AUT UNIVERSITY COMPUTING + MATHEMATICAL SCIENCES

Web Traffic Prediction for Online Advertising

Rojaa Ramani Matlakunta
(Student ID: 0785495)

This thesis is submitted as part fulfillment of the Degree of Master of Computer and Information Sciences at the Auckland University of Technology.

March 2011

Table of Contents

Attestation of Authorship

“I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree of diploma of a university or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.”

Yours sincerely,

(Rojaa Ramani Matlakunta)

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Dr. Russel Pears, Senior Lecturer in Computer Science and PhD Coordinator, AUT. His wide knowledge and his logical way of thinking have been of great value for me. His understanding, encouraging and guidance have provided a good basis for the present thesis. I am deeply grateful to my supervisor for his detailed and constructive comments, and for his important support throughout this work.

I owe my loving thanks to my lovely family. Without their encouragement and understanding, it would have been impossible for me to finish this work. My special gratitude goes to my Mother, Husband, Sai (older son) and little one Krish. I remember many sleepless nights with my little two years old son and my family constantly supported me throughout my studies.

Finally, I am ever grateful to God, the Creator, who is always with me and supporting. God at all times gave me the courage to face the complexities of life and complete this project successfully.

List of Abbreviations

- ARIMA : Auto Regressive Integrated and Moving Average
- ANN : Artificial Neural Network
- ANFIS : Adaptive Network based Fuzzy Inference System
- AR : Autoregressive
- BPN : Back-propagation network
- DENFIS : Dynamic Evolving Neuro-Fuzzy Inference Systems
- DOW : Day of the Week
- DFT : Discrete Fourier Transform
- ECOS : Evolving Connectionist Systems
- ERNN : Evolving recurrent neural network
- EPS : Embedded phase space
- EFuNN : Evolving fuzzy neural systems
- ETL : Extract, Transform and Load
- FNN : Fuzzy neural network
- FIS : Fuzzy Inference System
- FFT : Fast Fourier Transform
- KDD : Knowledge discover in databases
- MAPE : Mean Absolute Percentage Error
- MLP : Multi-layer perceptron
- MANFIS : Multi-input-multioutput-ANFIS
- OLAP : Online Analytical Processing
- RNN : Recurrent neural networks
- RMS : Root Mean Square
- SRN : Simultaneous recurrent neural network
- SOM : Self Organizing Map

List of Tables

List of Figures

Abstract

Online advertising is about publishing advertisements/commercials on the Web and helps advertisers to achieve their target on the Web. Online advertising maintains a set of popular websites on their network for each market/country. Therefore, they have to forecast the traffic of these websites. This information will be helpful for business analysts to propose the suitable Web sites to the marketers for advertising their product. The Business analysts have to analyse the user patterns; i.e., traffic data, demographics, etc., of various websites in their network before they propose a deal to the marketers. Most of the traffic on the websites is significantly steady. However, traffic data on few of the websites varies due to some periodic special events (like cricket world cup, rugby world cup, etc.) or sudden cases (like natural disasters) and some are seasonal websites (skiing websites, Christmas, etc.). All these factors have to be considered while forecasting the traffic of the Websites. Thus, online advertising have to predict the traffic of every website depending on their historical traffic data for planning or for scheduling commercials for Clients.

Current research mainly concentrates on the data present on World Wide Web (WWW). Employing various data mining schemes to unearth the underlying patters from the web is termed as Web mining. This stream of data mining processes the data that is present in form of web pages or web activities (for ex: server logs) (Dunham, 2003).

Web mining tasks can be divided into three types, which are *Web usage mining, Web content mining* and *Web structure mining.* This research is primarily concentrating on Web usage mining. Web usage mining mainly involves the automatic discovery of user access patterns from one or more Web servers (Mobasher, 1997). The analysis of such data can help the organizations to determine the life time value of customers, cross marketing strategies across products, and effectiveness of promotional campaigns. Finally, for organizations that sell advertising on the World Wide Web, analysing user access patterns helps in targeting ads to specific groups of users (Mobasher, 1997). Therefore,

Web usage patterns can be used to acquire business intelligence to improve sales and advertisement on the Web.

The main objectives of this research are to mine four years historical data and identify the hot spots of websites, to discover the intensity and the time span of hot spots, and how these spots can be used in future traffic prediction. In addition, we are interested to research how these hot spots are recurring year after year. Current research mainly studies the historical traffic data of the websites and attempts to predict the future traffic by using the data mining models. Each and every data mining model has certain advantages and disadvantages. Some are suitable to certain domains and some are inappropriate to others. Therefore, the challenge is to determine the suitable data mining model for the current domain.

Chapter 1 Introduction

1.1 Motivation for the Research

The term *data mining* has mostly been used by statisticians, data analysts, and the management information systems (MIS) communities. The phrase *knowledge discovery in databases* (KDD) refers to the overall process of discovering useful knowledge from data, and *data mining* refers to a particular step in this process (Fayyad et.al., 1996). In this research, data mining techniques will be applied to the data on Web advertising in order to discover the knowledge of user traffic patterns.

The ability to forecast the future, based on the past data is an important leverage that can push the organization forward. Time Series forecasting, the forecast of a time ordered variable, is an important tool under this scenario, where the research goal is to predict the behaviour of complex systems solely by looking into the past data (Paulo et. al., 2006). A time series is a collection of periodic ordered observations (x1, x2….xt) that appears in a wide set of domains such as agriculture, finance, media, etc., just to name a few (Paulo et. al., 2006). A time series model assumes that the observations are *dependent***;** that is past patterns will occur in future. Therefore, this research is mainly focused on prediction, which is time series forecasting.

Online advertising is dependent on the predicted traffic data to project future traffic which is ultimately used to set the advertising rate for a given website which in turn is used in production of the *rate card*. A rate card lists advertising prices and functions as a promotion copy. It merely project the costs to advertise on the site but also projects general information on the industry covered and a profile of the people using the site, demographic information, policies, additional fees and artwork requirements. Apart from the rate card, the business analysts have to analyse the user patterns too i.e., traffic data, demographics etc., of various websites in their network. Hence, prediction plays a vital role in managing the online advertising process.

Because of the issues stated above, data mining techniques will be applied to traffic data. Firstly, it should be stated that the current system in operation at a major media company in New Zealand is very primitive in its prediction process, thus negatively impacting the rate card application, scheduling application and

several other reporting services [Private Communication]. The current system does not use any rigorous method for prediction. It basically scans the historical traffic data, calculates the average and projects the values for the future. Due to the overly simplified nature of the technique used, prediction is inaccurate and affects other systems as well. As the prediction is inaccurate, revenue prediction is inaccurate. It negatively impacts on setting advertising prices to marketers, projection of traffic figures and various reporting systems. Hence, there is a considerable drop in the revenue generation.

This research aims to improve the prediction process by employing techniques from the statistical and data mining areas to improve the prediction accuracy. In this context, the aim is to predict the future traffic load on the network based on historical traffic data which is collected periodically in time steps of hours, days, months and years. Time series prediction is based on the idea that the time series carry within them the potential for predicting their future behaviour. Analysing observed data produced by a system, can give good insight into the system, and knowledge about the laws underlying the data. With this knowledge gained good predictions of the system's future behaviour can be made (Geva, 1998). Box-Jenkin's ARIMA time series analysis is attractive for the reason that it can capture complex arrival patterns, including those that are stationary, non-stationary and seasonal (periodic) ones (Tran, 2001). Therefore, these factors influenced the choice to include such methods in this research.

1.2 Research Objective

To be successful in a competitive world, any media service must not only maintain a good understanding of their present state of network, but be able to forecast the future as accurately and precisely as possible. With so much of the future subject to unknown and random events, it is not surprising that medium and long term forecasts are regularly wrong. However, accurate prediction is highly desirable in Online Advertising.

Currently the traffic data is collected on an hourly basis for every website in the network. Once when the data is collected, it is not instantly accessible for prediction; it is available only a week later for forecasting because it has to be transferred to a different repository for further processing. In other words,

yesterday's data is not available for forecasting the next day's traffic; hence, a requirement is to do long range forecasting to predict traffic volume over time horizons that exceed a week.

Various methods have been proposed to model and predict the future behavior of time-series. Analysis of historical trends in traffic data with a view to making predictions is therefore a key task, and neural network technology is well suited (Zhao, 2004). Statistical models such as moving average and exponential smoothing methods, autoregressive models (AR), linear regression models, autoregressive moving average (ARMA) models, and Kalman filtering based methods have been widely used in practice (Park, 2009).

The main goal of research is to study the historical traffic data of the websites and attempt to predict the future traffic by using data mining models. Each and every data mining model has certain strengths and limitations. Some are suitable to certain domains and some are inappropriate to others. Therefore, the challenge is to determine the data mining model suitable to the application domain of online advertising. The main research goals are:

- To determine the suitable data mining model for the domain under investigation.
- To predict the hourly traffic data of the websites for at least a time horizon of two months.
- To identity hot spots in the network. Traffic data per hour of a website is referred to as a spot in media terminology. Certain spots of a website show unusual traffic numbers due to many reasons like activity, entertainment, popularity, event, interest or natural disaster, etc. Those spots are known as hot spots.
- To identity the seasonal behaviour of the websites.

1.3 Structure of the thesis

This chapter has discussed the current issues in online advertising and the goals of the research. Further, the motivation behind the research was discussed by assessing the management needs of online advertising.

In the next chapter, we will survey past work done in the area of neural networks and statistical methods for time series prediction. We will emphasize on the strengths and limitations of the individual techniques that are covered.

In Chapter 3 we will present our methodology which will include an analysis of the data and its patterns, the pre-processing strategy used for spectrum analysis and the computation of the Fast Fourier Transform.

Chapter 4 describes the various prediction models that are covered.

A plan for the experimental study will be presented in Chapter 5.

Chapter 6 presents the experimental results. We will detail the various experiments that we ran and compare results with previous work, where appropriate.

Chapter 7 concludes the thesis with a discussion of the key achievements of the research and discusses several different directions in which future work can be undertaken to further improve the prediction.

Chapter 2 Literature Review

2.1 Introduction

Data analysis is a process of cleaning, transforming and modeling data with the goal of highlighting useful information, suggesting conclusions and supporting decision making. Data mining is a particular data analysis technique that focuses on modeling and knowledge discovering for predictive rather than purely descriptive purposes. Predictive analytics focuses on application of statistical or structural models for predictive forecasting. It encompasses a variety of techniques from statistics, data mining and game theory that analyze current and historical facts to make predictions about future events (Dunham, 2003). The approaches and techniques used to conduct predictive analysis can be broadly grouped into *regression* techniques and *machine learning* techniques.

Regression analysis is widely used for forecasting and prediction, where its use has substantial overlap with the field of machine learning. This analysis is a statistical approach to forecast change in a dependent variable on the basis of change in one or more independent variables. The most commonly used form of regression is *linear regression.* Machine learning is a branch of artificial intelligence which was originally employed to develop techniques to enable computers to learn. It includes a number of advanced statistical methods of regression and classification. In artificial intelligence, *artificial neural networks* have been applied to problems of prediction, classification or control in a wide spectrum of fields (Dunham, 2003).

Current research handles time series data. Hence, the research emphasis on techniques of time series analysis and forecasting. Time is a phenomenon which is very complex and also very important in many real-world problems. That almost every kind of data contains time-dependent information, either explicitly in the form of time stamps or implicitly in the way the data is collected from a process that varies with time (Walgampaya, 2006). There are various techniques available for time series analysis and predictions. In the following section we are analyzing few common techniques on regression analysis and

neural networks, and some recent research show successful real world applications of these techniques.

2.2 Time series prediction

The fundamental problem in traffic prediction is to find efficient self-similar models to predict future traffic variations precisely with good predictability. (Walgampaya, 2006) used three common forecasting techniques, Support Vector Machines (SVMs), Multilayer Perceptron (MLP), and Multiple Regression (MR). Based on their results, MR technique gives the best model for the prediction and it outperforms the MLP and SVM.

(Tang and Fishwick, 2002) studied neural networks as a model for forecasting and compared results with Box-Jenkins method for both long and short term forecast. They have used around 16 different time series of varying complexity and concluded that results from both methodologies are comparable for short term forecasting. However, their work shows that neural networks are superior for series with long term memory. Work in (Chakraborty et al, 1992) used neural network to forecast flour prices. They compared their results with the autoregressive moving average (ARMA) model and showed that a better accuracy is achieved with neural networks. Similarly (Kajitani et al, 2005) have compared the forecasting abilities of neural network models with other forecasting techniques in time series forecasting. (He, 2009) introduced the Graphics Processing Units (GPU) based computing to accelerate the short term load forecasting with MLP. This forecasting method is tested with the Queensland electricity market demand time series. The results show that this approach is much better than the CPU based parallel computing mainly in terms of speed. GPU computation cost is only one tenth of the CPU's cost.

(Khotanzad, 2003) applied two different artificial neural network (ANN) architectures, multilayer perceptron and fuzzy neural network to predict one-step ahead value of the MPEG and JPEG video, Ethernet and Internet traffic data. MLP has been used to predict the variable bit rate traffic or simulated traffic using auto regression or chaotic models. The number of bytes per frame or per group of frames for the MPEG and JPEG video data is predicted. Their work combines the individual forecasts made by ANN and FNN predictors which

can enhance prediction accuracy, improve generalization and lower dependence on the training set. They used two stage prediction systems; the first stage includes the individual predictors, MLP and FNN running in parallel and the second stage consists of a combiner that produces the final forecast. Finally (Khotanzad, 2003) concluded that MLP and FNN predictors give better results compared to the AR model using the same number of the lagged traffic values.

(Welch, 2009) compared three types of neural networks for short term prediction of wind speed; the MLP, Elman recurrent neural network and simultaneous recurrent neural network (SRN). Their results show that while the recurrent neural networks outperform the MLP in the best and average case with a lower overall mean squared error, the MLP performance is comparable. While the SRN performance is superior, the increase in required training time for the SRN over the other networks may be a constraining factor, depending on the application. Over the past decades, neural networks have demonstrated great potential for time series prediction when such series follow a nonlinear trajectory. MLP and its variants were used frequently for non-linear time series prediction (Ma, 2007). However, MLP is not appropriate for chaotic time series prediction. Ma et al (Ma, 2007) concluded that RNNs were computationally more powerful than feed-forward networks and the valuable results could be obtained for chaotic time series prediction.

(Sabry, 2007) investigated two time series forecasting techniques, namely ARIMA and logistic regression to predict daily traffic volume at three Egyptian intercity roads. Their research used the SPSS statistical package. The average annual, monthly and weekly daily traffic volumes were calculated for both logistic and ARIMA models. The forecasted traffic volumes were then compared with the actual traffic volumes on the standard error deviation. Their research concluded that the ARIMA model seems to be the best method for forecasting traffic volume.

A neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from artificial neural network theory to determine its parameters (fuzzy sets and rules) by processing data samples. Sony and Kasabov (2002) introduce a fuzzy

inference system known as DENFIS for adaptive on-line learning and dynamic time series prediction. The DENFIS model has been applied to predict the future values of chaotic time series – the Mackey-Glass data set. DENFIS is similar to Evolving fuzzy neural networks (EFuNN) in some aspects. Sony and Kasabov (2002) developed the DENFIS model with an idea that, depending on the position of the input vector in the input space, a fuzzy inference system for calculating the output value is formed dynamically bases on *m* fuzzy rules created after the learning phase. Their experiment included 3000 learning data points which were used in the learning process and 500 testing data points which were used in the recall process. In recall process, DENFIS produced a satisfactory result. They compared the DENFIS with some existing on-line learning models such as neural gas, resource-allocating network (RAN), evolving self-organizing maps (ESOM) and evolving fuzzy-neural network. Sony and Kasabov (2002) demonstrated that DENFIS can learn complex temporal sequences in an adaptive way and can outperform existing models.

Based on above study, current research focuses on the prediction methods such as ARIMA, MLP, RNN and DENFIS for online traffic prediction. These methods are explained in Chapter 4.

2.3 Online advertising

There has been much work in Online advertising in general and advertising in social networks in particular. In order to show the most relevant ads to a user, we need to predict how many users are expected to click on a particular advertisement. This in turn amounts to predicting how many users are expected to visit the network in that particular hour. (Yao, 2006) presents a novel and efficient method to predict web traffic. They combined wavelet analysis and neural networks for web traffic prediction. Their work concluded that wavelets can achieve high prediction accuracy for web traffic forecasting.

When the Web traffic volume is enormous and keeps on growing then the task of mining useful information becomes more challenging (Wang, 2004). Wang et al research dealt with large datasets and also covered different aspects like daily and hourly traffic data, page requests etc., Wang et al proposed a hybrid neuro-fuzzy model for mining Web usage patterns. Their work presents the

clustered Web data using Self Organizing Map (SOM) followed by modelling Fuzzy Inference System (FIS) to learn and predict the short-term and long-term usage patterns. Their experimental results also revealed the importance of the cluster information to improve the prediction accuracy of the FIS.

A back-propagation network (BPN) is a neural network that uses supervised learning method and feed-forward architecture. It is one of the most frequently utilized neural network techniques for classification and prediction (Wu et.al, 2006).

(Chabaa, 2009) have applied the adaptive neuro fuzzy inference system (ANFIS) for forecasting the internet traffic time series. ANFIS is a combination of fuzzy systems and neural networks. It has been applied to internet traffic data which is composed of 1000 time points. Their experimental results show that there is almost a complete agreement between measured and predicted data. Chabaa et al (Chabaa, 2009) concluded that the ANFIS model produce good accuracy in the prediction of internet traffic.

Based on above literatures, many researchers adopted neural network data mining methods in a wide variety of different application areas. The current research adopts the ARIMA and neural networks for web traffic prediction. There are pros and cons for every mining technique. The main advantage of neural networks for prediction is that they are able to learn from examples and after their learning is finished, they are able to catch hidden and strongly non-linear dependencies. The disadvantage of NNs is that the dependencies learnt are valid for only a certain period (Gerard, 2002). On the other hand ARIMA forecasting needs at least 100 observations to build a proper model. Its data cost are usually high. Unlike other simple naive models, there is no automatic updating feature. As new data become available the entire procedure must be repeated (Geurts, 1975). This model tends to be high cost because it requires large data, lacks convenient updating procedures and must be built using nonlinear estimation procedures. The main advantage of ARIMA is that it is suitable for short-run forecast with high frequency data. Keeping these features in mind, current research wants to adopt these techniques for traffic prediction.

2.4 Summary

This chapter provides a brief outline of time series prediction with the help of basic introduction and past research experiences on online traffic prediction. The following chapters will carry out a detailed analysis of the research methodologies, research issues followed by an analysis of experimental results in Chapter 6.

Chapter 3 Research Methodology

3.1 Introduction

The current chapter presents the research methodology to be used in this research. It gives an overview of the research paradigm and the research methods that were applied. Knowledge discovery is defined as the process of identifying valid, novel, and potentially useful patterns, rules, relationship, rare events, correlations and deviations in data (Fayyad et al., 1996). 'Time-Series Data Mining' is one of the important activities in mining knowledge in the form of rules, patterns and structured descriptions for a domain expert (Abe, 2007). It is concerned with discovery of universal laws that can be used to make predictions (Yinghong, 2009). These kinds of investigations are closely aligned with the research paradigm known as 'Positivist'.

The positivist view is shared by researchers who believe that knowledge can be acquired through observation and experimentation. The positivist believes in *empiricism*. Empiricism relies on practical experience and experiments, rather than on theories (Yinghong, 2009). Current research understands the domain through observation and experimentation which is the case in data mining as well. Therefore, the research is mainly dependant on the positivist paradigm.

3.2 Research Approach

Research approach refers to the approach that has been adopted to conduct the research. A Positivist approach is usually associated with quantitative data collection methods and statistical analysis (Probst, 2003). On the other hand, Interpretivism or the qualitative approach is a way to gain insights through discovering meanings by improving our comprehension of the whole. Interpretivist research design is ethnography and the main methods are ones that help researchers understand social life from the point of view of those being studied, such as unstructured observation, unstructured interviews and personal documents. Positivist research is guided by the scientific criteria of measuring instruments of quantification, systematic collection of evidence, reliability and transparency. Positivist research designs tend to be those that are closest to the logic of natural science research and mainly involve the use of surveys or

experimental designs (Taylor, 2003). Therefore, the current research aligns with the Positivist approach rather than the Interpretivist approach.

