation, the traffic flow data was randomly generated using the Matlab randperm function. Since the data was randomly generated as shown in experiment 4, manual clustering of data was done before training the MLP model. Manual clustering of these random traffic data helps to order vehicles speeds in such a way as to portray the vehicle traffic flow as was observed on the Ben Schoeman freeway. During clustering each instance is an averaged speed value arising from multiple speeds in the last 5 minutes. In order to ensure that all the four classes (congested, into congested, out congested and normal) are properly represented during training, the dataset of 326 instances was used to train and validate the MLP model.

The data was then pre-processed as follows:

December and January were excluded. In December there are more tourists in Gauteng due to Christmas holidays and in January most students look for new schools. In order to avoid the training model from being affected, both months were excluded. In the rest of the dissertations students from schools and tertiary institutions shall be called students.

November, February and March: Traffic pattern is homogenous. A total of 104 instances will be used. Traffic pattern for 4 weeks chosen randomly was used to represent both months. Mondays afternoons and Fridays mornings will use 4 instances each to represent off-peak traffic (32 instances) and another 4 instances will be used to represent peak traffic (32 instances). During the 3 months, off-peak traffic pattern on Tuesdays to Thursdays is similar, as a result, 4 instances will be used at once at any day between Tuesdays to Thursday's of the week. Two weeks will be used consecutively to represent this off-peak traffic (8 instances). Tuesdays to Thursday's (mornings and afternoons) peak traffic will be represented by 4 instances each for 4 consecutive weeks (32 instances).

April was also excluded due to Easter holidays and closing of schools.

May, June and July: Mayday and any other holiday (where banks are closed) are classified under weekends. It is winter season with similar traffic pattern. 108 instances were used to represent the whole 3 months. Traffic pattern for 4 weeks chosen randomly was used to represent these months. Within 4 chosen weeks Mondays afternoons will use 4 instances each week to represent off-peak traffic hours (16 instances) and another 4 instances will be used to represent peak traffic (16 instances) hours. For Friday mornings 8 instances will be used every week for the first 3 weeks (24 instances) to represent morning peak and morning off-peak traffic hours. Then, on the last Friday of any chosen 4th week, only the morning peak traffic will be represented (4 instances). This is due to the traffic that normally happens towards lunch time at the end of June when schools goes to holidays. 16 instances (4 instances each week for 4 weeks) will be used to represent off-peak hours, 16 instances for morning peak hours and another 16 instances for afternoon peak hours from Tuesday to Thursday during the 4 weeks that represent May, June and July.

August, September and October: Traffic pattern is sporadic and then becomes homogenous. From August and the beginning of September when the weather is windy people prefer to use transport. As a result days where more transport is being used, bursts of traffic will be experienced. From the 2nd week of September traffic starts to be homogenous until the end of October. Thus, 112 instances will be used to represent these 3 months. 1 week in August is used. 2 weeks in September, just before holidays and after holidays are also used. Finally 1 week in October will be used to complete 4 weeks used for the 3 months. Mondays afternoons and Fridays mornings will use 4 instances each to represent off-peak traffic (32 instances) and another 4 instances will be used to represent peak traffic (32 instances). Tuesday to Thursday will use 12 instances once in a particular day of the chosen week (4 morning peak, 4 afternoon peak and 4 off-peak hours). This will be repeated for 4 weeks, resulting in 16 morning peak instances, 16 off peak instances and 16 afternoon peak instances.

In order to eliminate the influence of the weekend, the travel pattern for Monday morning was left out and also the travel pattern for Friday afternoon was also left out. The model will also not be used to predict traffic status over the weekend. A rationale being that 90 percent of organizations do not open during weekends and thus their impact on the economy might be minimal.

4.6.2 Creating an artificial neural network model

In this experiment a MLP was used to predict the traffic condition for the next 5 minutes using historical data from the traffic data collected over a period of 9 months (total of 402 instances). Out of these 402 instances 326 were stored inside MySQL RTCC-database as shown in experiment 4. This 326 instances were used during training and validation of the chosen multilayer perceptron model, thereafter the remaining 76 instances (stored separately on an excel spreadsheet) were used to test the model thus simulating the previous 5 minutes of data to be used to predict the traffic condition in the next 5 minutes. Each minute has an instance, thus in 5 minutes there are 5 instances. The 5 instances are an aggregate of the vehicle traffic data collected in each minute. Within this 5 instances data, vehicle speed data was discreticised into 4 thresholds', very slow (velocity is $v \leq$ 45.4km/h), below-thresh (velocity is 45.5km/h $\leq v \leq$ 60.4km/h), above-thresh (velocity is 60.5km/h $\leq v \leq$ 80.4km/h) and right (velocity is 80.5km/h $\leq v \leq$ 120.4km/h) as shown in Figure 4.24 inside column *speed_parameter*. The MLP model is presented with these 4 discreticised vehicle speeds together with the values of the weather, accidents and road works information. Feeding the MLP model with discreticised vehicle speed allows the MLP model to predict a discreticised vehicle speed range likely to happen on the highway in the next 5 minutes. The South Africa Weather Services continuous weather broadcasts together with SANRAL video monitoring feeds are used to provide weather, accidents and road works information to the model prior to vehicle traffic prediction as shown in Figure 4.24. The inclusion of road works as one of the attributes is due to the effects on traffic flow of continuous road development and maintenance taking place around Gauteng main highways. Each of these 7 instances is presented to the MLP and an outcome for each of them is given.

A complete artificial neural network model, the MLP is shown in Figure 4.25, where t_1 - t_5 is time in minutes from the 1st minute to the 5th minute and where $P_{t_1 - t_5}$ is vehicle input pattern feed to the MLP model from the 1st minute to the 5th minute. The 7 inputs feed to the model during the first 5 minutes includes discreticised vehicle speeds from each of the four classes (Vbelow thresh$t_1 - t_5$, Vabove thresh $t_1 - t_5$,Vvery slow$t_1 - t_5$ and Vright$t_1 - t_5$**)**, was it raining in the last 5 minutes? i.e. weather information (rainy$t_1 - t_5$), any accidents in the last 5 minutes? i.e. accident information (accidents$t_1 - t_5$**)** and presence of people who do road maintenance on the highway in the last 5 minutes (roadworkers$t_1 - t_5$).

The aim of this MLP model in Figure 4.25 is to predict either "normal", or "into congestion", or "out congestion" or "congestion" as a traffic status condition likely to happen in the next 5 minutes ($P_{t_{10}}$ **).** In Figure 4.26 the X- axis shows time in minutes (from 1 to 5 minutes) and the Y- axis shows vehicle speed in kilometres per hour (from 0 to 120.4km/h).

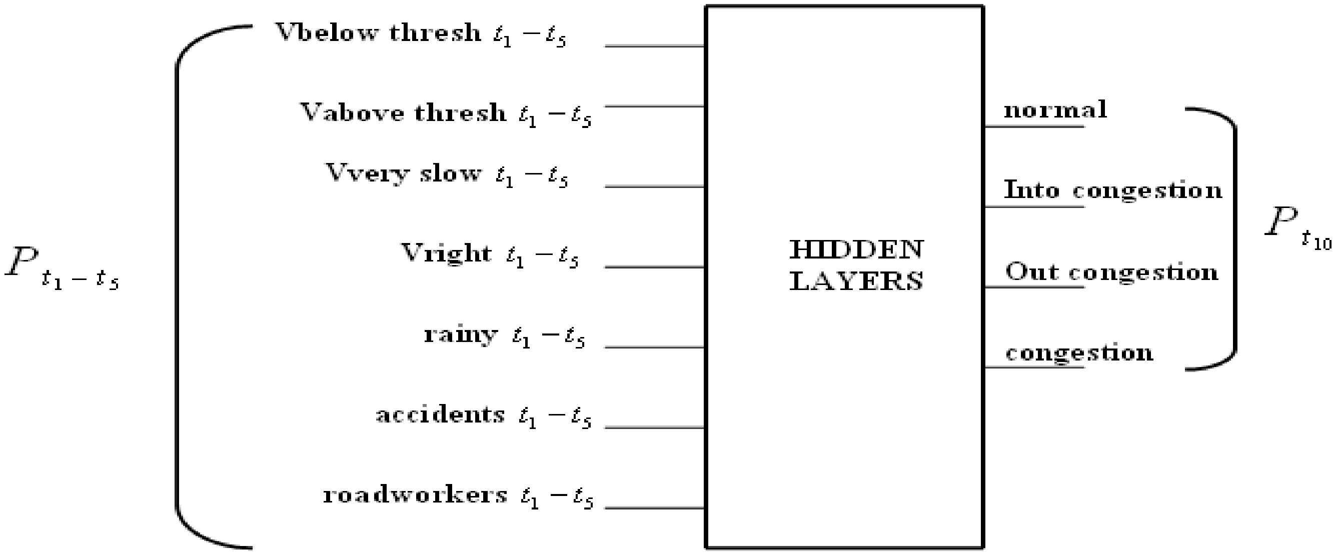


Figure 4.25: MLP prediction model used for 5 minutes ahead prediction

The inputs feed to the trained MLP model were applied as they are shown in Figure 4.28. The four conditions A to D shown in Figure 4.26 are conditions to be predicted. Together they are used to predict the next 5 minutes' traffic status. Figure 4.26 also shows different times from 1st minute to 5th minute during the course of the day for which the vehicle traffic data was captured on the highway. In Figure 4.26 if the predicted traffic condition is A (Congestions) it means that vehicle speed on the highway will be between 0 to 45.4km/h. If the predicted traffic condition is B (Into congestion) it means that vehicle speed on the highway will be between 45.5 to 60.5km/h. If the predicted traffic condition is C (Out of Congestion) it means that vehicle speed on the highway will be between 60.5 to 80.4km/h. If the predicted traffic condition is D (normal highway traffic) it means that vehicle speed on the highway will be between 80.5 to 120.4km/h.

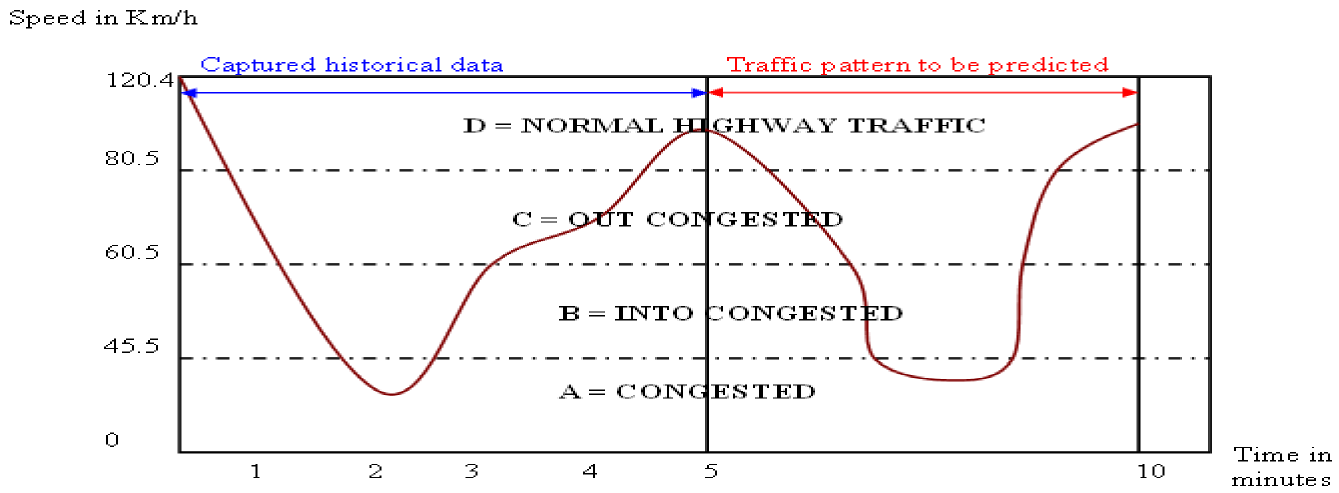


Figure 4.26: Traffic conditions to be predicted on the Ben Schoeman highway

4.6.3 Training the neural network

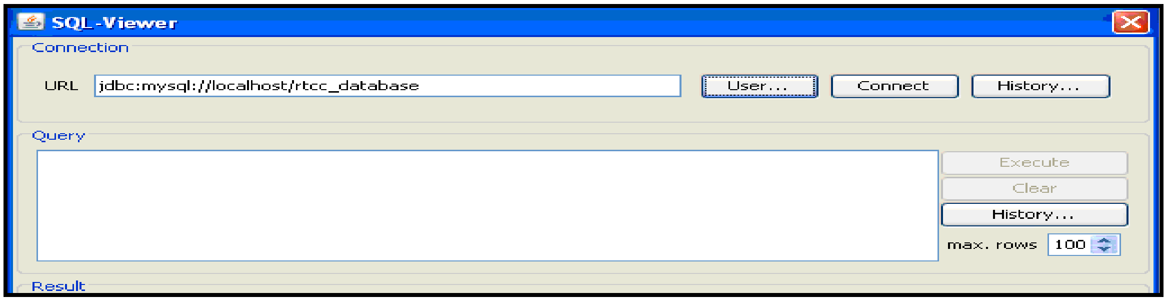
The following steps were followed in this experiment:

- Launch WEKA explorer and create a connection to MySQL database
- Retrieve vehicle traffic data from MySQL database
- From WEKA select MLP
- Split the data into 66 percent, 15 percent validation and the remaining 19 percent was used to test the model
- Start training the MLP and test the resulting model
- View classification output
- Save the trained MLP model
- Present the novel instance (data) to the saved MLP model (to predict novel traffic data)

Step 1: Loading data from MySQL database

In order to load data from the MySQL database (RTCC-database created in Figure 4.27), java database connectivity driver (JDBC) was used to connect to a local host path (the RTCC-database which is specified with a URL path of WEKA SQL viewer).

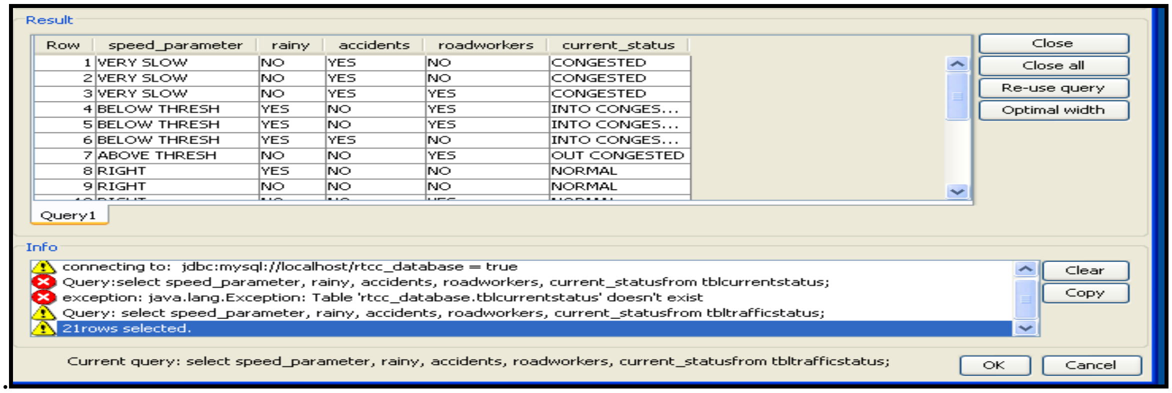
Supply MySQL root username and password to create a connection to the database. Once a connection was established, the SQL query to SELECT the training data FROM tbltrafficstatus was executed to retrieve data pertaining to the columns shown in Figure 4.28



URL jdbc:mysql://localhost/rtcc_database

max. rows 100

Figure 4.27: Supplying a path to RTCC MySQL database



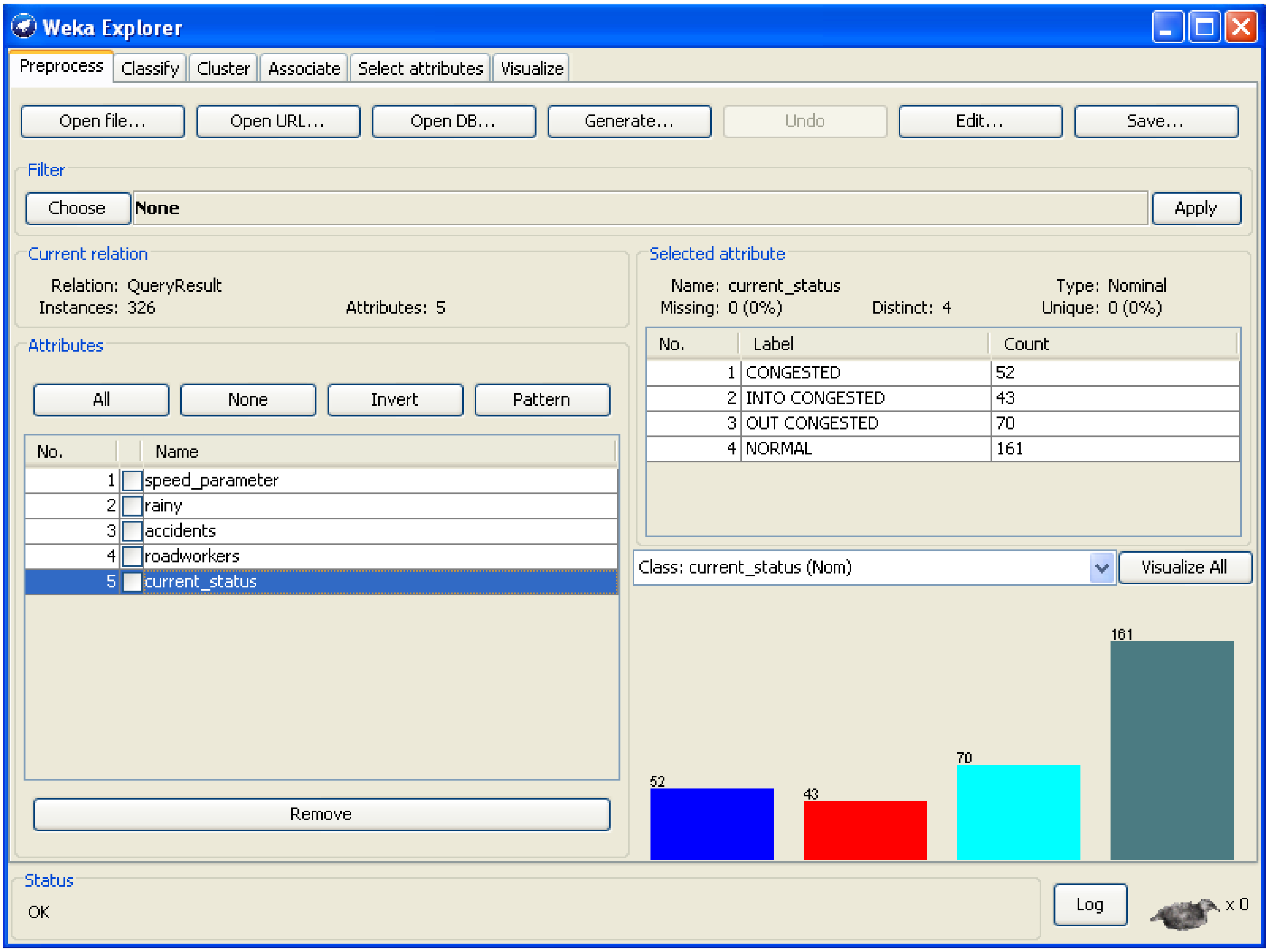
Row	speed_parameter	rainy	accidents	roadworkers	current_status
1	VERY SLOW	NO	YES	NO	CONGESTED
2	VERY SLOW	NO	YES	NO	CONGESTED
3	VERY SLOW	NO	YES	YES	CONGESTED
4	BELOW THRESH	YES	NO	YES	INTO CONGES...
5	BELOW THRESH	YES	NO	YES	INTO CONGES...
6	BELOW THRESH	YES	YES	NO	INTO CONGES...
7	ABOVE THRESH	NO	NO	YES	OUT CONGESTED
8	RIGHT	YES	NO	NO	NORMAL
9	RIGHT	NO	NO	NO	NORMAL

```
connecting to:  jdbc:mysql://localhost/rtcc_database = true
Query:select speed_parameter, rainy, accidents, roadworkers, current_statusfrom tblcurrentstatus;
exception: java.lang.Exception: Table 'rtcc_database.tblcurrentstatus' doesn't exist
Query: select speed_parameter, rainy, accidents, roadworkers, current_statusfrom tbltrafficstatus;
21rows selected.
```

Current query: select speed_parameter, rainy, accidents, roadworkers, current_statusfrom tbltrafficstatus;

Figure 4.28: Loading data in WEKA from RTCC MySQL database

Clicking "OK" in Figure 4.28 means accepting the data retrieved from MySQL and displays a screen shown in Figure 4.29. In Figure 4.29 the total number of instances and all the attributes including statistical information about each attribute of the data loaded from MySQL are shown. Five attributes (speed_parameter, rainy, accidents, roadworkers and current_status) loaded from MySQL are shown by NO 1, 2, 3, 4 and 5 in Figure 4.29. The histogram shows attribute distribution for each selected attribute. On the top left corner of each bar in the histogram the numbers 52, 43, 70 and 161 respectively show the number of instances with similar characteristics inside the current_status field. The label field in Figure 4.29 identifies these characteristics as congested, into congestion, out of congestion and normal respectively for each of the 4 displayed bars of the histogram. The histogram shown in Figure 4.29 is as a result of clicking attribute 5 (current_status). Clicking on a different attribute will display a different histogram with different numbers on top.



Filter

None

Current relation

Relation: QueryResult
Instances: 326 Attributes: 5

Selected attribute

Name: current_status Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count
1	CONGESTED	52
2	INTO CONGESTED	43
3	OUT CONGESTED	70
4	NORMAL	161

Attributes

No.		Name
1	□	speed_parameter
2	□	rainy
3	□	accidents
4	□	roadworkers
5	□	current_status

Class: current_status (Nom)

Status

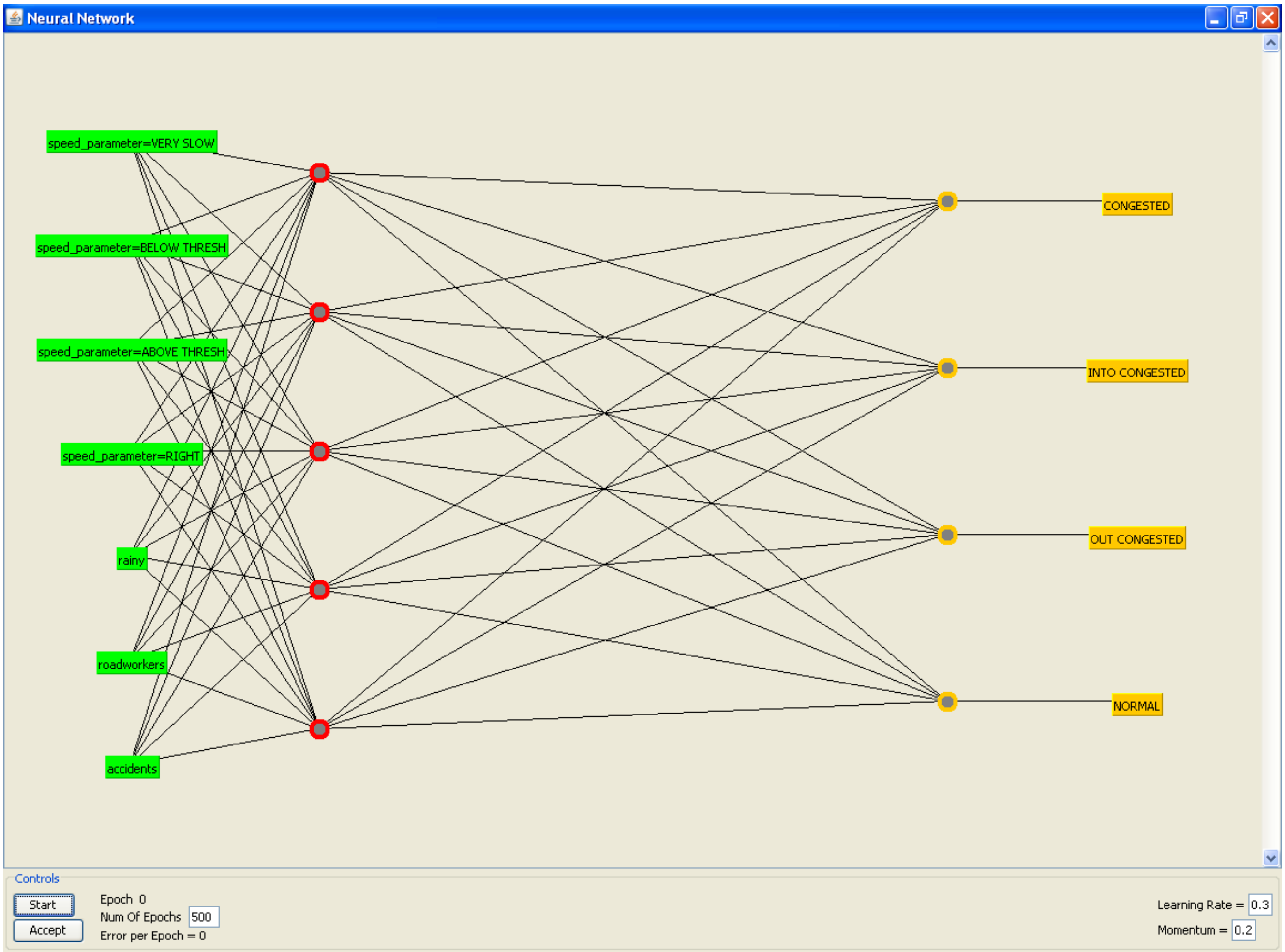
OK

Figure 4.29: View of data in WEKA explorer

Step 2: Training the Classifier

- Referring to Figure 4.29, click the Classify tab to select the Classifier
- Select MLP classifier from functions in WEKA
- Load data from MySQL RTCC-database.
- Split the data (training, validation and test data): The dataset of 402 instances was divided as follows: 66 percent training (266 instances), 15 percent validation (60 instances) and 19 percent testing (76 instances). 326 instances shown in Figure 4.29 are instances used to train and validate the MLP model.

The result is the architecture of the neural network MLP shown in Figure 4.30. This is the model that was saved. It will be used later to classify novel instances comprising 76 instances whose targets are known but these instances have never been presented to the model before. On testing the targets were removed. The objective was to find out how many of these instances the model was going to correctly classify as shown in section 4.6.4.



Epoch 0
Num Of Epochs 500
Error per Epoch = 0
Learning Rate = 0.3
Momentum = 0.2

Figure 4.30: The MLP architecture

The experiments were carried out using different architectures as shown in Table 4.1 through Table 4.4 before the final winning model shown in Figure 4.30 was chosen. The remaining 19 percent of vehicle traffic data (separate data saved on an excel spreadsheet) was used to test each of the architectures. At the beginning, experiments were carried out by varying the number of inputs as shown in Table 4.2, Table 4.3 and Table 4.4. Furthermore, the numbers of hidden units were also varied. When varying the number of inputs not all the inputs could be varied since from intuition it is known that the vehicle speed pattern forms the integral part of the prediction model. When varying the number of inputs, *speed_parameter = below thresh*, *very slow*, *above thresh* and *right* were not touched. *Roadworkers*, *rainy* and *accidents* are the inputs which were randomly varied in order to see how the model fares when any of these inputs are omitted. The experiments started by using all 7 inputs while varying the number of hidden units as shown in Table 4.1.

Table 4.1: The results of varying the number of hidden units, with the number of inputs and outputs remaining constant

MLP architecture	***Training performance***			***Testing Accuracy in %***
	Training accuracy in %	***RMSE***	***Time taken to train the model in seconds***	
7:1:4	*86*	*0.23*	*0.4*	*48*
7:2:4	*99*	*0.02*	*0.3*	*84*
7:3:4	*90*	*0.07*	*0.6*	*80*
7:4:4	*95*	*0.09*	*0.7*	*83*
7:5:4	*100*	*0.01*	*1.0*	*100*
7:6:4	*89*	*0.27*	*0.8*	*79*
7:7:4	*87*	*0.26*	*0.8*	*83*
7:14:4	*79*	*0.31*	*1.02*	*68*

The criterion used for selecting the best model was a root mean square error (RMSE) value and the accuracy shown by the MLP model during training. If a model performed poorly (having a large RMSE), the adjustment of the number of hidden units was done. At the end of the experiments in Table 4.1, the architecture 7:5:4 showed better performance with 100 percent accuracy during both training and testing while maintaining a low RMSE value of 0.01. Thus this indicates that using 5 hidden units the model provides a better performance.

Once the architecture 7:5:4 was identified, in the next experiments shown in Table 4.2 the inputs: *roadworkers*, *accidents* and *rainy* were randomly varied in order to see if the same performance can be achieved when any of these inputs were omitted by the model. In the 1st experiment in Table 4.2, *roadworkers* is the feature that was left out. In the 2nd experiment, *roadworkers* and *rainy* are the inputs which were left out. In the 3rd experiment, *roadworkers*, *rainy* and *accidents* were all left out.

Table 4.2: The results of using 5 hidden units while varying the number of inputs with outputs remaining constant

MLP architecture	***Training performance***			***Testing Accuracy in %***
	Training accuracy in %	***RMSE***	***Time taken to train the model in seconds***	
6:5:4	*91*	*0.23*	*0.9*	*81*
5:5:4	*83*	*0.29*	*0.7*	*76*
4:5:4	*76*	*0.37*	*0.5*	*68*

Varying the number of inputs did not improve the model results as shown in Figure 4.2. Decreasing the number of inputs resulted in the increase of the RMSE value thus affecting the accuracy of the model during training and testing.

Seen that the results for the architectures 7:2:4 (99 percent) and 7:4:4 (95 percent) during training were not far away from 7:5:4 (100 percent) as shown in Table 4.1, in Table 4.3 and Table 4.4 the number of inputs were also varied for these 2 architectures. The same sequence as in Table 4.2 was followed when varying the number of inputs (In the 1st experiment, *roadworkers* is the feature that was left out. In the 2nd experiment, *roadworkers* and *rainy* are the inputs which were left out. In the 3rd experiment, *roadworkers, rainy* and *accidents* were all left out). The results are shown in Table 4.3 and Table 4.4.

