Fuzzy Inference System for Network Traffic Load Prediction

Dissertation submitted to the
Jawaharlal Nehru University, New Delhi
In Partial fulfillment of the requirements for the award of the degree of

Master of Technology
in
Computer Science and Technology

By

Chandresh Kumar Maurya

Under the supervision of

Prof. Sonajharia Minz

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067, INDIA
JULY, 2012

जवाहरलाल नेहरु विश्वविद्यालय
SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWHARLAL NEHRU UNIVERSITY
NEW DELHI-67

CERTIFICATE

This is to certify that the dissertation entitled **"Fuzzy Inference System for Network Traffic Load Prediction"** being submitted by **Mr. Chandresh Kumar Maurya** to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi,** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Technology,** is a record of bona fide work carried out by him under the supervision of **Professor Sonajharia Minz**.

This work has not been submitted in part or full to any university or institution for the award of any degree or diploma.

(Prof. Karmeshu)
Dean

(Prof. Sonajharia Minz)
Supervisor

जवाहरलाल नेहरु विश्वविद्यालय
SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWHARLAL NEHRU UNIVERSITY
NEW DELHI-67

DECLARATION

This is to certify that the dissertation entitled "**Fuzzy Inference System for Network Traffic Load Prediction**" is being submitted to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi,** in partial fulfillment of the requirements for the award of degree of **Master of Technology** in **Computer Science and Technology,** is a record of bonafide work carried out by me.

The matter embodied in the dissertation has not been submitted in part or full to any university or institution for the award of any degree or diploma.

(Chandresh Kumar Maurya)
M. Tech (Computer Science and Technology)
(2010-2012)
SC&SS, JNU
New Delhi-110067

Dedicated to:

My Parents, Brothers, Sisters & Sweet Nephew

Acknowledgements

I would like to express my sincere thanks to my supervisor **Prof. Sonajharia Minz** for her encouragement, enthusiasm and direction provided to me during the entire duration of the proposed work. Without her valuable thoughts, recommendation, and patience, I would have never been able to complete my work.

I wish to thank all my fellow colleagues for their help and support. I gratefully acknowledge the unselfish help given to me by them who inspired me greatly through many interesting discussions, support, and feedbacks.

Also, I am thankful to my parents for their dogged persistence and motivations in getting me complete my degree.
Finally, I want to thank the almighty whose choicest blessings have sailed me through this research effort.

Chandresh Kumar Maurya

Abstract

Traditional statistical analysis of Internet traffic data is often employed to determine traffic distribution, to summarize user's behavior patterns, or to predict future network traffic. Mining of Internet traffic data may be used to discover hidden user groups, to detect payment fraud, or to identify network abnormalities like flow control and congestion control. The work carried out in this dissertation utilizes computational intelligence techniques like fuzzy logic to predict Internet traffic data pattern. A fuzzy inference system (FIS) is constructed. The FIS is first trained and then tested on numerically generated data by Mackey-Glass delay differential equation. Data collected over JNU LAN for twenty four hours is preprocessed. The cleaned data is subsequently given as input to the FIS for prediction. As a result, it is found that the predicted traffic agree with actual Internet traffic pattern. In addition, Hurst parameter estimate shows degree of self-similarity in JNU LAN traffic data to be about 0.92 that is much closer to Perfect self-similarity (H=1).

.

Contents

Abstract --vi

List of Figures --viii

List of Tables---ix

Chapter 1 – Introduction---1

1.1 Motivation--2
1.2 Problem definition---3
1.3 Outline of the Thesis---4

Chapter 2 – Related Work--5

Chapter 3 - Preprocessing of Network Traffic Data--------------------8

3.1 Network traffic data--8
3.2 Publicly Available Measurements---------------------------------10
3.3 JNU LAN traffic data--11