3.3 Research Objective

Chapter 1 has highlighted issues related to web traffic prediction and problems associated with online advertising. We have provided a brief outline of data mining techniques in time series prediction with the past research experiences in Chapter 2. In spite of extensive research in the area of time series prediction, it is always uncertain which data mining technique is suitable for predicting future traffic as its usage in the application domain of online advertising has not been studied rigorously before. The main objective of research is to establish which prediction method performs best out of the range of methods that we will be applying, namely ARIMA, Multi Layer Perceptron, Recurrent Neural Network and DENFIS.

3.4 Research Methodology

Knowledge discovery in databases (KDD) is a complex process which involves many stages such as choosing the target data, data pre-processing, data transformation (if required), performing data mining to extract patterns and finally assessing and interpreting the discovered structures (Hand et.al., 2007). The time-series data mining environment consists of time-series pattern extraction, rule induction and rule evaluation through visualization and evaluation interfaces (Abe et.al, 2007). Signal processing methods, Fourier transform, Wavelet, and fractal analysis methods have been developed to analyze time-series data. (Abe et.al, 2007) has identified three phases for mining time series which includes: Data pre-processing, Mining and Post-processing of mined results.

Pre-processing data includes data cleaning, filtering irrelevant data and noise removal. The exact set of activities performed in this phase depends very much on the given time-series and also on the mining method that is to be deployed. Real world data generally contains errors, incomplete information and may also contain some discrepancies in codes etc. Hence, it is very important to cleanse the data before mining in order to get productive results. Healthy input leads to healthy and accurate results which in turn produces quality decisions.

3.4.1 Data Cleaning

Data Cleaning is a process which fills in missing values, removes noise and corrects the inconsistency data (Han, 2001).

The online advertising dataset has been analysed and missing values have been identified. After clarification with domain experts, it was decided that the arithmetic mean was to be used to replace missing traffic values. The Media Company embeds a unique code on every page of the Website to track the Internet traffic data. We have observed the traffic data was missing on certain pages of the website. In other words, system was unable to track the traffic data on certain places of the website. This could be interpreted in several different ways such as: inability to track traffic as the code was not embedded on the respective page, incorrect code, or page was temporarily halted for a while.

Noise Removal

Noise is fundamentally a random error or variance in a measure variable. These incorrect attribute values may be due to data entry problems, faulty data collection, inconsistency in naming convention or technology limitation (Han, 2001).

Current research has opted for a semi-automated approach to noise detection. With the help of domain experts we identified suspect values and checked them by hand. As mentioned in the previous section, code in each page of the website is helpful for tracking the online traffic data. However, there are instances in which few of the pages or sections might load incorrect code; hence attributing the traffic to incorrect websites. This happens due to data entry and it is hard to spot such critical issues except for someone who constantly monitors the traffic intensity of websites. This explains why the current research has opted for the manual tedious check to remove noise with the assistance of domain experts.

Data transformation

In data transformation, the collected data are transformed or consolidated into forms appropriate for mining. *Aggregation* is one of the methods in *data transformation,* where summary or aggregation operations are applied to the

data. The Media Company collects the traffic data on an hourly basis. For mining purposes, the hourly data was aggregated to the weekly and monthly levels of granularity in order to deduce the seasonal patterns that occur at these two levels.

Time Series Analysis

According to (Nagpual, 2005) a time series comprises four components namely *Trend Component, Seasonal Component, Cyclic Component* and *Irregular Component.* Basically, there are two main objectives of time series analysis: (Nagpual, 2005)

- Understand the underlying structure of the time series by breaking it down to its components.
- To fit a mathematical model and then proceed to forecast the future.

There are two main approaches to time series analysis, which are associated with the time domain (i.e. trend component) or the frequency domain (i.e periodic component). The techniques used in the frequency domain fall under spectral analysis, which is a practical tool to gain insight into the periodicities of the data. The main objective of spectral analysis is to detect unknown hidden frequencies in the periodic time series, to provide useful descriptive statistics and to act as a diagnostic tool to indicate which further analysis might be relevant. Therefore, the current research utilises Spectral analysis, as detailed in Chapter 4.

3.4.2 Mining Techniques

The ability to extract hidden knowledge in data and to act on the knowledge is becoming increasingly important in today's competitive world. The entire process of applying computer based methodology, including new techniques for discovering knowledge from data is called data mining (Kantardzic, 2003).

Two primary goals of data mining tend to be "predictive" and "descriptive" (Kantardzic, 2003). *Prediction* involves using some variables or attributes in the data set to predict unknown or future values of other attributes of interest. On the other hand, *description* focuses on finding patterns describing the data that can be interpreted by humans.

The Predictive modeling is based on the use of the historical data. Predictive model data mining tasks include classification, regression, time series, and prediction. Unlike the predictive modeling approach, a descriptive model serves as a way to explore the properties of the data examined, not to predict new properties (Dunham, 2003). Clustering, summarization, association rules, and sequence discovery are viewed as descriptive in nature (shown in Figure 3-1: Dunham, 2003).

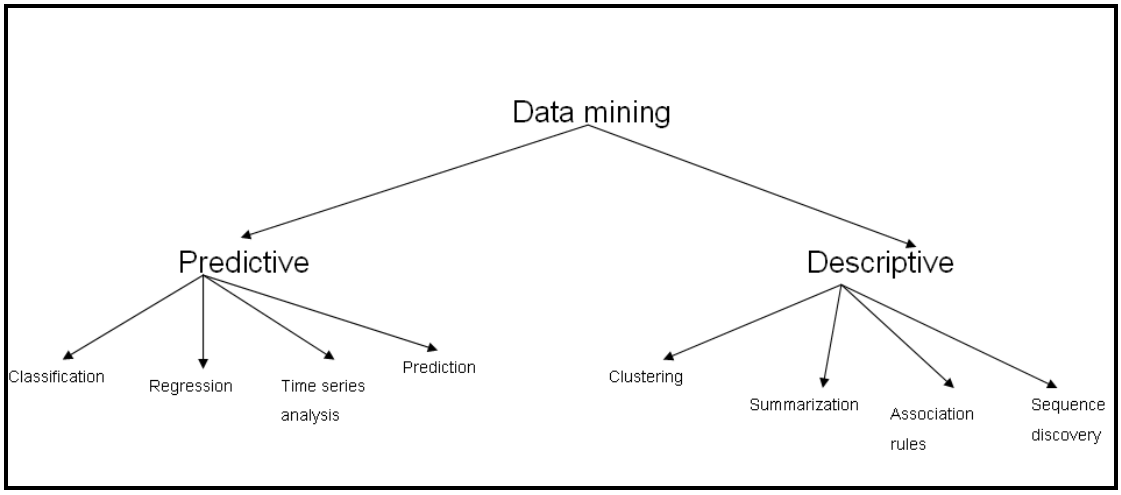


Figure 3-1: Data mining models and tasks (Dunham, 2003)

The main objective of this research is to predict the future online traffic of the web sites depending on the historical data, which aligns appropriately with techniques such as *regression, time series analysis* and *prediction*. This paper focuses on the prediction methods like ARIMA, MLP, RNN and DENFIS for online traffic prediction, which are explained in the next Chapter.

3.5 Summary

In this chapter, we outlined and discussed research approaches and objectives, along with a general classification scheme that outlined the different types of mining models. We provided an overall framework for our research approach and next chapter will discuss in detail the prediction models to be used in this research.

Chapter 4 Prediction Methods

4.1 Introduction

Current research focuses on the prediction methods, namely ARIMA, MLP, RNN and DENFIS for online traffic prediction. This chapter explains each of these prediction models in detail.

4.2 Periodogram

A basic idea in statistics and mathematics is to take a complex object (such as a time series) and break it up into the sum of simple objects that can be studied separately, see which ones can be thrown away as being unimportant, and then adding what's left back together again to obtain an approximation to the original object. The periodogram of a time series is the result of such a procedure (Newton, 1999). The periodogram is an estimate of the spectral density of a signal. By looking at the spectral density, we can identity seasonal components, and decide to which extent we should filter noise. The spectral representation of a time series $\{X_t\}$, (t=1,**...**,n), decomposes $\{X_t\}$ into a sum of sinusoidal components with uncorrelated random coefficients. From there we can obtain decomposition the autocovariance and autocorrelation functions into sinusoids (XLSTAT, 2010).

The spectral density corresponds to the transform of a continuous time series. However, we usually have access to only a limited number of equally spaced data points, and therefore, we need to obtain first the discrete Fourier coordinates (cosine and sine transforms), and then the periodogram. With the help of a smoothing function applied on the periodogram, we can obtain a spectral density estimate which is a better estimator of the spectrum (XLSTAT, 2010). A periodogram is a graphical data analysis technique for examining frequency-domain models of an equi-spaced time series. The periodogram is the Fourier transform of the autocovariance function. The periodogram for a time series x_t is (Jenkis, 1968):

$$S(f) = \frac{\Delta}{n}\left(\left(\sum_{t=-n}^{n-1} x_t \cos(2\pi f t \Delta)\right)^2 + \left(\sum_{t=-n}^{n-1} x_t \sin(2\pi f t \Delta)\right)^2\right)$$

where *f* is the frequency, *n* is the number of observations in the time series, Δ equals (n+1)/2 for odd values of n and (n+2)/2 for even values of n (Jenkis, 1968). The Vertical axis represents the spectrum estimate at the given frequency while the Horizontal axis consists of the Fourier frequencies (1/n, 2/n, 3/n, …, (n/2)/n) where n is the number of observations in the time series.

According to Chatfield (2004, p.136), two factors have led to increasing use of the smoothed periodogram. First, is the advent of high-speed computers. Second, is the discovery of the fast Fourier Transform (FFT) which greatly speeded up computations. In practice, the periodogram is often computed from a finite length digital sequence using the fast Fourier Transform. FFT is a fast algorithm for computing the Discrete Fourier Transform (DFT).

The FFT functions (fft, fft2, fftn) are based on a library called FFTW. To compute an *n*-point DFT where *n* is composite ($n = n_1 n_2$), the FFTW library decomposes the problem using the Cooley-Tukey algorithm, which first computes n_1 transforms of size n_2, and then computes n_2 transforms of size n_1 (MathWorks, 2010). The FFT is a faster implementation of the DFT. The FFT algorithm reduces an *n*-point Fourier transform to about $(n/2) \log_2 (n)$ complex multiplications. For example, a DFT algorithm on 1024 data points would require 1,048,576 multiplications. The FFT algorithm reduces to 5120 multiplications.

The spectrum of a time series is not only an important theoretical concept but it is also an important practical tool to gain insight into the periodicities of the data (Nagpual, 2005). Therefore, current research implements periodogram to detect unknown hidden frequencies in the time series. According to (Nagpual, 2005), an inherent problem of periodogram estimates is that the variance is large, of the order of power spectral density (PSD) squared. Moreover, the variance doesn't decrease as $n \rightarrow \infty$.

4.3 Multi Layer Perceptron

Gerard (2002) defines a Neural network as a system which is composed of many simple processing elements operating in parallel whose function is determined by the network structure, connection strengths, and the processing performed at computing elements or nodes. NN is a powerful data modeling tool which is able of capturing and representing complicated input/output relationships. NN acquires knowledge through learning. Hence, networks resemble the human brain at some simplistic level (Gerard, 2002). The main advantage of the usage of neural networks for prediction is that they are able to learn from examples and capture hidden, strongly non-linear dependencies. Another big advantage is they can handle problems with many parameters. The disadvantage of NNs is that they are notoriously slow, especially in the training phase but also in the application phase.

MLP is a feed forward network and it is a common form of neural network. It is also known as a supervised network. This network requires a desired output in order to learn. MLP is mainly used to create a model that maps correctly the input to the output using historical data. Thus, a model can be used to produce the output when the desired output is unknown. MLP involves two steps, training and testing. One assumes that a training set is available, given by the historical data that contains the inputs and corresponding desired outputs. In this learning method, MLP constructs an input-output mapping; adjusts the weights and biases at each iteration based on the minimization of error measure between the output produced and the desired output. Therefore, learning entails an optimization process and this is repeated until an acceptable criterion is reached (He, 2009). The structure of MLP is shown in Figure 4-1.

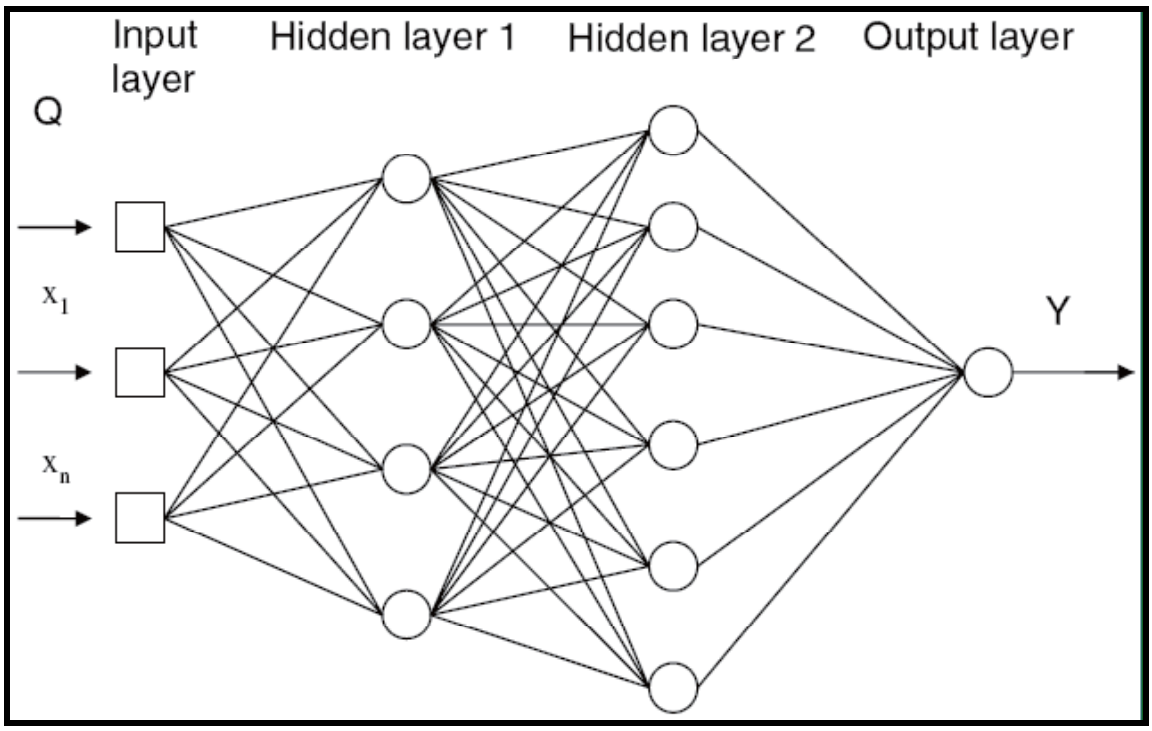


Figure 4-1: The structure of multi-layer neural networks.

It consists of input, hidden layers and an output layer. Each layer is comprised of several neurons. Each neuron is connected to every neuron in adjacent layers before they are been introduced as input to the neuron in the next layer by connection weight, which in turn determines the strength of the relationship between two connected neurons. Each neuron adds up all the inputs that it receives, and this sum has been converted to an output value which is based on a predefined activation (He, 2009). Therefore, the sum of x_i *(i = 1, 2,…,n)* multiplied with corresponded weight factor w_i and critical value, *b* formed the neuron output *y* thorough transformed function, *f*

$$y = f(net)$$

$$net = \sum_{i=1}^{n} x_i w_i + b$$

Where *n* is the number of neurons of input, hidden and output layers, *y* is model output, *f* is transformed function, w_i is weight and *b* is bias.

The main advantage of MLP's when compared to other neural model structures are that it is easy to implement and it can approximate any input/output map. The main disadvantages are that they are trained slowly and require large amounts of training data. One rule of thumb is that the training set size should be 10 times the network weights to accurately classify data with 90% accuracy

(Principe et al. 1999). According to Soltic et.al (2004), multilayer perceptron has some drawbacks such as absence of incremental learning, no facility for extracting knowledge (rules) and often, not good generalization. The major disadvantage of artificial neural networks is they cannot result in a simple probabilistic formula of classification (Yeh, 2009), encompass a slow training time, harder interpretation, and a difficult implementation in terms of the optimal number of nodes (Mitra, 2003).

4.4 Auto Regressive Integrated Moving Average (ARIMA)

The ARIMA modelling and forecasting approach is also known as the Box-Jenkins approach. The ARIMA model is a widely used for univariate time series. It combines three processes, namely, the Autoregressive (AR), differencing to strip off the integration (I) of the series and Moving Averages (MA) (Sabry, 2007). The AR term is linear regression of the current value of the series against one or more prior values of the series. The MA term is introduced to capture the influence of random shocks to the future (Liu, 2005). The combination of AR and MA term is called ARMA model. The ARMA model assumes that the data are stationary. However, this assumption doesn't hold in most of the real time data series. Hence, the **I**ntegrated term has been introduced to remove the impact of non-stationary data by differencing. The three processes, AR (p), I (d) and MA (q) are combined and interacting among each other and recomposed into the ARIMA (p, q, d) model.

A simple ARIMA (0, 0, 0) model without any of the three processes above is written as:

$$\mathbf{Y_t = a_t}$$

The *autoregression process* [ARIMA (p, 0, 0)] refers to how important previous values are to the current one over time. A data value at t_1 may affect the data value of the series at t_2 and t_3. But the data value at t_1 will decrease on an exponential basis as time passes so that the effect will decrease to near zero. It should be pointed out that ϕ is constrained between -1 and 1 and as it becomes larger, the effects at all subsequent lags increase (Box, 1994). Autoregression coefficients are denoted by ϕ.

$$Y_t = \phi_1 Y_{t-1} + a_t$$

The *integration process* [ARIMA (0, d, 0)] is differenced to remove the trend and drift of the data (i.e. makes non-stationary to data stationary). The first observation is subtracted from the second and the second from the third and so on. So the final form without AR or MA processes is the ARIMA (0, 1, 0) model (Box, 1994):

$$Y_t = Y_{t-1} + a_t$$

The order of the process rarely exceeds one (d < 2 in most situations).

The *moving average process* [ARIMA (0, 0, q)] is used for serial correlated data. The process is composed of the current random shock and portions of the q previous shocks. An ARIMA (0, 0, 1) model is described as (Box, 1994):

$$Y_t = a_t - \theta_1 a_{t-1}$$

θ theta is vector of MA coefficients (the coefficient of the lagged forecast error is denoted by the “theta” and it is conventionally written with a negative sign for reasons of mathematical symmetry.

As with the integration process, the MA process rarely exceeds the first order. ARIMA forecasting needs at least 100 observations to build a proper model. Its data cost are usually high. Unlike other simple naïve models, there is no automatic updating feature. As new data become available the entire modeling procedure must be repeated (Geurts, 1975). This model tends to be high cost because it requires large data, lack of convenient updating procedures and the fact that they must be estimated using nonlinear estimation procedures. The main advantage of ARIMA model is it works for short-run forecast with high frequency data. It is flexible and can be used for various applications with different features. For example: it can model time series with a wide variety of features such as trend and seasonality by incorporating the AR term, the Integrated term and MA term together and by adjusting the parameters of each term (Liu, 2005).

4.5 Recurrent Neural Networks

A neural network is made up of individual units termed as neurons. Each neuron has a weight, connected to every neuron in adjacent layers and associated with each input. Each neuron sums up all the inputs it receives, and is converted to an output value which is based on a predefined activation or a transfer function (Kim, 2009). The Neural networks are in two categories; the feed forward neural network and the recurrent neural network. The feed forward NN's have no loops and their output mainly depends on present input layer. But RNN involves the previous states as well as current states. RNN is a modification to feed forward NN to allow for temporal classification. In this case, a *context* layer is added to the network, which stores the information between observations. When new inputs are fed into the RNN at each step, and then the previous contents of the hidden layer are fed into the context layer is shown in Figure 4-2. These then feed back into the hidden layer in the next step (Kim, 2009).