Varying the number of inputs for the two competing models as shown in Table 4.3 and Table 4.4 did not improve these models accuracy, as a result 7:5:4 architecture was chosen as the best model to be used to forecast the next 5 minutes vehicle traffic status.

Table 4.3: The results of using 2 hidden units while varying the number of inputs with outputs remaining constant

MLP architecture	***Training performance***			***Testing Accuracy in %***
	Training accuracy in %	***RMSE***	***Time taken to train the model in seconds***	
6:2:4	*86*	*0.37*	*0.5*	*67*
5:2:4	*80*	*0.30*	*0.4*	*67*
4:2:4	*80*	*0.29*	*0.2*	*65*

Table 4.4: The results of using 4 hidden units while varying the number of inputs with outputs remaining constant

MLP architecture	***Training performance***			***Testing Accuracy in %***
	Training accuracy in %	***RMSE***	***Time taken to train the model in seconds***	
6:4:4	*91*	*0.38*	*0.8*	*76*
5:4:4	*90*	*0.38*	*0.5*	*71*
4:4:4	*86*	*0.37*	*0.4*	*67*

The chosen MLP model in Table 4.1 was made up of 7 inputs, 5 hidden nodes and 4 outputs (7:3:4 model). The MLP model was constructed using 66 percent training data (more data during learning) with 15 percent of data used as a validation set to fine tune the parameters of the classifier. Thereafter 19 percent of the remaining data (not used during training) was used to test the model.

The training of the MLP using the back-propagation algorithm was used and 326 instances loaded in Figure 4.29 were used to train the network. The parameters chosen were a momentum of 0.2, a learning rate of 0.3, a validation set of 15 percent, validation threshold of 20 and 500 epochs. The proposed model used the early stopping approach to prevent noise which can cause over-fitting. The network was configured to automatically reset when a lower learning rate was obtained.

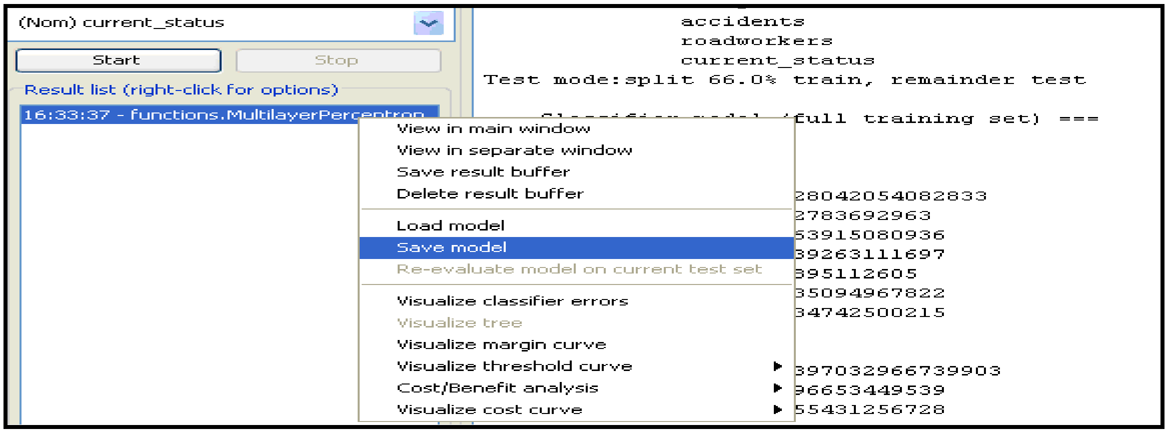
Cross validations using 10 folds were used in order to mitigate any bias which might be caused by a particular sample. Using 10 folds means repeating training and testing 10 times using 10 different random samples. In each iteration 66 percent of the data was randomly selected for training. Each of the 10 models was tested and the performance metrics of all the 10 tests were averaged to give the final model.

Step 3: View classification output

The results of the winning trained model were 100 percent accuracy and the detailed results are in annexure B.

Step 4: Save the model

Once the performance of the model is satisfactory "right click" on top of the model listed inside the result list pane and save the model (Figure 4.31). The saved model will be used to classify new vehicle traffic data.

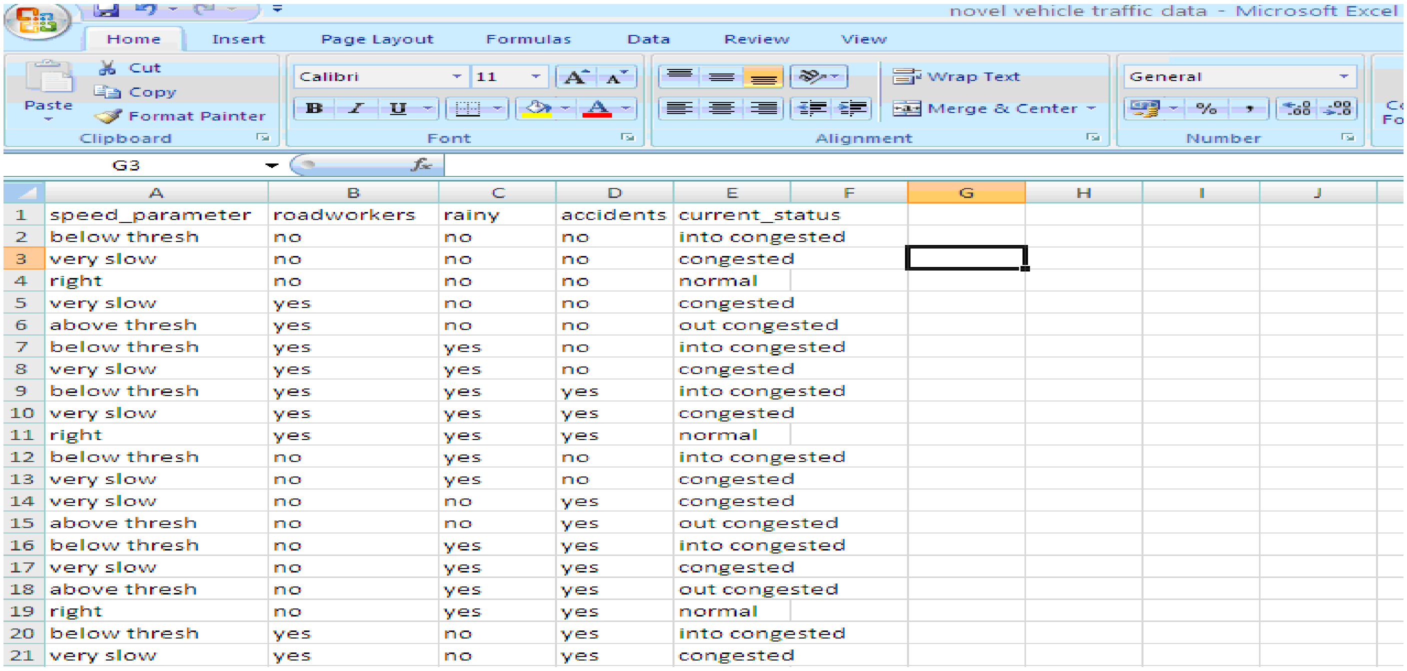


```
                      accidents
                      roadworkers
                      current_status
Test mode:split 66.0% train, remainder test

                                  full training set) ===



                                  28042054082833
                                  2783692963
                                  53915080936
                                  39263111697
                                  395112605
                                  35094967822
                                  34742500215

                                  397032966739903
                                  96653449539
                                  55431256728
```

Figure 4.31: Saving the trained MLP model

4.6.4 Using the trained MLP model to predict novel traffic data

An excel spreadsheet shown in Figure 4.32 was used to provide new vehicle traffic data not used during training (76 instances). To classify these novel instances whose targets are known, the objective was to find out how many of these instances the model was going to correctly classify.



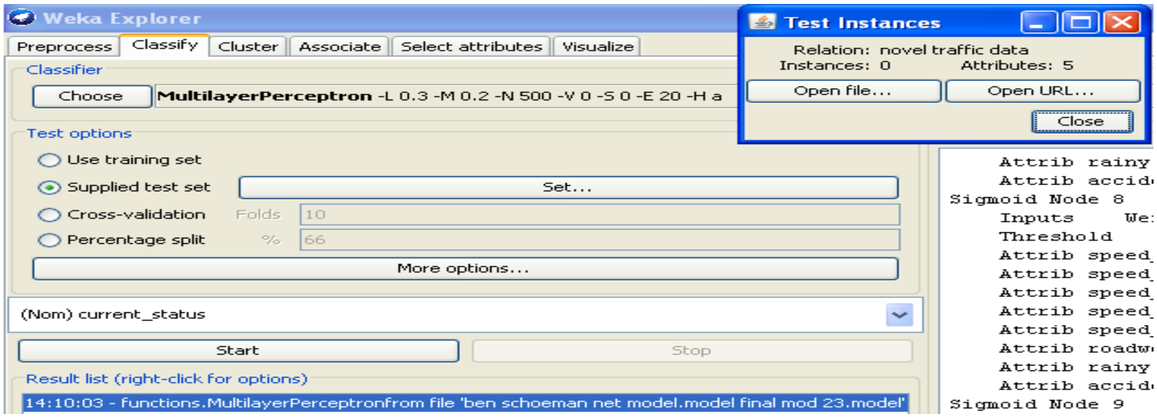
	A	B	C	D	E	F	G	H	I	J
1	speed_parameter	roadworkers	rainy	accidents	current_status					
2	below thresh	no	no	no	into congested					
3	very slow	no	no	no	congested					
4	right	no	no	no	normal					
5	very slow	yes	no	no	congested					
6	above thresh	yes	no	no	out congested					
7	below thresh	yes	yes	no	into congested					
8	very slow	yes	yes	no	congested					
9	below thresh	yes	yes	yes	into congested					
10	very slow	yes	yes	yes	congested					
11	right	yes	yes	yes	normal					
12	below thresh	no	yes	no	into congested					
13	very slow	no	yes	no	congested					
14	very slow	no	no	yes	congested					
15	above thresh	no	no	yes	out congested					
16	below thresh	no	yes	yes	into congested					
17	very slow	no	yes	yes	congested					
18	above thresh	no	yes	yes	out congested					
19	right	no	yes	yes	normal					
20	below thresh	yes	no	yes	into congested					
21	very slow	yes	no	yes	congested					

Figure 4.32: New vehicle traffic data in excel spreadsheet

4.6.4.1 Loading the trained MLP model

Right click anywhere inside the result list pane shown in Figure 4.31, select “Load model” and browse to the location where the trained MLP model was saved.

Under “test options” in Figure 4.33 select “supplied test set” and click on “set” button. Click “Open file” from Test Instances pane which is visible after clicking a “set” button as shown on the top right corner in Figure 4.33.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Relation: novel traffic data
Instances: 0 Attributes: 5

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

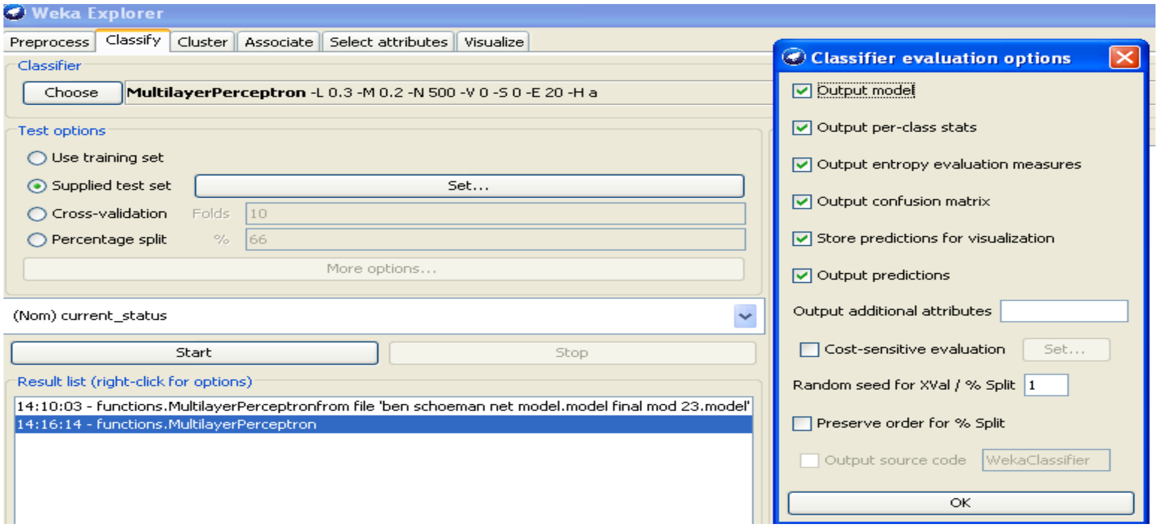
14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'

```
    Attrib rainy
    Attrib accid
Sigmoid Node 8
    Inputs    We:
    Threshold
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib roadw
    Attrib rainy
    Attrib accid
Sigmoid Node 9
```

Figure 4.33: Loading new vehicle traffic data

Browse to the location containing the arff file (excel spreadsheet converted to arff format, detailed instructions on how to convert are in annexure C).

Click “more options” button in Figure 4.33 and ensure that output predictions, store predictions for visualization, output confusion matrix, output entropy evaluation measures, output per class stats and output model are all selected as shown in Figure 4.34. Click “OK” button when done and click “start” button to begin testing.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

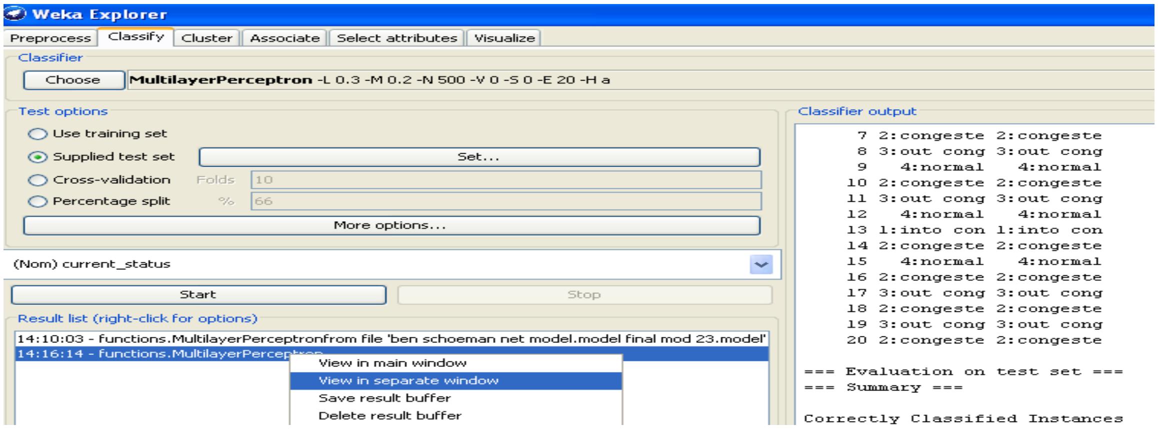
(Nom) current_status

14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'
14:16:14 - functions.MultilayerPerceptron

☑ Output model
☑ Output per-class stats
☑ Output entropy evaluation measures
☑ Output confusion matrix
☑ Store predictions for visualization
☑ Output predictions
Output additional attributes
☐ Cost-sensitive evaluation
Random seed for XVal / % Split 1
☐ Preserve order for % Split
☐ Output source code WekaClassifier

Figure 4.34: Modifying the classifier evaluation options

When testing was complete the results were shown inside the classifier output pane on the right hand side in Figure 4.35. To view detailed results in a separate window right click on top of the multilayer perceptron model within the result list pane shown in Figure 4.35 and select “view in separate window”.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

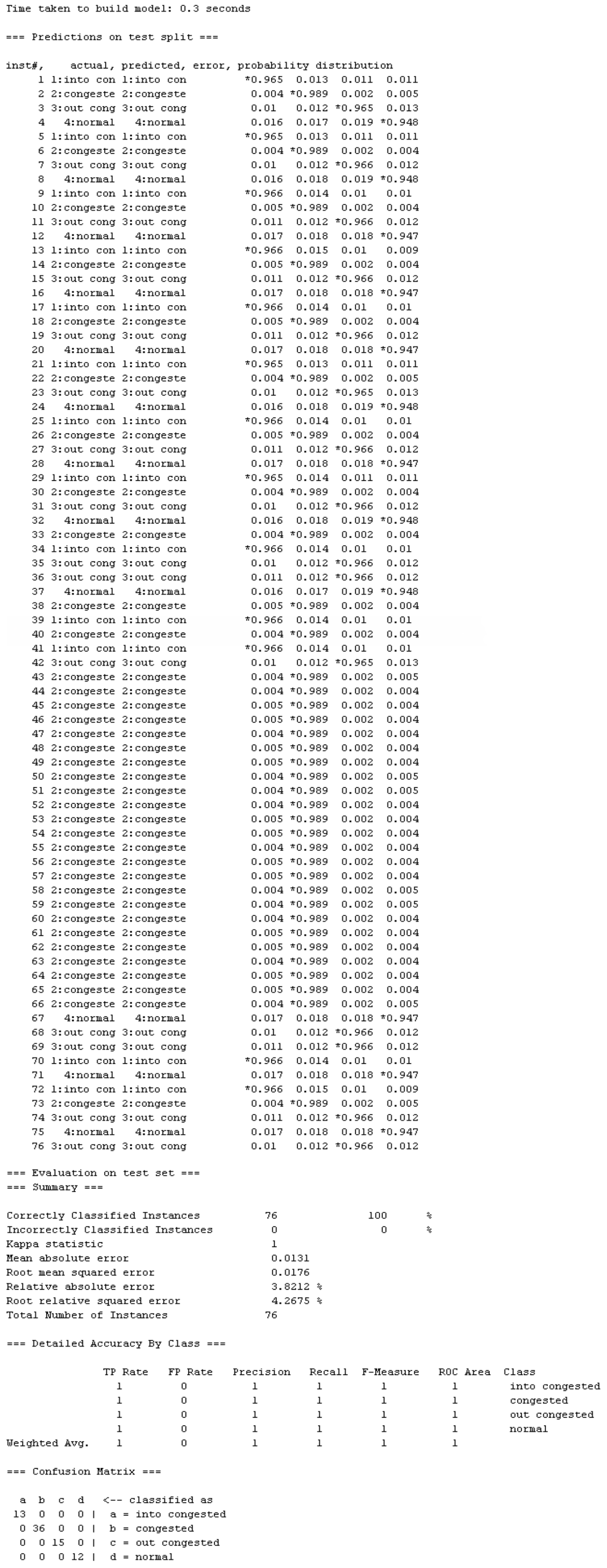
(Nom) current_status

14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'
14:16:14 - functions.MultilayerPerceptron

```
      7 2:congeste 2:congeste
      8 3:out cong 3:out cong
      9   4:normal   4:normal
     10 2:congeste 2:congeste
     11 3:out cong 3:out cong
     12   4:normal   4:normal
     13 1:into con 1:into con
     14 2:congeste 2:congeste
     15   4:normal   4:normal
     16 2:congeste 2:congeste
     17 3:out cong 3:out cong
     18 2:congeste 2:congeste
     19 3:out cong 3:out cong
     20 2:congeste 2:congeste

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances
```

Figure 4.35: Viewing testing output results in a separate window

Detailed results are shown in Figure 4.36.



```
Time taken to build model: 0.3 seconds

=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1 1:into con 1:into con        *0.965  0.013  0.011  0.011
     2 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
     3 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
     4   4:normal   4:normal         0.016  0.017  0.019 *0.948
     5 1:into con 1:into con        *0.965  0.013  0.011  0.011
     6 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
     7 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
     8   4:normal   4:normal         0.016  0.018  0.019 *0.948
     9 1:into con 1:into con        *0.966  0.014  0.01   0.01
    10 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    11 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    12   4:normal   4:normal         0.017  0.018  0.018 *0.947
    13 1:into con 1:into con        *0.966  0.015  0.01   0.009
    14 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    15 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    16   4:normal   4:normal         0.017  0.018  0.018 *0.947
    17 1:into con 1:into con        *0.966  0.014  0.01   0.01
    18 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    19 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    20   4:normal   4:normal         0.017  0.018  0.018 *0.947
    21 1:into con 1:into con        *0.965  0.013  0.011  0.011
    22 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    23 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
    24   4:normal   4:normal         0.016  0.018  0.019 *0.948
    25 1:into con 1:into con        *0.966  0.014  0.01   0.01
    26 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    27 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    28   4:normal   4:normal         0.017  0.018  0.018 *0.947
    29 1:into con 1:into con        *0.965  0.014  0.011  0.011
    30 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    31 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    32   4:normal   4:normal         0.016  0.018  0.019 *0.948
    33 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    34 1:into con 1:into con        *0.966  0.014  0.01   0.01
    35 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    36 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    37   4:normal   4:normal         0.016  0.017  0.019 *0.948
    38 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    39 1:into con 1:into con        *0.966  0.014  0.01   0.01
    40 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    41 1:into con 1:into con        *0.966  0.014  0.01   0.01
    42 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
    43 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    44 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    45 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    46 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    47 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    48 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    49 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    50 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    51 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    52 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    53 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    54 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    55 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    56 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    57 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    58 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    59 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    60 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    61 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    62 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    63 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    64 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    65 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    66 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    67   4:normal   4:normal         0.017  0.018  0.018 *0.947
    68 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    69 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    70 1:into con 1:into con        *0.966  0.014  0.01   0.01
    71   4:normal   4:normal         0.017  0.018  0.018 *0.947
    72 1:into con 1:into con        *0.966  0.015  0.01   0.009
    73 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    74 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    75   4:normal   4:normal         0.017  0.018  0.018 *0.947
    76 3:out cong 3:out cong         0.01   0.012 *0.966  0.012

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          76              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0131
Root mean squared error                  0.0176
Relative absolute error                  3.8212 %
Root relative squared error              4.2675 %
Total Number of Instances               76

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        into congested
                 1         0          1         1         1          1        congested
                 1         0          1         1         1          1        out congested
                 1         0          1         1         1          1        normal
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 13  0  0  0 |  a = into congested
  0 36  0  0 |  b = congested
  0  0 15  0 |  c = out congested
  0  0  0 12 |  d = normal
```

Figure 4.36: Testing output results

4.6.4.2 Interpreting testing output results

During testing the trained MLP model was used to predict the likelihood of the occurrence of x in equation (4.2) for each instance of sample P when y has already occurred. Equation (4.2) reads as follows: what is the probability of observing x given that y has already occurred.

$$P(x|y) , \tag{4.2}$$

where y is actual data and x is predicted data.

There are four parts shown on the testing results output in Figure 4.36 namely "prediction on test split", "evaluation on test set summary", "detailed accuracy by class" and the "confusion matrix".

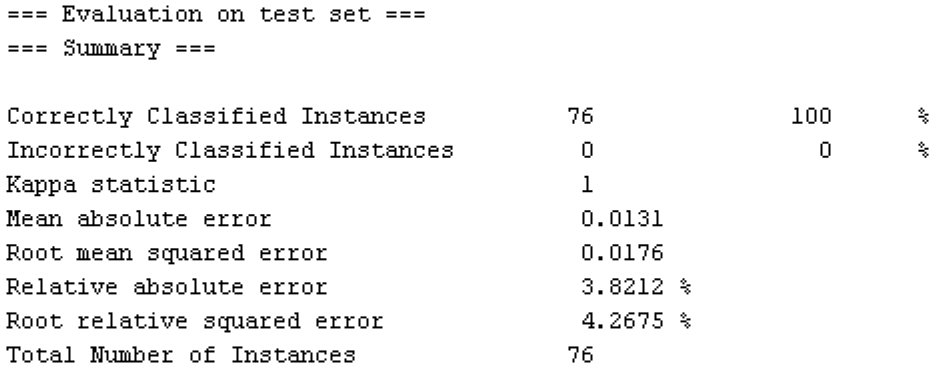
The "confusion matrix" is used for post-processing the results. "Evaluation on test set summary" shows the summary of results achieved during testing by the trained MLP model. Prediction of each novel instance is shown under "predictions on test split". Towards the end of the results output in Figure 4.36 "detailed accuracy by class" section outlines the performance metric of the model.

Each of the 76 instances loaded from excel spreadsheet were predicted. The overall results summary is shown in Table 4.5. The first two lines show the accuracy and error rate of the model. All 76 instances were classified correctly. A Kappa statistic value is 0 for the lack of any relation and 1 for very strong statistical relation. Kappa statistics of 1 shown in Table 4.5 show that the model has very strong statistical relations between the class label and the conjunction of constraints of attributes (instances). The mean absolute error, the root mean square error (RMSE), the relative absolute error and the root relative square error are most useful during numeric prediction. These errors depict the detailed error rate encountered by the MLP model during testing. The mean absolute error is the sum of errors from all incorrectly classified instances, i.e. is the sum of the difference between the predicted and the actual value for each incorrectly predicted instance.

The RMSE is the square of mean absolute errors. The mean absolute error and the RMSE are used to determine the learning rate of the model. The error rate close to zero in Table 4.5 (0.0131 and 0.0176 for mean absolute and root mean square respectively) shows that the knowledge a model obtained during training is good. The closer the error rate is to zero the more accurate is the model during prediction.

Relative absolute error and root relative square error give an idea of the scale of the error compared to how varied the actual values are i.e. the more varied the values, the harder the task of prediction. With 4 percent for both relative absolute error and root relative square error in Table 4.5 it is shown that the prediction task was not difficult due to strong statistical relations between the predicted classes and the actual instances.

Table 4.5: Evaluation on test set summary



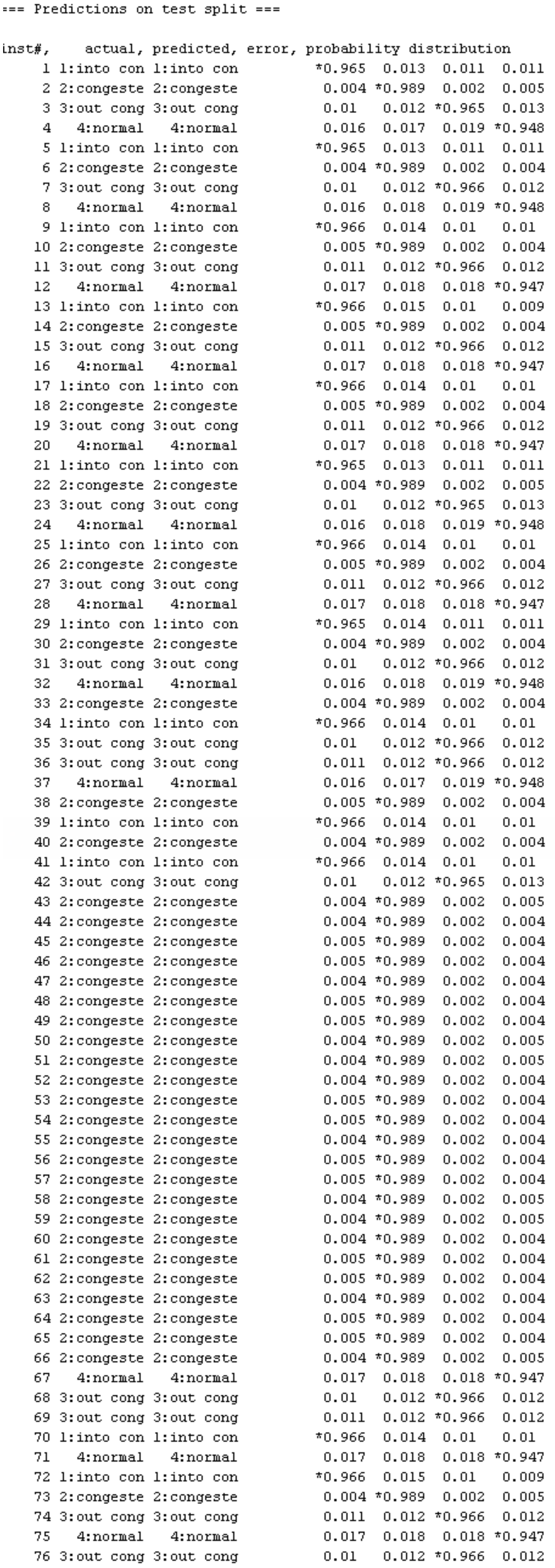
```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          76              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0131
Root mean squared error                  0.0176
Relative absolute error                  3.8212 %
Root relative squared error              4.2675 %
Total Number of Instances               76
```

Table 4.6 shows prediction results for each of the instances loaded from an excel spreadsheet in Figure 4.33. The column heads are identified by inst# (instance number), actual, predicted, error and probability distribution.