Chapter 4 - Methodology for Time Series Prediction----------------14

4.1 Methodology---14
4.2 Designing Fuzzy Inference System-------------------------------15
4.3 Self-Similarity---23

Chapter 5 – Implementation, Experiment Results and Analysis--25

5.1 Implementation ---25
5.1.1 System Specification--25
5.1.2 Implementation Details--25
5.2 Data Description---26
5.3 Experiment and Results--28
5.3.1 Prediction of Mackey-Glass series----------------------------29
5.3.2 Prediction of JNU LAN traffic load--------------------------32
5.4 Self-Similarity in the experimental data--------------------------37
5.5 Analysis--40

Chapter 6 - Conclusion and Future Work------------------------------41

6.1 Conclusion--41
6.2 Future Work--41

REFERENCES
APPENDIX

List of Figures

3.1 Packet capturing infrastructure--11
3.2 Packet sniffing using Smartsniff---------------------------------------12
3.3 Plot between time and number of bytes per second-----------------13
3.4 Plot of number of packets per second against time------------------13
4.1 Fuzzy inference system framework for time series prediction-------16
4.2 Extracting linguistic summary---17
4.3 Membership functions--18
4.4 Triangular Fuzzy membership functions-------------------------------20
4.5 The form of a fuzzy rule base---22
5.1 Mackey-Glass series ---27
5.2 (a) Number of packets/second (b) Number of Bytes/second--------28
5.3 Triangular fuzzy sets for all inputs and one output-------------------29
5.4 True vs. predicted data plot (a) Wang method (b) WCVmethod -30
5.5 True Vs. predicted data plot (a) Wang method (b) WCVmethod----30
5.6 Prediction of Mackey-Glass series with two inputs and five fuzzy Sets (a) True Vs. predicted (b) Absolute error plot. ------------------31
5.7 Prediction of Mackey-Glass series with three inputs and five fuzzy sets (a) True Vs. predicted (b) Absolute error plot. -------------------31
5.8 (a) JNU network traffic load prediction plot (b) MSE with 3 fuzzy sets and 4 predictor---33
5.9 (a) JNU network traffic load prediction plot (b) MSE with 5 fuzzy sets and 4 predictor--34
5.10 (a) JNU network traffic load prediction plot (b) MSE with 7 fuzzy sets and 4 predictor--35
5.11 (a) JNU network traffic load prediction plot (b) MSE with 5 fuzzy sets and 3 predictor--36
5.12 Variance–time plot for Mackey-Glass series and calculation of Hurst parameter--38

5.13 Estimation of Hurst parameter for JNU LAN traffic data (a) Variance-time plot for number of packets per second (b) Variance-time plot for number of Bytes per second-----------------39

List of Tables

3.1 Number of packets per second and their capture timing-------------12

3.2 Data size in KB/sec carried by packets with their timing------------12

4.1 Linguistic summary for network traffic data----------------------------18

5.1 Comparision of WCV method with Wang method----------------------30

5.2 Comparision of WCV method with Wang method----------------------32

5.3 Comparision of WCV method with Wang method----------------------36

5.4 Comparision of WCV method with Wang method----------------------37

Chapter 1

INTRODUCTION

Mining Internet traffic data is a daunting task because of Massive data size, time varying patterns and topology. There are both spatial as well as temporal issues. Taking temporal dimension, one can have millions of IP packets (which are actually in GBs and TBs) by capturing at routers for an hour [1]. Regarding spatial issues, there could be thousands of routers, switches and end nodes even in small local area networks. To effectively mine the network traffic data, there should be well established infrastructure. The recent development of high performance hardware for IP packets can capture 10 Gb/s [2]. However, it is not feasible to use such vast amount of data for research and operation tasks. Filtering and preprocessing methods need to be applied. This has lead to the evolution of what researchers call Internet Science.

In fact, there are three major aspects that should be taken into account while modeling the Internet.

- Traffic
- Topology
- Effect of protocols on traffic and topology.

Analysis and data mining of topology related measurements are often executed offline with the help of network operators. The objective of these studies is to identify invariants that give insights into how topologies evolve. For instance, it has been found that any two randomly chosen documents on the web are 19 clicks away and any two nodes are 6 hopes away [3]. Major advances in Internet modeling have revealed identification of self-similarity, long range dependences (LRD) and use of power law to describe the global topology of the Internet.