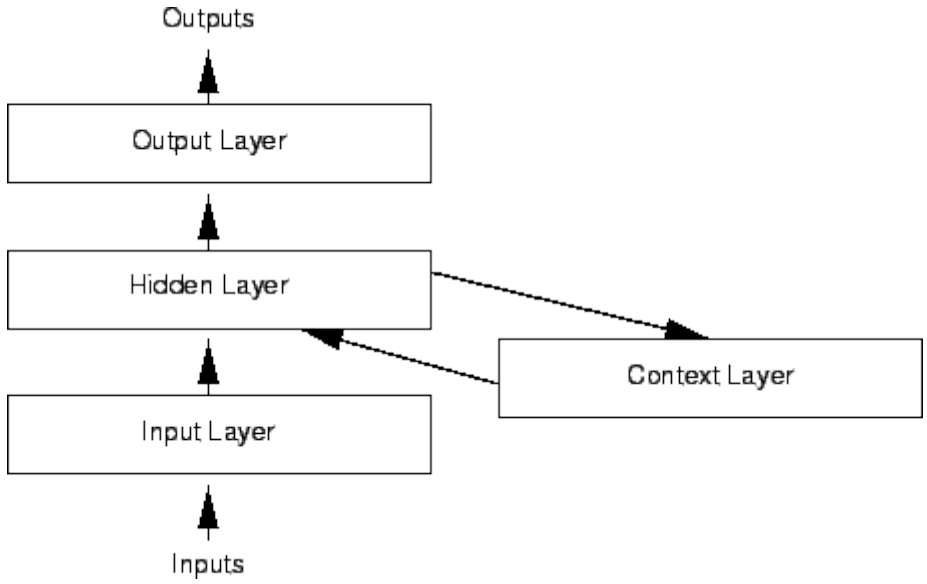


Figure 4-2: Recurrent neural network architecture.

RNN consists of internal states, inputs, outputs, weights, activation functions and feedback links. The current states are determined by the previous states, weights and inputs (Kim, 2009). RNNs have the capability to incorporate past experience due to internal recurrence (Ma, 2007). (Ma, 2007) addresses that the design of RNNs is difficult. The performance of neural networks is highly depends on the architecture and parameters of the networks. The main

disadvantage of RNNs is there are many parameters such as the number of units in hidden layer, learning rate, the encoding, and more (Kadous, 2002). Therefore, determining the parameters of a network greatly affect the performance criteria i.e., learning speed, accuracy of learning, noise resistance and generalization ability.

4.6 Dynamic Evolving Neuro-Fuzzy Inference Systems

Evolving Connectionist Systems (ECOS) are multi modular, connectionist architectures that facilitate modelling of evolving processes and knowledge discovery (Kasabov, 2003). An ECOS is a neural network and have specific characteristics. They learn in on-line, incremental mode, with one pass through the data. They learn in a life-long learning mode. They have evolving structures and use constructive learning (Kasabov, 2003).

ECOS facilitate different kind of knowledge representation and extraction, mostly memory based on statistical and symbolic knowledge. One of the ECOS model is called Evolving Fuzzy Neural Network (EfuNN) which consists of five layers namely input layer, fuzzy input membership functions layer, rule (case) node layer, fuzzy output membership functions layer and output layer as shown in Figure 4-3.

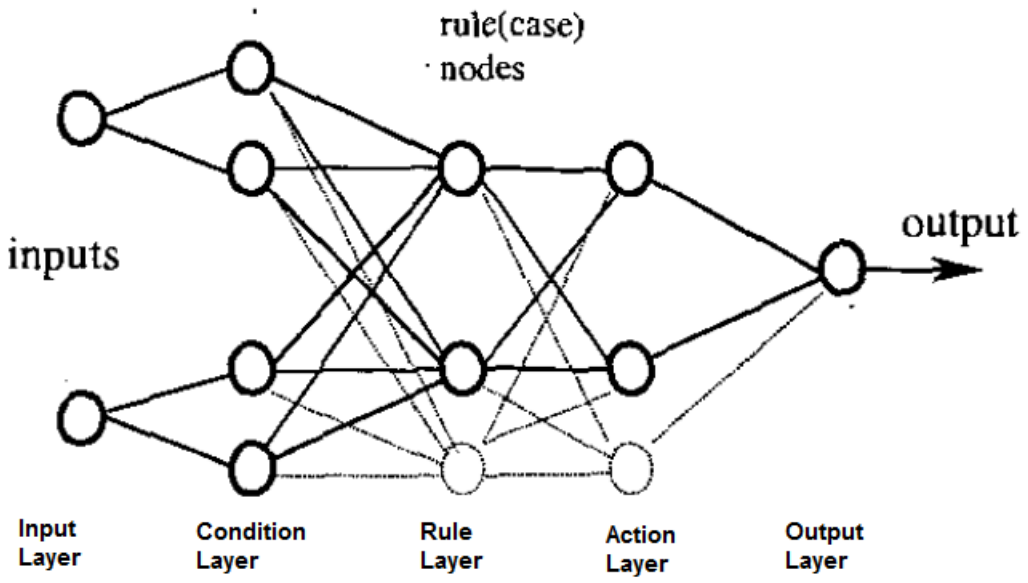


Figure 4-3: A block diagram of EfuNN (Kasabov, 2003).

Sony and Kasabov (2002) proposed a model called dynamic evolving neural fuzzy inference system (DENFIS). It inherits and develops EfuNN's dynamic

features that make DENFIS suitable for on-line adaptive systems. It uses Takagi-Sugeno type of fuzzy inference engine. This engine is composed of *m* fuzzy rules indicated as follows:

If x_1 is R_{m1} and x_2 is R_{m2} and … and x_q is $R_{mq,}$ then

y is $f_m(x_1,x_2, \ldots, x_q)$

where "x_j is R_{ij}", i = 1, 2, … m; j = 1, 2, … q, are m x q fuzzy propositions as m antecedents for m fuzzy rules respectively; x_j, j = 1, 2, …, q, are antecedent variables defined over universe of discourse X_j, $j = 1, 2, \ldots, q$, and R_{ij}, i = 1, 2, …, m; j = 1, 2, …,q, are fuzzy sets defined by their fuzzy membership functions $\mu_{Aij} : X_j$ -> [0, 1], i = 1, 2, …, m; j = 1, 2, …, q. In the consequent parts, y is a consequent variable, and crisp linear functions f_i, i = 1, 2, …, m, are employed.

Soltic et.al (2004) states that the DENFIS based model is recommended for online prediction applications. When new or unseen data becomes available, the DENFIS will adapt its structure and produce output to accommodate the new input data. The model creates rules during learning process.

4.7 Summary

This chapter outlined the characteristics of prediction methods, their strengths and limitations towards online prediction. These methods will be further discussed in the following chapter on experimental design.

Chapter 5 Experimental Design

5.1 Introduction

This chapter will focus on describing the experiments which are designed to analyze the variations in online traffic activity and to implement the proposed traffic prediction solution using various prediction methods. The main intention behind these experiments is to find a model which is capable of accurately predict websites traffic for up to four to five weeks in advance.

5.2 Datasets

The Online traffic repository for the Media Company under study maintains traffic data of websites from several different countries. For analysis and experimentation purposes, the raw dataset of five popular websites in the network for a single country will be used in the research. The dataset consists of following attributes:

Attribute name	Description	Data Type
ID	Website Unique identification number	Integer
MediaUnit_ID	Timestamp in yyyymmddhh	Big Integer
Country_ID	Traffic data of a Country	Integer
Year	Year in *yyyy* format	Integer
Month	Month in *mm* format	Integer
Day	Day in *dd* format	Integer
Hour	Hour in *hh* format	Integer
DayofWeek	Day of the week (1-Mon …. 7-Sun)	Integer
Week	Week of the year (1 .. 52)	Integer
TrafficVolume	Traffic volume (users)	Integer

Table 5-1: Dataset structure

5.2.1 Data Stream Sliding Window model

The need to process large amount of data has motivated the field of data mining. The data mining approach may allow handling of large data sets, but it still does not address the problem of a continuous supply of data. The data stream paradigm has emerged in response to the continuous data problem

(Bifet, 2009). Recently data intensive applications are widely recognized, the data to be processed in those applications is not static, but it is as continuous data stream. Such applications like network traffic analysis, online transaction analysis, networking monitoring, security, telecommunications data management, web applications, sensor networks and others (Tsai, 2009).

The time models for data stream mining include the *landmark model,* the *tilted-time window model* and the *sliding window model.* The landmark model mainly considers all the data from a specific period of time to the current time. The tilted-time model as well considers the data from the start of streams to the current time, but the time period is divided into multiple slots. The sliding model is different from other models; it focuses on the recent data from the current model back to a specified time period. The size of the window could be fixed time period or number of transactions (Tsai, 2009).

Current research uses the sliding window technique since it comprises of several attractive properties. It is well defined and easily understood. Most importantly, sliding window technique emphasises on recent data; it evaluates the query not over the entire historical data from the start, but rather only sliding windows of recent data from the streams (Chen, 2008).

(Chen, 2008) states three types of sliding window operators namely *tuple-based, time-based and partitioned.* A tuple-based sliding window on a stream *S* takes a positive integer *N* as a parameter and outputs a relation *R.* A time-based sliding window on a stream *S* takes a time interval *T* as a parameter and outputs a Relation *R.* A partitioned sliding window on a stream *S* takes an integer *N* and set of attributes {A1, ..., Ak} of *S* as parameters, and is specified by following *S* with “[Partition by A1, …, Ak Rows N]”. Current research domain imposes the time-based sliding window; it suits as it requires concentrating on most recent data for a given period for forecasting. For online traffic prediction, it considers last twelve weeks of data for predicting current week data (Figure 5-1). In practice, the domain experts from Media Company considers past three months (12 weeks) of data in prediction process.

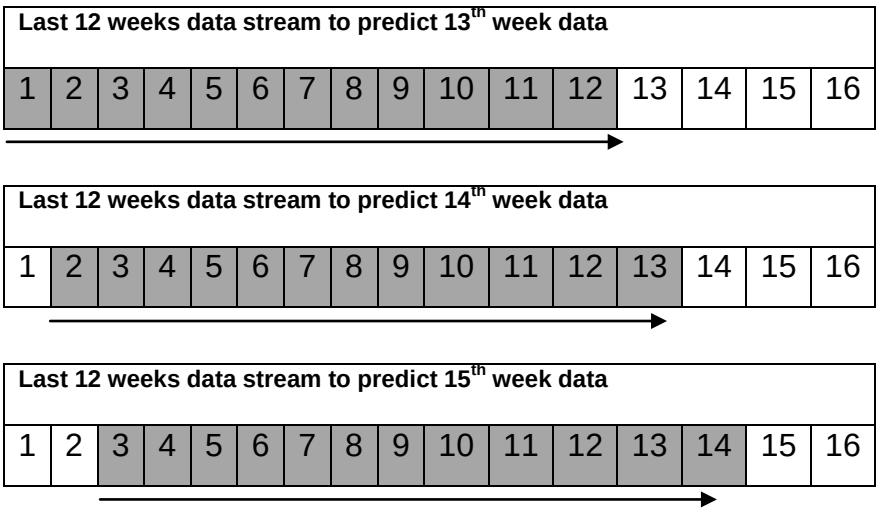


Figure 5-1: Sliding window model

Following describes a generic routine that is used to mine the data.

Routine: mineTrafficData()

Parameters:

- Let 'historicalWeeks' be the recent number of weeks, those weeks traffic data is considered for input.

Method:

For all the websites in data file, do the following

a. Prepare the data stream for every website,
b. Neural network produces output from input values,
c. Compare forecast values against the actual values. Change settings to reduce forecast error.

End for

5.3 Tools

This section highlights the statistical tools that have been used in current research; these tools are mainly student evaluation versions. Hence, they support limited number of records for running experiments.

5.3.1 Matlab

MATLAB (MATrix LABaoratory) is a tool for numerical computation and visualization. It is a high-level technical computing language and interactive environment for algorithm development, data analysis, data visualization and numeric computation. It contains Mathematical, Statistical and Engineering functions to support all common Engineering and Science operations. It

supports interactive development without the need to perform low-level administrative tasks, such as declaring variables and allocating memory. It helps to apply concepts in a wide range of engineering, mathematics and science applications including signal and image processing, control design, communications, financial modeling etc., (Mathworks, 2010).

MATLAB supports the entire data analysis process, right from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis to producing presentation-quality output. It provides Fourier analysis and filtering, Data analysis and statistics and various other functions for performing mathematical operations and analyzing data (Mathworks, 2010). Student version for download is available at www.mathworks.com.

Current research utilizes MATLAB for identifying pattern recurrence and to run experiments on neural networks for traffic prediction.

5.3.2 NeuralWare

NeuralWorks Predict is an integrated tool for rapidly creating and deploying prediction and classification applications. This package mainly combines neural network technology with genetic algorithms, statistics, and fuzzy logic to find optimal or near-optimal solutions for a wide range of problems. This tool has been used in classrooms and academic research facilities around the globe in Engineering, Medicine, Business, Computer Science and other disciplines (Neuralware, 2010).

The complete NeuralWorks system has five main components. *The Train/Test Selection* component selects the training and test sets for model building. *The Data Analysis and Transformation component* analyses data and transforms it into forms suitable for Neural Networks. *The Input Variable Selection Component* uses a generic algorithm to search the input variables which are good predictors of the output. *The Neural Net Component* of Predict supports two non-linear feed-forward constructive algorithms. One is based on non-linear Kalman filter learning rule and is mainly designed for regression problems. The other is general purpose algorithm which is based on Adaptive Gradient

learning rule. *The Flash Code Component* converts the completed model into C, FORTRAN or Visual Basic code (Neuralware, 2010).

In Microsoft Windows environment, NeuralWorks can be run either as an add-in for Microsoft Excel or as a command line program that offers batch mode processing. It is very easy to use and results are written back into the spreadsheet and the quick graph makes the result for suitable for presentation. Information about NeuralWare products and services can be found at www.neuralware.com.

Current research utilizes NeuralWare tool for Multi-Layer Perceptron model.

5.3.3 XLSTAT

XLSTAT is a powerful and user friendly data analysis and statistical solution tool. It offers comprehensive solutions for specific industries like Healthcare, Sensory analysis, Finance etc., It relies on Microsoft Excel for the input of data and as well as for the display of results, but the actual computation are executed by using its autonomous software components (XLSTAT, 2010).

This tool provides several components; one of the components called XLSTAT-Time is an add-in that has been developed to provide XLSTAT-Pro users with a powerful solution for time series analysis and forecasting. It provides features like Fourier transformation, Spectral analysis, Descriptive statistics, Smoothing, Series Transformation, ARIMA models and Homogeneity tests. The fully functional 30 days evaluation version is available on XLSTAT website www.xlstat.com.

Current research utilizes XLSTAT for ARIMA model prediction.

5.3.4 NeuCom

NeuCom is a generic environment for data analysis, modeling and knowledge discovery developed by the Knowledge Engineering and Discovered Research Institute (KEDRI). NeuCom is self programmable, learning and reasoning computer environment which is based on connectionist (Neurocomputing) modules. It is founded on the theory of Evolving Connectionist Systems (ECOS). These modules gets familiarized with the incoming data in an online

incremental mode and extracts the meaningful rules that eventually helps the users to discover new knowledge in their respective domain (AUT, 2010).

Neucom is used to solve problems related to clustering, classification, data mining, prediction, and pattern discovery. It is also a development environment where new intelligent systems for decision support and data analysis can be created across discipline (AUT, 2010). It can provide solutions in areas of Science, Engineering, Business, Bio-informatics, Medicine, Arts and Education. Student version of this software can be downloaded from the AUT website http://www.aut.ac.nz/research/research-institutes/kedri/research-centres/centre-for-data-mining-and-decision-support-systems/neucom-project-home-page#download

Current research utilises Neucom tool to implement DENFIS model for capturing Evolving patterns.

5.3.5 Microsoft SQL Server 2005

Microsoft SQL Server 2005 is a comprehensive, integrated data management and analysis software that enables the organizations to reliably manage mission critical information and confidently run complex business applications. SQL Server 2005 includes several enhancements to enterprise data management in areas mainly *availability, scalability, security, manageability* and *interoperability (*Microsoft, 2010).

For developer productivity, SQL Server 2005 includes expanded language support too. It facilitates developers to choose a variety of languages to develop database applications. *Business Intelligence* is another attractive feature which includes ETL tool, online analytical processing (OLAP), data mining, data warehousing and reporting functionality.

Current research handles SQL server databases. Management Studio is used to build several SQL scripts to process the online traffic data which includes removing noise and preparing datasets for respective data mining models.

5.4 Performance Metrics

Current research examines and compares the effectiveness of data mining techniques namely MLP, Recurrent neural networks, DENFIS and ARIMA. This research distinguishes the mining tools from each other, and outlines their strengths and weaknesses in the context of online traffic prediction. As a measure to evaluate the performance of the forecast models, the root mean square is used.

The root mean square error (RMSE) is used to measure the differences between values predicted by a model and the values actually observed from the thing being modeled. The RMSE is a kind of generalized standard deviation. Standard deviation is a statistical measure of spread or variability. The standard deviation is the root mean square deviation of the values from their arithmetic mean (Yao, 2006). RMSE formula is as follows:

$$RMSE = \sqrt{\sum_{t=1}^{N} (x_t - \hat{x}_t)^2 / N}$$

Where x_t and $\hat{x}_t$ are respectively the observed data and their predicted values, *N* is the number of predicted values concerned. Expressing the formula in words, the difference between the values predicted by a model and the values actually observed are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors.

5.5 Experimental Plan

The experiments were performed on a Windows XP PC with a Celeron ® CPU 2.80 GHz, 2 GB of RAM and 80 GB of hard disk space. Microsoft SQL Server Management Studio (2005) has been used to access SQL database of online traffic repository. Various SQL scripts are written to extract different types of datasets to facilitate these experiments. The following sections describe about the design of experiments in detail.

5.5.1 Experiment 1: Use of Periodogram for identifying Pattern Recurrence

This experiment is designed to analyze the variation in online traffic activity. The periodogram was used to study the time series data pattern. MATLAB and XLSTAT software tools have been used to run this experiment. Primarily, we have taken the raw dataset as mentioned in Section 5.2 and processed the data to fit into this experiment to analyze the variation in traffic activity. As a result, we have dropped all attributes in dataset except HOUR and TRAFFICDATA columns for about 500 records approximately for each of the five websites. This limitation of records is because some of the tools (as explained in section 5.3) are student evaluation versions and they support limited numbers of records. Since the data is collected in hourly intervals, the HOUR column records values from hour 00 (12am) to 23 (11pm) for a particular day. To identify pattern recurrence, the data has been altered in the HOUR column in the following manner.

Hour = (Hour + 24) * (Day -1)

The above pre-processing performed from the *second day* onwards for 30 days. We obtained five different datasets from the five websites that were chosen for this research.

In practice, the traffic data is collected in hourly intervals by the media company. Since we would also like to monitor the daily traffic patterns, SQL scripts were written to aggregate the hourly traffic data into daily date for the five websites.