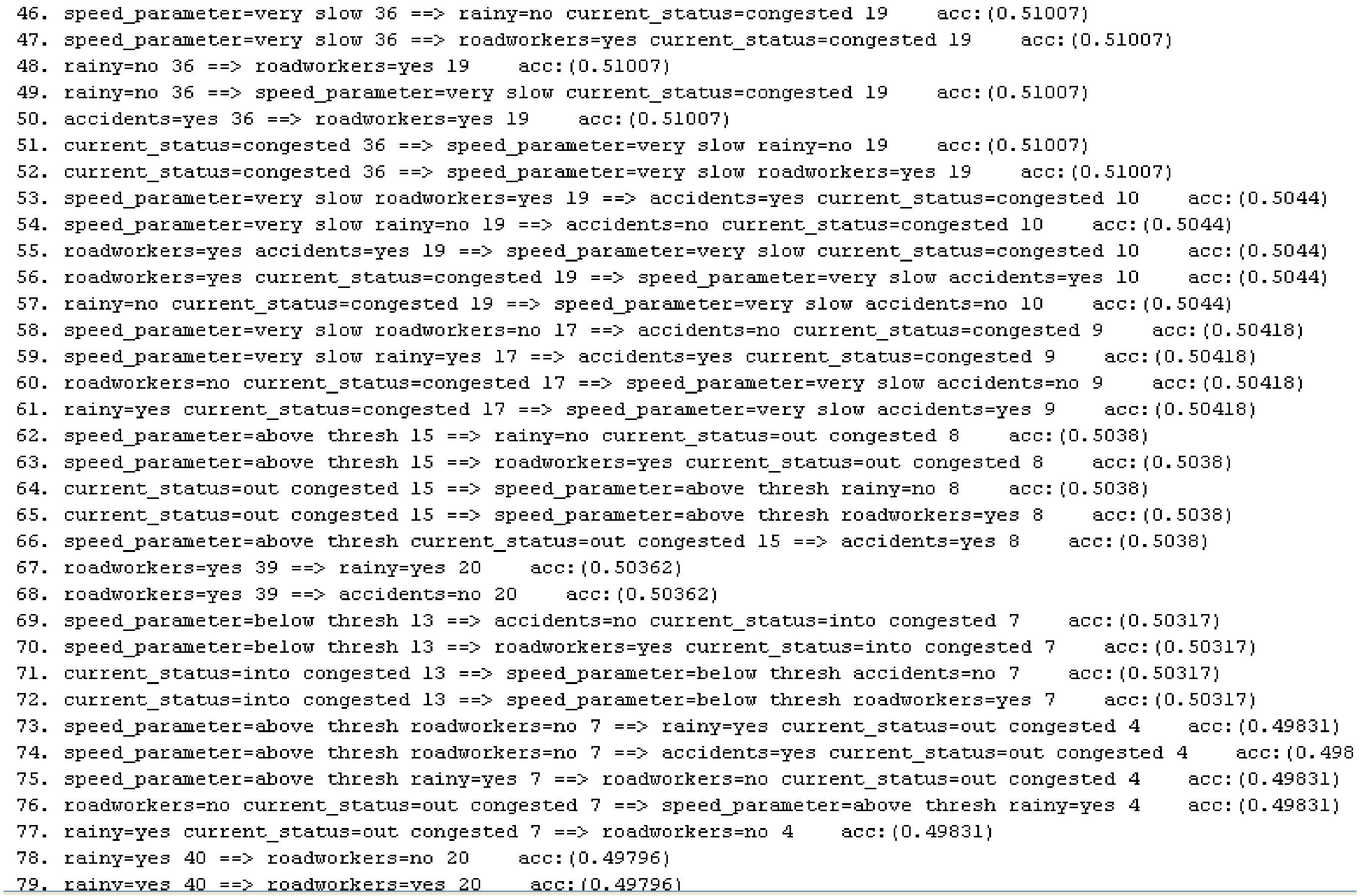
Underneath inst# column are all the 76 instances to be predicted. Actual column shows actual discriticised traffic status based on traffic properties of each instance. The predicted column shows predicted traffic status based on the prior knowledge the model gained during training (actual traffic status column). The error column identifies instances incorrectly classified. The probability distribution has four columns representing four classes to be predicted (from left to right: into congested, congested, out congested and normal). A plus (+) sign inside the error column identifies instances incorrectly classified. A plus (+) sign is not visible and as a result none of the instances were misclassified.

Selecting associate tab in Figure 4.29 gave predictive apriori associator rules shown in Figure 4.37. This tab was used to extract rules (created during training of the MLP model) to be used during testing to predict the four classes (congested, out congested, into congested, normal). It is evident from these rules that the trained MLP model takes into consideration all the 7 inputs (*speed_parameter* = *very slow, speed_parameter* = *below thresh, speed_parameter* = *above thresh, speed_parameter* = *right, rainy, roadworkers* and *accidents*) shown in Figure 4.25 during prediction of new vehicle traffic data.

Table 4.6: Predictions on test data



```
=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1 1:into con 1:into con        *0.965  0.013  0.011  0.011
     2 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
     3 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
     4   4:normal   4:normal         0.016  0.017  0.019 *0.948
     5 1:into con 1:into con        *0.965  0.013  0.011  0.011
     6 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
     7 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
     8   4:normal   4:normal         0.016  0.018  0.019 *0.948
     9 1:into con 1:into con        *0.966  0.014  0.01   0.01
    10 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    11 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    12   4:normal   4:normal         0.017  0.018  0.018 *0.947
    13 1:into con 1:into con        *0.966  0.015  0.01   0.009
    14 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    15 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    16   4:normal   4:normal         0.017  0.018  0.018 *0.947
    17 1:into con 1:into con        *0.966  0.014  0.01   0.01
    18 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    19 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    20   4:normal   4:normal         0.017  0.018  0.018 *0.947
    21 1:into con 1:into con        *0.965  0.013  0.011  0.011
    22 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    23 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
    24   4:normal   4:normal         0.016  0.018  0.019 *0.948
    25 1:into con 1:into con        *0.966  0.014  0.01   0.01
    26 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    27 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    28   4:normal   4:normal         0.017  0.018  0.018 *0.947
    29 1:into con 1:into con        *0.965  0.014  0.011  0.011
    30 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    31 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    32   4:normal   4:normal         0.016  0.018  0.019 *0.948
    33 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    34 1:into con 1:into con        *0.966  0.014  0.01   0.01
    35 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    36 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    37   4:normal   4:normal         0.016  0.017  0.019 *0.948
    38 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    39 1:into con 1:into con        *0.966  0.014  0.01   0.01
    40 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    41 1:into con 1:into con        *0.966  0.014  0.01   0.01
    42 3:out cong 3:out cong         0.01   0.012 *0.965  0.013
    43 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    44 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    45 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    46 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    47 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    48 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    49 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    50 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    51 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    52 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    53 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    54 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    55 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    56 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    57 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    58 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    59 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    60 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    61 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    62 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    63 2:congeste 2:congeste         0.004 *0.989  0.002  0.004
    64 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    65 2:congeste 2:congeste         0.005 *0.989  0.002  0.004
    66 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    67   4:normal   4:normal         0.017  0.018  0.018 *0.947
    68 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
    69 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    70 1:into con 1:into con        *0.966  0.014  0.01   0.01
    71   4:normal   4:normal         0.017  0.018  0.018 *0.947
    72 1:into con 1:into con        *0.966  0.015  0.01   0.009
    73 2:congeste 2:congeste         0.004 *0.989  0.002  0.005
    74 3:out cong 3:out cong         0.011  0.012 *0.966  0.012
    75   4:normal   4:normal         0.017  0.018  0.018 *0.947
    76 3:out cong 3:out cong         0.01   0.012 *0.966  0.012
```



```
46. speed_parameter=very slow 36 ==> rainy=no current_status=congested 19    acc:(0.51007)
47. speed_parameter=very slow 36 ==> roadworkers=yes current_status=congested 19    acc:(0.51007)
48. rainy=no 36 ==> roadworkers=yes 19    acc:(0.51007)
49. rainy=no 36 ==> speed_parameter=very slow current_status=congested 19    acc:(0.51007)
50. accidents=yes 36 ==> roadworkers=yes 19    acc:(0.51007)
51. current_status=congested 36 ==> speed_parameter=very slow rainy=no 19    acc:(0.51007)
52. current_status=congested 36 ==> speed_parameter=very slow roadworkers=yes 19    acc:(0.51007)
53. speed_parameter=very slow roadworkers=yes 19 ==> accidents=yes current_status=congested 10    acc:(0.5044)
54. speed_parameter=very slow rainy=no 19 ==> accidents=no current_status=congested 10    acc:(0.5044)
55. roadworkers=yes accidents=yes 19 ==> speed_parameter=very slow current_status=congested 10    acc:(0.5044)
56. roadworkers=yes current_status=congested 19 ==> speed_parameter=very slow accidents=yes 10    acc:(0.5044)
57. rainy=no current_status=congested 19 ==> speed_parameter=very slow accidents=no 10    acc:(0.5044)
58. speed_parameter=very slow roadworkers=no 17 ==> accidents=no current_status=congested 9    acc:(0.50418)
59. speed_parameter=very slow rainy=yes 17 ==> accidents=yes current_status=congested 9    acc:(0.50418)
60. roadworkers=no current_status=congested 17 ==> speed_parameter=very slow accidents=no 9    acc:(0.50418)
61. rainy=yes current_status=congested 17 ==> speed_parameter=very slow accidents=yes 9    acc:(0.50418)
62. speed_parameter=above thresh 15 ==> rainy=no current_status=out congested 8    acc:(0.5038)
63. speed_parameter=above thresh 15 ==> roadworkers=yes current_status=out congested 8    acc:(0.5038)
64. current_status=out congested 15 ==> speed_parameter=above thresh rainy=no 8    acc:(0.5038)
65. current_status=out congested 15 ==> speed_parameter=above thresh roadworkers=yes 8    acc:(0.5038)
66. speed_parameter=above thresh current_status=out congested 15 ==> accidents=yes 8    acc:(0.5038)
67. roadworkers=yes 39 ==> rainy=yes 20    acc:(0.50362)
68. roadworkers=yes 39 ==> accidents=no 20    acc:(0.50362)
69. speed_parameter=below thresh 13 ==> accidents=no current_status=into congested 7    acc:(0.50317)
70. speed_parameter=below thresh 13 ==> roadworkers=yes current_status=into congested 7    acc:(0.50317)
71. current_status=into congested 13 ==> speed_parameter=below thresh accidents=no 7    acc:(0.50317)
72. current_status=into congested 13 ==> speed_parameter=below thresh roadworkers=yes 7    acc:(0.50317)
73. speed_parameter=above thresh roadworkers=no 7 ==> rainy=yes current_status=out congested 4    acc:(0.49831)
74. speed_parameter=above thresh roadworkers=no 7 ==> accidents=yes current_status=out congested 4    acc:(0.498
75. speed_parameter=above thresh rainy=yes 7 ==> roadworkers=no current_status=out congested 4    acc:(0.49831)
76. roadworkers=no current_status=out congested 7 ==> speed_parameter=above thresh rainy=yes 4    acc:(0.49831)
77. rainy=yes current_status=out congested 7 ==> roadworkers=no 4    acc:(0.49831)
78. rainy=yes 40 ==> roadworkers=no 20    acc:(0.49796)
79. rainy=yes 40 ==> roadworkers=yes 20    acc:(0.49796)
```

Figure 4.37: Predictive Apriori rules used by the trained MLP model during classification

Figure 4.38 was generated from plotting the instances from Table 4.6 that have asterisks. Thirty six of these instances (with a maroon square) show 99 percent average success rate. These maroon squares instances are in the majority at this level. This high success rate is due to high probability of each of these instances shown in Table 4.6 in the 2nd column of "probability distribution" compared to others instances in the 1st, 3rd and 4th columns. This means that traffic in the next 5 minutes is likely to be congested.

In Figure 4.39 the margin curve shows the confidence level of the classifier in predicting all the instances. A margin curve is a plot illustrating the difference between probability predicted for the actual class and the highest probability predicted for the next most likely class for each instance. In Figure 4.39 instance was plotted against a margin value. Instances occupy margin values between -1 and 1. Any instances that lie on the negative region were misclassified by the model. The larger the margin (margin value closer to 1), the more confident the classifier in predicting the true class (traffic status: congestion, into congestion, out of congestion and normal). All instances in Figure 4.39 are occupying a value around 0.98 thus indicate more confidence in predicting the next most likely class.

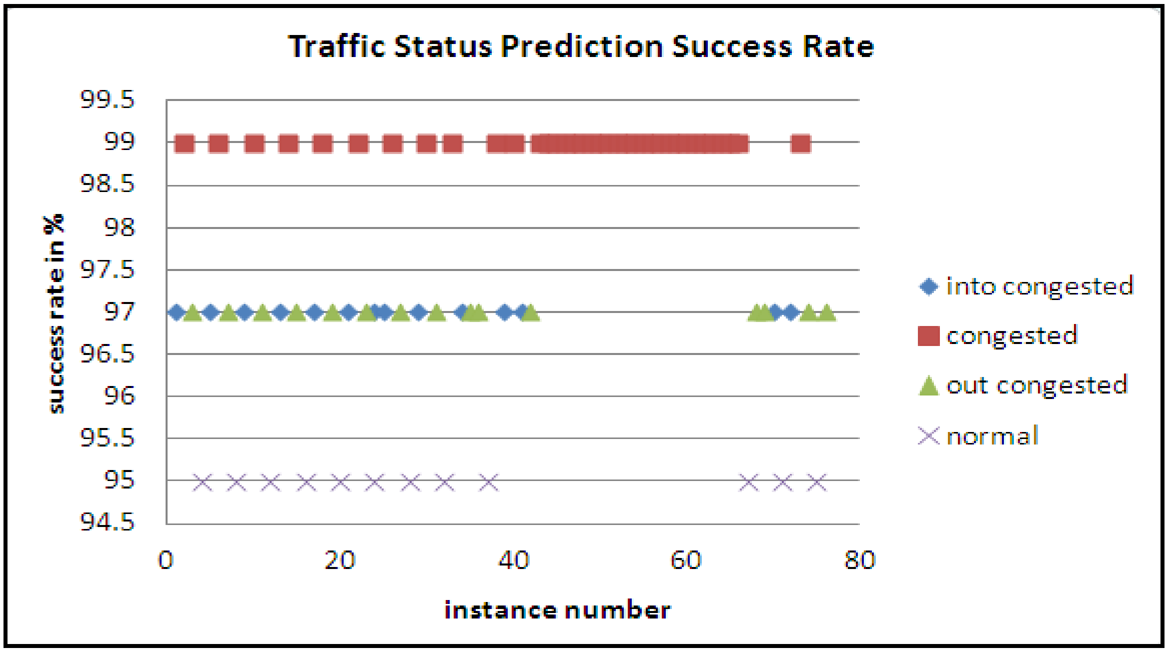
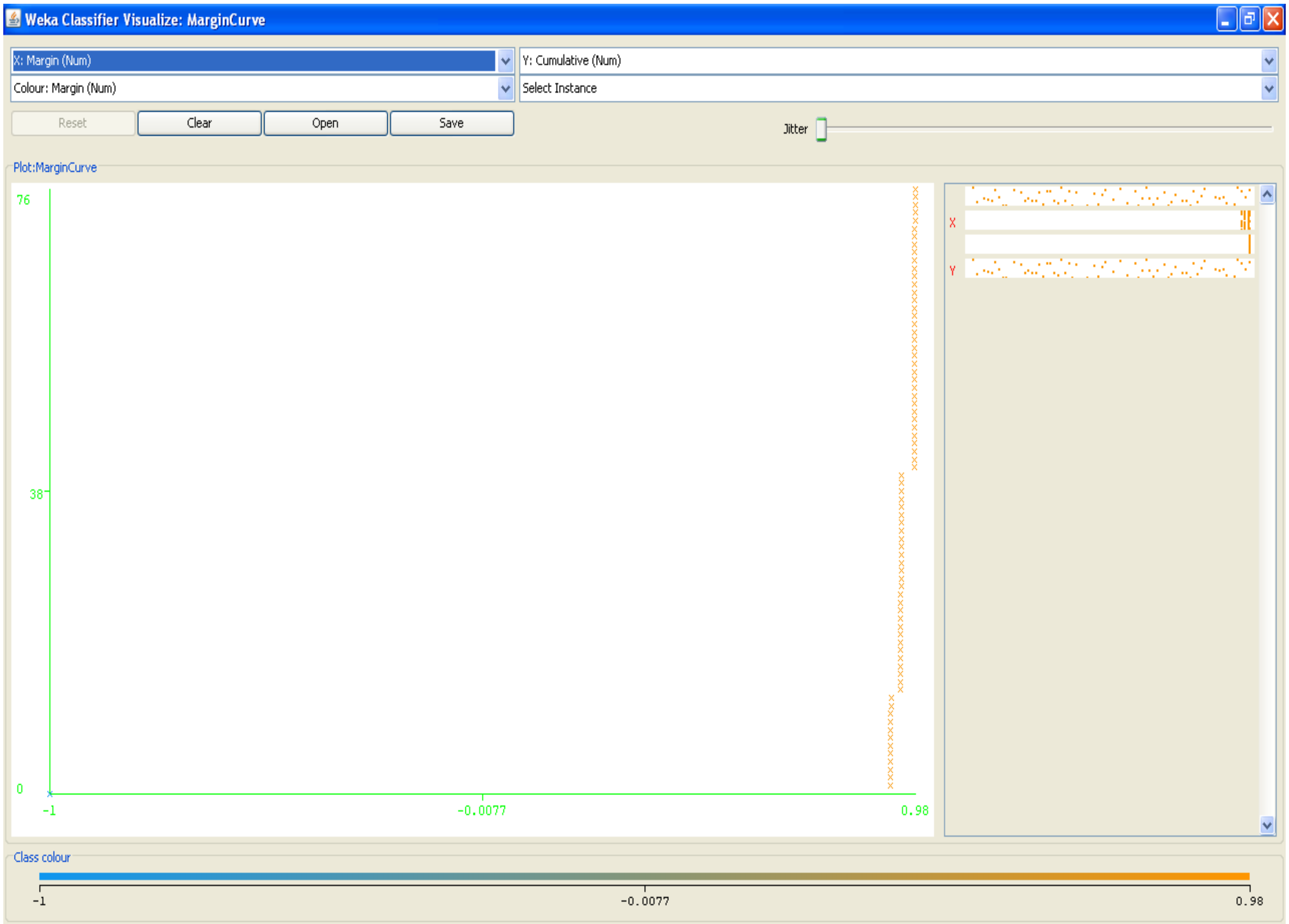


Figure 4.38: Instances with the highest success rate


Weka Classifier Visualize: MarginCurve

X: Margin (Num)

Y: Cumulative (Num)

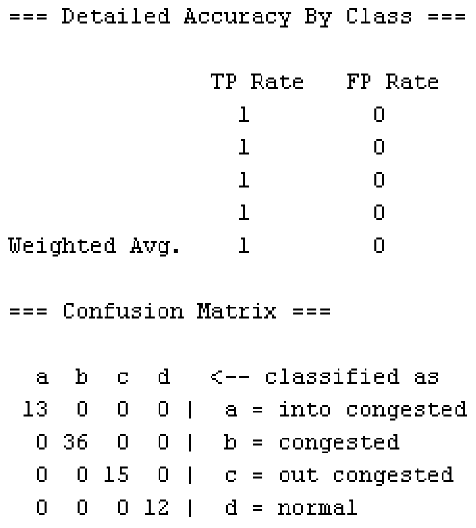
Colour: Margin (Num)

Select Instance

Figure 4.39: MLP margin curve

The accuracy of the algorithm in predicting each of the four classes is shown in Table 4.7. The four conditions are shown as classes in the “detailed accuracy by the class” table (Table 4.7). A “detailed accuracy by class” table makes use of a confusion matrix to determine the performance of the trained MLP model used in predicting each of the four classes (congested, into congested, out congested, normal). A confusion matrix shown in Figure 4.7 is a visualization tool typically used to present the results attained by the learner.

Table 4.7: Detailed accuracy of the model



```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1          1         1          1        into congested
                 1         0          1          1         1          1        congested
                 1         0          1          1         1          1        out congested
                 1         0          1          1         1          1        normal
Weighted Avg.    1         0          1          1         1          1

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 13  0  0  0 |  a = into congested
  0 36  0  0 |  b = congested
  0  0 15  0 |  c = out congested
  0  0  0 12 |  d = normal
```

Each row in a confusion matrix represents the instances in the actual class and each column represents the instances in a predicted class. A benefit of using confusion matrix is that it is easy to see if the system is confusing two classes i.e. mislabelling one class as the other class. Table 4.7 shows a 4x4 confusion matrix (a, b, c and d) representing into congested, congested, out congested and normal in actual and predicted classes respectively. The numbers from the top left corner down to the bottom right corner (13, 36, 15 and 12) represent a main (leading) diagonal. These numbers show all instances classified correctly for each predicted class. Any number(s) which appear above the main diagonal are called False Positive (FP) numbers i.e. it identifies the number of instances mislabelled as belonging to another different class. Any number(s) that appear below the main diagonal are called False Negative (FN) numbers. False negative numbers also identify incorrectly classified instances. The confusion matrix in Table 4.7 shows that the model made 76 correct predictions (13+ 36 + 15 + 12) as shown by the main diagonal numbers. There is no number above or below the main diagonal since the model predicted all 76 instances correctly.

The model accuracy is 100 percent, this value is confirmed by average True Positive (TP) rate of 1 outlined within “detailed accuracy by class” table in Table 4.7. Recall and precision rate in Table 4.7

measures the quality of the classification process. Recall refers to the portion of the positive examples retrieved during the classification process versus the total number of existing positive examples including positive examples not retrieved during classification. Recalls of 1 mean all positive (100 percent) examples were retrieved and classified as positive. As a result for all the four classes all positive examples were all retrieved. It is worth noting that the recall and TP rate values within "detailed accuracy by class" table in Table 4.7 are the same. This is due to the fact that the TP rate is a measure of positive examples which were classified as positive by the classifier.

Precision is defined as the portion of the positive examples that exist in the total number of examples retrieved i.e. out of all examples retrieved for each class how many were classified positive? 100 percent of examples were classified positive for congested, into congested, out congested and normal out of all examples retrieved as shown by a Precision value of 1 for each class.

Receiver Operating Characteristics (ROC) area shown inside "detailed accuracy by class" table in Table 4.7 is a metric used to gain insight into the decision making ability of the model i.e. how likely is it that the model can accurately predict the negative or the positive class? ROC measures the impact of changes in the probability threshold. The probability threshold is the decision point used by the model for classification. The default classification threshold for binary (positive and negative) classification is 0.5. When the probability of prediction is 50 percent or more, the model predicts the class. When the probability is less than 50 percent the other class is predicted. In a multiclass classification the predicted class is the one with the highest probability. Looking at the ROC column, a value of 1 shows that this trained MLP model decision making is 100 percent spot on in identifying positive and negative examples.

4.7 Experiment 6: Displaying traffic status information on the highway

Predicted highway traffic status based on novel traffic data in experiment 5 needs to be displayed on the billboards along the highway for the motorists to read and based on traffic status help the motorists to make use of alternative routes.

Equipment and software used

This experiment was carried out using the following equipment and software:

- Lenovo Pentium 4 dual core E8400 computer with 3 GHz CPU and 2 GB of RAM

Operating Systems

- Windows XP version 5.1.2600 service pack 2 build 2600

Application software

- Proteus 7 professional
- PIC-C compiler version 4.038

Objective

This experiment simulates how traffic status information can be broadcast on using a LCD for the motorists on the simulated highway to be able to view or read.

In this experiment a Programmable Input Controller (PIC) was used together with the Liquid Crystal Display (LCD) to display prediction results on the simulated highway. The LCD received traffic status information from a remote Server (RTCC server) through a wireless medium. The remote server used artificial intelligence (multilayer perceptron in experiment 5) to analyse data it received from vehicle speed sensors (placed on the highway) and determined the traffic status shown in Figure 4.36.

Traffic status was discretised as follows:

- Congestion - vehicle speed value below 45.4km/h.
- Into congestion - vehicle speed value between 45.5 – 60.4 km/h
- Out of congestion - vehicle speed value between 60.5 – 80.4 km/h
- Normal traffic - vehicle speed value between 80.5 – 120.4km/h.

During the normal traffic condition the LCD displays a message to indicate that traffic is operating normal.

The following steps (recipe) were followed when carrying out this experiment:

1. Draw schematic presentation of virtual devices using Proteus ISIS professional software and create connections between devices using devices’ driver file and datasheets diagrams
2. Use PIC-C compiler wizard to configure initial micro chip settings
3. Code the PIC program

4. Modify the Proteus schematic properties diagram to match with the PIC-Compiler settings and thereafter upload coded source file.

Step 1

Figure 4.40 shows the schematic presentation of virtual devices used in the design of the display board. The schematic diagram in Figure 4.40 was done using Proteus ISIS professional. The connections between devices were done with reference to PIC18F4320, 40 x 2 LM018L LCD and 4x3 keypad's datasheet diagrams.

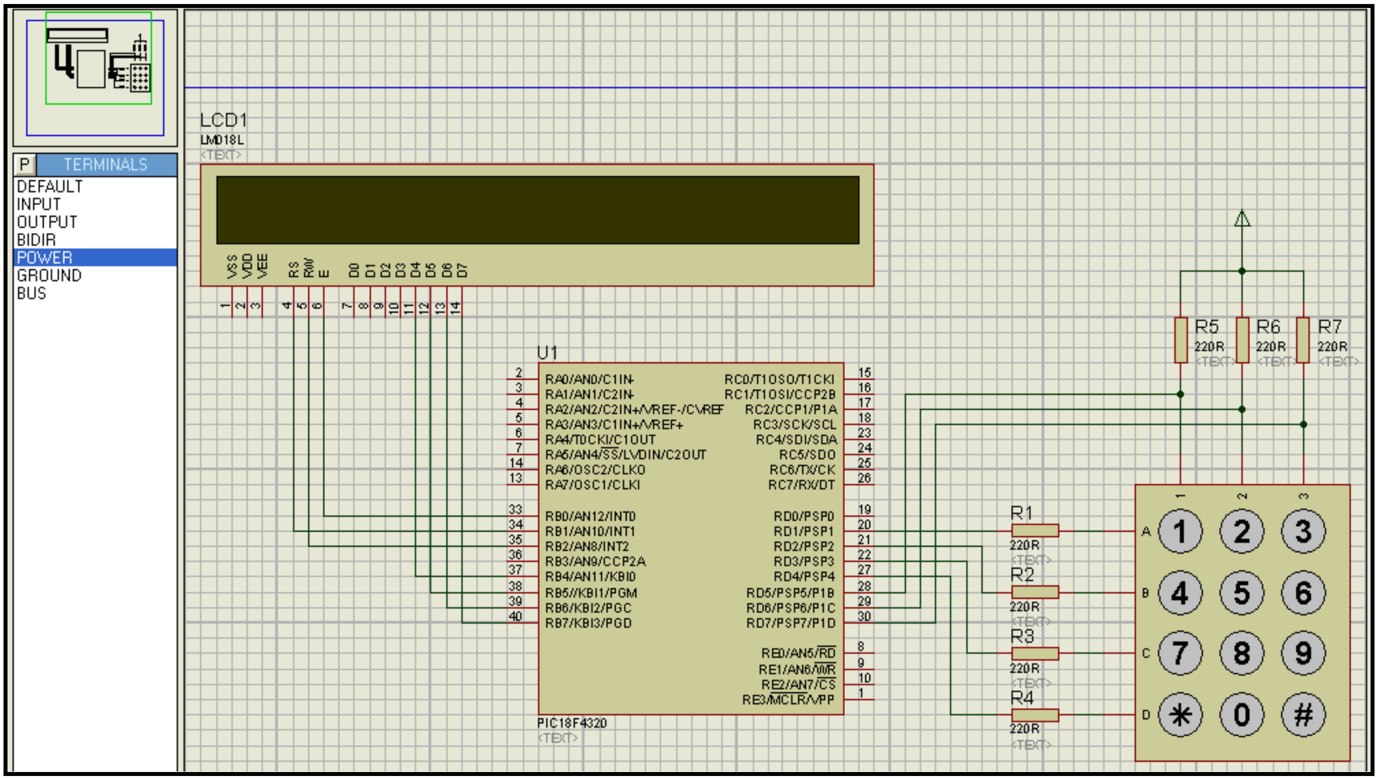


Figure 4.40: Schematic diagram for LCD display components drawn using Proteus ISIS professional

In Figure 4.40 the microcontroller (PIC18F4320) was used to read predicted traffic status information from the RTCC server and write that information on the 40 x 2 LM018L LCD screen. A keypad (manually) was used as a RTCC server to send predicted traffic status information to a LCD screen based on discretised speed values. Regulation of current between a microcontroller and the keypad was done using METALFILM220R resistors.

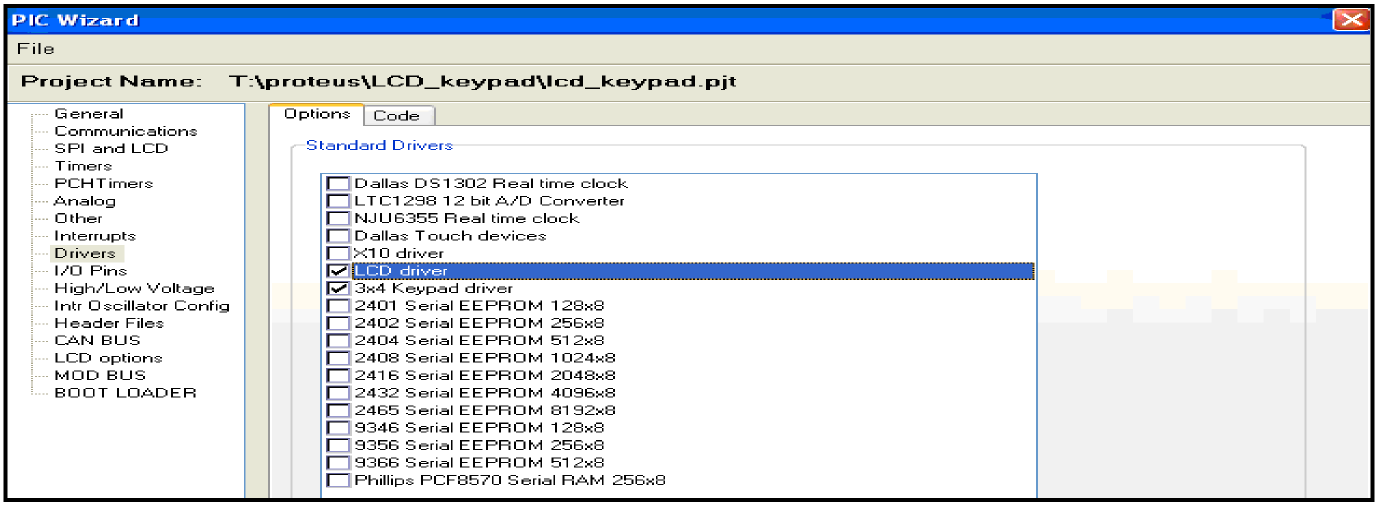
Step 2

Once the schematic in Figure 4.40 was saved, the PIC-C compiler software was launched. PIC-C in this experiment was used to provide settings required for the components shown in Figure 4.40 to allow communication between each of them.

In the general properties screen of PIC-C wizard the following settings were done:

- Controller: PIC18F4320
- Oscillation frequency to 8MHz
- Select Internal RC Osc, no CLKOUT
- Clear all fuses

The PIC-C compiler has built-in support for the keyboard and the LCD as a result in Figure 4.41, keypad and LCD drivers were loaded so that the compiler can recognize these components.