Because of inherent nonlinear dynamics, many statistical models have failed to characterize the traffic invariants Therefore, there is the need of developing models that are nonlinear and can capture above mentioned behavior. In this direction, fuzzy theory has proved vital. The fuzzy logic technique is also important for approximation of function and modeling static and dynamic systems. As a result of the research, fuzzy systems now in addition to taking the linguistic information (linguistic rules) from human experts, can adapt itself using numerical data (input/output pairs) to have a good performance. The fuzzy modeling of dynamic systems is addressed, as well as the methods to construct fuzzy models from knowledge and data. Furthermore, the fuzzy inference system is mapped onto a neural network-like architecture. This has lead to the development of Neuro-fuzzy systems.

1.1 Motivation

Today, the Internet has evolved enormously and in an unleashed way because of TCP/IP protocol. Millions of TCP/IP packets move in a single day across the campus local area network (LAN). Several studies have shown that Internet traffic exhibit complex nonlinear behavior. Many classes of dynamical behavior have been described, including regular predictable and unpredictable behavior, transient and intermittent chaos, narrow-band and broad-band chaos, pseudo-randomness and superposition of several basic patterns [4]. In some cases, however, studies have ascertained that network traffic shows pattern that can be properly classified within regular predictable phenomena. In this aspect, we can view analysis of network traffic data as analysis of time series data. In these cases, the theory of nonlinear dynamics provides a proper framework for the analysis, identification and prediction of network traffic time series.

Network traffic prediction finds application in variety of domains like congestion and admission control, network management and anomaly detection as well as quality of service (QoS) provisioning. The idea behind network traffic prediction is to predict network traffic for the next control or action period based upon active or passive measurement.

There are services like network weather service (*http://nws.cs.ucsb.edu*) that has become important for adaptive applications in recent years. Similarly, Grid computing depends on the availability of measurement and predictions of network conditions in order to optimize performance.

This has resulted in the development of grid-based services for predicting TCP/IP end-to-end throughput and latency [5].

1.2 The Problem Definition

Time series prediction and analysis in general is a recurrent problem virtually in all areas of natural and social sciences as well as in engineering. In the field of time series prediction, prediction accuracy is not the only major goal. Understanding the behavior of time series and gaining insights into their underlying dynamics is a highly desired capability of time series prediction methods [6].

In the past, conventional statistical techniques such as AR and ARMA models have been extensively used for forecasting [7]. An accurate traffic prediction model should have the ability to capture prominent traffic characteristics, such as long-range dependence (LRD) and self-similarity in the large time scale, multifractal in small time scale. However, these techniques have limited capabilities for modeling time series data, and more advanced nonlinear methods including artificial neural networks have been frequently applied with success [8]. Some authors have also applied fuzzy techniques for the prediction of network traffic series with considerable success. Because of the novelty of the technique, FIS is developed and used for prediction purpose. In this dissertation, it is assumed that network traffic data

Objective:

To develop a Fuzzy inference system that can predict Internet traffic load pattern so as to optimize the network resources and handle flow and congestion control dynamically.

1.3 Organization of Thesis

The thesis is organized as follows: Chapter 2 talk about the related work done in the area of network traffic prediction. Chapter 3 discusses various steps taken to prepare data for prediction purpose. Methodology for time series prediction is given in chapter 4. Chapter 5 talks about the implementation details and simulation results and conclusion is discussed in chapter 6.

Chapter 2

RELATED WORK

There exist statistical techniques for modeling and predicting network traffic but they could not capture complex nonlinear behavior exhibited by network traffic data. Earlier models used for network traffic prediction includes Markov model, Auto Regressive (AR) and Auto Regressive Integrated Moving Average (ARIMA). The exponential decay of the autocorrelation function of these models gives them the ability to capture the short-range dependence (SRD) characteristics only. Such kinds of models are also known as Short-Memory Stochastic models. However, it has been shown that the traffic data exhibited a high degree of long-range dependence (LRD) characteristics besides SRD. Thus, such models cannot characterize the network traffic well, and unsuitable for traffic prediction.