5.5.2 Experiment 2: Use of MLP for Prediction

Differencing is a popular and effective method of removing trend from a time series. It offers a clear picture of a true underlying behaviour of time series (Easton, 1997). (Williams, 2003) states that the d*ifferencing* creates a transformed series which consists of the differences between lagged series and observations. The differencing operator with the single lag is represented with the symbol ∇. The first order differences of time series values x_1, x_2, x_3... x_N are given by new series y_1, y_2… $y_{N-1.}$ Therefore,

$y_1 = x_2 - x_1$

$y_2 = x_3 - x_2$

$y_3 = x_4 - x_3$

$y_{N-1} = x_N - x_{N-1.}$

The operation $y_t = x_t - x_{t-1} = \nabla x_t$ is called the first difference. Sometimes the first differencing is not enough to remove the trend. In such instances, we need further differencing. Second order difference is given by (Williams, 2003)

$Zt = \nabla^2 y_t = \nabla x_t - \nabla x_{t-1} = x_t - 2x_{t-1} + x_{t-2}$

Through the observation of data (from Experiment1 results), we found the number of users have been increasing over time, but we have also seen that there is seasonality too. To make this series stationary, we first must pre-transform the data and then must do regular differencing. Hence, we would like to implement following proposed traffic prediction solution (Figure: 5-2) using MLP neural networks.

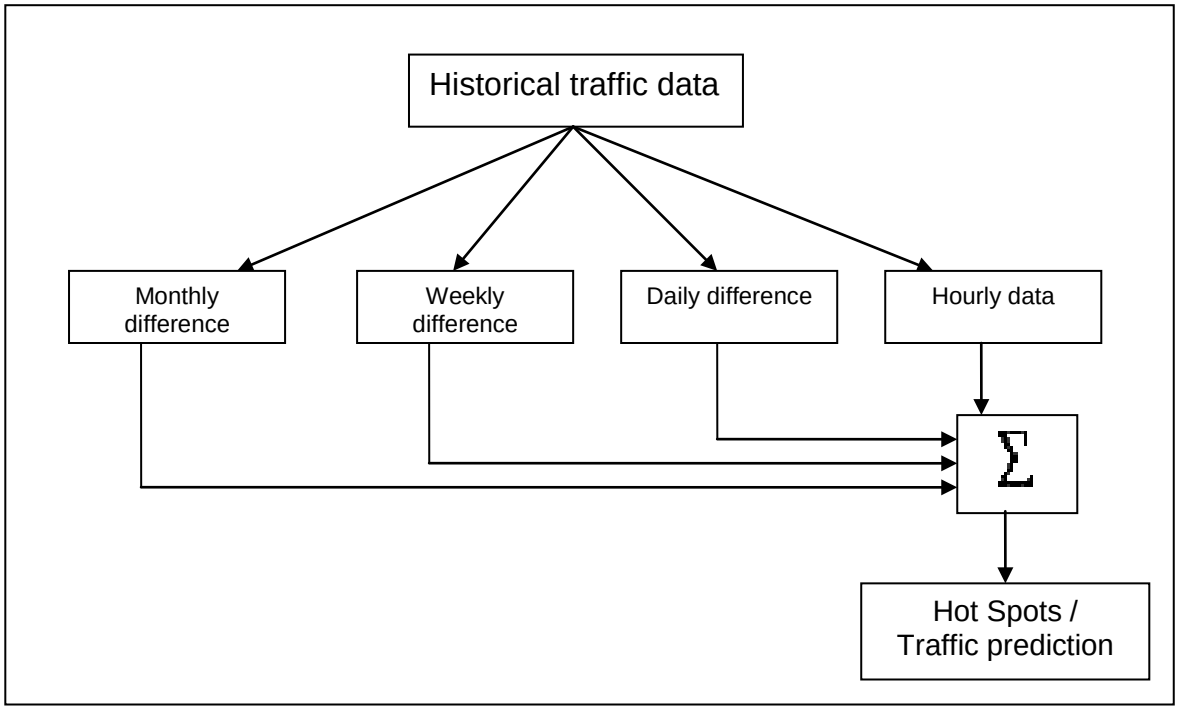


Figure 5-2: Proposed Traffic Prediction Solution

NeuralWare tool is used to run this experiment.

Data pre-processing

We have used one month traffic data to prepare datasets for MLP analysis. For illustration, the following Table 5-2 shows the Monday's traffic data at hourly level for Website1.

Calculating of Daily difference:

When working with time series data, we need to refer the values of a series in previous periods. We need the current and the previous values of the series to model a time series. As a result, daily difference finds the difference of the current day and previous day traffic values. The daily traffic time series values $d_1, d_2, d_3, \ldots\ldots, d_N$ are given by new series $y_1, y_2, \ldots\ldots y_{N-1}$..

The operation $y_t = d_t - d_{t-1} = \nabla d_t$ is called the *daily difference*.

For instance, in order to find out the daily difference value for an hour 09 (9am) of Monday, we find the difference of current Monday's traffic data with the previous Monday's traffic data.

For example to calculate daily difference value for 9 am, formula is as follows:

d_1 = Sum up the traffic gathered till 9am of the previous Monday i.e., 12am to 9am.

d_2 = Sum up the traffic gathered till 9am of current Monday i.e., 12am to 9am.

Daily difference $y_1 = (d_2 - d_1)$

Similarly, this process is repeated from Monday to Sunday for all 24 hours for approximately about three week's data for analysis.

Calculating of Weekly difference:

Weekly difference finds the difference of the current week and previous week traffic values. The weekly traffic time series values $w_1, w_2, w_3, \ldots\ldots, w_N$ are given by new series $y_1, y_2, \ldots\ldots y_{N-1}$. The operation $y_t = w_t - w_{t-1} = \nabla w_t$ is called the weekly *difference*.

For instance to calculate weekly difference value for 9am of Tuesday, formula is as follows:

w_1= Sum up the traffic gathered till 9am of the previous Tuesday i.e., previous week's traffic from 12am of Monday to 9am of Tuesday.

W_2= Sum up the traffic gathered till 9am of current Tuesday i.e., current week 's traffic from 12am of Monday to 9am of Tuesday.

Weekly difference $y_1 = (W_2 - W_1)$

Similarly, this process is repeated from Monday to Sunday for all 24 hours for approximately about three week's data for analysis.

Table 5-2: MLP dataset (Website1 – Monday's traffic data)

Weekly Difference	Daily Difference	Hour	Day Of Week	Actual Traffic
29800	35	0	1	681
29802	37	1	1	318
29836	71	2	1	225
29876	111	3	1	215
29827	62	4	1	273
29841	76	5	1	873
29598	-167	6	1	2675
29651	-114	7	1	6565
30078	313	8	1	10508
30636	871	9	1	10802
32979	3214	10	1	11226
35100	5335	11	1	10345
37688	7923	12	1	13849
41688	11923	13	1	13491
45683	15918	14	1	11297
46979	17214	15	1	11567
49145	19380	16	1	11548
49772	20007	17	1	6894
48867	19102	18	1	3771
48205	18440	19	1	3901
46839	17074	20	1	4211
45545	15780	21	1	4222
44710	14945	22	1	2958
44493	14728	23	1	1401

Table 5-2 shows five datasets that were prepared from the five different websites containing the daily and weekly differences. These datasets were prepared to run experiments using the MLP. We ran the MLP experiments with the following configuration: the minimum number of hidden units to add at one time is set to 1 and the maximum number of hidden units to add at one time is set to 2; evaluation mode was used to evaluate the performance of the generated model with data from the test set; Root Mean Power Error was chosen as the error metric.

During variable selection, the training and test sets were combined and then partitioned into cross-validation sets. The number of cross-validation folds was set to 3.

5.5.3 Experiment 3: Use of ARIMA Model for Prediction

ARIMA models are flexible and widely use in time series analysis. It combines three processes autoregressive (AR), differencing to strip off the integration (I) of the series and moving average (MA) (Sabry, 2007). ARIMA can capture complex patterns including three major types of processes: stationary, non-stationary and seasonal (Tran, 2001).

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation etc., are all constant over time. (Tran, 2001) describes that the stationary series represents a process in statistical equilibrium, with observations fluctuating around a fixed mean with constant variance, and covariances that depend only on the number of separating time steps.

A non-stationary series has no natural mean but tends to increase or decrease over time (Tran, 2001). Mostly in business, economic and finance time series, trend is produced by evolving preferences, technologies and demographics. Their behaviour could be upwards or backwards. Such kind of trending pattern is non-stationary.

Seasonal series capture periodicity (Tran, 2001). For example, retail sales tend to increase during September to December and then decline after the holidays.

ARIMA (p, d, q) Models attempt to describe the systematic pattern of a time series by three parameters namely (Sabry, 2007):

- **p** : Number of autoregressive terms (AR-terms) in a time series.
- **d** : Number of differences to achieve stationarity of a time series
- **q** : Number of moving average terms (MA-terms) in a time series.

Each of above process types has its own characteristic way of responding to a random disturbance. The ARIMA model parameters generally take values of either 1 or 2, while values greater than two are very rarely required in practice.

Data pre-processing

Microsoft SQL Server 2005 was used to read and extract data from the online traffic repository. SQL scripts were written to process three years worth of raw data. Raw dataset structure is already explained in section 5.2. Basically this data is collected at each hour and every record holds a timestamp. The timestamp is used to group the data by the week it occurs in the year. Five datasets were prepared for ARIMA prediction; the dataset for Website1 for year 2008 has been illustrated in Appendix B for reference. The hourly data was aggregated to the weekly and monthly levels of granularity in order to deduce the seasonal patterns that occur at these two levels.

XLSTAT tool requires the data that correspond to the time series for prediction. Important model parameters are p, d, q values; these are the orders of the model. *p* is order of the autoregressive part of the model. For eg: input as 1 for an ARIMA(0, 1, 2) model. *d* is a differencing order of the model. *q* is a order of the moving average part of the model. For eg: input 2 for a MA(2) model or for an ARIMA(1,1,2) model. Experiments will be carried out with different combinations of p, d and q.

5.5.4 Experiment 4: Use of Recurrent Networks for Traffic Prediction

The feed forward NN's have no loops and their output mainly depends on the input layer. But RNNs take into account the previous states as well as current states, which is desirable from a prediction point of view. RNNs have the capability to incorporate past experience due to internal recurrence (Ma, 2007). Due to the fact that the RNN differs from conventional feed forward networks and has potential to capture recurrent patterns, it was included in the experimentation for the purposes of comparison with the feed forward variety of neural networks (Ma, 2007).

MATLAB tool is used to implement RNN model for traffic prediction. Datasets that were prepared for Experiment 2 will be utilized for RNN analysis. Elman network '*newelm'* is used to run experiments. The Elman network differs from two layers network in that the first layer has recurrent connection. Elman networks are two layer back-propagation networks, with addition of a feedback connection from the output of the hidden layer to its input (Mathworks, 2010). Network is trained and the parameter *net.trainParam.*epochs, which is the

maximum number of times the complete data set may be used for training is set to 3000. Experiments use six hidden layer *tansig* (Tan-Sigmoid Transfer function) neurons and a single *logsig* (Log-Sigmoid Transfer function) output layer. For training purposes, 70% of data is set aside and the rest is used for testing.

5.5.5 Experiment 5: Use of Dynamic Evolving Neuro-Fuzzy Inference Systems for Capturing Evolving Patterns

Sony and Kasabov (2002) proposed a model called Dynamic Evolving Neural Fuzzy System which is similar to the Evolving fuzzy-neural network in some aspects. Basically, DENFIS model is based on Takagi-Sugeno fuzzy rules and fuzzy inference. It evolves through incremental, hybrid learning, and accommodate new input data, including new features, new classes etc., through local element tuning. This model has been applied to Mackay-Glass time series prediction. Kasabov (2002) concluded that DENFIS can effectively learn complex temporal sequences in an adaptive way and outperform some existing models such as Resource-allocation network (RAN), Evolving fuzzy-neural network (EFuNN) and Evolving Self-Organizing maps (ESOM). DENFIS is included in experimentation mainly for the sake of comparison with other neural network models that are involved in current research for the time-series prediction.

Neucom tool is used to implement Dynamic Evolving Neuro-Fuzzy Inference Systems for Capturing Evolving Patterns. Datasets that were prepared for Experiment 2 will be utilized for analysing DENFIS model. Experiments use the default parameters of Neucom tool like *epochs* is set to 2 and *number of nodes* (MofN) to 3.

5.6 Summary

This chapter focused on the plan of experiments. It illustrated various datasets, preparation of datasets, tools used in the research and performance metrics to be used to evaluate results.

Chapter 6 Research Findings

6.1 Introduction

The previous chapters have described the background behind online advertising, formulated research objectives, identified suitable methods for time series prediction, and designed experiments to test the effectiveness of the prediction methods to be used. The current chapter focuses on the results of the experimental study.

6.2 Findings from Experiment 1 (Use of Periodogram)

The periodogram was one of the earliest statistical tools for studying periodic tendencies in time series. We have chosen the MATLAB tool to run this experiment. In this experiment, we use the MATLAB fast Fourier transform (FFT) function to analyze the variations in online traffic activity. We used three years of hourly web traffic data for *five* popular websites as explained previously in Chapter 5. Figures 6-1 and 6-2 show network traffic for the two websites in a 24 hour time period. The traffic traces for the other three websites can be found in Appendix-A.

Figures 6-1 and 6-2 shows that the 24-hour time period encapsulating a day to be the first harmonic.

We observed a cyclic pattern in daily network traffic data. On a weekly scale, peaks were generally observed in the five working days, followed by a drastic reduction in traffic during the weekend, as shown in Figure 6-1. In a few instances, the opposite behaviour was observed, whereby traffic drops off during week days only to pick up during the weekends. The presence of a second harmonic suggests some asymmetry of network traffic across a 7 day time period.

We have also monitored hourly traffic data and observed that it steadily rises at the start of each day, reaches a peak during mid-day and collapses at the end of the day. The exact behaviour or traffic pattern of a website depends on the content of the website.

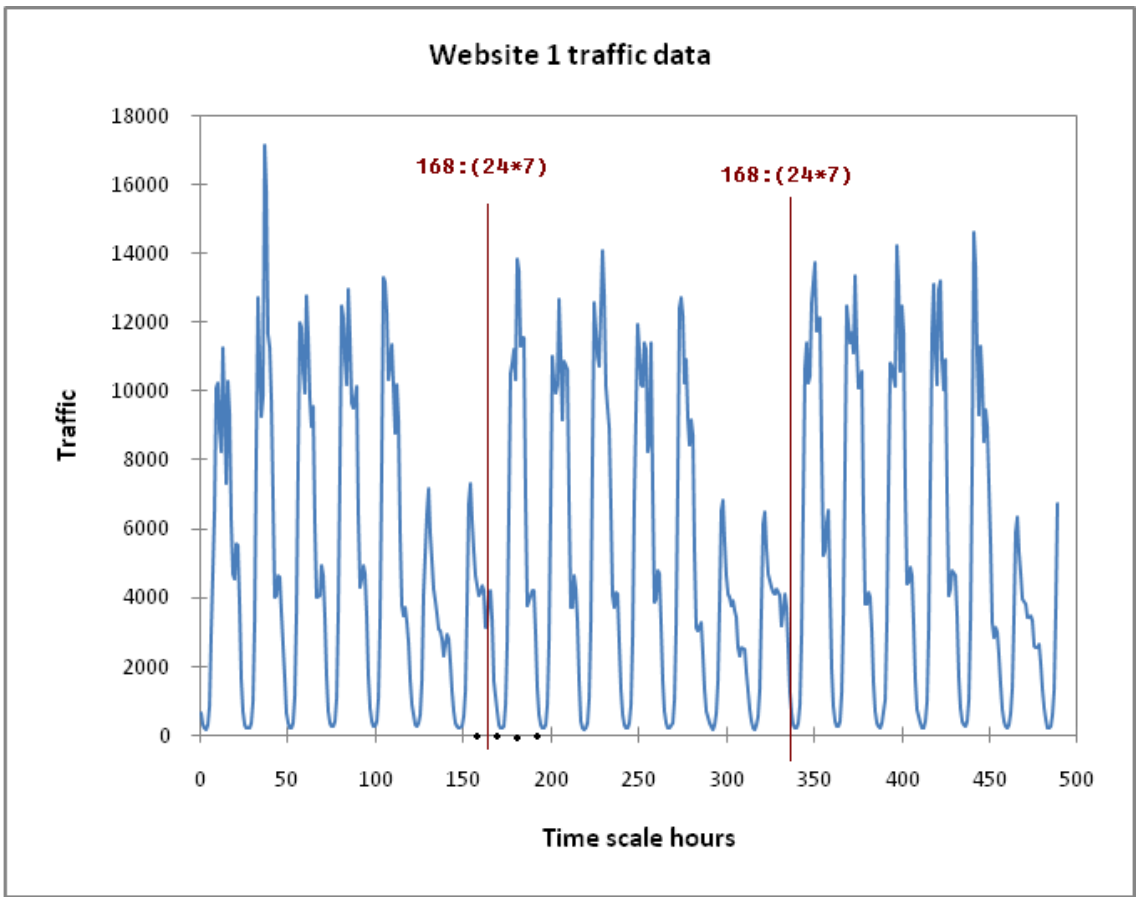


Figure 6-1: Website1 Hourly traffic trace (Experiment 1)

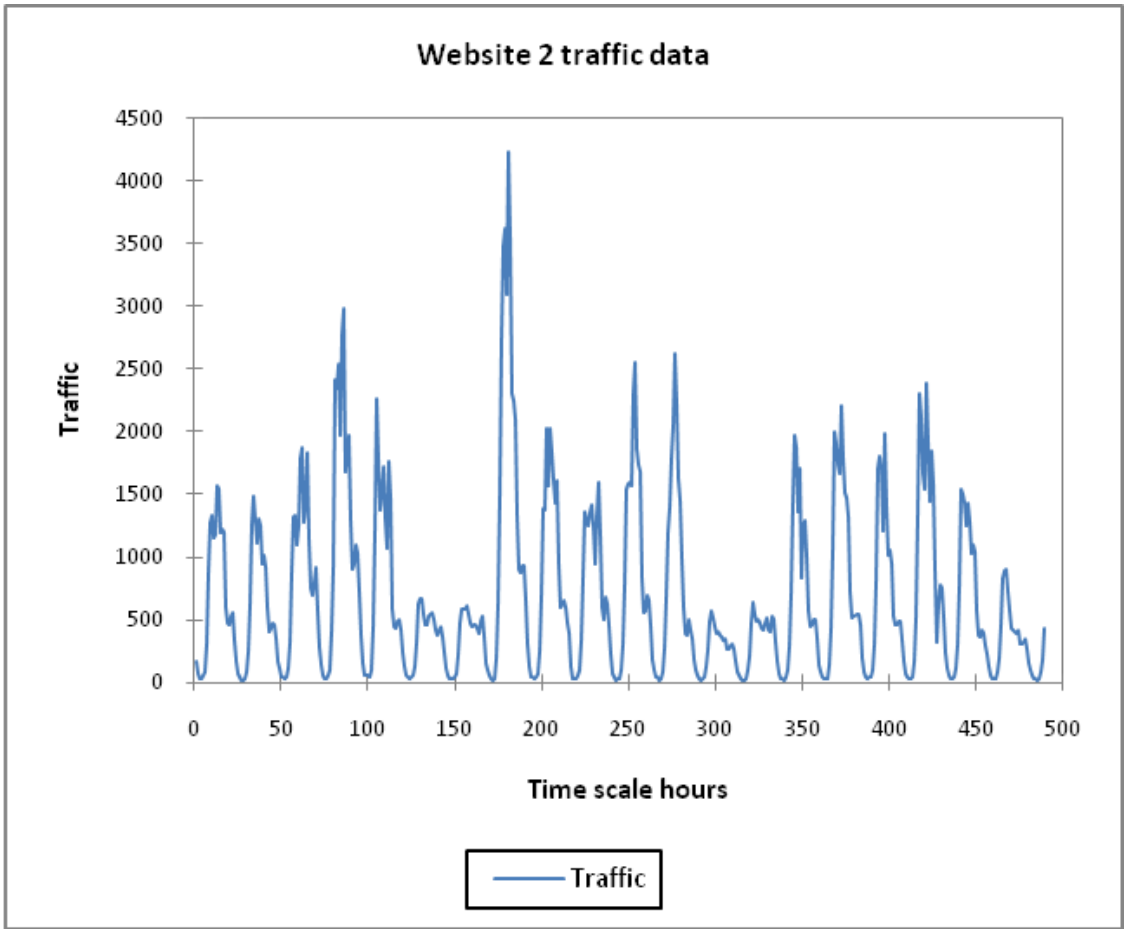


Figure 6-2: Website2 Hourly traffic trace (Experiment 1)

We used daily traffic volumes collected over a one month (30 days period) and the resulting periodograms for three of the websites are shown in Figures 6-3, 6-4 and 6-5. We started by taking the FFT of the website traffic data into vector Y which is given by:

Y = *fft*(trafficdata);

The result of this transform is the complex vector Y. The magnitude of Y squared is called the estimated power spectrum. A plot of the estimated power spectrum versus frequency makes up a *periodogram.* The plots in Figures 6-3, 6-4 and 6-5 confirm the cyclical nature of website traffic activity, which reaches a peak about every 7 days.