Project Name: T:\proteus\LCD_keypad\lcd_keypad.pjt

- ☐ Dallas DS1302 Real time clock
- ☐ LTC1298 12 bit A/D Converter
- ☐ NJU6355 Real time clock
- ☐ Dallas Touch devices
- ☐ X10 driver
- ☑ LCD driver
- ☑ 3x4 Keypad driver
- ☐ 2401 Serial EEPROM 128x8
- ☐ 2402 Serial EEPROM 256x8
- ☐ 2404 Serial EEPROM 512x8
- ☐ 2408 Serial EEPROM 1024x8
- ☐ 2416 Serial EEPROM 2048x8
- ☐ 2432 Serial EEPROM 4096x8
- ☐ 2465 Serial EEPROM 8192x8
- ☐ 9346 Serial EEPROM 128x8
- ☐ 9356 Serial EEPROM 256x8
- ☐ 9366 Serial EEPROM 512x8
- ☐ Phillips PCF8570 Serial RAM 256x8

Figure 4.41: Loading a keyboard and LCD drivers

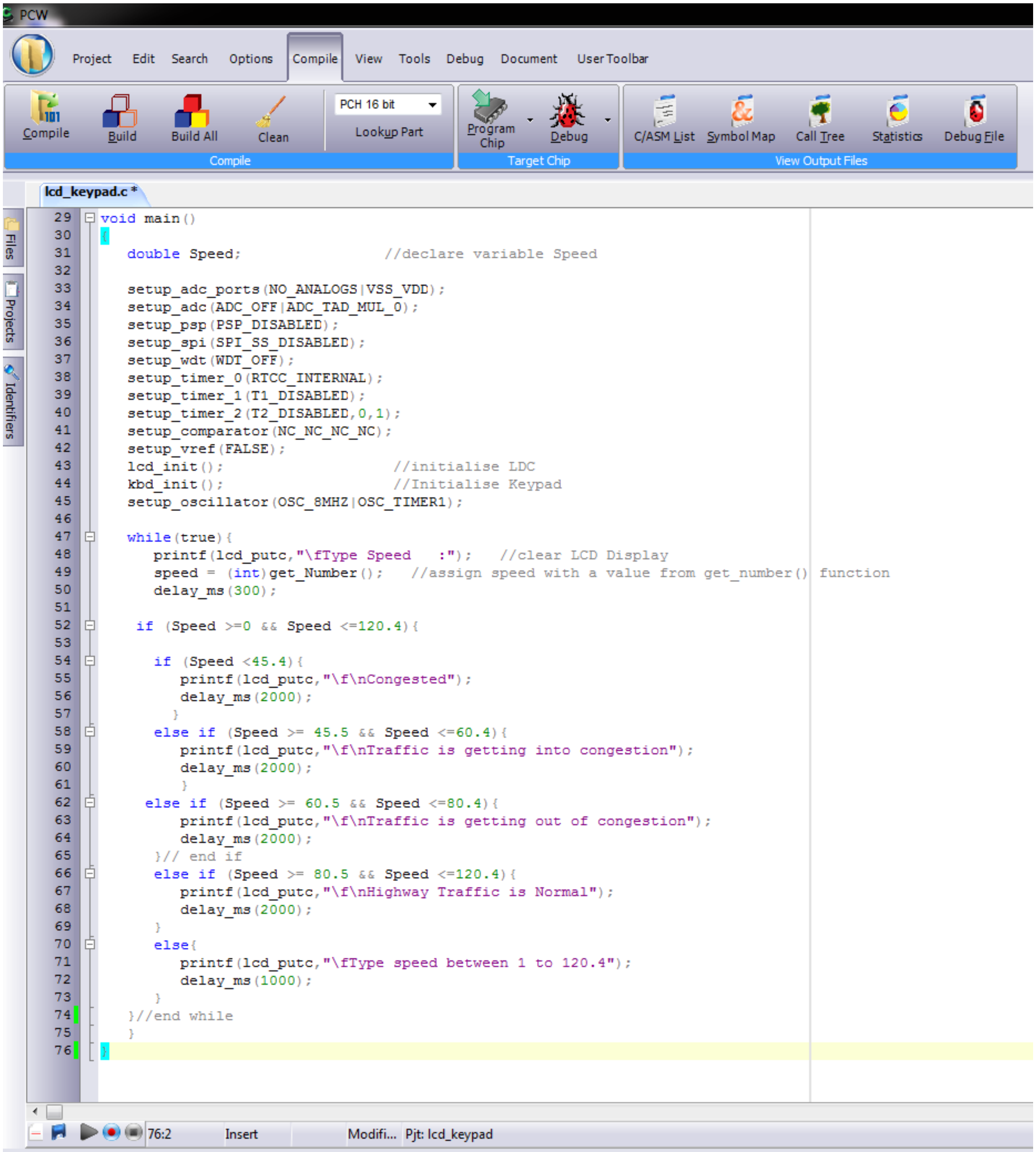
Step 3

C programming language was used to program the source code inside PIC-C as shown in Figure 4.42.

- In line 43 and 44, both the keyboard and the LCD are initialised.
- Inside the driver file, the LCD was set to be connected to the PORTB of the PIC18F4320 and the keyboard was set to be connected to PORTD of the PIC18F4320
- Line 52 to line 67 are conditions of discretised speed to be displayed on the highway

- Line 71 the keyboard is used to provide the discretised input speed as predicted by multilayer perceptron in experiment 5

Once coding was complete the program was built and compiled in order to generate the machine code (HEX file) to be loaded into the PIC.

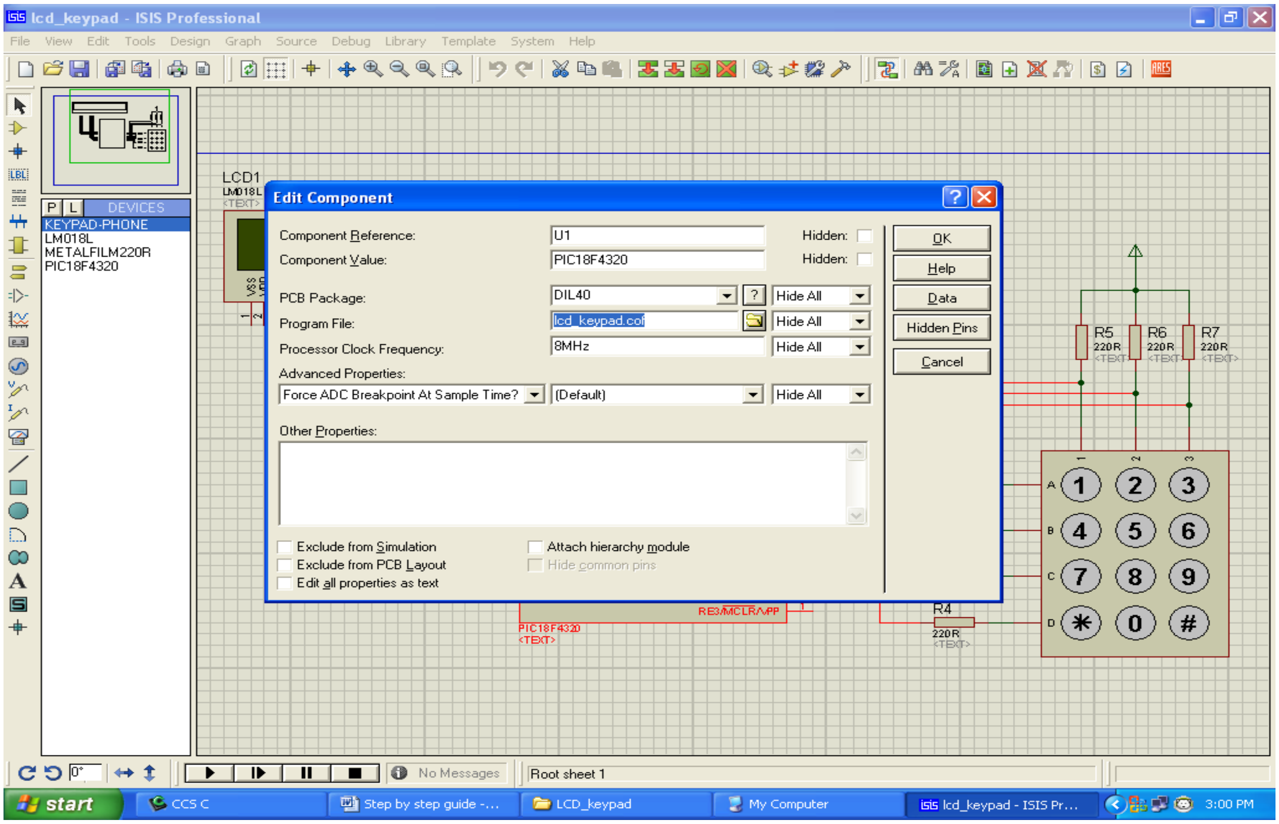


```
29  void main()
30  {
31     double Speed;                  //declare variable Speed
32
33     setup_adc_ports(NO_ANALOGS|VSS_VDD);
34     setup_adc(ADC_OFF|ADC_TAD_MUL_0);
35     setup_psp(PSP_DISABLED);
36     setup_spi(SPI_SS_DISABLED);
37     setup_wdt(WDT_OFF);
38     setup_timer_0(RTCC_INTERNAL);
39     setup_timer_1(T1_DISABLED);
40     setup_timer_2(T2_DISABLED,0,1);
41     setup_comparator(NC_NC_NC_NC);
42     setup_vref(FALSE);
43     lcd_init();                     //initialise LDC
44     kbd_init();                     //Initialise Keypad
45     setup_oscillator(OSC_8MHZ|OSC_TIMER1);
46
47     while(true){
48        printf(lcd_putc,"\fType Speed    :");   //clear LCD Display
49        speed = (int)get_Number();   //assign speed with a value from get_number() function
50        delay_ms(300);
51
52      if (Speed >=0 && Speed <=120.4){
53
54        if (Speed <45.4){
55           printf(lcd_putc,"\f\nCongested");
56           delay_ms(2000);
57          }
58        else if (Speed >= 45.5 && Speed <=60.4){
59           printf(lcd_putc,"\f\nTraffic is getting into congestion");
60           delay_ms(2000);
61           }
62       else if (Speed >= 60.5 && Speed <=80.4){
63           printf(lcd_putc,"\f\nTraffic is getting out of congestion");
64           delay_ms(2000);
65        }// end if
66        else if (Speed >= 80.5 && Speed <=120.4){
67           printf(lcd_putc,"\f\nHighway Traffic is Normal");
68           delay_ms(2000);
69        }
70        else{
71           printf(lcd_putc,"\fType speed between 1 to 120.4");
72           delay_ms(1000);
73        }
74     }//end while
75     }
76  }
```

Figure 4.42: The compiler source code

Step 4

In this step the Proteus schematic microcontroller settings and the actual PIC18F4320 microcontroller settings were compared in order to ensure that both settings match. This comparison is required since in this experiment the schematic in Figure 4.40 was drawn using Proteus software and source code programmed using C programming inside PIC-C development software. In order for the settings to match, the Proteas "component value" should point to use PIC18F44320 as shown in Figure 4.43. The "processor clock frequency" should also be set to 8 MHz so that the same clock rate is used by both the microcontroller in Proteus and PIC-C softwares.



Edit Component

Component Reference: U1 Hidden: □
Component Value: PIC18F4320 Hidden: □
PCB Package: DIL40 Hide All
Program File: lcd_keypad.cof Hide All
Processor Clock Frequency: 8MHz Hide All
Advanced Properties:
Force ADC Breakpoint At Sample Time? (Default) Hide All
Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Edit all properties as text
□ Attach hierarchy module
□ Hide common pins

Figure 4.43: Matching the Proteus schematic diagram property settings

The source code was loaded from the program file path specified in Figure 4.43. Once loaded a play radio button shown on the bottom left corner was used to start a simulation. A keypad in Figure 4.40 was used to send results of prediction done using MLP inside the RTCC-database server to LCD screen located on the simulated highway. Based on the forecasted traffic status (congestion, into congestion, out of congestion or normal highway traffic) a keypad was used to manually type discretised values 0 to 45.4

representing congestion, 45.5 to 60.4 representing into congestion, 60.5 to 80.4 representing out of congestion and 80.5 to 120.4 representing normal traffic on the highway.

4.7.1 Results

Based on the results of prediction shown in Figure 4.36, the keypad representing the RTCC-database server in Figure 4.44 was used to send a discretised value (value between 0 and 45.4) to the LCD screen thus a message in Figure 4.44 was displayed on the LCD screen installed on simulated Ben Schoeman highway for motorists to read.

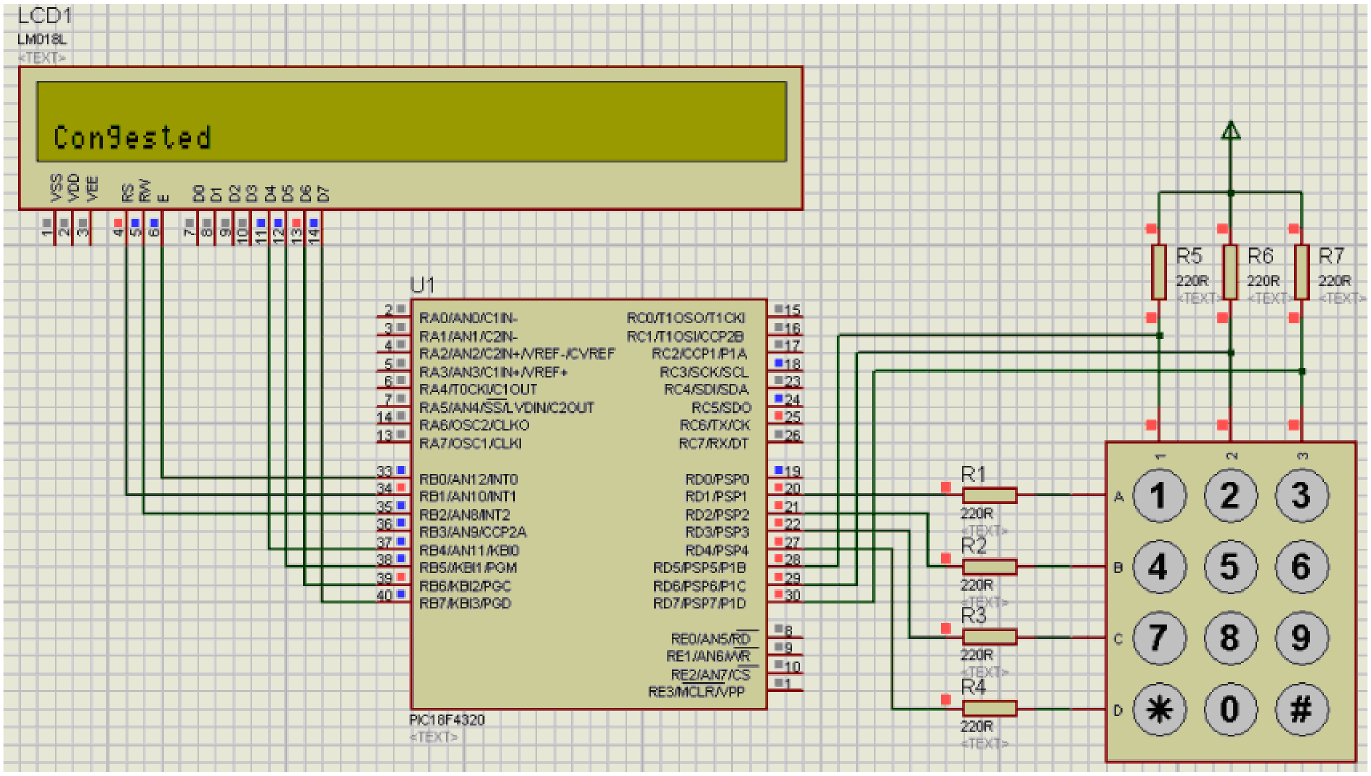


Figure 4.44: Traffic status displayed by LCD placed along the simulated Ben Schoeman highway

The results shown in Figure 4.44 indicate that in the next 5 minutes vehicles will be travelling at the speed between 0 to 45.4km/h. As a result motorists who are racing against time can choose to switch to an alternative route. The ability to forecast traffic status in the next 5 minutes shows the intelligence part of this traffic jam project. On the real LCD system, wireless medium would be used to create a connection to the RTCC-database.

4.8 Experiment 7: Securing the entire RTCC network including wireless links using PEAP with EAP-TLS

Equipment and software used

This experiment was carried out using the following equipment and software:

- 5 Lenovo dual core E8400 Pentium 4 computers with 3 GHz CPU and 2 GB of RAM
- 802.11n Cisco Linksys wireless router

Operating Systems installed inside the computers

- Windows 2003 enterprise server with service pack 1 (all servers)
- Windows XP version 5.1.2600 service pack 2 build 2600 (wireless client)

Objective:

This experiment was carried out in order to secure access to the RTCC network including access to the wireless link that is a conduit for vehicle speed sensors to send vehicle speed to the RTCC-database server. Securing this wireless link will ensure that the RTCC-database server does not receive vehicle speed data from spoofed speed sensors. Spoofed sensors here refer to the sensors which a hacker can use if the wireless link is not secure to send false vehicle speed data to the RTCC-database. The RTCC-database will accept this vehicle speed data hoping that it comes from the legitimate sensors placed on the motorway. Furthermore a hacker can take advantage of an unsecured network to change captured vehicle speed records inside the RTCC-database. The consequence of both cases can be disastrous.

In this experiment user and computer certificates will be used for authentication.

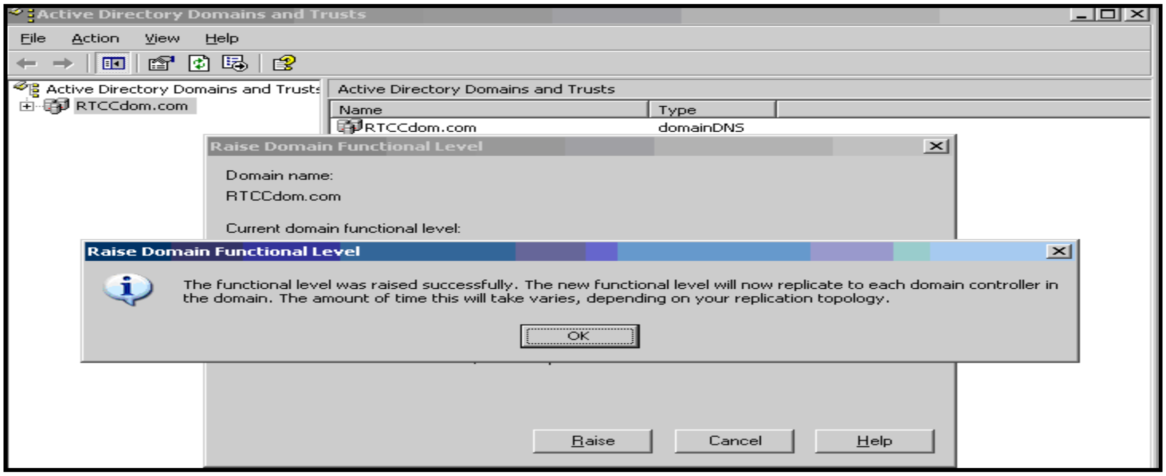
The following steps were followed to conduct this experiment:

1. Configure the Domain Name System (DNS) server and raise the domain functionality
2. Configure the Certification Authority (CA)
3. Setup the Internet Authentication Service (IAS) server
4. Configure the Wireless Access Point (WAP)
5. Connect a wireless client to the secured RTCC network

Step 1: Configuring the Domain Name System (DNS) server and raising the domain functionality

The domain controller named RTCCDOM together with the active directory was installed inside the DNS server (Figure 4.2) by issuing out a command *dcpromo* from "run" windows textbox. The RTCCDOM in this project is a computer running Windows 2003 enterprise server with service pack 1 included. In this work it was used to centrally administer domain user accounts, computer accounts and the allocation of permissions to all network resources within the RTCC network. Usernames and password used to login to the domain was created inside the RTCCDOM. Computers in the RTCC network all join the domain name RTCCDOM in order for them to be able to communicate between one another. The allocation of permission to users and computers is one of the major functions of this domain controller.

When the first windows server 2003 based domain controller is deployed in a domain, the domain operates in native mode which is the default lowest functional level. A domain functional level provides a means of enabling additional domains and forest up to date wide range of active directory features. The domain in native mode only supports Windows 2000 based domain controllers which are outdated features. In order to take advantage of a new set of features available, domain functionality level in Figure 4.45 was raised from Windows 2000 mixed mode to the highest level available in Windows which is Windows 2003 server mode. This upgrade improves active directory performance and security.


Active Directory Domains and Trusts
File
Action
View
Help
Active Directory Domains and Trusts
RTCCdom.com
Active Directory Domains and Trusts
Name
Type
RTCCdom.com
domainDNS
Raise Domain Functional Level
Raise Domain Functional Level
OK
Raise
Cancel
Help

Domain name:
RTCCdom.com

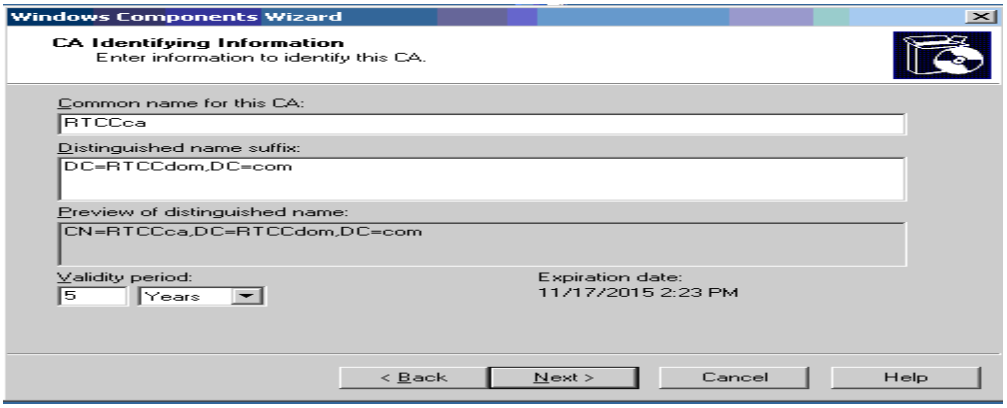
Current domain functional level:

The functional level was raised successfully. The new functional level will now replicate to each domain controller in the domain. The amount of time this will take varies, depending on your replication topology.

Figure 4.45: Successful completion of RTCCDOM functionality level upgrading

Step 2: Configuring the Certification Authority (CA)

Once the domain controller was set up the "add or remove" windows components were used to install certificate services inside the CERTIFICATION server (Figure 4.2). The certification server in this experiment was a computer with Windows 2003 enterprise server installed. The certification server was used to manage issuing, renewing and validation of digital certificate within RTCCDOM domain. These certificates were issued to RTCCDOM domain users and computers. The name of the trusted root Certification Authority (CA) RTCCca which was created is shown in Figure 4.46. All the computers in the Local Area Network (LAN) that comprise the traffic monitoring system had to have a certificate to be used during identity validation. This was done through the RTCCDOM domain controller.



CA Identifying Information
Enter information to identify this CA.

Common name for this CA:
RTCCca

Distinguished name suffix:
DC=RTCCdom,DC=com

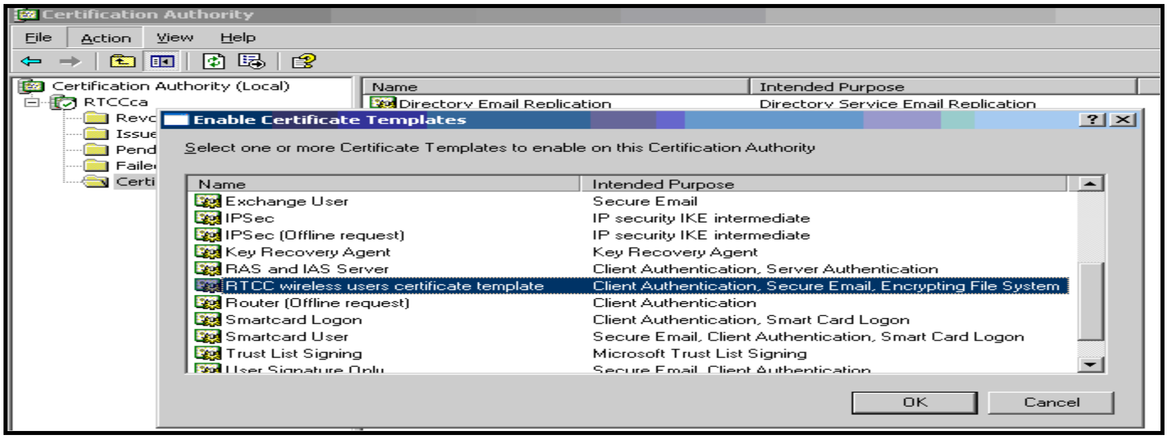
Preview of distinguished name:
CN=RTCCca,DC=RTCCdom,DC=com

Validity period: 5 Years

Expiration date:
11/17/2015 2:23 PM

Figure 4.46: Trusted certification authority RTCCca to manage certificates within the RTCCDOM domain

In Figure 4.47 a template name “RTCC wireless users certificate” was enabled. This template name identifies the certificate template to be issued to RTCCDOM domain users. All RTCCDOM domain users obtained this certificate on logging into the network. In regard to this work, the user accounts used to login to the RTCCDOM from IIS, IAS, RTCC-database server and wireless clients (speed sensors) used the certificate template shown in Figure 4.47 during authentication to identify themselves to the root CA RTCCca.



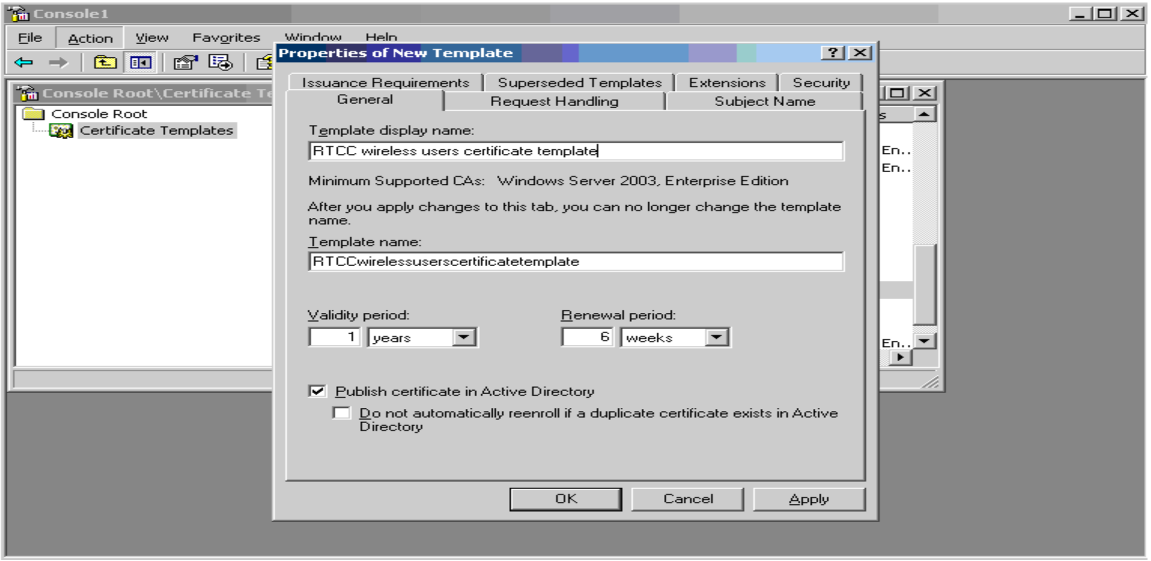
Name	Intended Purpose
Directory Email Replication	Directory Service Email Replication

Select one or more Certificate Templates to enable on this Certification Authority

Name	Intended Purpose
Exchange User	Secure Email
IPSec	IP security IKE intermediate
IPSec (Offline request)	IP security IKE intermediate
Key Recovery Agent	Key Recovery Agent
RAS and IAS Server	Client Authentication, Server Authentication
RTCC wireless users certificate template	Client Authentication, Secure Email, Encrypting File System
Router (Offline request)	Client Authentication
Smartcard Logon	Client Authentication, Smart Card Logon
Smartcard User	Secure Email, Client Authentication, Smart Card Logon
Trust List Signing	Microsoft Trust List Signing
User Signature Only	Secure Email, Client Authentication

Figure 4.47: Enabling the certificate template issued to RTCCDOM domain users by the Certification authority RTCCca

The certificate template enabled in Figure 4.47 was published in active directory as shown by a tick inside the check box in Figure 4.48. Publishing the certificate in active directory makes the certificate to be available to RTCCDOM domain users.



Template display name:
RTCC wireless users certificate template

Minimum Supported CAs: Windows Server 2003, Enterprise Edition

After you apply changes to this tab, you can no longer change the template name.

Template name:
RTCCwirelessuserscertificatetemplate

Validity period: 1 years

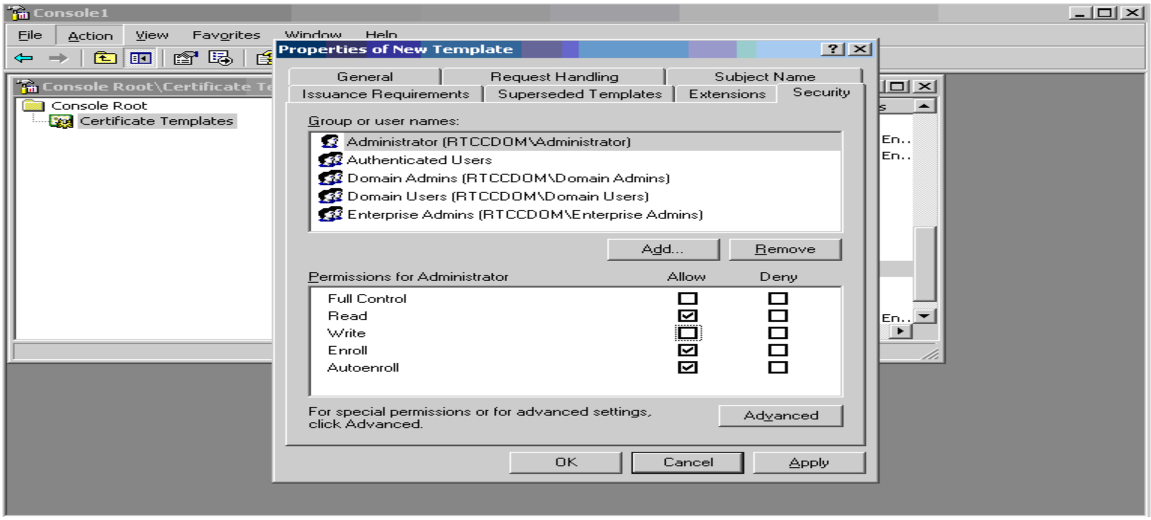
Renewal period: 6 weeks

☑ Publish certificate in Active Directory

□ Do not automatically reenroll if a duplicate certificate exists in Active Directory

Figure 4.48: Publishing of a certificate template in the active directory

The RTCCDOM administrator is used to request a certificate on behalf of each valid domain user during the first domain logging process. In order for the RTCCDOM.COM administrator to fulfil this request the RTCCDOM administrator was given read, enrol and auto- enrolment permission as shown in Figure 4.49. A valid domain user refers to a user which appears within a list of RTCCDOM domain users' database.