More recent models known as the Fractional Auto Regressive Integrated Moving Average (FARIMA) model, Fractional Brownian Motion(fBm), Fractional Gaussian Noise and Generalized ARMA(GARMA) capture both SRD and LRD and has been used to model and predict traffic data [9][10]. Since these models can predict LRD phenomena, they are called Long-Memory Stochastic Models. Nonetheless, these models fail to capture multifractal which has been observed in network traffic at small time scale. For this reason, another Multifractal Wavelet model (MWM) has been introduced to solve this problem. MWM model can capture multifractal but cannot predict traffic [11]. On the other hand, it is found that traffic exhibited non-stationary and non-linear properties and threshold autoregressive (TAR) model [12] has been proposed to model such properties. In [12], the authors have developed the first network measurement system which integrated prediction and they have also proposed running multiple predictors simultaneously and forecasting one which exhibiting the smallest prediction error produced on its measurements.

Apart from the above mentioned model-based prediction schemes, [13] has reported that non-model-based prediction provides better prediction than model-based prediction as long as the traffic is LRD or self-similar. However, the authors only compared their non-model-based prediction model with the FARIMA and FBM models. Both these two models cannot capture bursty traffic very well and this bursty characteristic affects traffic prediction accuracy. Another significant prediction research work is introduced in [6], which analyzed the prospects for multi-step prediction of network traffic using ARMA and MMPP models. Their analysis is based on continuous time ARMA and MMPP models driven by Gaussian noise sources. Some authors as in [14] have used combined approach of ARIMA/GARCH to model and forecast network traffic. However, its predication methodology is quite complex and unstable and does not scale well. The Volterra models[15] and the Higher Order Cumulant (HOC) techniques[16] are used also respectively for analyzing IP network and modeling time series of internet traffic.

On the other hand, in [17] the author has proposed seasonal autoregressive-generalized autoregressive conditional heteroscedasticity (AR-GARCH) models in order to capture seasonal pattern manifested by Internet traffic. However, it works only for hourly and higher resolution and does not scale well for low time resolution like minutes and seconds. In addition, [17] uses RMSE criterion for error measurement in prediction; for network traffic, it has been pointed out [18] that a generalized-cost function approach that penalizes under-forecasting is an important consideration in predicting network traffic.

Because of massive data size, varying topology and protocols governing traffic and topology, there is the need of methodology that can scale for dataset of any size and can capture self-similarity, multifractal behavior prominently found in network traffic data. Hence, research community is now using computational intelligence based methods and models for network traffic measurement and prediction purpose. Among these are: Fuzzy logic based techniques, support vector machines, neural network and their combined approach as discussed in [20][19].

Techniques of fuzzy logic are widely used because of their rule interpretability, ability to solve system identification and predication problems. Indeed, the theory of fuzzy logic provides a mathematical framework to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning[22,21]. Furthermore, the application of this theory has increased in recent years and has multi-disciplinary in nature, includes automatic control, signal processing and time series prediction.

Chapter 3

Preprocessing of Network Traffic Data

Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is pre-processed. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset. Data preprocessing methods are divided into following three categories [23].

- Data Cleaning
- Data Integration and Transformation
- Data Reduction

Data cleaning removes observation with noise and missing values. Data integration merges data from multiple sources into coherent form whereas transformation operation changes data into form appropriate for mining. Reduced data is split into two sets, the “training set” and the “test set”. The training set is used to “to train” the data mining algorithm, while the test set is used to verify the accuracy of any pattern found. The section 3.1 discusses Network traffic data whereas section 3.2 talks about the publicly available measurement systems and infrastructure.

3.1 Network Traffic Data

The traffic data contains event log tables recording activities occurred in the network. They are aggregated from the distributed database of the network management system.

Measurement Systems and Infrastructure

Network performance depends on and can be measured in terms of a number of parameters such as capacity, available bandwidth, delay, jitter, and packet loss and packet disorder. These and other network parameters are related in a complex manner and to a varying extent. Measuring the network is crucial to understanding the Internet behavior and designing control mechanisms for improving performance.