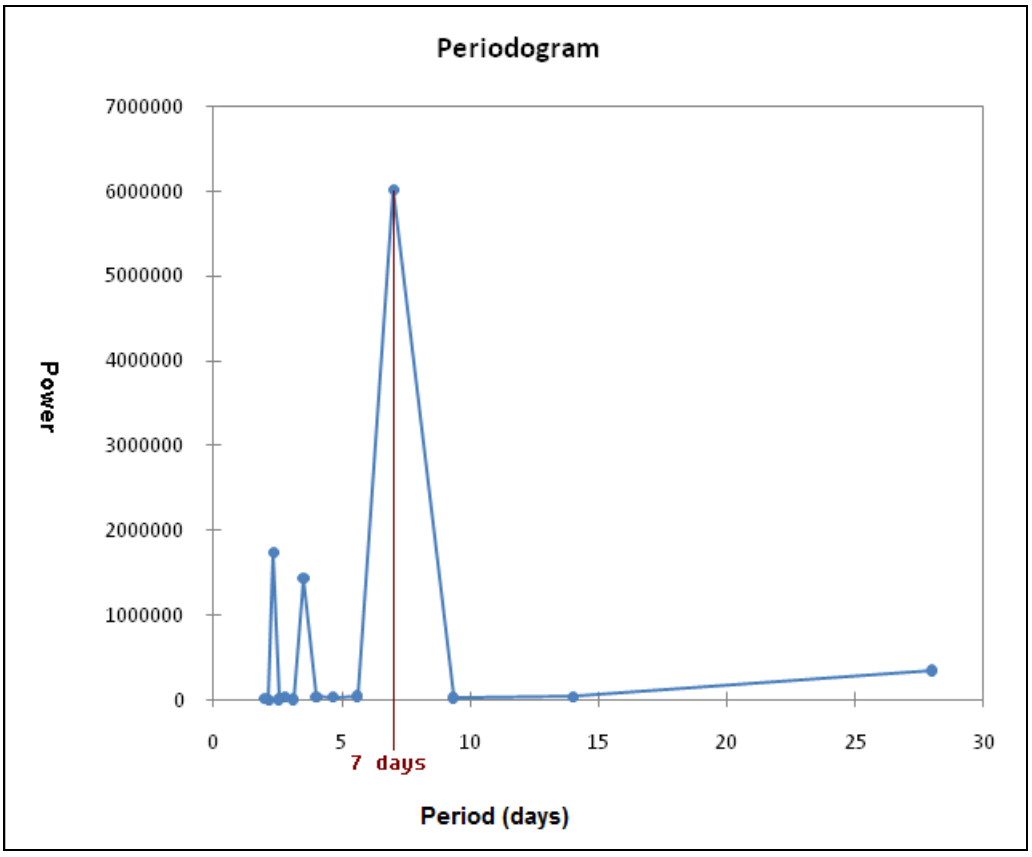


Figure 6-3: Website1 Daily traffic trace (Experiment 1)

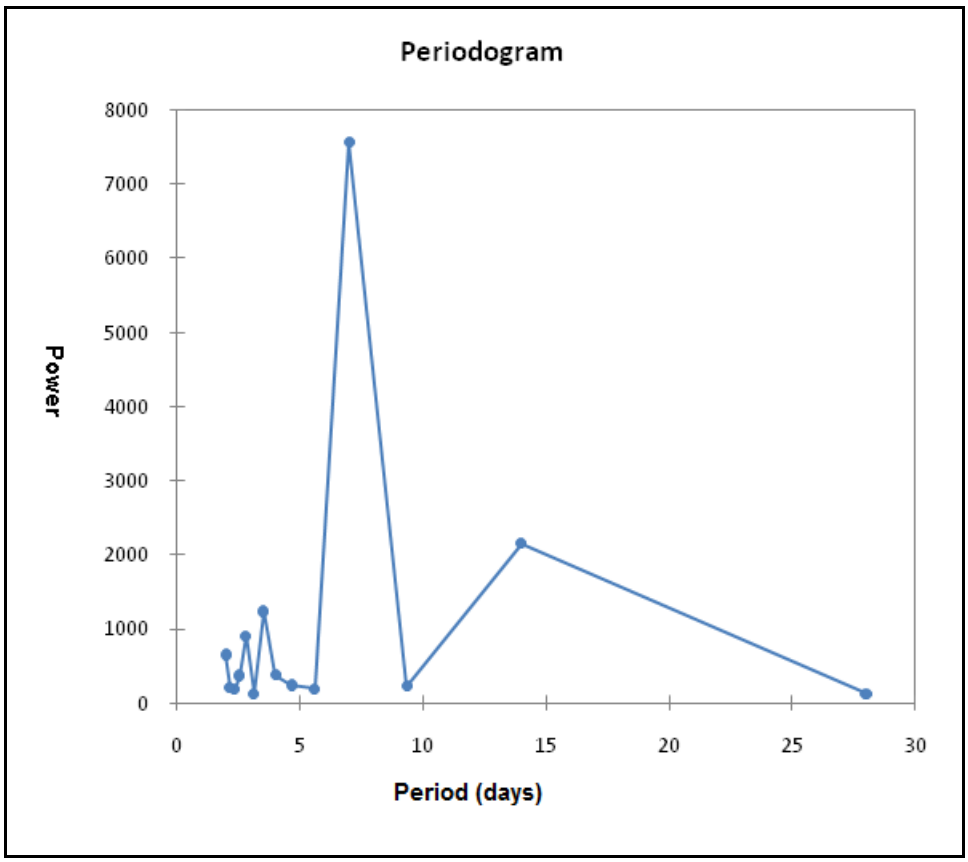


Figure 6-4: Website2 Daily traffic trace (Experiment 1)

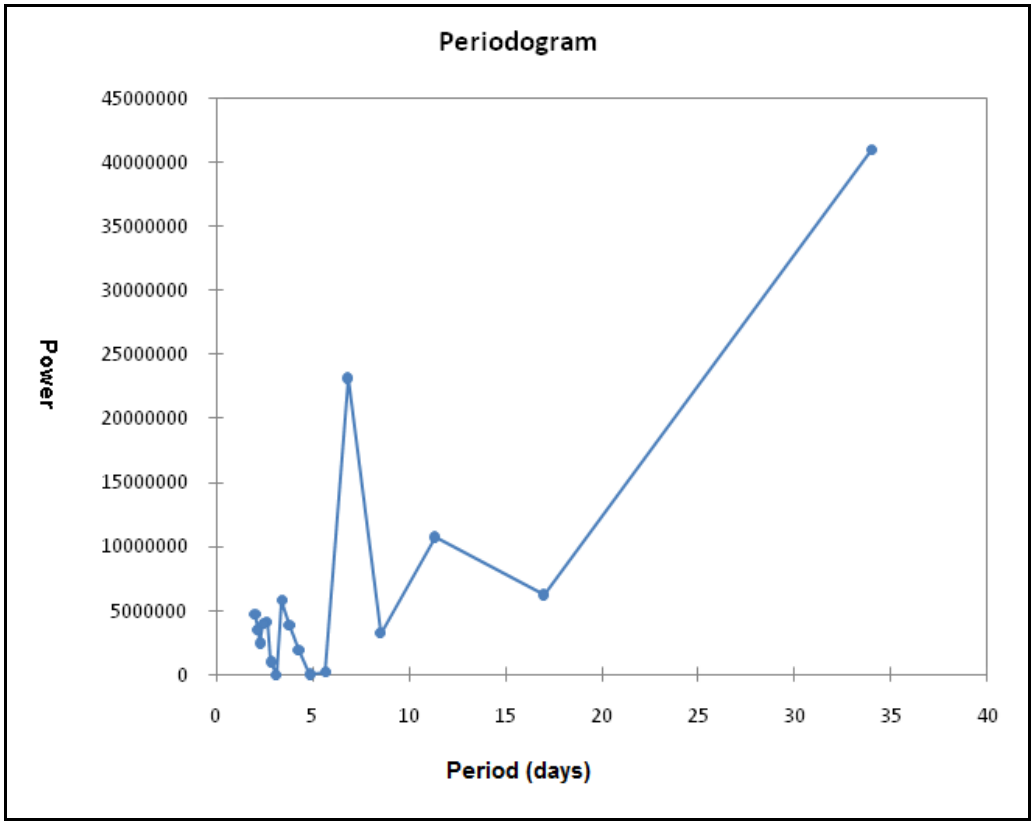


Figure 6-5: Website5 Daily traffic trace (Experiment 1)

The periodograms for the other two websites also showed that the first harmonic occurred at the 7 day mark. These experimental findings, together with plots of daily and monthly traffic volumes provided the empirical evidence needed to support the model that was presented in Section 5.5.2

6.3 Findings from Experiment 2 (Use of MLP for Prediction)

The results of periodogram analysis suggested that a prediction model based on daily and weekly differencing, as presented in Section 5.5.2 would be effective in predicting future traffic values.

MLP analysis was performed on datasets produced by the five different websites that we used in the research and their hourly predicted traffic values are closely examined. After careful inspection and discussion with domain experts, a minor discrepancy in predicted values for certain hours was detected, mainly for the hours from 10 pm to 12 am. This anomalous behaviour has been observed for all five websites. We have observed that the afternoon hourly predicted traffic values consistently underestimate the actual values. This has even influenced the predictions on the midday hours as well (Figures 6-6 & 6-7). Predicted values for Website1 are shown in Table 6-1.

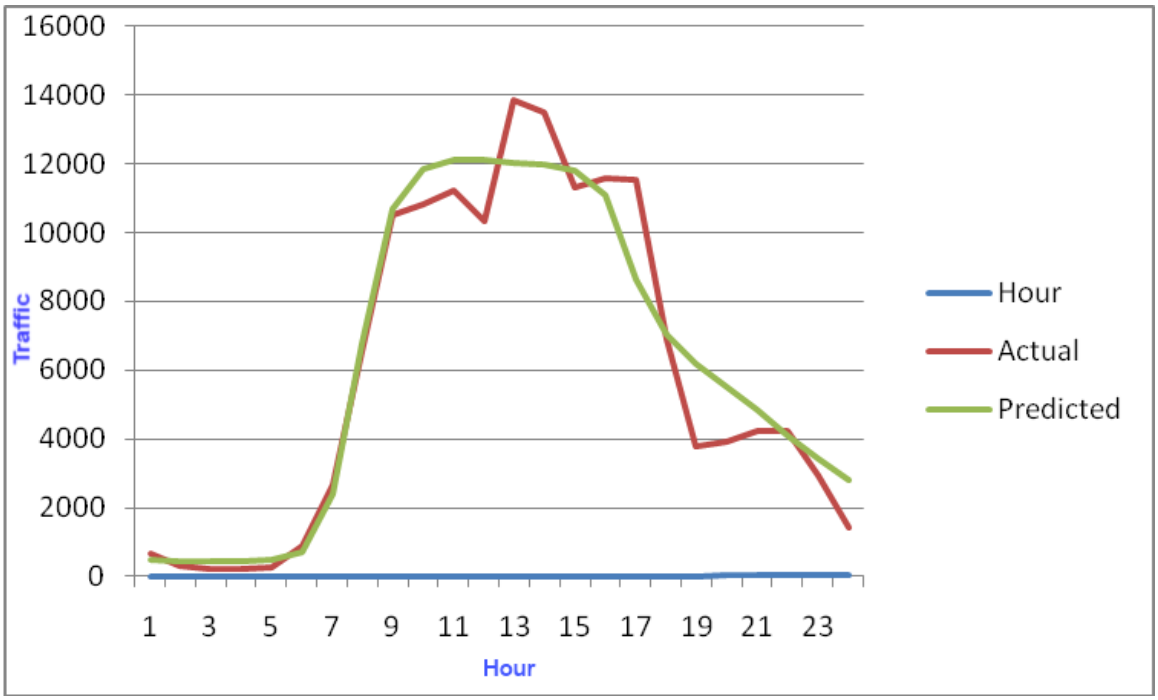


Figure 6-6: Website1 Monday's hourly traffic data prediction (Experiment 2)

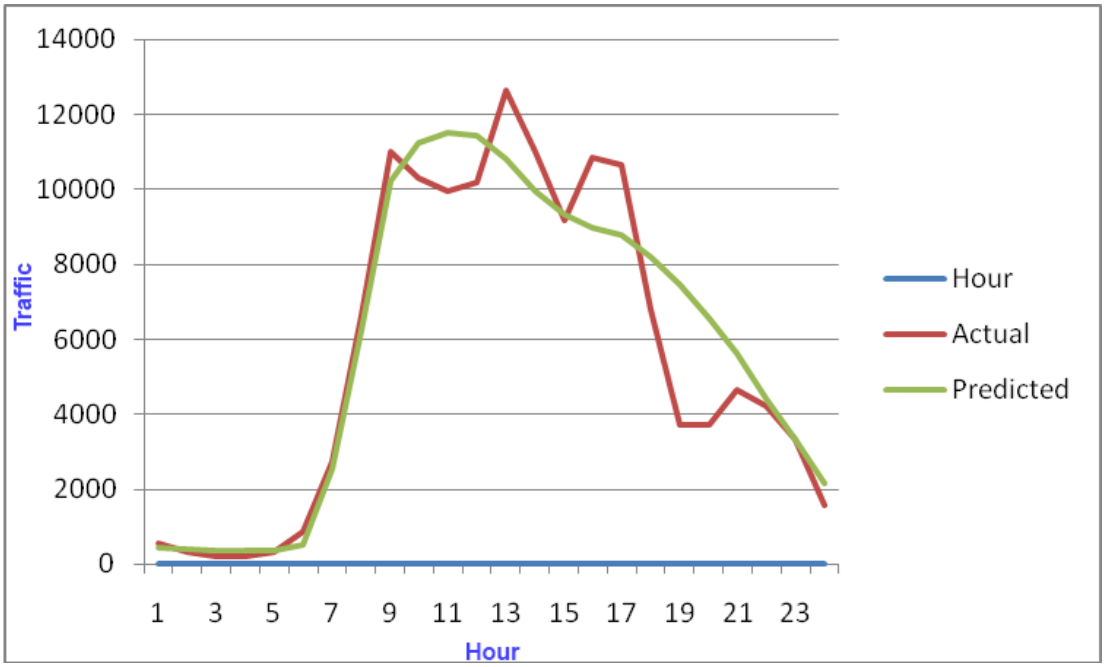


Figure 6-7: Website1 Tuesday's hourly traffic data prediction (Experiment 2)

This examination enabled us to recognize that certain day parts of the hourly traffic data were adversely influencing prediction on other parts of the day. This suggested that we should split the data into different day parts for further analysis. Our day split consisted of:

- 12 am to 7 am
- 8 am to 3 pm
- 4 pm to 11 pm

We ran MLP analysis separately on each day part for three times on five websites. We noted that the predicted values are now considerably more accurate than with the previous version of the dataset. These website data have an obvious pattern; they are significantly low in early morning, start to pick up during working hours and gradually slow down at night. Since we fed 24 hours data to the Neural Ware tool for MLP analysis, the high traffic data of certain hours have impact on the following hours in terms of prediction. Therefore, when we segregate the data into different day parts, the prediction becomes more accurate. Predicted traffic values for day parts are shown in Table 6-2 and their Root Mean Square values can be compared in Table 6-3. Predicted traffic values before day split for Website1 is shown in Table 6-1.

Table 6-1: MLP analysis (Website1 Monday's traffic prediction before day split) (Experiment 2)

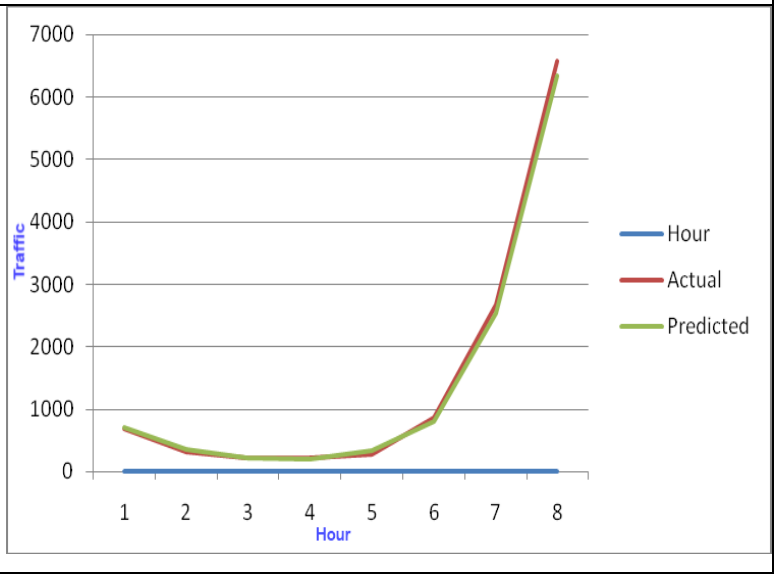
Hour	Day Of Week	Actual Traffic	Prediction
0	1	681	484.2423
1	1	318	452.7574
2	1	225	432.8721
3	1	215	429.9717
4	1	273	464.2106
5	1	873	701.2065
6	1	2675	2412.85
7	1	6565	6780.04
8	1	10508	10673.31
9	1	10802	11841.71
10	1	11226	12099.08
11	1	10345	12096.31
12	1	13849	12032.95
13	1	13491	11966.67
14	1	11297	11806.37
15	1	11567	11065.91
16	1	11548	8636.461
17	1	6894	7025.242
18	1	3771	6169.809
19	1	3901	5518.907
20	1	4211	4835.912
21	1	4222	4111.381
22	1	2958	3410.54
23	1	1401	2805.396

Table 6-2: MLP analysis on Monday's day parts of Website1 (Experiment 2)

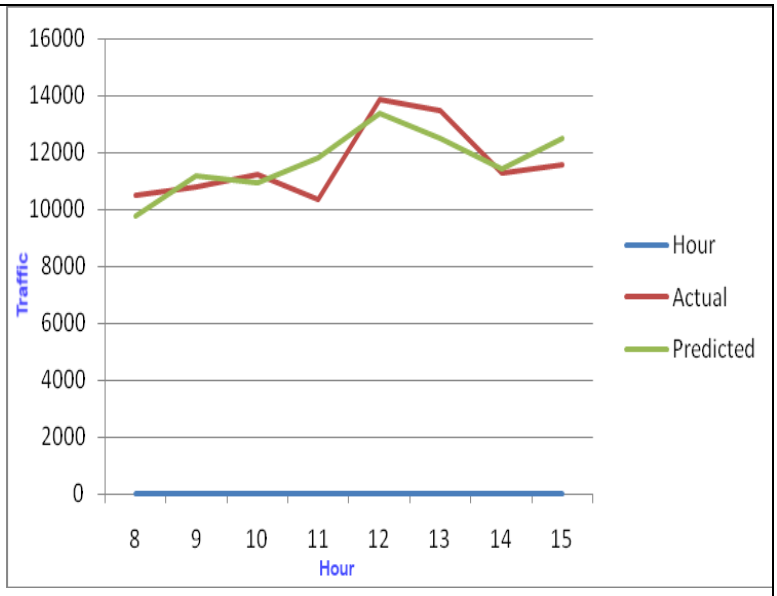
Day part data

Hour	Actual	Predicted
0	681	713
1	318	361
2	225	211
3	215	207
4	273	332
5	873	800
6	2675	2534
7	6565	6346