Group or user names:

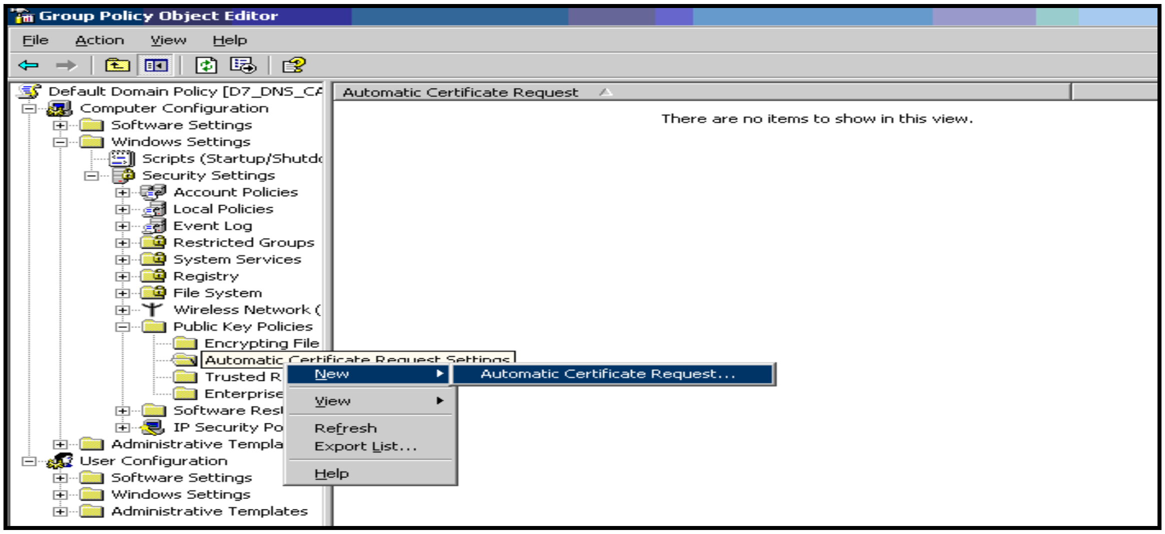
- Administrator (RTCCDOM\Administrator)
- Authenticated Users
- Domain Admins (RTCCDOM\Domain Admins)
- Domain Users (RTCCDOM\Domain Users)
- Enterprise Admins (RTCCDOM\Enterprise Admins)

Permissions for Administrator	Allow	Deny
Full Control	□	□
Read	☑	□
Write	□	□
Enroll	☑	□
Autoenroll	☑	□

For special permissions or for advanced settings, click Advanced.

Figure 4.49: Assigning auto-enrol permissions to RTCCDOM administrator

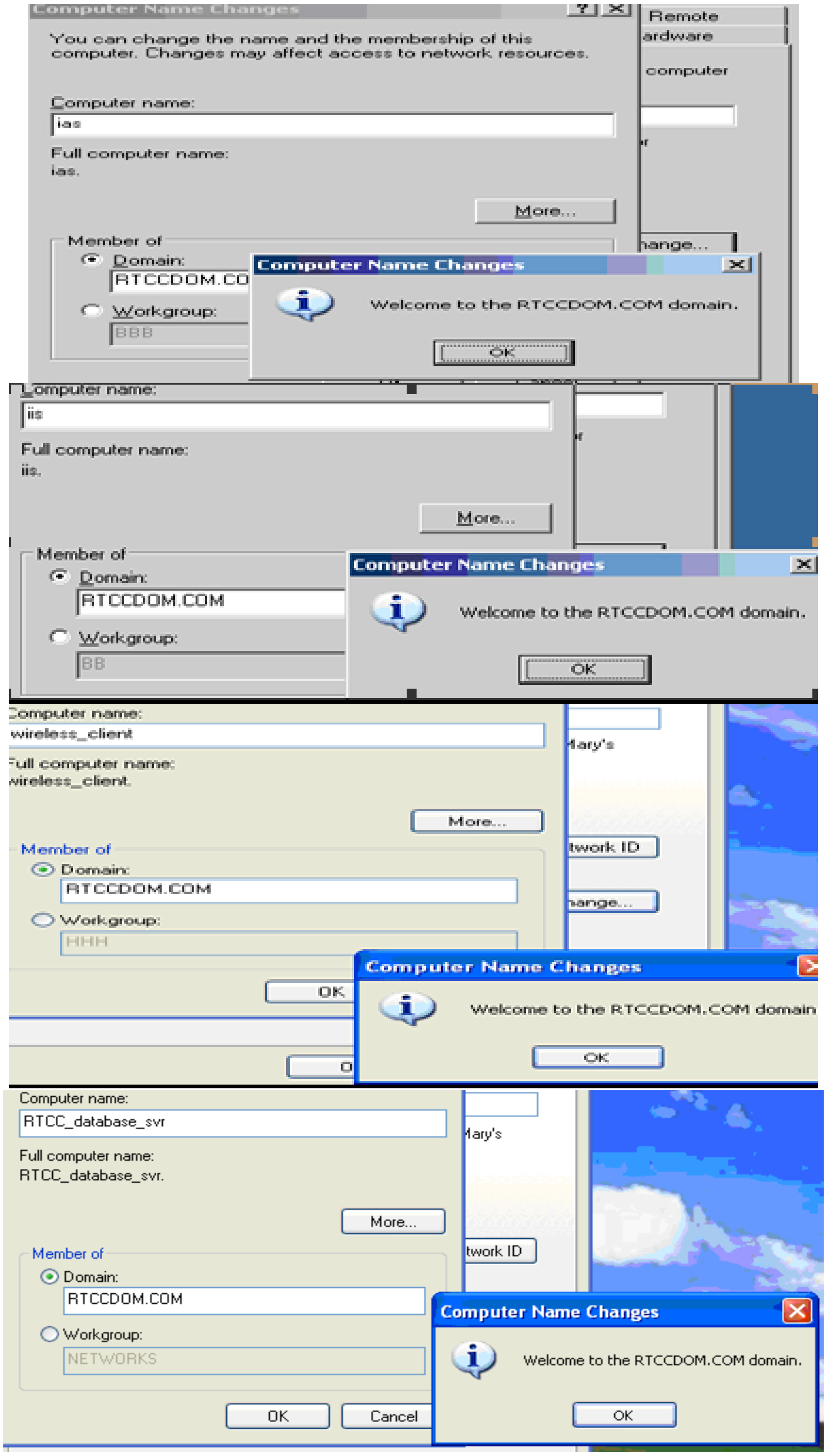
Mutual authentication used by PEAP requires that the domain computers also obtain certificates to be used during the authentication process. In Figure 4.50 computer configuration public key policy was used to import published certificates thus making the certificate available to a RTCCDOM domain computer.



There are no items to show in this view.

Figure 4.50: Obtaining a digital certificate using an automatic certificate request

The computers and users obtain certificates when logging into the domain for the first time. As a result once the configuration which allows a domain user and computer to obtain certificates was done a computer name tab which appears as one of the computer properties was used to join IAS, IIS, wireless client and RTCC-database server to the RTCCDOM domain as shown in Figure 4.51. By joining the RTCCDOM domain registers IAS, IIS, wireless client and RTCC-database server as legitimate domain members who can gain access to network resources within RTCCDOM. At the same time they are given digital certificates to be used during domain logging authentication.



Computer Name Changes

You can change the name and the membership of this computer. Changes may affect access to network resources.

Computer name: ias

Full computer name:
ias.

Member of
● Domain: RTCCDOM.CO
○ Workgroup: BBB

Computer Name Changes

Welcome to the RTCCDOM.COM domain.

Computer name: iis

Full computer name:
iis.

Member of
● Domain: RTCCDOM.COM
○ Workgroup: BB

Computer Name Changes

Welcome to the RTCCDOM.COM domain.

Computer name: wireless_client

Full computer name:
wireless_client.

Member of
● Domain: RTCCDOM.COM
○ Workgroup: HHH

Computer Name Changes

Welcome to the RTCCDOM.COM domain

Computer name: RTCC_database_svr

Full computer name:
RTCC_database_svr.

Member of
● Domain: RTCCDOM.COM
○ Workgroup: NETWORKS

Computer Name Changes

Welcome to the RTCCDOM.COM domain.

Figure 4.51: Joining IAS, IIS, wireless client and RTCC-database server to RTCCDOM.COM domain

All the computers (the IAS, IIS, wireless client and RTCC-database server) welcomed to the RTCCDOM network in Figure 4.51 formed a secured RTCC network controlled by RTCCDOM domain controller. Upon joining the RTCCDOM domain the certificate shown in Figure 4.52 was issued out to IAS, IIS, wireless client and RTCC-database server.

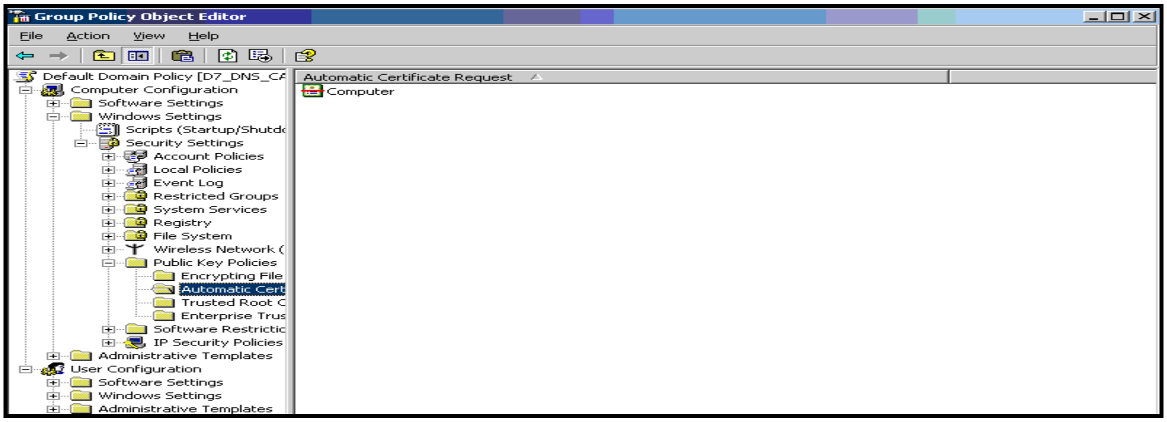
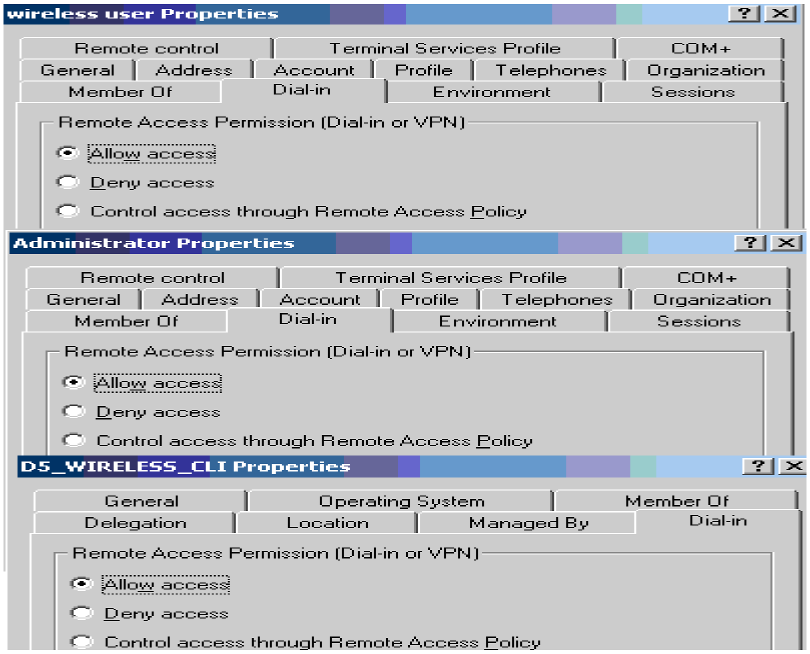


Figure 4.52: A certificate issued to computers

- **Enable wireless client access to RTCC network through the wireless medium**

A user account named wireless to be used to login from a wireless client computer and a group called RTCCwirelessGROUP was created within the RTCCDOM domain. The wireless user account, the RTCCDOM administrator and the wireless client computer accounts in Figure 4.53 were allocated remote access dial in permissions and became members of RTCCwirelessGROUP. The remote dial-in permissions are permissions which allow RTCCwirelessGROUP members to connect to the RTCC network using a wireless medium. In this project the wireless clients are the vehicle speed sensors installed on the highway. The dial-in permissions were allocated to these vehicle speed sensors so they could use the wireless link to send vehicle traffic speeds. Each vehicle speed sensor was made to be a member of RTCCwirelessGROUP by the traffic monitoring authority before it could be allowed to use the wireless medium.



wireless user Properties

Remote control | Terminal Services Profile | COM+
General | Address | Account | Profile | Telephones | Organization
Member Of | Dial-in | Environment | Sessions

Remote Access Permission (Dial-in or VPN)

● Allow access
○ Deny access
○ Control access through Remote Access Policy

Administrator Properties

Remote control | Terminal Services Profile | COM+
General | Address | Account | Profile | Telephones | Organization
Member Of | Dial-in | Environment | Sessions

Remote Access Permission (Dial-in or VPN)

● Allow access
○ Deny access
○ Control access through Remote Access Policy

DS_WIRELESS_CLI Properties

General | Operating System | Member Of
Delegation | Location | Managed By | Dial-in

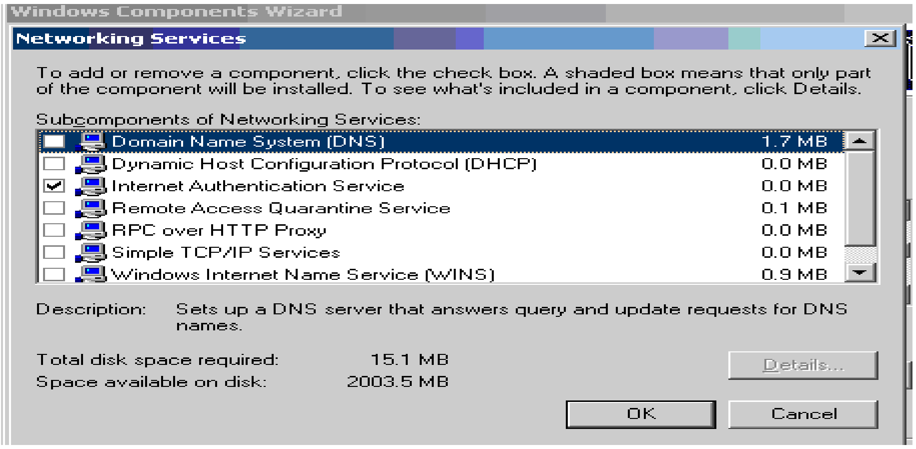
Remote Access Permission (Dial-in or VPN)

● Allow access
○ Deny access
○ Control access through Remote Access Policy

Figure 4.53: Configuring the remote permissions for the wireless user, the RTCCDOM administrator and the wireless client

Step 3: Setting up the Internet Authentication Server (IAS)

The Internet Authentication Service (IAS) shown ticked in Figure 4.54 was installed on the IAS computer in Figure 4.2. The IAS server is a computer running Windows 2003 enterprise server operating system. The IAS server was used in this experiment to authenticate wireless clients before they are allowed to connect to the wireless part of the RTCC network. The wireless clients in this project refer to vehicle speed sensors which are linked wirelessly to the RTCC network.



Windows Components Wizard

Networking Services

To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Networking Services:

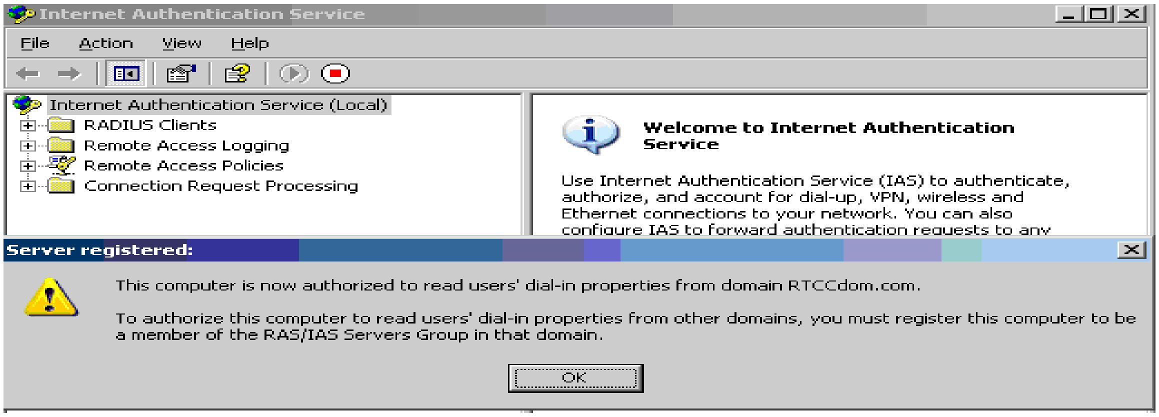
□	Domain Name System (DNS)	1.7 MB
□	Dynamic Host Configuration Protocol (DHCP)	0.0 MB
☑	Internet Authentication Service	0.0 MB
□	Remote Access Quarantine Service	0.1 MB
□	RPC over HTTP Proxy	0.0 MB
□	Simple TCP/IP Services	0.0 MB
□	Windows Internet Name Service (WINS)	0.9 MB

Description: Sets up a DNS server that answers query and update requests for DNS names.

Total disk space required: 15.1 MB
Space available on disk: 2003.5 MB

Figure 4.54: Installing the Internet Authentication Service inside IAS computer

After the IAS was installed, a computer (IAS in Figure 4.2) running IAS was registered in the Active directory and the dial-in permissions were granted. Thus the IAS computer could query the user and computer accounts database in active directory during wireless user and computer accounts validation. With dial-permissions granted to the IAS server, the IAS server could further use the RTCCDOM domain controller to compare the user and computer permissions in the domain database which are used to log into the RTCC network using a wireless medium. Only user and computer accounts with dial-in permissions are allowed access. The dial-in permissions in this project are used when vehicle speed sensors request access to the wireless network. This process also adds the IAS computer to the domain pre-existing Remote Access Servers (RAS) / IAS server's security group as shown in Figure 4.55.


Internet Authentication Service
File
Action
View
Help
Internet Authentication Service (Local)
RADIUS Clients
Remote Access Logging
Remote Access Policies
Connection Request Processing
Welcome to Internet Authentication Service
Server registered:
OK

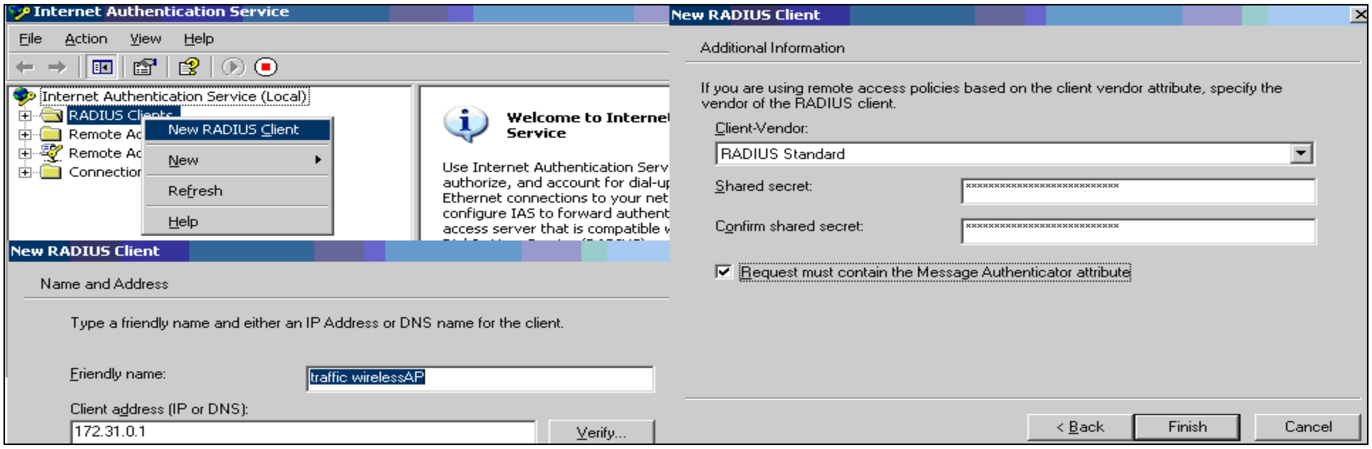
Use Internet Authentication Service (IAS) to authenticate, authorize, and account for dial-up, VPN, wireless and Ethernet connections to your network. You can also configure IAS to forward authentication requests to any

This computer is now authorized to read users' dial-in properties from domain RTCCdom.com.

To authorize this computer to read users' dial-in properties from other domains, you must register this computer to be a member of the RAS/IAS Servers Group in that domain.

Figure 4.55: Configuring the dial-in permissions for the IAS

While configuring the IAS properties inside the IAS server the wireless router (wireless access point) in Figure 4.2 was registered as a RADIUS client as shown by the client IP address 172.31.0.1 in Figure 4.56. A RADIUS client (WAP) in this project was used to forward wireless clients' (speed sensors) requests before wireless clients are allowed to connect to the wireless network to the RADIUS server (IAS server) for authentication purposes. With the dial-in permissions already granted to the IAS server in Figure 4.55, the IAS (RADIUS) server could read and compare permissions of wireless login user and computer accounts against permissions in the RTCCDOM domain controller database before a wireless connection is established.


Internet Authentication Service
File Action View Help
Internet Authentication Service (Local)
RADIUS Clients
Remote Ac
Remote Ac
Connection
New RADIUS Client
New
Refresh
Help
Welcome to Internet Authentication Service
Use Internet Authentication Serv authorize, and account for dial-up Ethernet connections to your net configure IAS to forward authent access server that is compatible w
< Back
Finish
Cancel
Verify...

New RADIUS Client

Additional Information

If you are using remote access policies based on the client vendor attribute, specify the vendor of the RADIUS client.

Client-Vendor: RADIUS Standard

Shared secret: ***************************

Confirm shared secret: ***************************

☑ Request must contain the Message Authenticator attribute

New RADIUS Client

Name and Address

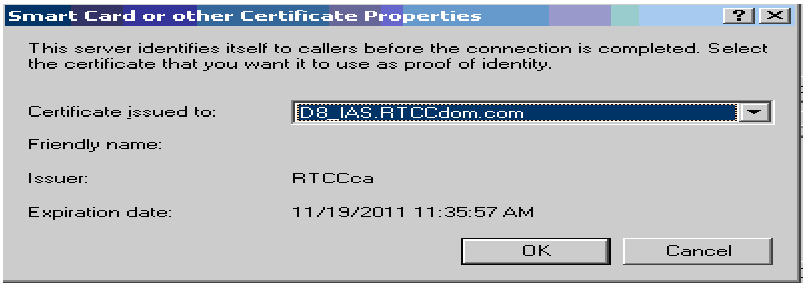
Type a friendly name and either an IP Address or DNS name for the client.

Friendly name: traffic wirelessAP

Client address (IP or DNS): 172.31.0.1

Figure 4.56: Specifying the RADIUS client to the IAS

The remote access policy was also created within the IAS computer. This policy is used together with RTCCwirelessGROUP to filter allowed wireless users (credentials required by the vehicle speed sensors before connection to wireless network is granted) during Protected Extensible Authentication Protocol (PEAP) authentication process. In this study PEAP was used to create a secure encrypted channel during wireless client user account and computer account validation. The wireless clients who are members of the RTCCwirelessGROUP and have valid computer certificates were allowed to connect to the RTCC network using a secured wireless network. The IAS server in the RTCC network also identifies itself to the wireless clients using a certificate shown in Figure 4.57. This certificate is validated by the wireless clients in order to ensure that they are not communicating with a spoofed IAS server.



This server identifies itself to callers before the connection is completed. Select the certificate that you want it to use as proof of identity.

Certificate issued to: D8_IAS.RTCCdom.com

Friendly name:

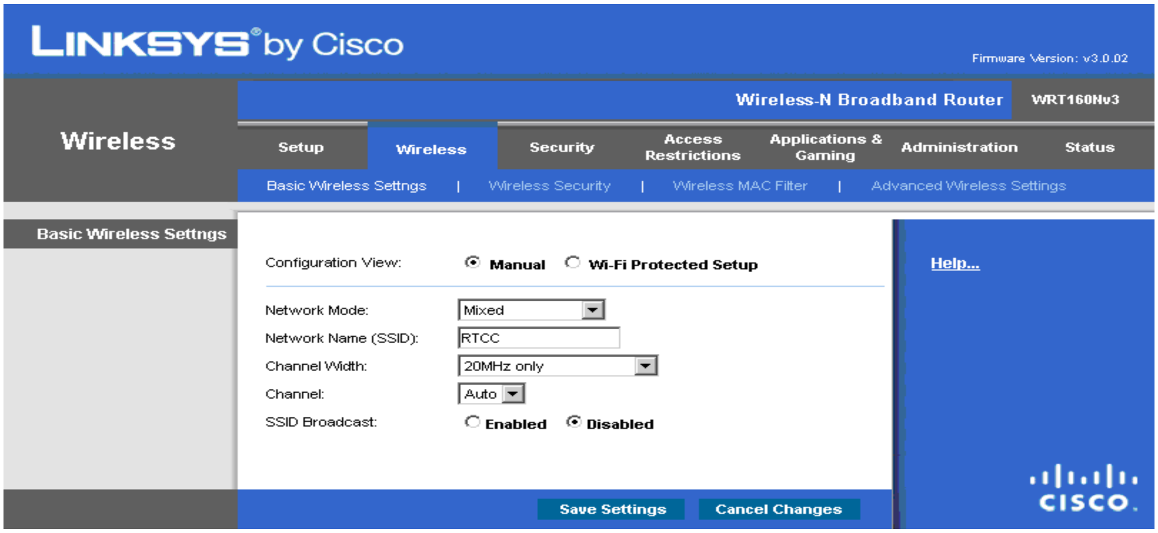
Issuer: RTCCca

Expiration date: 11/19/2011 11:35:57 AM

Figure 4.57: IAS servers' authentication certificate

Step 4: Configuration of the RADIUS client (Wireless Access Point (WAP))

In Figure 4.4 the RTCC wireless network was created. By default the RTCC network name broadcast is enabled as shown in Figure 4.4. The broadcasting of the RTCC network name allowed other wireless users to view the RTCC network name as one of the available wireless networks within the wireless spectrum. As a result the RTCC wireless network could become a target to unauthorised access. In order to hide the RTCC wireless network name in Figure 4.58 the Service Set Identification (SSID) broadcast was disabled.



Configuration View: ● Manual ○ Wi-Fi Protected Setup

Network Mode: Mixed

Network Name (SSID): RTCC

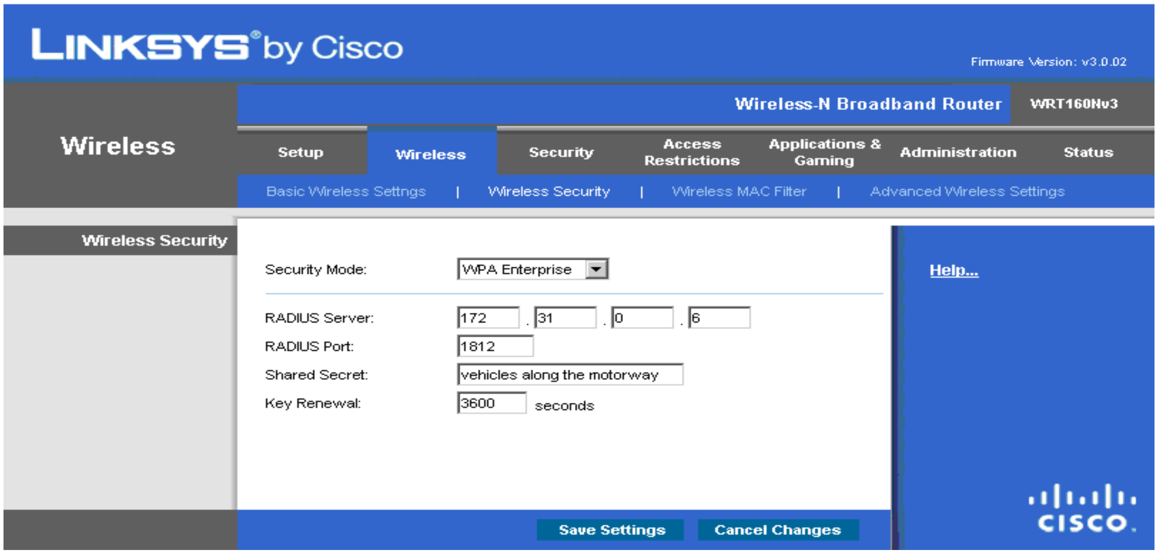
Channel Width: 20MHz only

Channel: Auto

SSID Broadcast: ○ Enabled ● Disabled

Figure 4.58: Disabling of the RTCC SSID broadcast

The Wifi Protected Access (WPA) security mode was used in this experiment as shown in Figure 4.59. With WPA a new key is generated each time a wireless client re-establishes connection with the wireless access point unlike using WEP which makes use of the same key for every connection. Using the same key repeatedly makes it easy for a hacker to use available free software to figure out a key being used to connect to a wireless network.