Current measurement systems [24] can be classified into two main types: active and passive. The former are of a distributed nature and are usually accessible to end users and applications. The latter are centralized and often restricted to network operators and engineers. The current challenges in this area are to increase the maturity of these systems, to deploy measurement infrastructures and to enable generalized macroscopic analysis of the Internet.

Active Systems

Active measurement systems work by sending probe traffic from an end node in order to measure parameters such as round-trip time and packet loss percentage [25]. Active measurement tools inject probe packets into the network and analyze the response. Some active systems are variable packet size (VPS), packet pairs, packet trains, packet tailgating, ALBP (Asymmetric Link Bandwidth Probing), self-loading streams, to name a few.

Passive Systems

Passive measurement systems are based on recording data at a network node, i.e., no probe packets are sent. While passive systems do not require cooperation or coordination among end nodes, the quality and relevance of data decisively depends on the location of the measurement point. Thus, cooperation between network operators [25] is a prerequisite of passive measurement infrastructures.

3.2 Publicly Available Measurements

Traces are one of the main outcomes of measurement infrastructures. The use of common traces recorded by both active and passive measurement infrastructures are key reproducible research and comparison of results in general. Traces may comprise data about topology, traffic, specific applications and a variety of heterogeneous measurements. In this sense, the recent availability traffic traces of high-speed networks, especially at OC48 and OC192 speeds, require a great deal of effort and cooperation among different agents. Cooperative measurement projects and infrastructures also allows for wide scale analysis of networks.

In most of the experiment, the researchers have used a wide set of publicly available network traffic traces obtained through passive monitoring. These traces are usually made of a sequence of packet headers (possibly including part or the entire payload as well). Some other traces only provide a restricted set of data about each received packet, in particular the arrival time and size, as well as some other especially relevant data such as TCP flags. Some traces have an historical relevance such as the Bellcore traces and the traces taken at the Lawrence Berkeley National Laboratory. The first were the empirical basis to find self-similarity and long-range dependence in Ethernet traffic [26] whereas the second were instrumental in showing that the Poisson model fails to capture the general behavior of traffic in wide area networks.

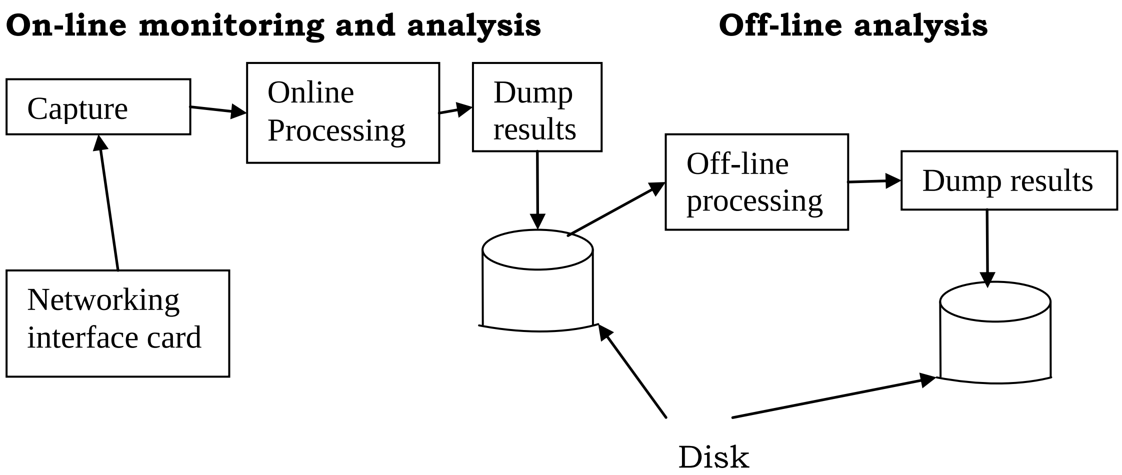


Fig 3.1 Packet capturing infrastructure

3.3 JNU LAN traffic data

For the purpose of internet traffic analysis, TCP/IP packets were captured using SmartSniff (http://www.SmartSniff.com). One million packets were captured from 2PM to 3PM on April 12, 2012 over JNU LAN. About 0.01% of these packets were dropped. One snapshot of this is shown in the figure 3.1. Next, of the twelve fields, only the most relevant field were taken into account