Graph



Hour	Actual	Predicted
8	10508	9754
9	10802	11188
10	11226	10909
11	10345	11801
12	13849	13375
13	13491	12480
14	11297	11435
15	11567	12501



Hour	Actual	Predicted
16	11548	10301
17	6894	5770
18	3771	4185
19	3901	5101
20	4211	5189
21	4222	4349
22	2958	3119
23	1401	1971

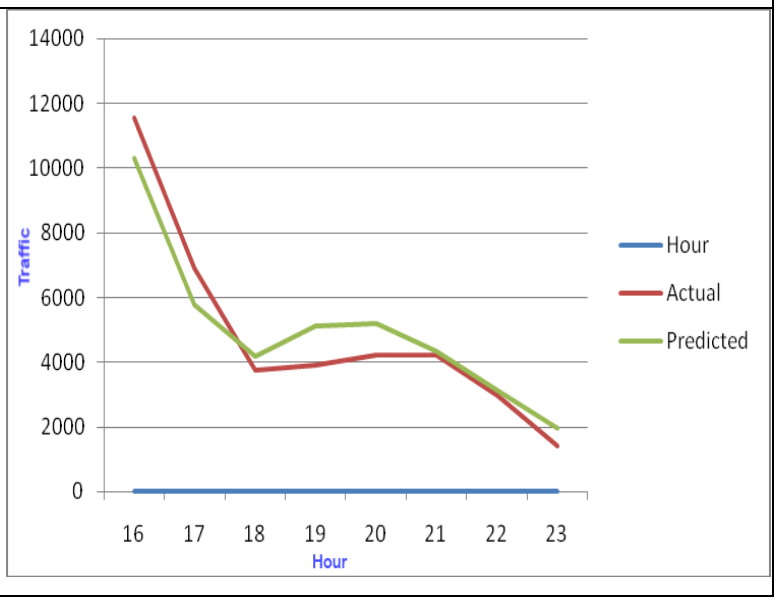


Table 6-3: Root Mean Square values for MLP analysis (Experiment 2)

RMS	Website1	Website2	Website3	Website4	Website5
Without – Day part data	1178.4	297.3	783.7	93.1	538.4
12am - 7am	410.9	53.4	132	34.8	70.5
8am – 3pm	1295.8	446.3	973	69.8	808.7
4pm – 11pm	1003.5	206.4	483	81.4	436.5

6.4 Findings from Experiment 3 (Use of ARIMA Model for Prediction)

Datasets prepared to support ARIMA analysis for five websites was illustrated in Chapter 5. We ran XLSTAT tool for ARIMA weekly traffic prediction. Our initial exploration with predicting hourly traffic with ARIMA showed that the predicting accuracy was much lower than that of MLP. Table 2 (Appendix B) shows the comparison of prediction values between the ARIMA and MLP analysis for

Website1 Hourly traffic data. Section 6.2 has already explained the improvement in MLP prediction when we segregate the data into different day parts. Hence, we tried using same day parts dataset for ARIMA analysis and results are compared against the MLP model results, shown in Table 3 (Appendix B). Since the prediction accuracy was much lower than the MLP, we thus decided to test the ARIMA's capability for longer term forecasting – i.e., at the weekly level of granularity.

We applied ARIMA modelling with different combinations of (p, d, q) values since they influence the prediction and the effects on accuracy were monitored. To fit the model, we supply the number of AR terms, MA terms and differencing to strip off the integration (I) of the series. Comments are tabularised in Table 6-4 and their graphs presented in Appendix-B.

Table 6-4: (p, d, q) Parameters influence in ARIMA prediction (Experiment 3)

p	d	q	Hessian Standard error	Comments on weekly prediction depending on the output values.
1	0	0	0.052	Inconsistent.
1	1	0	0.089	Good.
1	1	1		Failed as process is close to non-stationarity
1	0	1	0.049	Under estimated
0	0	1	0.056	Over estimated
0	1	1	0.215	Few values predicted exactly, some are abnormal.
1	2	0	0.078	Over estimated mostly, sometimes values are under estimated. Inconsistent behavior.

From Table 6-4 it appears that the ARIMA model (1, 1, 0) is the best. Its weekly predicted traffic values are closely matching the actual values (Figure 6-10). This is a first-order autoregressive, or "AR (1)", model with one order of non-seasonal differencing and a constant term i.e., an ARIMA (1, 1, 0) model with constant (Sabry, 2007).

$$\hat{Y}(t) = \mu + Y(t-1) + \phi\,(Y(t-1) - Y(t-2))$$

In this formula, the constant term is denoted by μ and the autoregressive coefficient is denoted by Φ. We have experimented with various different combinations of (p, d, q) values and finally concluded that ARIMA (1, 1, 0) model produced the best overall predictions. The main challenge and most

important step in fitting an ARIMA model is the determination of the order of differencing needed to stationarize the series. The optimal order of differencing is often the order of differencing at which the standard deviation is lowest (Decision411, 2010). Mild *under* differencing can be compensated for by adding AR terms to the model, while mild *over* differencing can be compensated for by adding MA terms (Decision411, 2010). Thus, we ran experiments with different combination of differencing, AR and MA terms for prediction and fulfilled that ARIMA (1, 1, 0) model prediction is superior.

Figure 6-8: Website1 ARIMA (1, 1, 0) weekly traffic prediction - Experiment 3

Seasonal prediction:

For seasonal prediction ARIMA additional attributes P, D and Q values for prediction apart from (p, d, q) is required, and the ARIMA(p,d,q)(P,D,Q) model is then represented by:

$$\begin{cases} Y_t = (1-B)^d \left(1-B^s\right)^D X_t - \mu \\ \phi(B)\Phi\left(B^s\right) Y_t = \theta(B)\Theta\left(B^s\right) Z_t, \quad Z_t \propto N(0,\sigma^2) \end{cases}$$

with

$$\begin{cases} \phi(z) = 1 - \sum_{i=1}^{p} \phi_i z^i, \quad \Phi(z) = 1 - \sum_{i=1}^{P} \Phi_i z^i \\ \theta(z) = 1 + \sum_{i=1}^{q} \theta_i z^i, \quad \Theta(z) = 1 + \sum_{i=1}^{Q} \Theta_i z^i \end{cases}$$

D is the differencing order of the seasonal part of the model.

s is the period of the model.

P is the order of the autoregressive seasonal part of the model.

Q is the order of the moving average seasonal part of the model.

μ is the mean term

B is the backshift operator, i.e, $BX_t = X_{t-1}$

Z is the vector of lagged regressors $(Z_{t-1}, Z_{t-2},\ldots, Z_{t-p})$

We implemented the model with following parameter combination: p=0, d=1, q=1, P=0, D=1, Q=1 and s=12 for seasonal monthly prediction. However, the prediction is not as accurate as expected as it requires a long range of data to capture the annual seasonal effect. Since the dataset is holding only three years data, it is not possible to train the data properly to extract the seasonal effect.

6.5 Findings from Experiment 4 (Use of Recurrent Networks)

MATLAB has been used to implement RNN as explained in Chapter 5. Prediction for Website 4 weekly traffic data is shown in Figure 6-9. Unfortunately, the output predicted values are abnormal. We ran this experiment against datasets from other websites and their behaviour is also similar; the relevant figures are listed in Appendix-C for reference. We also experimented with hourly level traffic data. These predictions too were of poor quality, as seen in Figure 6-10. Weekly and hourly traffic prediction using the RNN model was not effective for any of the websites that we experimented against.

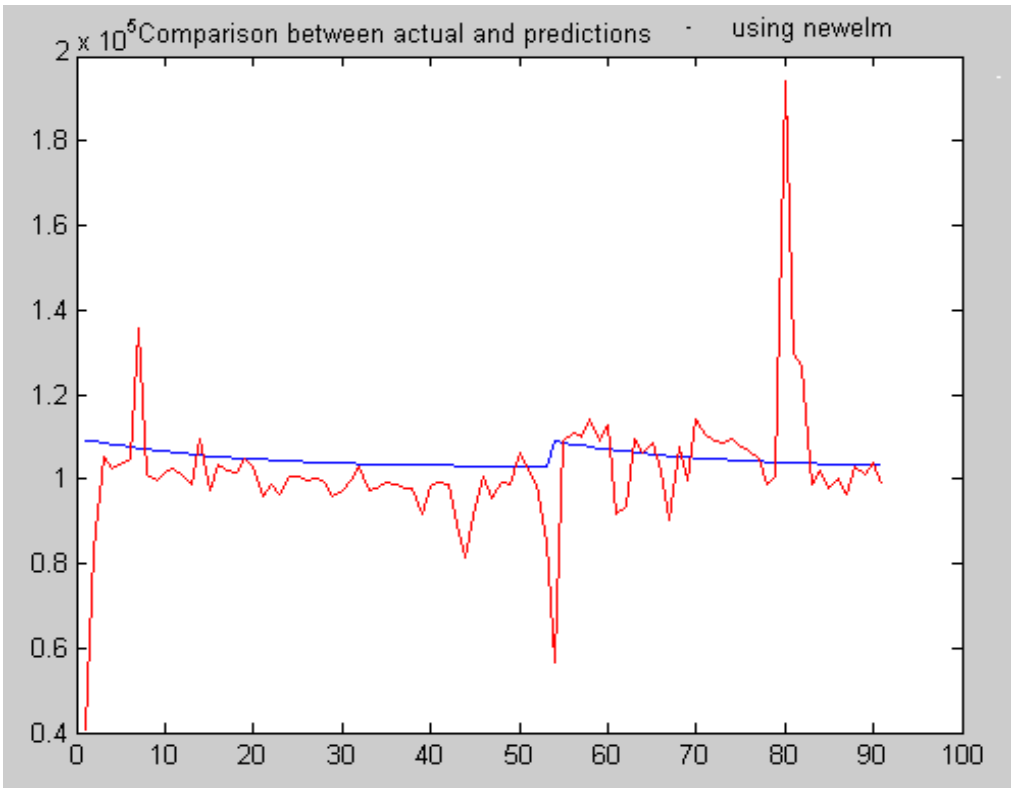


Figure 6-9: Website4 RNN's weekly prediction – Experiment 4

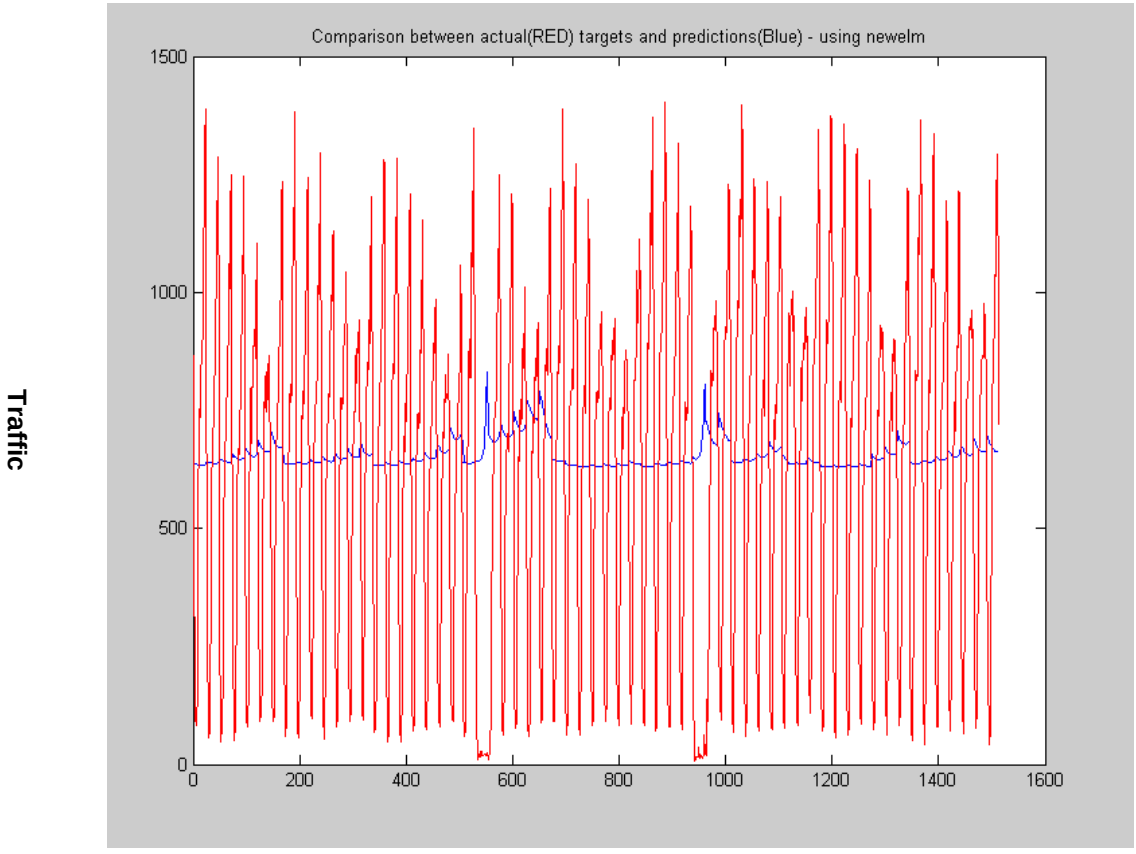


Figure 6-10: Website4 RNN's hourly prediction – Experiment 4

6.6 Findings from Experiment 5 (Use of DENFIS)

Neucom tool has been used to implement DENFIS as explained in Chapter 5. Figure 6-11 shows the hourly prediction of Website1. The *Desired* symbol in the figure indicates the predicted output and *Actual* indicates the actual traffic values. Predicted traffic values are overestimated in a few cases and underestimated in others. Prediction is satisfactory but quite as accurate as MLP for the hourly traffic prediction scenario. Table 6.5 shows that the RMS errors for websites 1 and 2 were slightly higher than that of MLP for the same websites (as given in Table 6-3). We experimented with other websites and the results were similar (see Appendix-C). We also used different values for parameters such as the number of epochs, number of nodes, etc., but no significant difference in results was observed.

However DENFIS has the capability to evolve its prediction model incrementally when patterns change in a dynamic fashion, unlike MLP which suffers from the problem of catastrophic forgetting. We note that the datasets used in this research has quite regular patterns and this explains why MLP was able to produce relatively accurate results. On the other hand we would expect DENFIS to outperform MLP on time series data that exhibits more chaotic behaviour.

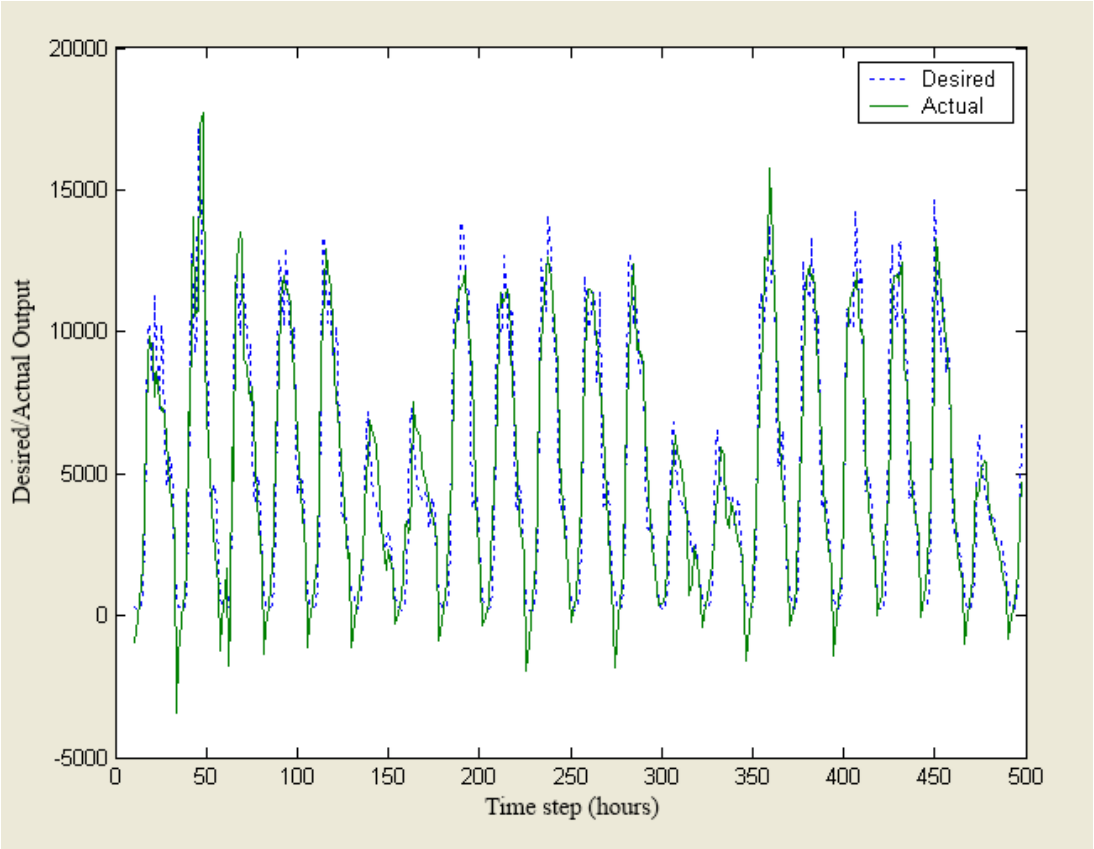


Figure 6-11: Website1 DENFIS hourly prediction – Experiment 5

Table 6-5: Root Mean Square values for DENFIS model (Experiment 5)

Website name	RMS
Website 1	1508
Website 2	382

6.7 Summary

This chapter presented the research findings together with an analysis of the comparative strengths and weaknesses of the 4 prediction methods that we investigated. The results showed that the MLP technique has performed well in predicting hourly traffic data only when data was segregated into different day segments. ARIMA model performed well in terms of weekly traffic prediction. DENFIS was quite versatile and performed equally well at both daily and weekly prediction time horizons.

Chapter 7 Conclusions and Future Work

This research addressed the issue of predicting traffic volumes for online advertising. From the research literature on time series prediction, four main methods were identified as suitable candidates for traffic prediction in the online advertising environment. Our study recognized that the selection of the prediction method depends on several factors such as the availability of historical data, time horizon to be forecasted, and the context of forecasts.

The main objective of this research was to observe how a well established statistical method such as the ARIMA model performs in comparison to computational methods such as the MLP, DENFIS and Recurrent neural networks in traffic prediction.

The research showed clearly that neural network methods (with the exception of the Recurrent Neural Network) outperformed the ARIMA method for short-term forecasting. One possible reason for this is that there is much more variability present in daily data when compared to weekly data. Statistical methods such as ARIMA operate best with stationary time series which explains why ARIMA underperformed on daily data but did better on the more stable weekly datasets. On the other hand neural network methods such as the MLP and DENFIS are better equipped to deal with data that exhibit a high degree of variance as they can capture highly non-linear patterns.

In the online advertising environments both short-term and long term forecasting are equally valuable in the management of the advertising budget and hence this research recommends that either the MLP or DENFIS methods be used for forecasting as they both performed well across the two different time horizons. Our preference is for DENFIS over MLP as DENFIS has two important advantages over its MLP counterpart. First, it has the ability to evolve its model and adapt to changes in traffic patterns. Second, it provides knowledge in the form of rules which can be easily understood by management unlike the black box output produced by the MLP.

The one unexpected finding in this research was the very poor performance of the Recurrent Neural Network. Despite extensive experimentation which involved tuning various parameters, the predictions produced were consistently poor in terms of prediction accuracy, no matter which time horizon (daily or weekly) was used. One possible area for future investigation is to investigate whether a different implementation of the RNN in a more robust machine learning toolkit would produce better results.

Future Work

Although much work remains to be done, this study helped to identify suitable techniques for online traffic prediction. The current research was unable to capture seasonal trends due to insufficient volume of data. However, future work should focus on identifying websites whose traffic is influenced by seasons; the main focus is to identify the intensity and its duration.