Security Mode: WPA Enterprise

RADIUS Server: 172 . 31 . 0 . 6

RADIUS Port: 1812

Shared Secret: vehicles along the motorway

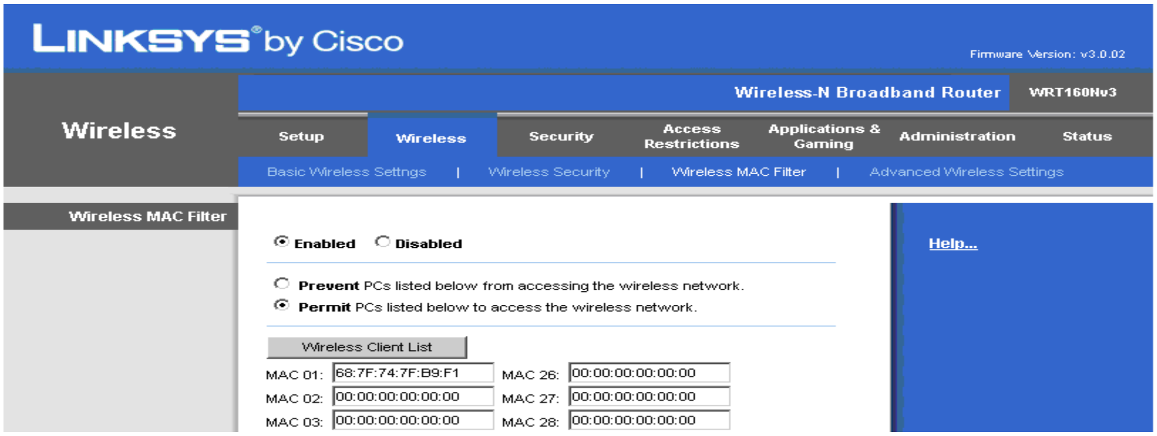
Key Renewal: 3600 seconds

Figure 4.59: Setting up of the RTCC wireless security mode

While setting up the IAS server in Figure 4.56, the wireless router (WAP) was registered as a RADIUS client. In Figure 4.59 a RADIUS server was also indicated to the WAP so that it knows where to forward all wireless connection requests for validation purposes. In Figure 4.59 the RADIUS server which is the IAS server is indicated by the IP address 172.31.0.6 thus any wireless connection requests were forwarded to the IAS server. The 1812 port shown in Figure 4.59 was used to forward validation information received from each wireless client (wireless speed sensor on the highway) to the RADIUS server before access to the wireless network is granted. This port was only used to send computer certificates, usernames and passwords between the RADIUS client and the RADIUS server. The IAS servers would then validate the information regarding each wireless connection request against the created remote access policy.

A shared secret passphrase set in Figure 4.56 and Figure 4.59 is used by the RADIUS server (IAS) to authenticate the RADIUS client (WAP) thus ensuring that the wireless connection requests are received from legitimate RADIUS clients. In other words, the RADIUS client identifies itself to the RADIUS server using the shared secret passphrase. Each wireless connection request when forwarded by the RADIUS client is wrapped with this shared secret passphrase and thus ensuring that the RADIUS server is communicating with the correct RADIUS client. The RADIUS server (IAS) acts as a firewall (permit or deny access) between the vehicle speed sensors installed on the highway and the storage database, RTCC-database server.

On top of allowing access to the wireless network based on remote access policy, a wireless MAC filter was enabled inside the wireless router (WAP) in Figure 4.60. MAC filtering was used for specifying the MAC addresses of wireless computers (vehicle speed sensors) to be permitted to connect to the wireless network. Only wireless computers of which their MAC addresses appear inside the list of permitted MAC addresses were allowed to connect to the wireless network. The wireless computer with a MAC address which corresponds to the one shown in Figure 4.60 was allowed to connect. These MAC addresses are for vehicle speed sensors installed on the highway. These sensors reside on the wireless part of the RTCC network and thus they need to be allowed access to the wired part of RTCC network where the vehicle speed data will be sent.


LINKSYS® by Cisco
Firmware Version: v3.0.02
Wireless-N Broadband Router
WRT160Nv3
Wireless
Setup
Wireless
Security
Access Restrictions
Applications & Gaming
Administration
Status
Basic Wireless Settngs | Wireless Security | Wireless MAC Filter | Advanced Wireless Settings
Wireless MAC Filter
Help...

● Enabled ○ Disabled

○ **Prevent** PCs listed below from accessing the wireless network.
● **Permit** PCs listed below to access the wireless network.

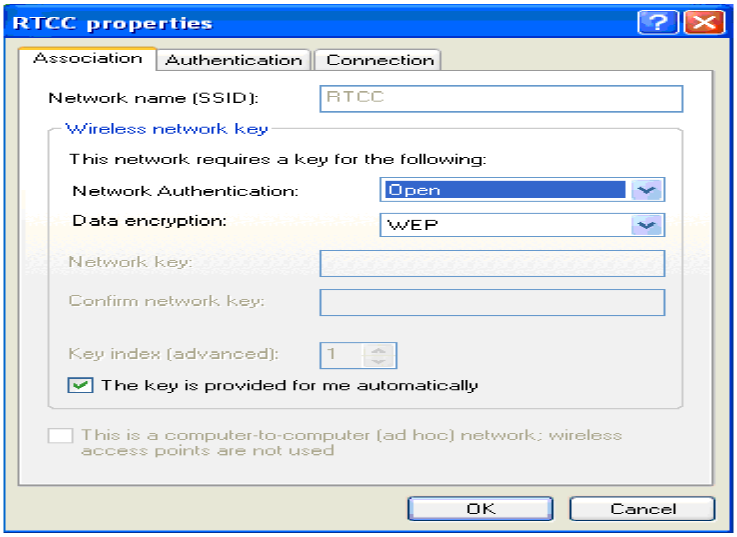
Wireless Client List

MAC 01:	68:7F:74:7F:B9:F1	MAC 26:	00:00:00:00:00:00
MAC 02:	00:00:00:00:00:00	MAC 27:	00:00:00:00:00:00
MAC 03:	00:00:00:00:00:00	MAC 28:	00:00:00:00:00:00

Figure 4.60: Configuring MAC address filtering for the RTCC wireless network

Step 5: Connecting a wireless client to RTCC wireless network

The wireless network name RTCC created in Figure 4.4 was manually added as the preferred wireless network for the wireless client (vehicle speed sensors). The open network authentication using WEP as a data encryption protocol was used as shown in Figure 4.61. Using open authentication allowed each wireless client to have access to connect and communicate with the RADIUS client (WAP) so that login credentials (username, password, computer certificate and user certificate) could be forwarded to the RADIUS server (IAS) before connection to the RTCC network is granted. RADIUS client also checks the MAC address database to validate the MAC address of each wireless client before login credentials are submitted to the RADIUS server. The WEP was used to encrypt any information which includes vehicle speeds to be sent through a wireless channel. The tick inside a check box “The key is provided for me automatically” in Figure 4.61 refers to the shared secret passphrase setup in Figure 4.56 inside the IAS server and Figure 4.59 inside the RADIUS client (WAP). The wireless client does not need to supply this shared secret passphrase but the RADIUS client (WAP) will supply the shared secret passphrase automatically when forwarding wireless connection requests with the required logging credentials to the RADIUS server. The RADIUS client will wrap each wireless connection request with a shared secret passphrase on behalf of the wireless client on sending each wireless client request to the RADIUS server (IAS).


RTCC properties

Association | Authentication | Connection

Network name (SSID): RTCC

Wireless network key

This network requires a key for the following:

Network Authentication: Open

Data encryption: WEP

Network key:

Confirm network key:

Key index (advanced): 1

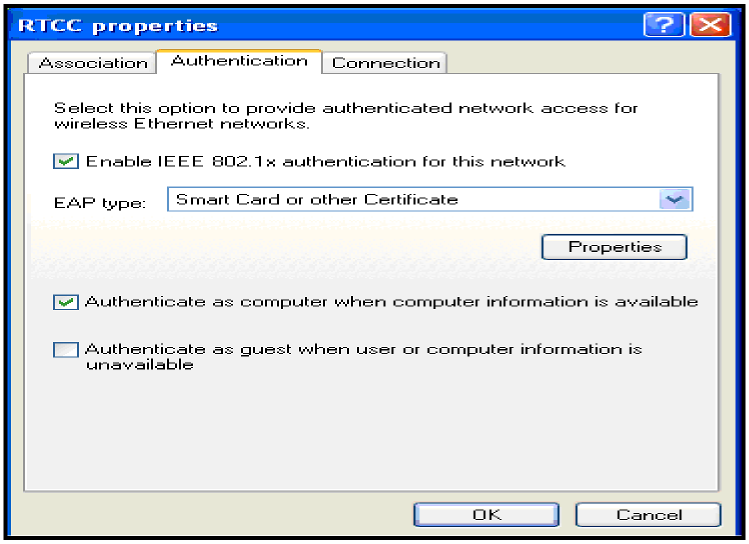
☑ The key is provided for me automatically

□ This is a computer-to-computer (ad hoc) network; wireless access points are not used

OK | Cancel

Figure 4.61: Wireless client association settings

In Figure 4.62 the wireless client was configured to prove the identity using EAP IEEE 802.1X authentication to the IAS server. The IEEE 802.1X protocol use digital certificates as a method of proving identity.


RTCC properties

Association | Authentication | Connection

Select this option to provide authenticated network access for wireless Ethernet networks.

☑ Enable IEEE 802.1x authentication for this network

EAP type: Smart Card or other Certificate

Properties

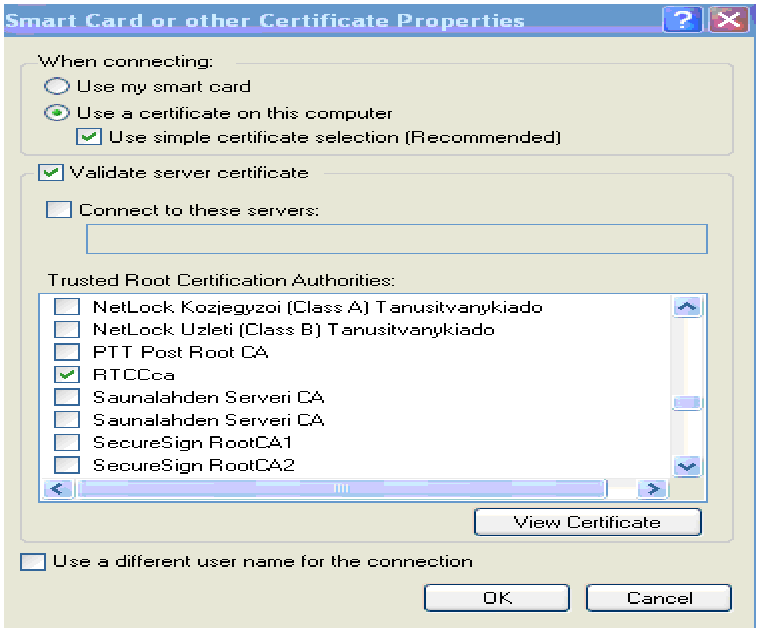
☑ Authenticate as computer when computer information is available

□ Authenticate as guest when user or computer information is unavailable

OK | Cancel

Figure 4.62: Specifying a wireless client authentication type

In this experiment the wireless client was not the only one to be validated by the RADIUS server before connecting to the RTCC wireless network but the RADIUS server (IAS) was also validated by the wireless client as indicated by a tick inside validate server certificate checkbox. During the authentication process the RADIUS server forwards its own computer certificate shown in Figure 4.57 to the wireless client to validate. The root CA RTCCca was used to perform this validation as shown in Figure 4.63. By validating the RADIUS server (IAS) the wireless client ensures that it is not communicating with the spoofed IAS server.



Smart Card or other Certificate Properties

When connecting:

○ Use my smart card

● Use a certificate on this computer

☑ Use simple certificate selection (Recommended)

☑ Validate server certificate

□ Connect to these servers:

Trusted Root Certification Authorities:

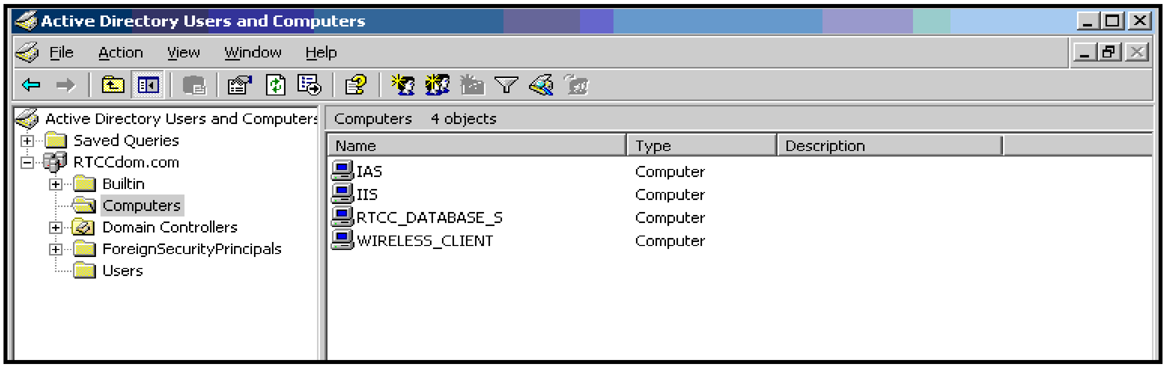
□ NetLock Kozjegyzoi (Class A) Tanusitvanykiado
□ NetLock Uzleti (Class B) Tanusitvanykiado
□ PTT Post Root CA
☑ RTCCca
□ Saunalahden Serveri CA
□ Saunalahden Serveri CA
□ SecureSign RootCA1
□ SecureSign RootCA2

□ Use a different user name for the connection

Figure 4.63: Specifying the trusted root certification authority RTCCca

4.8.1 Results

Figure 4.64 shows the active directory database showing computers which joined RTCCDOM domain. The IAS server, IIS server, wireless client and the RTCC-database server became members of RTCCDOM domain as shown in Figure 4.51.



Active Directory Users and Computers

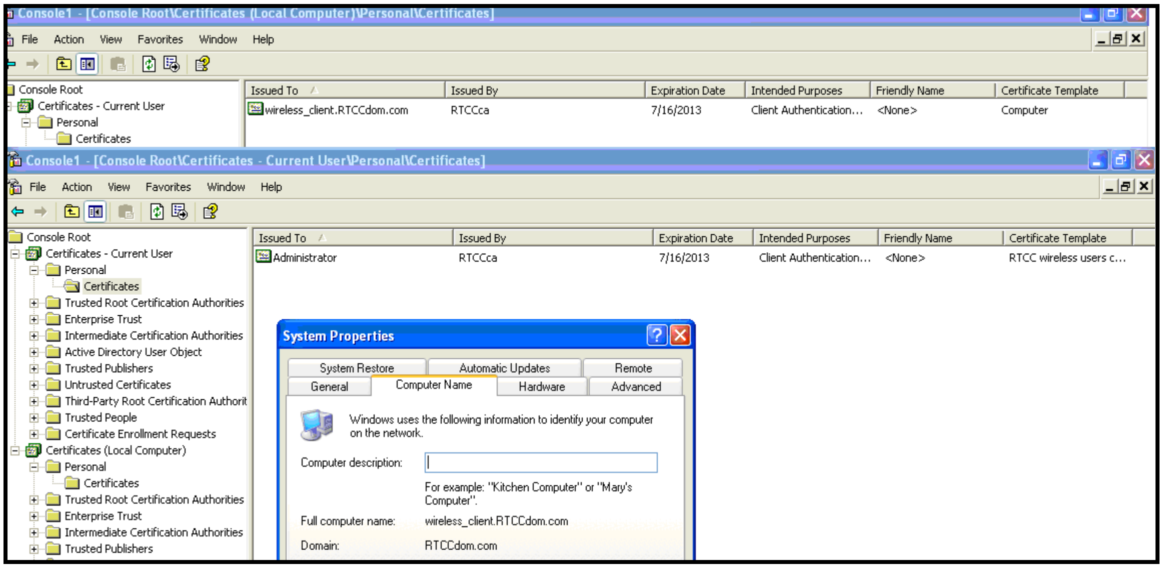
Active Directory Users and Computers
- Saved Queries
- RTCCdom.com
  - Builtin
  - Computers
  - Domain Controllers
  - ForeignSecurityPrincipals
  - Users

Computers 4 objects

Name	Type	Description
IAS	Computer	
IIS	Computer	
RTCC_DATABASE_S	Computer	
WIRELESS_CLIENT	Computer	

Figure 4.64: Registered computer accounts by the RTCCDOM.COM domain controller

On joining the RTCCDOM domain each computer was allocated a computer and user digital certificate by the root certification authority RTCCca to be used to authenticate them when logging into the domain (Figure 4.65).



Console1 - [Console Root\Certificates (Local Computer)\Personal\Certificates]

File Action View Favorites Window Help

Console Root
Certificates - Current User
Personal
Certificates

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Certificate Template
wireless_client.RTCCdom.com	RTCCca	7/16/2013	Client Authentication...	<None>	Computer

Console1 - [Console Root\Certificates - Current User\Personal\Certificates]

File Action View Favorites Window Help

Console Root
Certificates - Current User
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authorit
Trusted People
Certificate Enrollment Requests
Certificates (Local Computer)
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Certificate Template
Administrator	RTCCca	7/16/2013	Client Authentication...	<None>	RTCC wireless users c...

System Properties

System Restore | Automatic Updates | Remote
General | Computer Name | Hardware | Advanced

Windows uses the following information to identify your computer on the network.

Computer description:

For example: "Kitchen Computer" or "Mary's Computer".

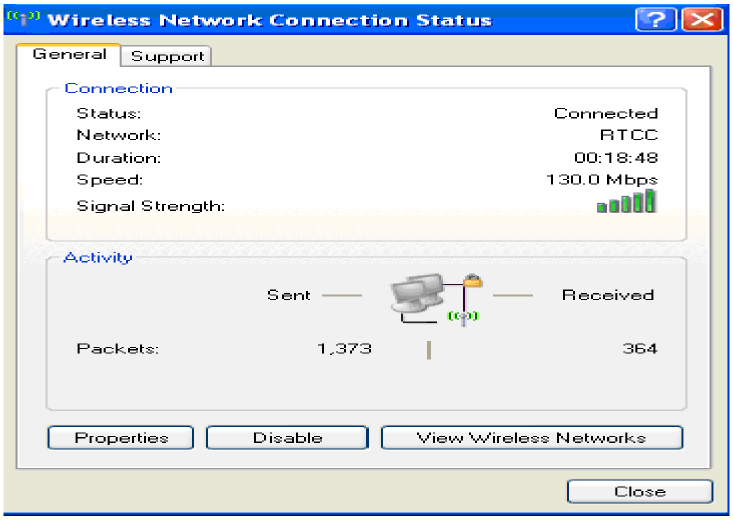
Full computer name: wireless_client.RTCCdom.com

Domain: RTCCdom.com

Figure 4.65: Wireless client certificates

The same procedure as in Figure 4.65 was repeated for securing DNS, IAS, IIS and RTCC-database servers.

In Figure 4.66 the wireless client could connect to the secured wireless network as shown by the wireless network name (Figure 4.4) and connection status.



Wireless Network Connection Status

General | Support

Connection

Status: Connected
Network: RTCC
Duration: 00:18:48
Speed: 130.0 Mbps
Signal Strength:

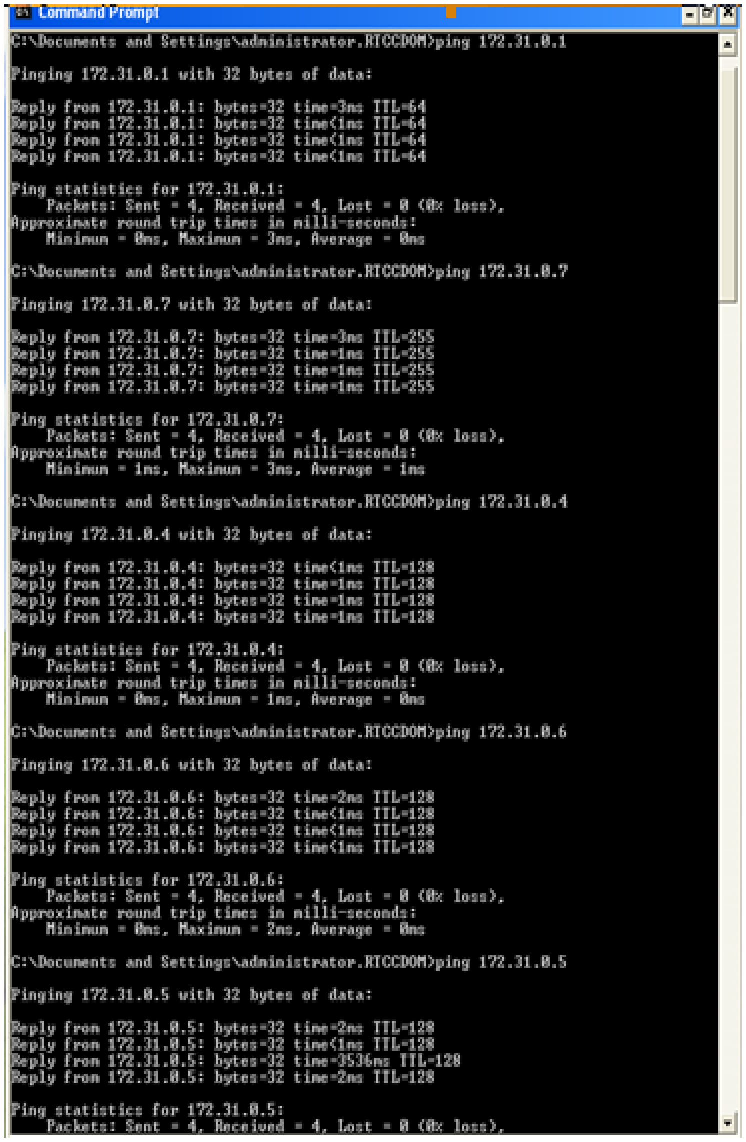
Activity

Sent — — Received

Packets: 1,373 | 364

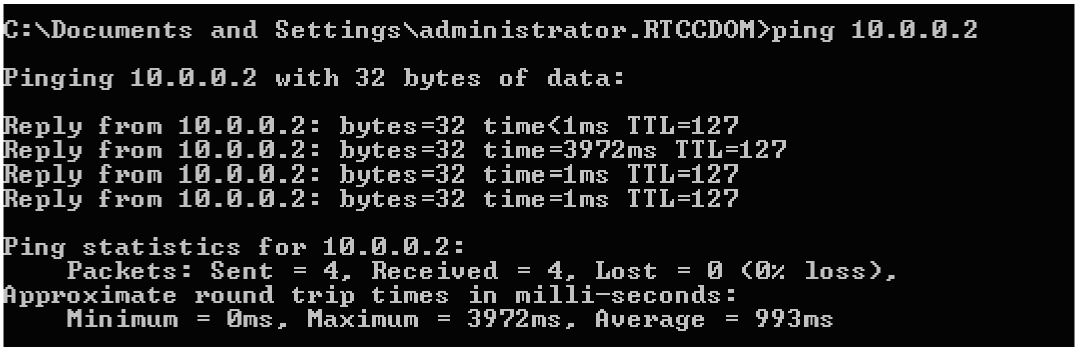
Figure 4.66: Status of wireless client connection

The connectivity test results showing end to end connectivity between the wireless client and the WAP, the router (default gateway), DNS, IAS and IIS are shown in Figure 4.67 and 4.68. Figure 4.67 further shows that once the wireless client was allowed access to connect to the wireless network (Figure 4.66) it could communicate with computers in the entire RTCC network. The 1st ping command in Figure 4.67 test connectivity between the wireless client and the RADIUS client (WAP). The 2nd ping command test connection to the router FA 0/1 port which is the default gateway of a wireless client computer. The 3rd ping command test connection to the DNS server. The 4th ping command test connection to the IAS server. The last ping command in Figure 4.67 test connection between the wireless client and the IIS server. Connectivity test to the RTCC-database server on a wired part of the RTCC network was also done as shown in Figure 4.68. All the tests were successful as indicated by the 4 messages received in each ping test and the vehicle speed data could be sent through a secured channel from the wireless speed sensors on the highway to the RTCC-database server.



```
C:\Documents and Settings\administrator.RTCCDOM>ping 172.31.0.1

Pinging 172.31.0.1 with 32 bytes of data:

Reply from 172.31.0.1: bytes=32 time=3ms TTL=64
Reply from 172.31.0.1: bytes=32 time<1ms TTL=64
Reply from 172.31.0.1: bytes=32 time<1ms TTL=64
Reply from 172.31.0.1: bytes=32 time<1ms TTL=64

Ping statistics for 172.31.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

C:\Documents and Settings\administrator.RTCCDOM>ping 172.31.0.7

Pinging 172.31.0.7 with 32 bytes of data:

Reply from 172.31.0.7: bytes=32 time=3ms TTL=255
Reply from 172.31.0.7: bytes=32 time=1ms TTL=255
Reply from 172.31.0.7: bytes=32 time=1ms TTL=255
Reply from 172.31.0.7: bytes=32 time=1ms TTL=255

Ping statistics for 172.31.0.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms

C:\Documents and Settings\administrator.RTCCDOM>ping 172.31.0.4

Pinging 172.31.0.4 with 32 bytes of data:

Reply from 172.31.0.4: bytes=32 time<1ms TTL=128
Reply from 172.31.0.4: bytes=32 time=1ms TTL=128
Reply from 172.31.0.4: bytes=32 time=1ms TTL=128
Reply from 172.31.0.4: bytes=32 time=1ms TTL=128

Ping statistics for 172.31.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Documents and Settings\administrator.RTCCDOM>ping 172.31.0.6

Pinging 172.31.0.6 with 32 bytes of data:

Reply from 172.31.0.6: bytes=32 time=2ms TTL=128
Reply from 172.31.0.6: bytes=32 time<1ms TTL=128
Reply from 172.31.0.6: bytes=32 time<1ms TTL=128
Reply from 172.31.0.6: bytes=32 time<1ms TTL=128

Ping statistics for 172.31.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Documents and Settings\administrator.RTCCDOM>ping 172.31.0.5

Pinging 172.31.0.5 with 32 bytes of data:

Reply from 172.31.0.5: bytes=32 time=2ms TTL=128
Reply from 172.31.0.5: bytes=32 time<1ms TTL=128
Reply from 172.31.0.5: bytes=32 time=3536ms TTL=128
Reply from 172.31.0.5: bytes=32 time=2ms TTL=128

Ping statistics for 172.31.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Figure 4.67: Wireless client connectivity tests to the IAS, IIS, DNS and the default gateways of the router



```
C:\Documents and Settings\administrator.RTCCDOM>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=127
Reply from 10.0.0.2: bytes=32 time=3972ms TTL=127
Reply from 10.0.0.2: bytes=32 time=1ms TTL=127
Reply from 10.0.0.2: bytes=32 time=1ms TTL=127