For network traffic load prediction, we have chosen number of packets per second, data size (in KB/Sec) and timing of the packet capture. The reason behind selecting these features only is that some of the fields were either non numeric (for example protocols like TCP/UDP or service name like domain/llmnr) or addresses (for example source and destination address of the form 172.16.0.1). The problem with these fields is that they cannot be fuzzified unless some modification is done in their representation. In addition, there contribution to network load is less significant as compared to selected fields. Thereafter, we collected the data with some missing values for data size and number of packets

Index	Protocol	Local Address	Remote Address	Local Port	Remote Port	Service Name	Packets	Data Size	Total Size	Capture Time	Data Speed	Last Packet Time
109	UDP	172.16.6.156	203.41.10.31	52138	53	domain	3 {3 ; 0}	129 Bytes {129...	284 Bytes {213...	10/11/2011 4:48:29...	0.0 KB/Sec	10/11/2011 4:48:36...
110	UDP	172.16.6.156	224.0.0.252	61479	5355	llmnr	4 {4 ; 0}	96 Bytes {96 ; 0}	260 Bytes {208...	10/11/2011 4:48:31...	1.0 KB/Sec	10/11/2011 4:48:32...
111	UDP	172.16.6.201	224.0.0.252	56797	5355	llmnr	2 {2 ; 0}	44 Bytes {44 ; 0}	150 Bytes {100...	10/11/2011 4:48:34...	0.4 KB/Sec	10/11/2011 4:48:34...
112	UDP	172.16.6.201	224.0.0.252	63862	5355	llmnr	2 {2 ; 0}	44 Bytes {44 ; 0}	150 Bytes {100...	10/11/2011 4:48:34...	0.4 KB/Sec	10/11/2011 4:48:34...
113	TCP	172.16.6.156	172.16.6.201	445	49270	microsoft-ds	34 {34 ...	4,522 Bytes {4,...	6,096 Bytes {5,...	10/11/2011 4:48:34...	5.6 KB/Sec	10/11/2011 4:48:35...
114	UDP	172.16.6.156	224.0.0.252	57424	5355	llmnr	4 {4 ; 0}	96 Bytes {96 ; 0}	260 Bytes {208...	10/11/2011 4:48:34...	0.9 KB/Sec	10/11/2011 4:48:34...

Fig 3.2 Packet sniffing using Smartsniff

Table 3.1: Number of packets per second and their capture timing

Capture time	No. of packets/sec
3:48:43	1548
3:48:44	42
3:48:44	1
3:48:45	14
3:48:45	5
3:48:48	11
3:48:48	5
3:48:49	30
...	...

Table 3.2 Data size in KB/sec carried by packets with their timing

Capture time	Data size in KB/sec
3:48:43	1520866
3:48:44	2702
3:48:44	98
3:48:45	5
3:48:45	595
3:48:48	4
3:48:48	1
3:48:49	13551
...	...

per second. After removing the missing values for data speed, we obtained around 1347 tuples. As shown in table 3.1and table 3.2. Since there is no advantage of having date, it is also removed from the dataset. Finally, to get the packets at an interval of seconds, millisecond timestamp was removed and what is left is the time of the format