Earlier Chapters have discussed the presence of spikes in the data; some of them are due to unknown or irregular events. As events influence the forecast process, our focus should be to investigate a mechanism that could identify these events behavior from historical data and guide the prediction process accordingly.

Recurrent neural networks have the capability to incorporate the past experience due to internal recurrence. In fact, RNNs are computationally more powerful than the feed-forward networks, and produce valuable approximation results for chaotic time series prediction (Ma, 2007). However, current research was unable to produce productive results with RNN. Therefore, we would like to work on evolving recurrent neural network such as DENFIS and (ERNN) proposed by (Ma, 2007) in our future work. ERNN is designed according to evolution of the reconstruction phase space and parameters of recurrent network structure.

In practice, current domain network traffic data shows cyclical characteristics. They are larger and shorter cyclical effects on the daily, weekly, monthly traffic data. We have monitored hourly traffic data and observed that it steadily rise at

the start of each day, reaches a peak during mid-day and collapse at end of the day. Similarly, we have observed a cyclic pattern in daily network traffic data. On a weekly scale peaks were generally observed in the five working days, followed by a drastic reduction in traffic during the weekend. So, another area of our future work would be to implement a forecasting model proposed by Zhao. (Zhao, 2004) research includes two models namely *load shape forecasting model* and *peak-load forecasting model.*

In load shape forecasting model, it takes the previous day's actual load pattern as a basis to predict the present day's load (Zhao, 2004). On other hand, the peak-load model only forecast the daily peak load. Zhao experimented load shape with 20, 30 and 40 neurons in hidden layer. They choose 15, 20 and 30 neurons in hidden layer in peak-load forecasting model. Current research has done similar experiments by splitting the data in different day parts which was implemented using MLP model. However, we did not choose different neurons in hidden layer in our experimentation. Hence, it would be worthwhile to test with different neurons to see whether Zhao results can be duplicated in the online advertising traffic domain.

In a review session with domain experts, a potential problem was been discovered which was to do with the prediction of traffic for a specific month of a year i.e., December. In general, to predict the traffic of the subsequent week, the research suggested that the past 12 to 15 weeks data should be used as input. However, for the festive period, the prediction process should not include preceding month data for forecasting. Domain experts proposed considering last year December month data as input data to predict the current year December traffic. This is another possibility to look into in our future work.

Appendix A

Experiment 1 (Section 6.2)

Periodogram figures

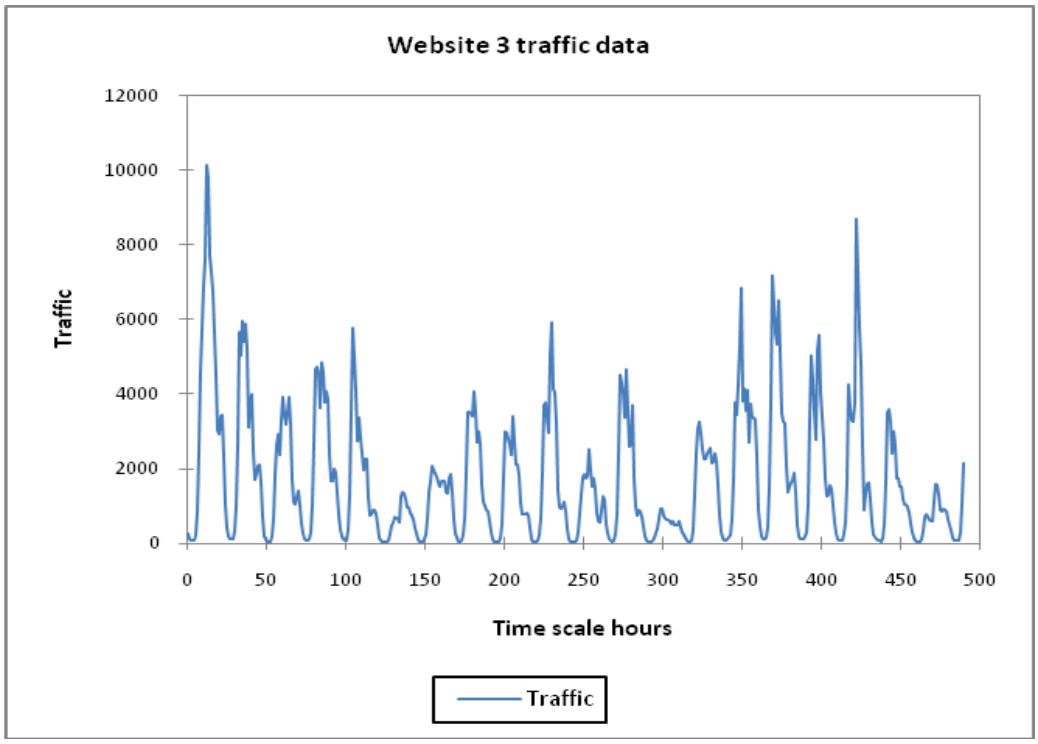


Figure 1: Website3 Hourly traffic trace (Experiment 1)

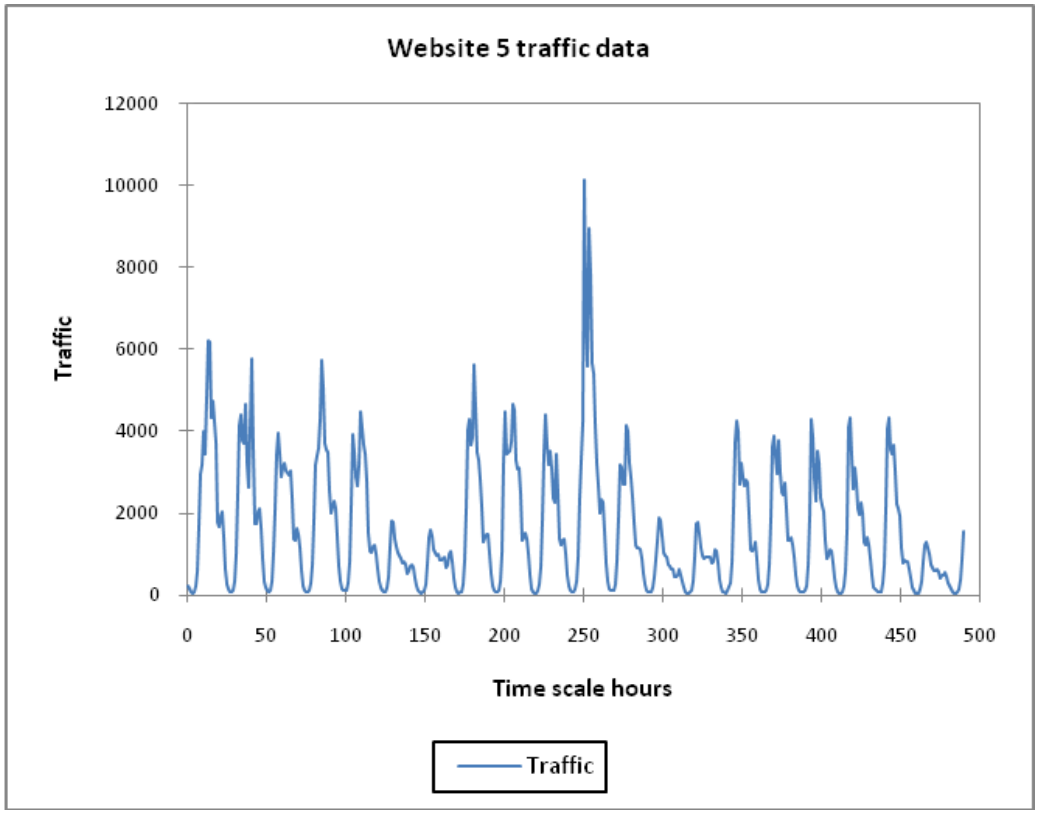


Figure 2: Website5 Hourly traffic trace (Experiment 1)

Appendix B

Experiment 3

2007 year data			2008 year data			2009 year data		
Year	Week	Actual Traffic	Year	Week	Actual Traffic	Year	Week	Actual Traffic
2007	6	246077	2008	1	548232	2009	1	257671
2007	7	419895	2008	2	644453	2009	2	750517
2007	8	442465	2008	3	690044	2009	3	857816
2007	9	437801	2008	4	693161	2009	4	861878
2007	10	412320	2008	5	653578	2009	5	806237
2007	11	426206	2008	6	891806	2009	6	813727
2007	12	401692	2008	7	692104	2009	7	896763
2007	13	470920	2008	8	664234	2009	8	707804
2007	14	378755	2008	9	636790	2009	9	692639
2007	15	378521	2008	10	619911	2009	10	849229
2007	16	428879	2008	11	675526	2009	11	827250
2007	17	446316	2008	12	580321	2009	12	863825
2007	18	469703	2008	13	566916	2009	13	821873
2007	19	488677	2008	14	668454	2009	14	674076
2007	20	501557	2008	15	606251	2009	15	722137
2007	21	470261	2008	16	744312	2009	16	687879
2007	22	490530	2008	17	632090	2009	17	829106
2007	23	449093	2008	18	709664	2009	18	928078
2007	24	481656	2008	19	677002	2009	19	1079053
2007	25	540198	2008	20	666080	2009	20	938682
2007	26	477128	2008	21	658736	2009	21	919951
2007	27	503190	2008	22	618345	2009	22	941997
2007	28	539390	2008	23	609685	2009	23	939644
2007	29	523544	2008	24	718996	2009	24	966842
2007	30	502662	2008	25	706117	2009	25	915102
2007	31	509620	2008	26	683358	2009	26	947396
2007	32	577302	2008	27	727384			
2007	33	567963	2008	28	720418			
2007	34	570068	2008	29	794159			
2007	35	511853	2008	30	869827			
2007	36	526253	2008	31	810557			
2007	37	542618	2008	32	682923			
2007	38	605469	2008	33	698582			
2007	39	587595	2008	34	749476			
2007	40	524775	2008	35	732232			
2007	41	556024	2008	36	775472			
2007	42	606726	2008	37	829514			
2007	43	515057	2008	38	683114			

2007	44	544661	2008	39	825287
2007	45	525730	2008	40	788276
2007	46	520967	2008	41	778939
2007	47	523610	2008	42	712680
2007	48	566570	2008	43	585740
2007	49	565973	2008	44	656740
2007	50	565136	2008	45	865322
2007	51	570525	2008	46	727706
2007	52	342322	2008	47	783885
			2008	48	807929
			2008	49	698908
			2008	50	741395
			2008	51	718080
			2008	52	497565

Table 1: Dataset of Website1 for Weekly traffic Prediction (using ARIMA)

Experiment 3 ARIMA figures

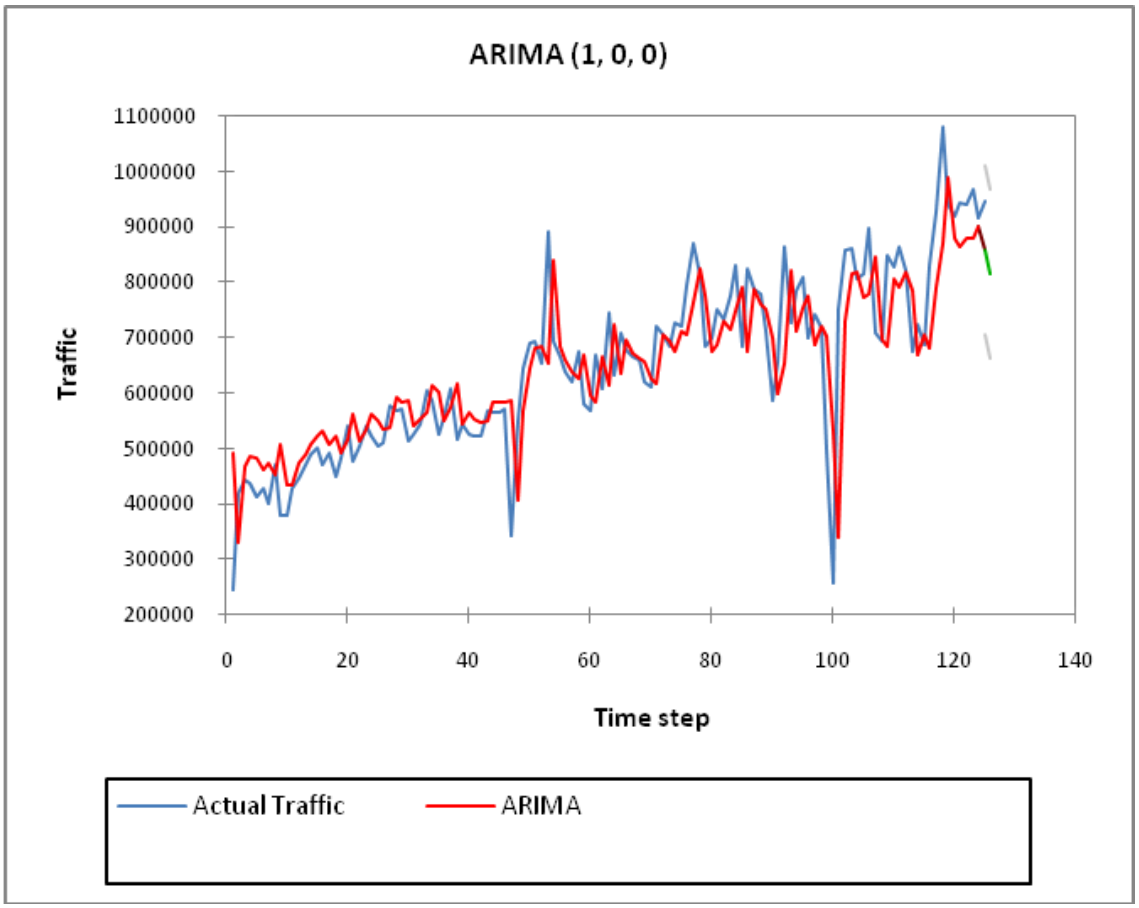


Figure 1: Website1 ARIMA (1, 0, 0) weekly traffic prediction - Experiment 3

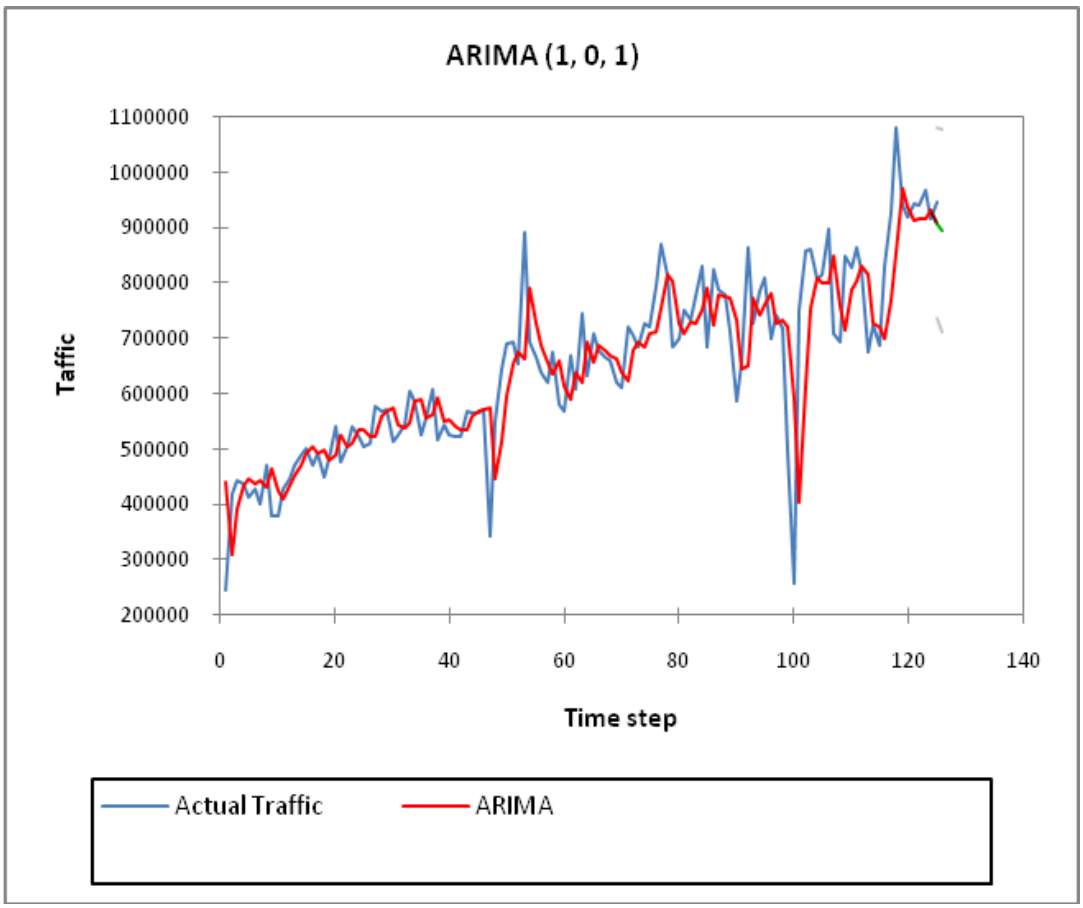


Figure 2: Website1 ARIMA (1, 0, 1) weekly traffic prediction - Experiment 3

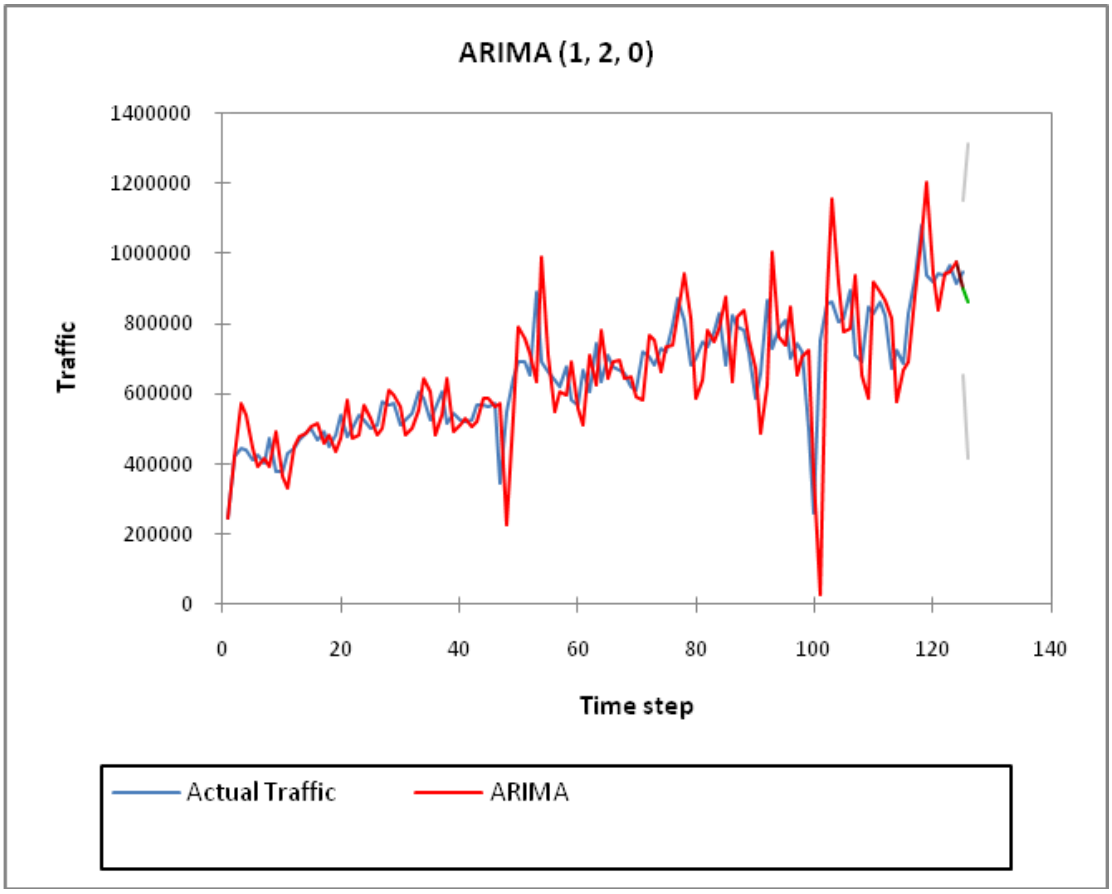


Figure 3: Website1 ARIMA (1, 2, 0) weekly traffic prediction - Experiment 3

Website 1 Hourly Traffic Prediction (ARIMA vs MLP)