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3972ms, Average = 993ms
```

Figure 4.68: Wireless connectivity test to the RTCC-database server

The results in this chapter indicate that before the vehicle speed sensors are installed on the highway to capture vehicle speeds they need to join the RTCCDOM domain and become members of the domain database. During the process of joining, vehicle speed sensors are issued with computer certificates. User accounts to be used by vehicle speed sensors to connect to the RTCCDOM domain network need to be created inside the active directory. These user accounts (with user certificate) are used during vehicle speed sensors authentication process together with the computer digital certificate. When the validation process is completed successfully the vehicle speed sensors are allowed to connect to the secured RTCC network using a wireless connection. Java sockets are then used to send vehicle speed data over a secured wireless connection to the RTCC-database server. The multilayer perceptron retrieves vehicle traffic data inside RTCC-database and forecasts traffic status. The results of prediction are then sent to the Proteus VSM over the already secured wireless connection to be displayed on the highway.

4.9 Chapter summary

In this chapter Matlab random generator function randperm was used to generate random vehicle speed on the simulated highway. The generated speed was sent to the RTCC-database server using java sockets. RTCC-database used MySQL to store the vehicle speed data received from speed sensors installed on the highway. Based on stored vehicle speed data inside a MySQL database, a multilayer perceptron network (MLP) included within WEKA workbench was used to predict traffic status in the next 5 minutes. CONGESTED traffic condition was predicted with the MLP model showing 100 percent accuracy. The predicted traffic condition was displayed on the highway for the motorists to read using Proteus Virtual System Modelling software. The entire RTCC network was secured using PEAP with EAP-TLS. In PEAP with EAP-TLS digital certificates were used during authentication by the IAS (RADIUS server) to validate domain user and computer accounts. PEAP with EAP-TLS employ the use of RADIUS clients during authentication of wireless speed sensors before they could gain access to the wired part of the

RTCC network. The RTCC-database server resides in a wired part where vehicle speed needs to be sent for storage inside the MySQL server. A RADIUS client (wireless router) was used to receive and forward wireless speed sensors request to connect to the RTCC network using a wireless link. These requests (which include username, password, computer certificate and user certificate) were forwarded to the IAS (RADIUS server) for validation purposes. The IAS server before granting access to wireless speed sensors, checks with other servers (DNS and CA) during which validation information was exchanged. The remote access policy within IAS server was also checked. The RADIUS client (wireless router) also checks the MAC filtering database for all permitted MAC addresses of wireless clients which could connect to the RTCC network using a wireless medium. When the validation process was complete the wireless client (speed sensors) could communicate with any computer within the secured RTCC network.

CHAPTER 5: DISCUSSION OF RESULTS

5.1 Introduction

In this chapter results of main experiments carried out in chapter 4 will be discussed.

5.2 Interpretation and analysis of the results

In experiment 5 the multilayer perceptron (MLP) was used to learn about a pattern in the data, thereafter forecast traffic status in the following 5 minutes. Results of testing the proposed MLP model using the novel vehicle traffic data are shown in Table 4.1 through Table 4.4. Several tables were built from varying the number of inputs (features) whilst having the hidden units remain constant. On the other hand, the inputs remained constant and the numbers of hidden units were varied. The results shown in Table 4.1 through Table 4.4 were the best models chosen through intuition obtained from the whole exercise.

Table 4.5 shows that all 76 instances were predicted correctly during the testing phase of the model. This resulted in 100 percent accuracy. The correlation co-efficient (Kappa statistic) of 1 was achieved by the best MLP model thus showing strong statistical relations between the instances and instance classes. The mean absolute and root mean square errors shown in Table 4.5 measure how close the predictions are to the eventual outcomes. The best MLP model shown as the 5th architecture in Table 4.1 achieved a low root mean square error (RMSE) of 10 percent (0.01) during training and 18 (0.018) percent during testing as compared to any other competing models in Table 4.1 through 4.4. This low RMSE was due to the proper learning which was acquired during training thus resulting in 100 prediction accuracy by the model.

This degree of knowledge is shown in Table 4.6. Inside the actual column are the actual instances to be predicted. Predicted instances are shown inside the predicted column. When averaging the probabilities of all instances with asterisks inside each predicted column (from left to right), into congested have 97 percent (1st column), congested have 99 percent (2nd column), out congested 97 percent (3rd column) and normal traffic 95 percent (4th column). The predicted traffic status as a class together with the probability associated with instances with high probabilities occupies the 2nd column. These instances in the 2nd column showed more confidence in predicting the true class (congested highway traffic) as shown by the margin curve in Figure 4.39. These are instances in Figure 4.39 which are lying very close to a margin value of 0.98 and they are showing more confidence in predicting the next most likely class (congested highway traffic).

What makes these results interesting is the exclusion of the influence of the weekend and also the bank holidays. Furthermore the actual vehicle speeds received from the highway on arrival were divided into discreticised values (normal, below thresh, above thresh, very slow) as shown within speed_parameter in Figure 4.24, and predicting the outcome of such discreticized values also shown within current_status columns in Figure 4.24. Using binominal values makes this model unique in that a range of speed that generalises a pattern on the highway is predicted instead of one fixed number. As a result the model accuracy is improved as seen by 100 percent accuracy during training and testing. Wang *et al.* (2005: 215) and Xu *et al.* (2009:1007) models came very close to the proposed model in this work by achieving 98 percent accuracy respectively during novel instance prediction. Wang *et al.* (2005: 215) used back-propagation neural network optimised using genetic algorithm to predict the changing trend of the future while Xu *et al.* (2009:1007) used a radial basis function (RBF) neural network and wavelet analysis in speed forecasting of Beijing urban freeways.

According to Xu *et al*. (2009:1007) radial basis function model in predicting free flow, transitional and congestion seems to work better in predicting congestion as shown in Xu *et al.* (2009:1007) as compared to predicting free flow and transition traffic status. Xu *et al.* (2009:1007) outline this confusion by showing identical traffic patterns predicted for both free flow and transition periods. In order to eliminate this confusion within the current study the transition period was divided into two parts namely: into congested and out congested. In this way it is easy for the proposed MLP model to differentiate between transitions periods which will result in congestion on the highway from transitions periods which will result in returning the vehicle traffic back to free flow.

Mao *et al*. (2011:274) optimized gray model and Wang *et al*. (2005: 215) combination of neural networks with genetic model both use traffic volumes obtained by counting the number of vehicles on the highway during specific times in a day to do predictions. Predicting traffic status using traffic volumes results in more prediction errors compared to using discreticised vehicle speeds thus affecting the algorithm accuracy in predicting the true class. Mao *et al.* (2011:274) predicted wrong traffic volumes as shown by the difference between real data and simulated data thus resulting in an average relative error of 70 percent (0.669). It is only after optimisation that the error rate decreased to 2 percent from 70 percent. Furthermore, an algorithm used by Wang *et al.* (2005:215) is also not good in predicting using real numbers, this is shown by the difference between predicted traffic volumes compared to expected traffic volumes. Expected traffic volumes of 37411, 34944, 32197 and 37729 were each predicted to be 38111, 34005, 31680 and 38108 respectively using a back-propagation (BP) training of 80 000 epochs thus obtaining 0.019 prediction errors. In trying to improve the accuracy of the prediction results Wang *et al.* (2005:215) used an advanced BP genetic algorithm trained using 50 generations. The advance BP-genetic algorithm resulted in a prediction error rate of 0.009 being achieved, which is much closer to expected traffic volume output in predicting 37 491, 34 961, 32 164 and 37 762 respectively. Thus the advance BP-genetic results show improvement even though not as 100 percent accurate as demonstrated by the proposed model within the current study during prediction of novel vehicle traffic data.

In using traffic volumes to predict future traffic status the models of Mao *et al*. (2011:274) and Wang *et al*. (2005:215) further create confusion in cases whereby there is high traffic volume but the vehicles are travelling at normal highway speed; a high traffic volume does not necessarily mean there is traffic congestion. Furthermore, in predicting that the highway traffic will have few vehicles does not mean there is a free flow on the highway since there might be a traffic bottleneck somewhere along the highway thus blocking the vehicle movements due to an accident or road works. In discreticising the vehicle speeds in this work, vehicle speeds are clustered according to the predefined ranges and during prediction the MLP model predicts a range of speed likely to happen on the highway also taking into consideration other factors besides vehicle speeds such as accidents and road works that influence the flow of traffic. Taking all this information into account the MLP model used in this work eliminates confusion, is decisive and shows strong statistical relation between attributes, instances and classes by predicting all instances correctly.

In experiment 6, the MLP traffic status prediction results are displayed on the LCD screen located on the simulated highway. Thus the motorists are alerted about the traffic condition for the coming 5 minutes. In Table 4.6 in the first column of probability distribution, 13 instances predicted into congestion (into congested) with a probability of 97 percent. In the second column 36 instances all predict that the highway will be congested with a probability of 99 percent. Inside the third column 15 instances predict that the highway will get out of congestion (out congested) with 97 percent probability. In the last column of probability distribution 12 instances predict normal highway traffic with 95 percent probability. Due to high probability a congested traffic status along the Ben Schoeman highway was predicted. As a result in Figure 4.42 compiler source code line 55 was executed by the PIC to be displayed by the LCD along Ben Schoeman highway as shown in Figure 4.44. Using the LCD to display the predicted traffic status provides a better way of continuous monitoring and control of the highway vehicle traffic. The continuous 5 minutes updates help in automatic diversion of traffic (controlling) to alternative routes during periods predicted to be congested since no motorist wants to remain in a route about to be congested. The use of alternative routes by the motorists before congestion periods further helps in highway traffic decongestion. The prediction models of Xu *et al*. (2009:1007), Wang *et al.* (2005:215) and Mao *et al.* (2011:274) did not incorporate a means whereby motorists are notified about the future traffic condition; instead Wang *et al.* (2005:211) use the results of prediction to deploy more policemen to the highway in anticipation that some accidents may be avoided.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

A secured intelligent system which addresses vehicle traffic congestion along the Ben Schoeman highway has been designed.

The proposed system uses a randperm matlab function to mimic magnetometer vehicle speed capturing sensors. This function generates random vehicle speeds which are typical of the real highway vehicle traffic behaviour. Java sockets are used to seamlessly relay generated vehicle speeds over a wireless medium to the central storage database (MySQL server that forms the RTCC shown in Figure 4.2). At the RTCC this data is processed and the recommended action taken.

In this project the multilayer perceptron (MLP) provides the intelligence part. At intervals of 5 minutes, the MLP collects the traffic data from MySQL database, analyses this data and based on the results of analysis predicts the traffic status for the next 5 minutes.

Billboards found along the highways can be used to display the 5 minute interval traffic status predicted by the MLP. This approach helps in the management of the highway traffic and thus alleviates vehicle traffic congestion. Put in other words, it promotes the free flow of traffic.

The entire RTCC network shown in Figure 4.2 is protected against unauthorised access using a combination of PEAP and EAP-TLS. This combination takes control during the authentication process. PEAP and EAP-TLS use digital certificates as an authentication method. Any computer or device that wants to connect to the RTCC network had to have its certificate validated by PEAP and EAP-TLS combination. Any user trying to log in through any of the computers that have been authenticated, must also have his certificate validated before he is granted access to the RTCC network resources. On top of using certificates, the wireless part of the network is further protected using a group policy and remote dial-in permissions validated by the IAS server and the DNS server respectively.

The implications of this work are the increase in productivity of workers and increased profits for companies. Consequently, this will make Gauteng an attractive destination for investment. The Logistics industry will benefit immensely as a result of free flow traffic.

6.2 Future work

Due to unavailability of funds actual speed capturing technologies were not used in this project instead a arbitrary function included within matlab library RANDPERM was used to generate random vehicle speeds. In future the use of actual speed capturing technologies will be looked at.

Incorporation of a system with recent road maps to be used to automatically calculate an alternative route for the motorists based on prediction, will also be considered in the future. This will be helpful to motorists who are not familiar with Johannesburg roads. In addition, these tourists or visitors will be made aware of the alternative routes they can use.

Bibliography:

AHMED, S.S., BAHAA, K.S. & MOHAMAD, M.E. 2010. “Intelligent Cross Roads Traffic Management System (ICTMS)”, *2nd* International *Conference on Computer Technology and Development* (ICCTD 2010).

ALAM, N., BALAIE, A.T. & DEMPSTER, A.G. 2011. “A DSC-based Traffic Flow Monitoring and Lane Detection System”. (IEEE 2011), Sydney, Australia.

AUNG, N.A., NEW, A.A., SOE, K.M., NAING, T.T. & THEIN, N.L. 2005. “Utilizing Multiple networks for Interprocess Communication in Cluster Computing”. (IEEE 2005).

BACHAN, P. & SINGH, B. 2010.”Perfomance Evaluation of Authentication Protocols for IEEE 802.11 Standard”.(International Conference on Computer & Communication Technology (ICCCT 2010).

BHORASKAR, R., VANKADHARA, N., RAMAN, B. & KULKARNI, P. 2012. “Wolverine: Traffic and Road Condition Estimation using Smartphone Sensors”. (IEEE 2012).

BOGGIA, G., CAMARDA, L., GRIECO, A. & ZACHEO, G. 2008. “Towards Wireless Networked Control System: an Experimental Study on Real-time Communications in 802.11 WLANs”. (IEEE 2008).

CHEN, Q., LAI, Y. & HAN, J. 2006. “An implementation for distributed backpropagation using corba architecture”. (IEEE 2006).

DEPARTMENT of Transport **see** REPUBLIC OF SOUTH AFRICA. Department of Transport.

GUILLERMO, L.T., TOURINO, J., DOALLO, R., LIN, Y. & HAN, J. 2009. “Efficient Java Communication Libraries over InfiniBand”. (2009 11th IEEE International Conference on High Performance Computing and Communications).

GUPTA, C.D., PATTANDER, S. & DE, D. 2012. “Performance Analysis of queuing models for Cooperative Content Distribution of Mobile Network”, (2012 Third International Conference on Computer and Communication Technology).

HAYKIN, S. 1994. NEURAL NETWORKS: A comprehensive foundation. Hamilton, Onterio, Canada: Macmillan College.

HOEPER, K. & CHEN, L. 2010.”An inconvenient Truth About Tunneled Authentications”. (35th Annual IEEE conference on Local Computer Networks, Denver, Colorado).

HU LI, D.L. & WEI, X. 2010. “Kalman Filtering-Based Cam-Shift Vehicle Tracking Algorithm for Highway Traffic Conditions”, (2010 International Conference on Computer Application and System modelling).

INGRAM, D., RESS, O. & NORMAN, A. 2006. “CORBA Transaction Through Firewalls”. (IEEE 2006).

KAITSA, K., STAVRAKAS, I., KONTOGIANNIS, T., DARADIMOS, I., PANAOUSIS, M. & TRIANTIS, D. 2007. “Load balancing incoming IP requests across a farm of clustered MySQL servers”. (IEEE 2007).

KOLAHI, S.S., QU, Z., SOORTY, B.K. & CHAND, N. 2009. “The Impact of security on the Perfomance of IPv4 and IPv6 Using 802.11n Wireless LAN”. (IEEE 2009).

LIN, O.C., KWEE, A.T. & TSAI, F.S. 2009. ”Database Optimization for Novelty Detection”. (ICICS 2009).

MAJSTOROVIC, N., SIRANOVIC, Z. & KAVRAN, K. 2012.”Tools for Grading students’ Exercises for Microsoft Access Applications”. MIPRO 2012, May 21-25, Opatija, Croatia.

MAN, T., WONG, S.C., XU, J.M., GUAN, Z.R. & ZHANG, P. 2009. “An Aggregation Approach to Short-Term Traffic Flow Prediction”. (IEEE 2009).

MAO, S., CHEN, Y & XIAO, X. 2011. “Short-term Traffic Flow Prediction Based on GM (1,1,Exp) model”. (IEEE 2011), China.

MINQIANG, C.A.I. 2012. “The Design Method of Network Chart System Based on Socket and Cloud Computing”. (2012 IEEE).

NGAMSURIYAROJ, S. & PORNPATTANA, R. 2010. “Performance Evaluation of TPC-H Queries on MySQL Cluster”, (2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops).

PRASHANT, S., PREETI, B. & AJITH, A. 2009. “Identification of Vehicle Class and Speed for Mixed Sensor Technology using Fuzzy-Neural & Genetic Algorithm: Design Approach”, *Proceedings of the International Workshop on Machine Intelligence Research* (2009 MIR Day) GHRCE, Nagpur, India.

REKHTER, Y. & LI, T. 1993.”An Architecture for IP Address Allocation with CIDR”, (RFC1518).

REPUBLIC OF SOUTH AFRICA. Gautrain. 2012. Annual report. Johannesburg: Government printer.

REPUBLIC OF SOUTH AFRICA. SANRAL. 2011. Annual report. Johannesburg: Government printer.

REHBORN, H. & PALMER, J. 2008. “ASDA/FOTO based on Kerner’s Three Phase Traffic Theory in North Rhine-Westphalia and its Intergration into vehicles”, (2008 IEEE Intelligent vehicles Symposium), Eindhoven, Netherlands. 4 – 6 June 2008.

REID, A. & LORENZ, J. 2008. Networking for Home and Small Businesses (*CCNA discovery learning guide)*. 2nd ed. United States of America: Cisco Press.

SAIGAL, S. & MEHROTRA, D. 2012. “Performance comparison of time series data using predictive data mining techniques”. *Advances in Information Mining*. 4(1):57 – 66.

SROKA, R. 2004. “Data Fusion based on Fuzzy Measures in Vehicle Classification Process”. (ICMT 2004), Como, Italy. 18 – 20 May 2004.

SHAHAMIRI, S.R., NASIR, W.M. & IBRAHIM, S. 2010. “An Automated Oracle Approach to Test Decision-making Structures”. (IEEE 2010).

SU, B. & WANG, L. 2010."Application of Proteus Virtual System Modelling (VSM) in Teaching of Microcontroller". (IEEE 2010).

THANH, V.T. & URANO, Y. 2010. "Mobile TCP socket for secure applications". (ICACT 2010).

TEKLI, J.M., DAMIANI, E., CHBEIR, R. & GIANINI, G. 2012. "SOAP Processing Perfomance and Enhancement", (IEEE TRANSACTIONS ON SERVICE COMPUTING, 5(3), July-September 2012).

VIBRA, L., VENKATESHA, M., PRASANTH, G.R., SUHAS, N., SHENOY, P.D., VENUGOPAL, K.R. & PATNAIK, L.M. 2008. " Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance", *Proceedings of the International Multiconference of Engineers and Computer Scientists.* 1. (IMECS 2008), Hong Kong, China.   19 – 21 March 2008.

WANG, Y., WANG, H. & XIA, L. 2005. "Highway Traffic Prediction with Neural Network and Genetic Algorithm". (IEEE 2005), Beijing, China.

Wikipedia. 2013. "MAISLOT". http://en.wikipedia.org/wiki/mailslot . accessed 29 March 2013.

WITTEN, I.H. & FRANK, E. 2005. Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed. United States of America.

XU, T., SUN, X., WU, Y. & XIE, C. 2009. "Artificial Neural Network and Wavelet Analysis Application in Speed Forecast of Beijing Urban Freeway". (IEEE 2009), P.R. China.

YASSIN, I.M., TAIB, M.N., ADNAN, R., KHAIRUL, M.S. & HAMZAH, M.K. 2012. "MySQL Database Lookup Table for Binary Particle Swarm Optimization-Based System Identification", (2012 IEEE Symposium on Industrial Electronics and Applications (ISIEA2012), September 23-26, Bandung, Indonesia).

YING, S., YANG, Y. & YING, S. 2008. " Study on Vehicle Navigation System with Real-time Traffic Information". (IEEE 2008).

ZHANG, F. 2012.”Mitigating Distributed Denial-of-Service Attacks: Application-Defense and Network-Defense Methods”. (IEEE 2012).

ZORATTI, I. 2006. MySQL Security Best Practices. Hetz , UK:IET.

Annexure A, Private subnets addresses used in the current study

In RFC 1918 the 32 bit IP version 4 IP addresses is divided into 5 classes; class A, B, C, D and E. Out of the 4 octets which make up an IP address, an IP address class is determined by the value inside the first octet. Each octet of an IP address is separated by a period. Class A IP addresses have a value between 1 to 127 inside the first octet. Class B IP addresses have a value between 128 to 191 inside the first octet. Class C IP Addresses have a value between 192 to 223 inside the first octet. Class D IP addresses have a value between 224 to 239 inside the first octet. Class E IP addresses have a value between 240 to 255 inside the first octet. Only class A, B and C IP addresses are assigned to hosts to communicate in commercial networks. Class D IP addresses are reserved for multicast purposes (sending network messages to hosts which belong to a particular group) and class E IP addresses are reserved for research and experiments. Within class A, B and C some of the IP addresses are private (obtained from Private network addresses) while some are public (obtained from Public network addresses). In class A IP addresses that start with a value 10 inside the first octet (network address 10.0.0.0) are private. In class B IP addresses that start with 172.16 to 172.31 (network addresses from 172.16.0.0 to 172.31.0.0) inside the first two octets are private. In class C IP addresses which start with 192.168.0 to 192.168.255 inside the first three octets (network addresses from 192.168.0.0 to 192.168.255.0) are private. In this project private network addresses (subnets) which start with 10 inside the first octet (network address 10.0.0.0) and a network address which starts with 172.31 inside the first two octets (network address 172.31.0.0) were used as can be seen in Figure 4.3.

Class E experimentation IP addresses could have been used but the devices used to build the entire road traffic control network (switches, routers and computers) only support the use of commercial IP addresses (IP addresses from class A, B or C network addresses). Any IP address which does not fall within the three identified private ranges (of class A: 10.0.0.0, class B: 172.16.0.0 to 172.31.0.0 and class C: 192.168.0.0 to 192.168.255.0) is public. Public IP addresses are allocated by the Internet Service Provider (ISP) to different devices which are used by different organisations to access the internet and as a result within the current study there was no need to use public IP addresses since connectivity to the internet was not required.

When assigning the IP addresses to the computers, even though class A and B subnets of 10.0.0.0 and 172.31.0.0 were used (Figure 4.3) a class C subnet mask of 255.255.255.0 was used. A subnet mask is assigned together with an IP address to a computer in order to identify a portion within an IP address which represents a network address and a portion which represents a host address. Computers belonging to the same subnet use the same numbers in a network portion of an IP address. Thus a used subnet mask is used to identified these network numbers in an IP address e.g. in Figure 4.3 two router interfaces Fa 0/0 and Fa 0/1 have been programmed to connect two different networks 10.0.0.0/24 and 172.31.0.0/24) respectively. Thus for both the interfaces a subnet mask 255.255.255.0 was used. A used subnet mask identifies a network portion as the first three octets for both subnets (10.0.0 and 172.31.0).

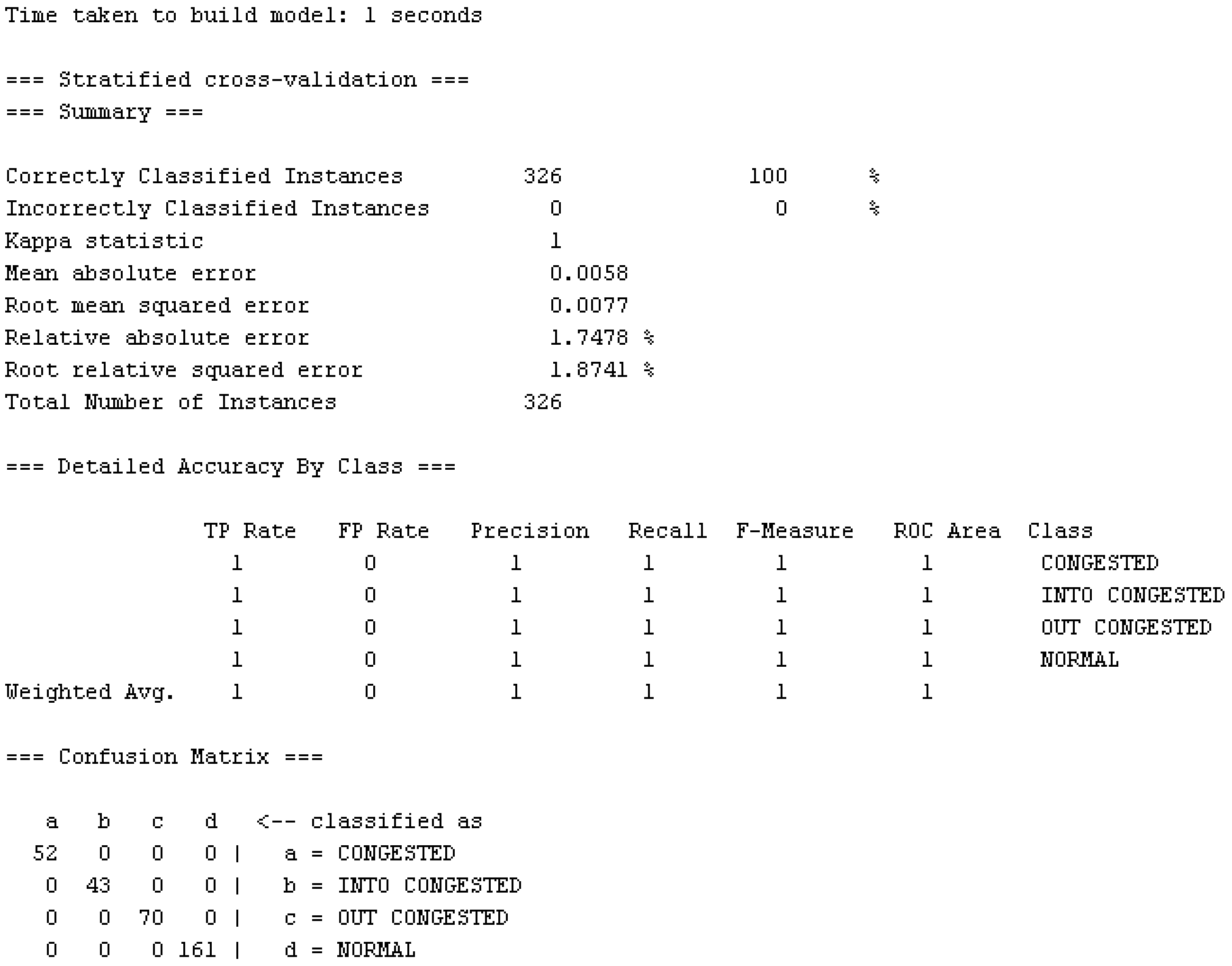
The octets with the 255's correspond to the octets within an IP address which represent a network i.e. the first three octets of the IP addresses assigned to Fa 0/0 (10.0.0.X) and Fa 0/1 (172.31.0.X) represent a network. All computers which belong to the same network as Fa 0/0 should all begin with values 10.0.0 when IP addresses are assigned. All computers which belong to the same network as Fa 0/1 should all start with values 172.31.0 when IP addresses are assigned. That is why all computers which are connected to Router0 Fa 0/1 (DNS and certification server, IIS, IAS and wireless client) including Router0 Fa 0/1, start with 172.31.0 within the first three octets of the IP address fields in Figure 4.3, 4.5 and 4.6. At the same time the RTCC-database server together with Router0 Fa 0/0 both start with 10.0.0 within the first three octets as shown in Figure 4.3, and Figure 4.8 A class C subnet mask was chosen to be used in this experiment due to a lower number of hosts it can accommodate in a single network. In order to obtain the number of hosts which can be accommodated, a default class C subnet mask of 255.255.255.0 was converted to a binary (i.e. representing each octet in binary) number. Since each octet is made up of eight binary bits, as a result a binary number 11111111.11111111.11111111.00000000 was obtained. With eight zeros on the last octet of a class C subnet mask, a formula $2^{x} - 2$ was used to calculate the number of hosts which can be accommodated by a class C mask, and 254 hosts was obtained after replacing an x with eight. Hosts refer to the number of IP addresses which correspond to the number of network devices to which IP addresses can be allocated. If all network devices are computers it means 254 computers in a network can each be given an IP address. Using a class C subnet mask for subnet 172.31.0.0 in the current project allows for connecting 246 wireless speed capturing sensors (excluding 6 IP addresses already assigned to the DNS and Certification server, IIS, IAS, switch3 vlan1, wireless router, Router0 fa 0/1) along the Ben schoeman highway to Pretoria. This number is sufficient since the current project is only concerned with monitoring traffic along the Ben Schoeman highway; not across all the highways in Gauteng. The same number of 254 hosts in a single network can also be accommodated by Router0 Fa 0/0 where RTCC-database server resides. Using 254 hosts is advantageous in that when there is a need for growth or expansion (upgrading) of the entire RTCC network unused IP addresses from this network segment (10.0.0.0) can be used to incorporate new devices into the network. A class A subnet mask of

255.0.0.0 (11111111.00000000.00000000.00000000) or class B subnet mask of 255.255.0.0 (11111111.11111111.00000000.00000000) could have been used if the entire project was intended to monitor traffic in all available routes in South Africa because more sensors would be required to be installed. In class A 26, 77721 hosts can be accommodated with class B accommodating 655534 hosts in each network.

In order to obtain a range from the first to the last IP address to be allocated to computers in network segment 172.31.0.0/24 subnetting rules were used since it is known that the first three octets 172.31.0.X (172.31.0.0) represent a network as identified by octets within a subnet mask with 255's (255.255.255.0). In order to calculate IP address ranges an octet with an X in 172.31.0.X was used. This octet corresponds to an octet with a zero (last octet) of the subnet mask (255.255.255.0). A subnet mask octet with a zero identifies an octet within an IP address that represents a host address. A host address is a number which uniquely identifies each host in a particular network. To calculate this host number so that it is unique for each host where an IP address is assigned, the last octet of 172.31.0.0 (172.31.0.X) was converted to a binary number resulting in 172.31.0.00000000.

In order to obtain the first valid IP address which can be assigned to the first host in network 172.31.0.0, the last binary bit of an octet with eight 0's was converted to a binary 1; thus resulting in 172.31.0.00000001 as the first IP address. With the rightmost bit in an octet having a value of 1 the values of the remaining bits, from right to left, are 2, 4, 8, 16, 32, 64 and 128. Converting the last octet 00000001 to decimal resulted in the decimal IP address 172.31.0.1 being the first IP address to be allocated to the first host in network 172.31.0.0/24. In order to determine the last IP address to be allocated to the last host in network 172.31.0.00000000, a subnetting rule further requires converting the host portion bits to ones (1's), except the last host bit (the opposite binary number to the one used when determining the first IP address) resulting in 172.31.0.11111110 (172.31.0.254 in decimal). As a result in this experiment the range of IP addresses used for hosts connected to Router0 Fa0/1 (172.31.0.0/24 subnet) start from 172.31.0.1 to 172.31.0.254.

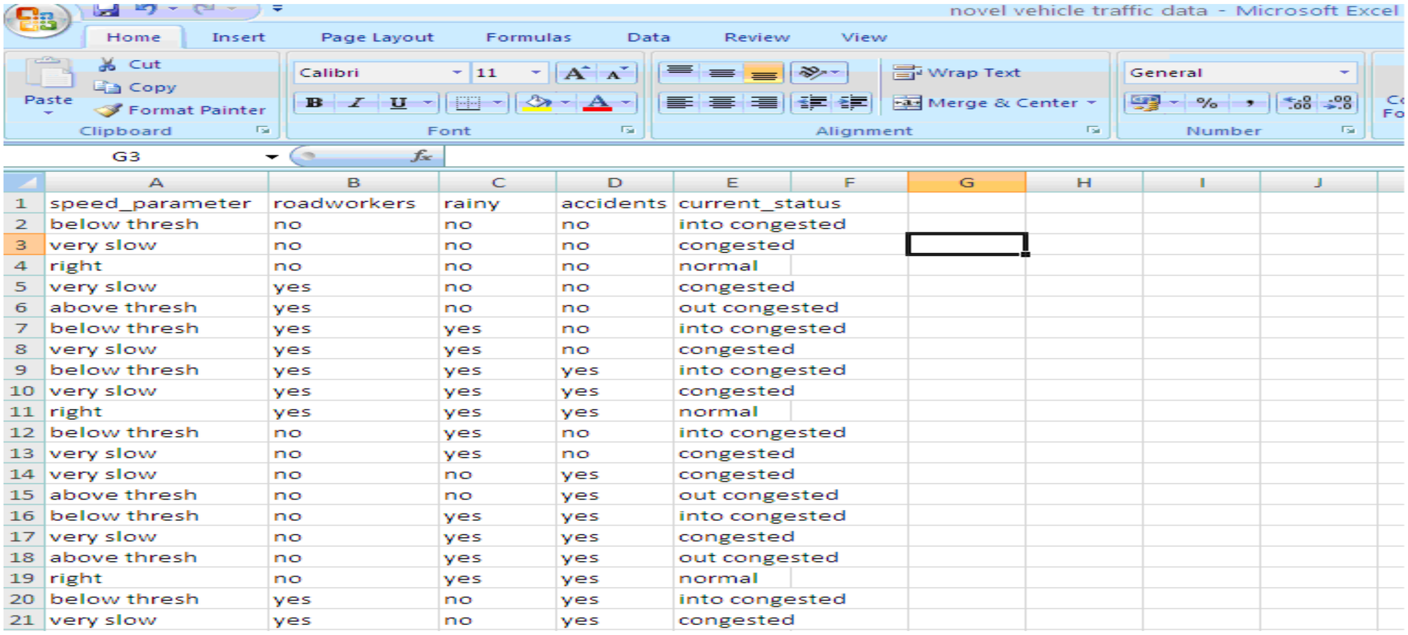
Annexure B, An output of a trained MLP model



```
Time taken to build model: 1 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         326              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0058
Root mean squared error                  0.0077
Relative absolute error                  1.7478 %
Root relative squared error              1.8741 %
Total Number of Instances              326

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1          1         1          1        CONGESTED
                 1         0          1          1         1          1        INTO CONGESTED
                 1         0          1          1         1          1        OUT CONGESTED
                 1         0          1          1         1          1        NORMAL
Weighted Avg.    1         0          1          1         1          1

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  52   0   0   0 |   a = CONGESTED
   0  43   0   0 |   b = INTO CONGESTED
   0   0  70   0 |   c = OUT CONGESTED
   0   0   0 161 |   d = NORMAL
```

Annexure C, Testing a trained MLP model with novel vehicle traffic data

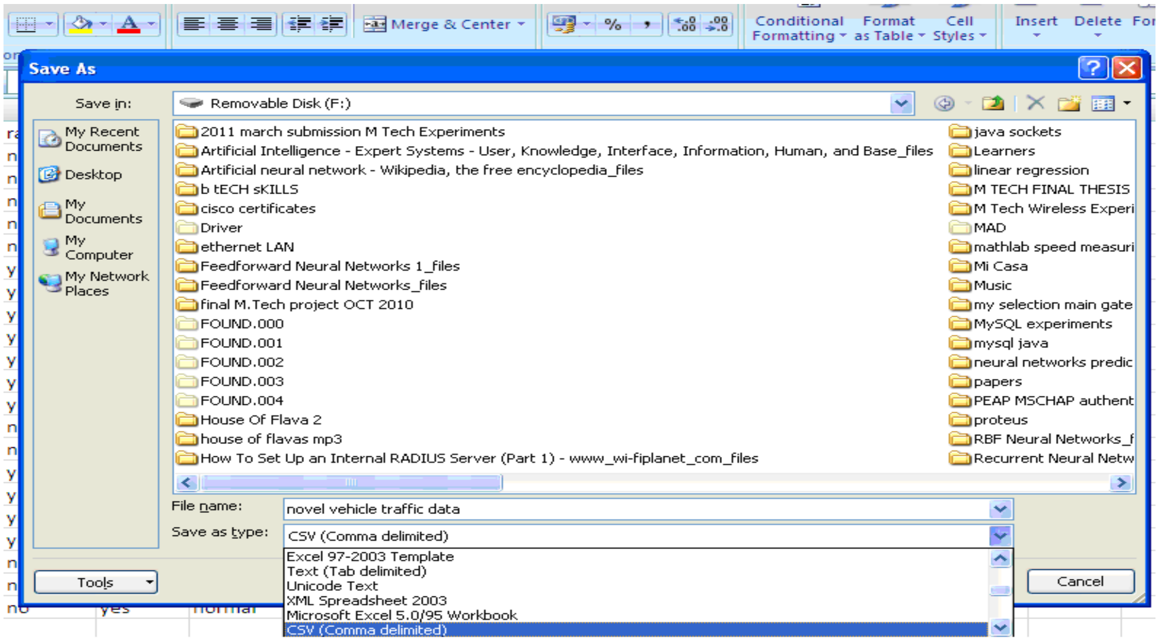
Vehicle traffic data was entered into an excel spreadsheet as shown in Figure 1.



	A	B	C	D	E
1	speed_parameter	roadworkers	rainy	accidents	current_status
2	below thresh	no	no	no	into congested
3	very slow	no	no	no	congested
4	right	no	no	no	normal
5	very slow	yes	no	no	congested
6	above thresh	yes	no	no	out congested
7	below thresh	yes	yes	no	into congested
8	very slow	yes	yes	no	congested
9	below thresh	yes	yes	yes	into congested
10	very slow	yes	yes	yes	congested
11	right	yes	yes	yes	normal
12	below thresh	no	yes	no	into congested
13	very slow	no	yes	no	congested
14	very slow	no	no	yes	congested
15	above thresh	no	no	yes	out congested
16	below thresh	no	yes	yes	into congested
17	very slow	no	yes	yes	congested
18	above thresh	no	yes	yes	out congested
19	right	no	yes	yes	normal
20	below thresh	yes	no	yes	into congested
21	very slow	yes	no	yes	congested

Figure 1: Shows vehicle speed data entered in excel spreadsheet

In Figure 2 the created spreadsheet was saved as a CSV file. Saving the spreadsheet as a CSV was to allow the WEKA workbench to be able to interpret the records within the spreadsheet.



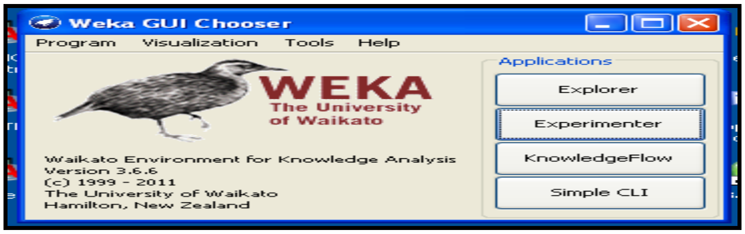
Save in: Removable Disk (F:)

File name: novel vehicle traffic data

Save as type: CSV (Comma delimited)

Figure 2: Shows saving an excel spreadsheet as a CSV file

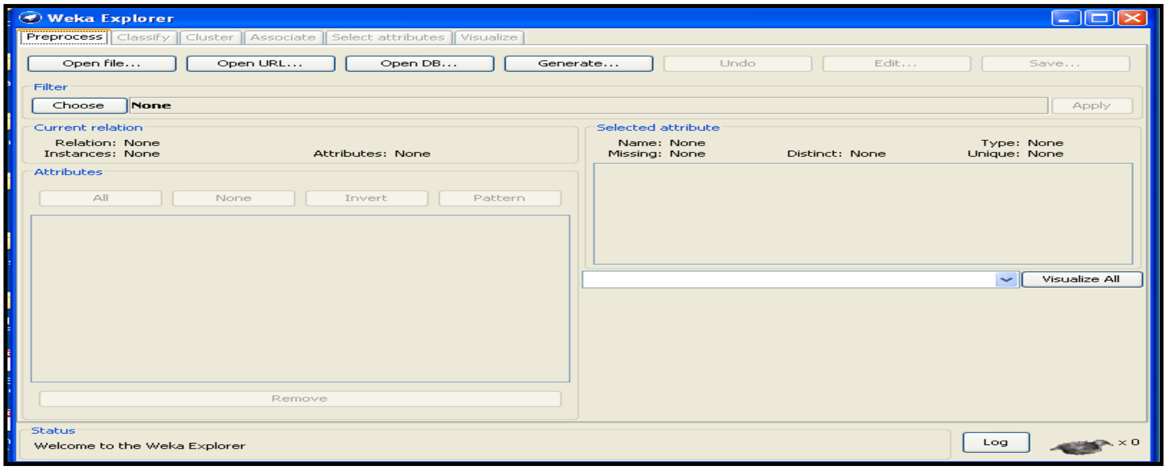
Launch WEKA toolbox and click on “Explorer” in Figure 3.



Waikato Environment for Knowledge Analysis
Version 3.6.6
(c) 1999 - 2011
The University of Waikato
Hamilton, New Zealand

Figure 3: Shows launching WEKA main window

Click “Open file” in Figure 4 to load excel spreadsheet saved in Figure 4.47.



Filter: None

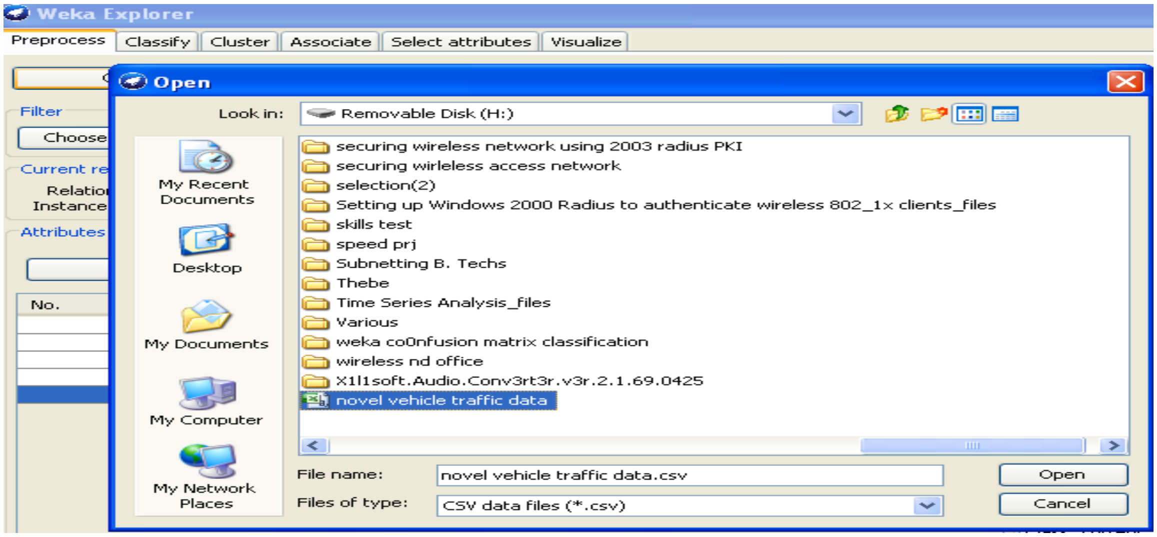
Relation: None
Instances: None
Attributes: None

Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Figure 4: Shows opening a spreadsheet file using WEKA Preprocessor window

In Figure 5 browse to the location where excel spreadsheet was saved and open a saved spreadsheet from WEKA explorer.



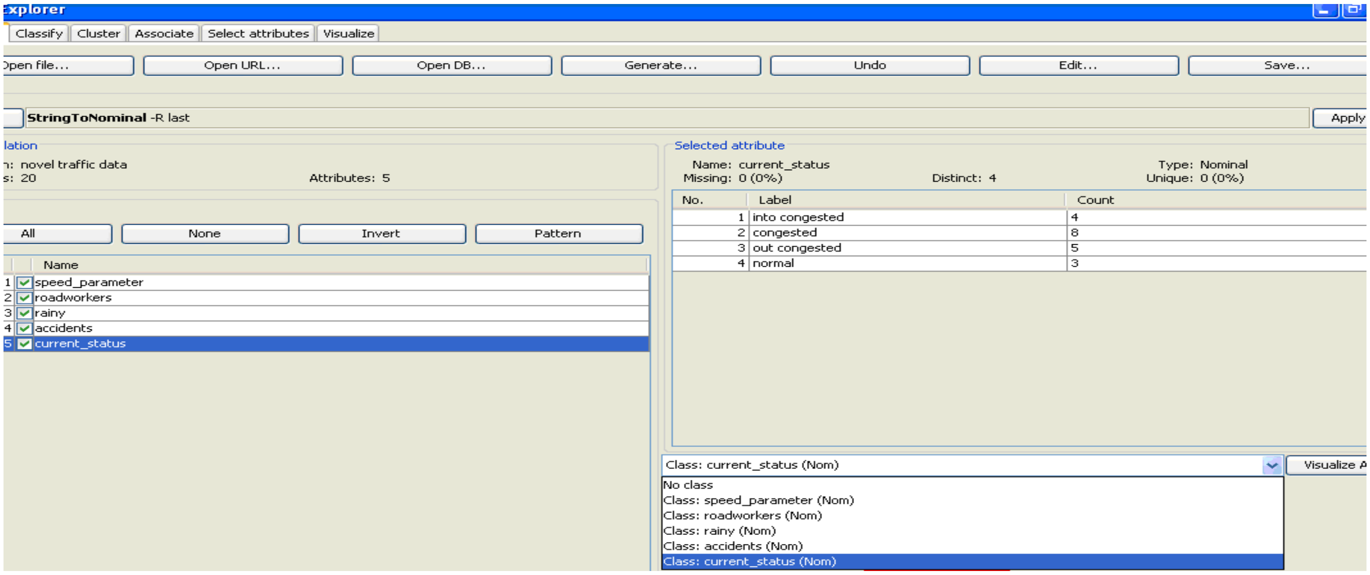
Look in: Removable Disk (H:)

securing wireless network using 2003 radius PKI
securing wirleless access network
selection(2)
Setting up Windows 2000 Radius to authenticate wireless 802_1x clients_files
skills test
speed prj
Subnetting B. Techs
Thebe
Time Series Analysis_files
Various
weka co0nfusion matrix classification
wireless nd office
X1l1soft.Audio.Conv3rt3r.v3r.2.1.69.0425
novel vehicle traffic data

File name: novel vehicle traffic data.csv
Files of type: CSV data files (*.csv)

Figure 5: Shows locating a spreadsheet file to be loaded using to WEKA

Click “ALL” button to select all attributes in Figure 6. Also select an attribute with classes to be predicted (current_status) from a dropdown list located on the bottom right of Figure 6. Click “Save” button to save the changes and automatically change the file type to arff as shown in Figure 7. Changing the file type ensures data compatibility between the trained MLP model and the spreadsheet file to be used during testing.



StringToNominal -R last

n: novel traffic data
s: 20 Attributes: 5

		Name
1	☑	speed_parameter
2	☑	roadworkers
3	☑	rainy
4	☑	accidents
5	☑	current_status

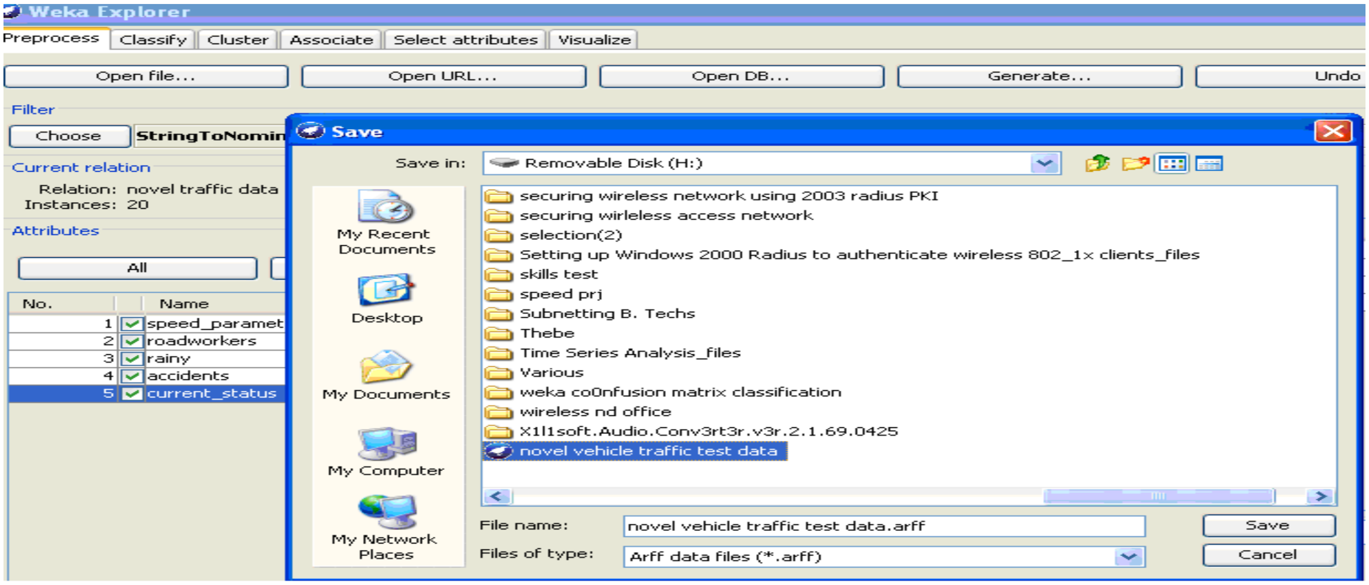
Name: current_status Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count
1	into congested	4
2	congested	8
3	out congested	5
4	normal	3

Class: current_status (Nom)

No class
Class: speed_parameter (Nom)
Class: roadworkers (Nom)
Class: rainy (Nom)
Class: accidents (Nom)
Class: current_status (Nom)

Figure 6: Shows selecting current_status as a class attribute



StringToNomin

Relation: novel traffic data
Instances: 20

No.		Name
1	☑	speed_paramet
2	☑	roadworkers
3	☑	rainy
4	☑	accidents
5	☑	current_status

Save in: Removable Disk (H:)

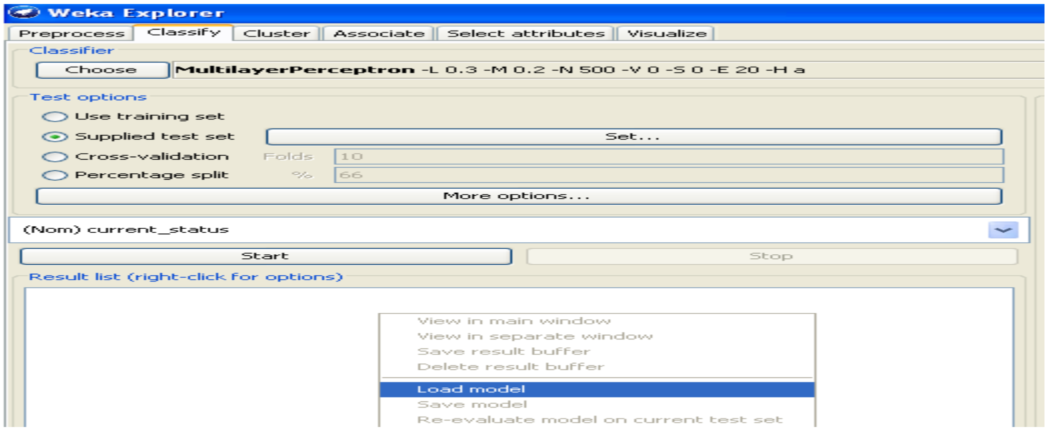
securing wireless network using 2003 radius PKI
securing wirleless access network
selection(2)
Setting up Windows 2000 Radius to authenticate wireless 802_1x clients_files
skills test
speed prj
Subnetting B. Techs
Thebe
Time Series Analysis_files
Various
weka co0nfusion matrix classification
wireless nd office
X1l1soft.Audio.Conv3rt3r.v3r.2.1.69.0425
novel vehicle traffic test data

File name: novel vehicle traffic test data.arff
Files of type: Arff data files (*.arff)

Figure 7: Shows saving changes made in WEKA and changing file type to arff type

After saving the changes made in Figure 6 and Figure 7, click on “Classify” tab. Click “Choose” button to select to use multilayer perceptron classifier. Then right click anywhere within result list pane to load a saved model as shown in Figure 8.

In Figure 9 the location where the trained multilayer perceptron model will be loaded from was identified and the model was opened to load.

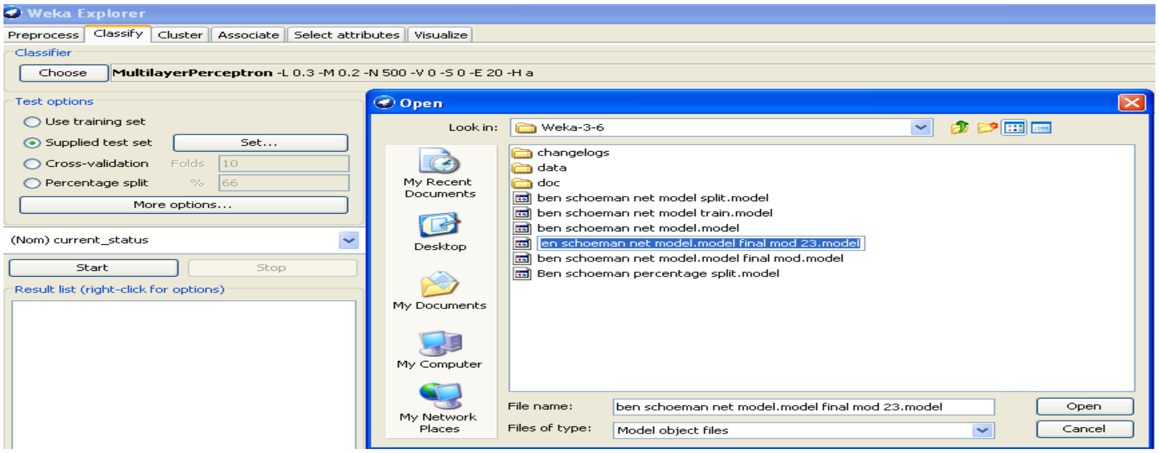


MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

Figure 8: Shows chosen multilayer perceptron as a classifier and loading a saved model



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

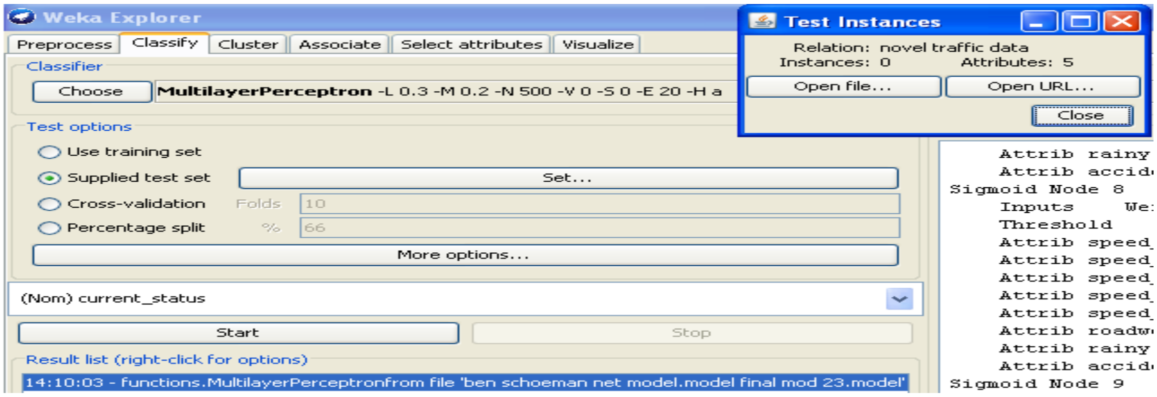
Look in: Weka-3-6

changelogs
data
doc
ben schoeman net model split.model
ben schoeman net model train.model
ben schoeman net model.model
en schoeman net model.model final mod 23.model
ben schoeman net model.model final mod.model
Ben schoeman percentage split.model

File name: ben schoeman net model.model final mod 23.model
Files of type: Model object files

Figure 9: Shows browsing to the location where the saved model was saved

In Figure 10 under “test options” select “supplied test set” and click on “set” button. Click “Open file” from Test Instances pane which is visible after clicking a “set” button as shown on the top right corner in Figure 10.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

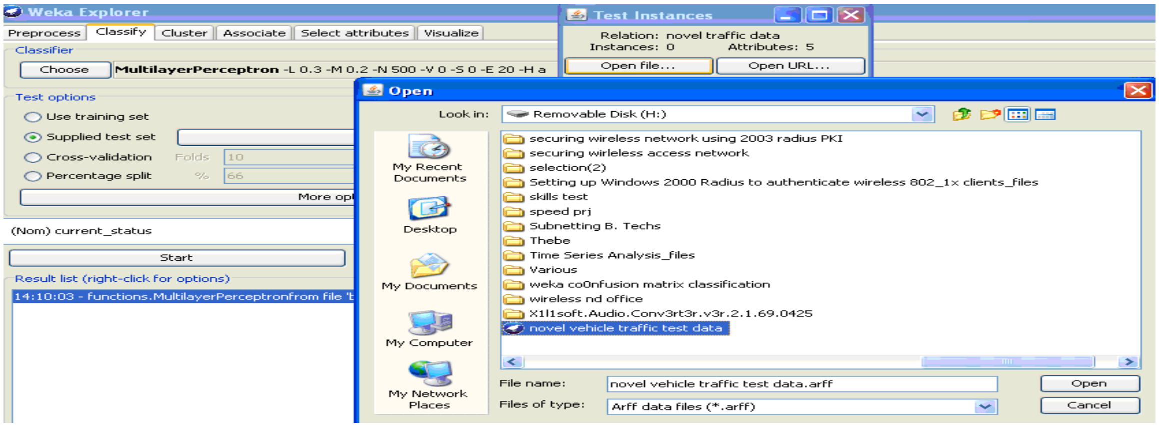
14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'

Relation: novel traffic data
Instances: 0 Attributes: 5

```
    Attrib rainy
    Attrib accid
Sigmoid Node 8
    Inputs    We
    Threshold
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib speed_
    Attrib roadw
    Attrib rainy
    Attrib accid
Sigmoid Node 9
```

Figure 10: Shows choosing a supplied test option and opening a file to be used to test the model

Browse to the location containing the arff file saved in Figure 7. Open the file as shown in Figure 11.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

14:10:03 - functions.MultilayerPerceptronfrom file 't

Relation: novel traffic data
Instances: 0 Attributes: 5

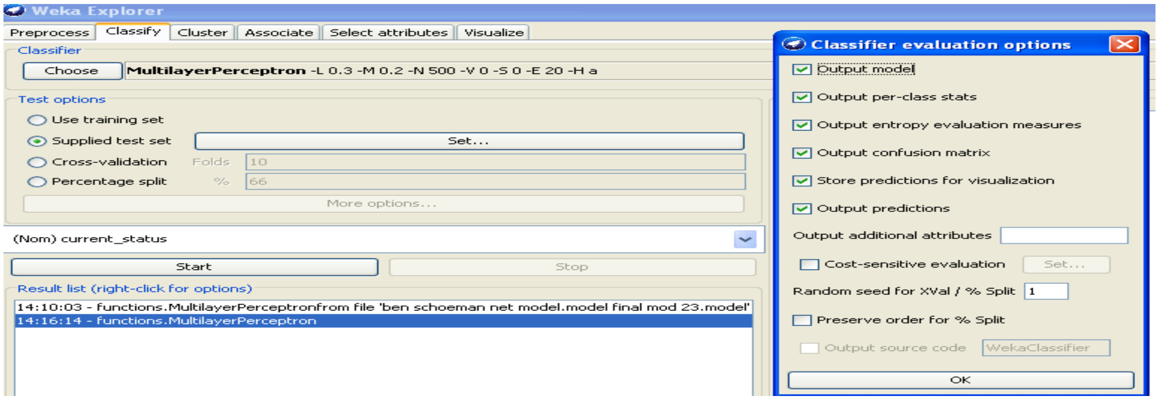
Look in: Removable Disk (H:)

securing wireless network using 2003 radius PKI
securing wireless access network
selection(2)
Setting up Windows 2000 Radius to authenticate wireless 802_1x clients_files
skills test
speed prj
Subnetting B. Techs
Thebe
Time Series Analysis_files
Various
weka co0nfusion matrix classification
wireless nd office
X1l1soft.Audio.Conv3rt3r.v3r.2.1.69.0425
novel vehicle traffic test data

File name: novel vehicle traffic test data.arff
Files of type: Arff data files (*.arff)

Figure 11: Shows browsing location to open the arff test file

Click "more option" button and make sure that output predictions, store predictions for visualization, output confusion matrix, output entropy evaluation measures, output per class stats and output model are all selected as shown in Figure 12. Click "OK" button when done and click "start" button to begin testing.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

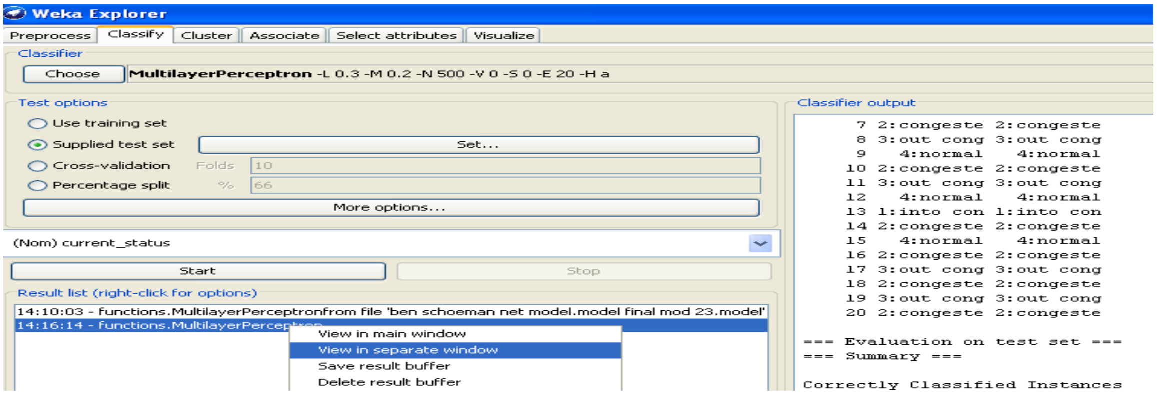
(Nom) current_status

14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'
14:16:14 - functions.MultilayerPerceptron

☑ Output model
☑ Output per-class stats
☑ Output entropy evaluation measures
☑ Output confusion matrix
☑ Store predictions for visualization
☑ Output predictions
Output additional attributes
☐ Cost-sensitive evaluation
Random seed for XVal / % Split 1
☐ Preserve order for % Split
☐ Output source code WekaClassifier

Figure 12: Shows selecting classifier evaluation options

When testing was complete the results were shown inside the classifier output pane on the right hand side in Figure 13. To view detailed results in a separate window right click on top of the multilayer perceptron model within the result list pane shown in Figure 13 and select “view in separate window”.



Classifier

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) current_status

Result list (right-click for options)

14:10:03 - functions.MultilayerPerceptronfrom file 'ben schoeman net model.model final mod 23.model'
14:16:14 - functions.MultilayerPercep

Classifier output

```
      7 2:congeste 2:congeste
      8 3:out cong 3:out cong
      9   4:normal   4:normal
     10 2:congeste 2:congeste
     11 3:out cong 3:out cong
     12   4:normal   4:normal
     13 1:into con 1:into con
     14 2:congeste 2:congeste
     15   4:normal   4:normal
     16 2:congeste 2:congeste
     17 3:out cong 3:out cong
     18 2:congeste 2:congeste
     19 3:out cong 3:out cong
     20 2:congeste 2:congeste

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances
```

Figure 13: Shows viewing output results in a separate window