Hour	Day Of Week	Actual	Prediction using ARIMA	Prediction using MLP
0	1	681.000	1853.793	484.2423
1	1	318.000	1060.853	452.7574
2	1	225.000	729.625	432.8721
3	1	215.000	644.765	429.9717
4	1	273.000	635.640	464.2106
5	1	873.000	688.563	701.2065
6	1	2675.000	1236.048	2412.85
7	1	6565.000	2880.328	6780.04
8	1	10508.000	6429.854	10673.31
9	1	10802.000	10027.742	11841.71
10	1	11226.000	10296.009	12099.08
11	1	10345.000	10682.899	12096.31
12	1	13849.000	9879.008	12032.95
13	1	13491.000	13076.320	11966.67
14	1	11297.000	12749.654	11806.37
15	1	11567.000	10747.684	11065.91
16	1	11548.000	10994.053	8636.461
17	1	6894.000	10976.716	7025.242
18	1	3771.000	6730.058	6169.809
19	1	3901.000	3880.400	5518.907
20	1	4211.000	3999.022	4835.912
21	1	4222.000	4281.889	4111.381
22	1	2958.000	4291.926	3410.54
23	1	1401.000	3138.558	2805.396
			ARIMA analysis Standard deviation: 4229.557	MLP analysis RMSE: 1178.4

Table 2: Monday's Hourly traffic Prediction of Website 1 (ARIMA vs MLP)

Hour	Actual	Prediction using ARIMA	Prediction using MLP
0	681	3186.377	713
1	318	1393.063	361
2	225	402.031	211
3	215	246.529	207
4	273	217.315	332
5	873	259.574	800
6	2675	734.106	2534
7	6565	2257.855	6346
		ARIMA analysis Standard deviation: 2131.650	MLP analysis RMSE: 410.9

Table 3: Website1 Hourly traffic prediction – day split (0 to 7am)

Appendix C

Experiment 4

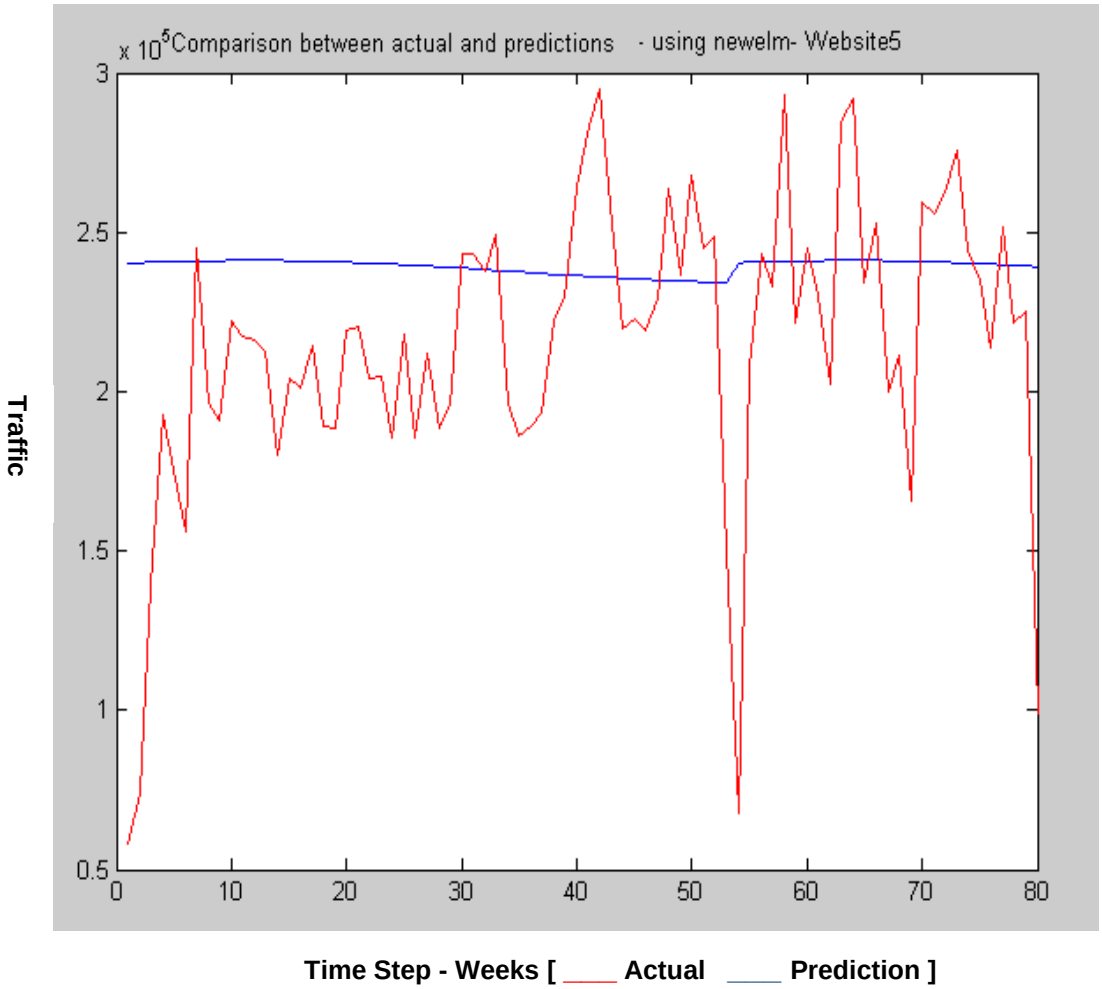


Figure 1: Website 5 RNN's weekly prediction – Experiment 4

Experiment 5

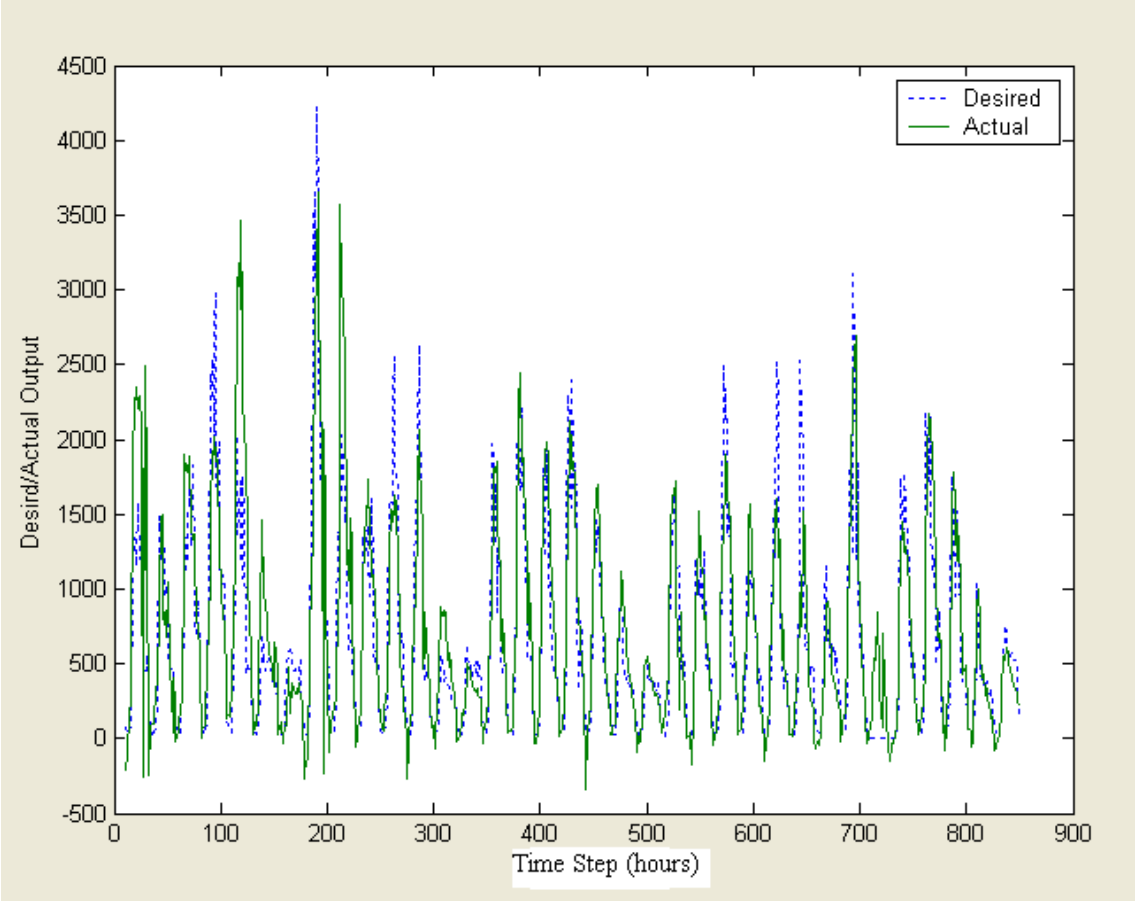


Figure 2: Website2 DENFIS hourly prediction – Experiment 5

References

Abe, H., Yokoi, H., Ohsaki, M., & Yamaguchi, T (2007). "Developing an Integrated Time-Series Data Mining Environment for Medical Data Mining." Paper presented at Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference(28-31 Oct. 2007): 127.

AUT. "Neucom Home page." Retrieved Dec 22, 2010, from http://www.aut.ac.nz/research/research-institutes/kedri/research-centres/centre-for-data-mining-and-decision-support-systems/neucom-project-home-page#download.

Bifet, A., Kirkby, R (2009). "Data Stream Mining A Partical Approach." Centre for Open Software Innovation.

Bloomfield, P. (1976). Fourier Analysis of Time Series: An Introduction, John Wiley & Sons Inc

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C (1994). Time series analysis – Forecasting and control. 3rd edition. NJ, USA, Prentice Hall.

Chabaa, S., Zeroual, A & Antari, J (2009). "ANFIS Method for Forecasting Internet Traffic Time Series." IEEE(15-17 Nov. 2009): 1-4.

Chakraborty, K., Mehrotra, K., Mohan, C., & Ranka, S (1992). "Forecasting the Behavior of Multivariate Time Series using Neural Networks." IEEE Transactions on Neural Networks 5: 961-970.

Chen, L., Lin, G (2008). "Extending Sliding-Window Semantics over Data Streams." Paper presented at Computer Science and Computational Technology, 2008. ISCSCT '08. 2.

Decision411. Retrieved Dec 22, 2010, from http://www.duke.edu/~rnau/411arim2.htm.

Dunham, M. H. (2003). Data Mining Introductory and Advanced Topics. New Jersey, USA, Pearson Education, Inc.

Easton, V. J., Mc Coll, J. H. . (1997). "Statistics Glossary." Retrieved Dec 21, 2010, from http://www.stats.gla.ac.uk/steps/glossary/time_series.html#diff.

Easycalculation. Retrieved Dec 22, 2010, from http://www.easycalculation.com/statistics/learn-standard-deviation.php.

Fayyad, U., Piatetsky, S. G & Smyth, P (1996). From Data Mining to Knowledge Discover in Databases. Cambrige, AAAI Press.

Frank, R. J., Davey, N & Hunt, S. P (1997). "Time Series Prediction and Neural Networks." Journal of Intelligent and Robotic Systems: 91-103.

Gerard, D. (2002). Neural Network Methodology and Applications; 1st edition., Springer.

Geurts, M. D. I., I. B (1975). "Comparing the Box-Jenkins Approach with the Exponentially Smoothed Forecasting Model." Journal of Marketing Research.

Geva, A. B. (1998). "ScaleNet – Multiscale Neural-Network Architecture for Time Series Prediction." Paper presented at Electrical and Electronics Engineers in Israel, 1996., Nineteenth Convention of 243 - 246

Gluszek, A., Kekez, M., & Rudzinski, F (2005). "Web traffic prediction with artificial neural networks." International Society for Optical Engineering 5775: 520-525.

Han, J., Kamber, M (2001). Data Mining: Concepts and Techniques. San Fransico, CA, Morgan Kaufmann.

Hand, D., Mannila, H and Smyth, P (2001). Principles of Data Mining, The MIT Press.

He, T., Dong, Z., Meng, K & Wang, H (2009). "Accelerating Multi-Layer Perceptron based Short Term Demand Forecasting Using Graphics Processing Units." Paper presented at Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009 1-4.

Jenkins, G., Watts, D (1968). Spectral Analysis and Its Applications, New York: Holden-Day.

Kajitani, Y., Hipel, K. W & McLeod, A. I (2005). "Forecasting Nonlinear Time Series with Feed-Forward Neural Networks: A Case Study of Canadian Lynx Data." Journal of Forecasting.

Kasabov, N., Song, Q and Nishikawa, I (2003). "Evolutionary Computation for Dynamic Parameter Optimisation of Evolving Connectionist." Journal Neural Networks 21(9).

Khotanzad, A., Nayera, S (2003). "Multi-Scale High-Speed Network Traffic Prediction using combination of neural networks." Paper presented at Neural Networks, 2003. Proceedings of the International Joint Conference on 2: 1071 - 1075.

Kim, M., Kim, Y., Sung, S and Yoo, Ch (2009). "Data-Driven Prediction Model of Indoor Air Quality by the Preprocessed Recurrent Neural Networks." Paper presented at ICCAS-SICE, 2009 1688 - 1692.

Liu, C., Wu, K & Tsao, M (2005). "Energy efficient information collection with the ARIMA model in wireless sensor networks." Paper presented at Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE 5: 2470--2474.

Ma, Q., Zheng, Q., Peng, H et. Al (2007). "Chaotic Time Series Prediction Based on Evolving Recurrent Neural Networks." Paper presented at Machine Learning and Cybernetics, 2007 International Conference on 6: 3496 - 3500

Mathworks. (2010). Retrieved Dec 22, 2010, from http://www.mathworks.com/.

Microsoft. Retrieved Dec 22, 2010, from http://www.microsoft.com/sqlserver/2005/en/us/features.aspx.

Mitra, S., Acharya, T (2003). Data Mining: Multimedia, Soft Computing, and Bioinformatics, Hoboken, New Jersey, John Wiley & Sons, Inc.

Mobasher, B. (1997). "A Taxonomy of Web Mining." Retrieved 2 May, 2008, from http://maya.cs.depaul.edu/~mobasher/webminer/survey/node6.html

Mohammed, W. K. (2002). "Recurrent Neural Networks " Retrieved 22-Dec, 2010, from http://www.cse.unsw.edu.au/~waleed/phd/html/node37.html.

Nagpual, P. S. (2005). "Time Series Analysis in WinIDAMS." Retrieved Dec 17, 2010, from http://portal.unesco.org/ci/en/files/18650/11133194701TimeSeriesAnal.pdf/TimeSeriesAnal.pdf.

Neuralware. Retrieved Dec 22, 2010, from http://www.neuralware.com/index.jsp.

Newton, H. J. (1999). "The Periodogram." Retrieved Dec 1, 2010, from http://www.stat.tamu.edu/~jnewton/stat626/topics/topics/topic4.pdf.

Park, D. C., Woo, D. M (2009). "Prediction of Network Traffic by using Dynamic BiLinear Recurrent Neural Network." Paper presented at Natural Computation, 2009. ICNC '09. Fifth International Conference on 2: 419 - 423.

Paulo, C., Miguel, R & Jose, N (2006). Time Series Forecasting by Evolutionary Neural Networks, Idea Group Inc.

Pavlykevych, M., Kostiv, O & Shatalova, O (2004). "The Analysis and Prediction of the WEB-Host External Channel Traffic by Mean of Time Series." Paper presented at Modern Problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the International Conference: 449 - 450.

Principe, J. C., Euliano, N. R., and Lefebvre, W. C (1999). Neural and Adaptive Systems: Fundamentals through Simulations, John Wiley and Sons Inc., New York

Probst, K., Hagmann, J (2003). "Understanding Participatory Research In The Context Of Natural Resource Management – Paradigms, Approaches and Typologies." Agricultural Research and Extension Network.

Richard, L. (1996). Retrieved Jan 3, 2011, from http://sundog.stsci.edu/rick/SCMA/node2.html.

Rutka, G. (2006). "Neural Network Models for Internet Traffic Prediction." Electronics and Electrical Engineering – Kaunas Technology 4(68): 55-58.

Sabry, M., Abd-El-Latif, H and Badra, N (2007). "Comparison Between Regression and Arima Models in Forecasting Traffic Volume." Australian Journal of Basic and Applied Sciences.

Soltic, S., Pang, S., Kasabov, N., Worner, S and Peackok, L (2004). "Dynamic Neuro-fuzzy Inference and Statistical Models for Risk Analysis of Pest Insect Establishment." Springer-Verlag Berlin Heidelberg: 971-976.

Song, Q., Kasabov, N (2002). "Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): On-line learning and Application for Time-Series Prediction." IEEE 10(2): 144 - 154.

Soule, A., Salamatian, K., Nucci, A & Taft, N (2005). "Traffic matrix tracking using Kalman filters." ACM 33: 24-31.

Srivastava, J., Cooley, R., Deshpande, M & Tan, P. N (2000). "Web usage mining: discovery and applications of usage patterns from web data." SIGKDD Explorations: 12-23.

Tang, Z., Fishwick, A (1993). "Feedforward Neural Networks as Models for Time Series Forecasting." ORSA Journal on Computing 5: 374-386.

Taylor, S. (2003). "Principles of sociology. Chapter 3 Theory and research." Retrieved 23 July 2010, from http://www.londonexternal.ac.uk/current_students/programme_resources/lse/lse_pdf/foundation_units/prin_soc/prinsoc_chapter3.pdf.

Tran, N. R., D. A (2001). "ARIMA Time Series Modeling and Forecasting for Adaptive I/O Prefetching." ACM.

Tsai, P. S. M. (2009). "Mining frequent itemsets in data streams using the weighted sliding window model." Expert Systems with Applications:An International Journal 36(9).

Walgampaya, C., Kantardzic, M (2006). "Selection of Distributed Sensors for Multiple Time Series Prediction." Paper presented at Neural Networks, 2006. IJCNN '06. International Joint Conference on 3152 - 3158.

Wang, X., Abraham, A & Smith, K (2004). "Intelligent web traffic mining and analysis." Journal of Network and Computer Applications: 147-165.

Welch, R., Ruffing, S and Venayagamoorthy, G (2009). "Comparison of Feedforward and Feedback Neural Network Architectures for Short Term Wind Speed Prediction " Paper presented at Neural Networks, 2009. IJCNN 2009. International Joint Conference on 3335 - 3340

Williams, B. M., Hoel, L. A (2003). "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. ." Journal of Transportation Engineering 129(6): 664-672

XLSTAT. (2010). Retrieved Dec 22, 2010, from http://www.xlstat.com.

Yao, S., Hu, Changzhen & Sun, Mingqian (2006). "Prediction of Web Traffic Based on Wavelet and Neural Network." Paper presented at Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on 1: 4026 - 4028

Yeh, I. C., Lien, C (2007). "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." Expert Systems with Applications.

Yeh, I. C., Lien, C (2009). "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." Expert Systems with Applications 36(2): 2473-2480.

Yinghong, Z., Yingchun, Z (2009). "The Paradigms for the Inquiry in Decision Support System (DSS) and a Design Framework of Cognitive DSS." Paper presented at Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on 1: 316.

Zhand, J., Chung, H. S & Lo, W (2008). "Chaotic Time Series Prediction Using a Neuro-Fuzzy System with Time-Delay Coordinates." IEEE 20(7): 956 - 964.

Zhao, G., Tang, H., XU, W & Zhang, Y (2004). "Application of Neural Network for Traffic Forecasting in Telecom Networks." Paper presented at Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on 4: 2607 - 2611.

Zhou, B., He, D & Sun, Z (2006). "Traffic Predictability based on ARIMA/GARCH Model." Paper presented at Next Generation Internet Design and Engineering.