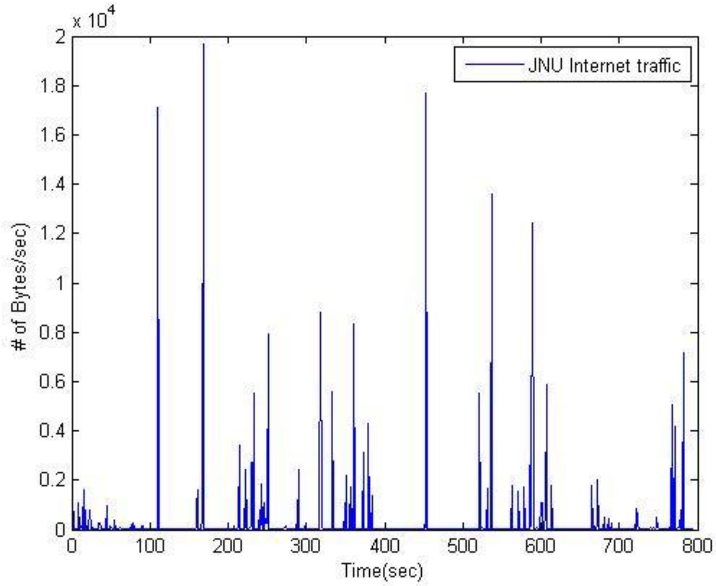


Fig. 3.3 Plot of number of Bytes with respect to time in seconds

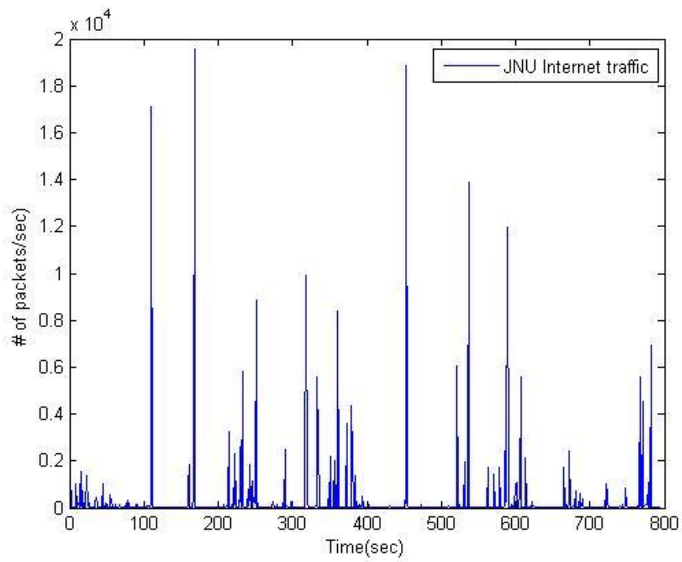


Fig. 3.4 Plot of number of packets with respect to time in seconds

HH:MM:SS and data size. A plot between time and number of bytes per second is shown below in figure 3.3. In figure 3.4, we have shown plot of number of packets per second against time. Note that removing millisecond from the timestamp will decrease accuracy of prediction. But, for the sake of simplicity of preprocessing, trade-off has been made between the accuracy and complexity.

Chapter 4
Methodology for Time Series Prediction

There are two methods for prediction: direct prediction and recursive prediction. In direct prediction, a prediction model is built for every prediction horizon. Thus, for maximum prediction horizon H, H direct models are built, one for each prediction horizon h. On the other hand, recursive prediction recommend using the same model recursively for predicting each successive future values. The former one suffers from high computational cost but giving an accurate model whereas the latter one does the opposite.

4.1 Methodology

Consider a discrete time series as a vector, $\bar{y} = x_1, x_2, \ldots, x_{t-1}, x_t$ that represents an ordered set of values, where t is the number of values in the series. Packard et al., [27] have demonstrated that an attractor may be reconstructed from a time-series by using a set of time delayed samples of the series. If Δ is the time delay, and m is an integer, then one may write for points on the attractor as

$$y(t+p)=f(\,x(t), x(t-\Delta), x(t-2\Delta), \ldots x(t-m\Delta)) \tag{4.1}$$

Where p is the prediction time in future and f( ) is a map. This may be viewed as m+1 dimensional surface. Thus, the “embedding dimension” d_E, is defined to be m+1. Taken [28] has proved that a least upper bound exist for which f( ) will be a smooth map. If the dimension of the attractor is defined to be, d_A, then one needs an embedding dimension less than or equal to $2d_A+1$ i.e.

$$d_E<=2d_A+1 \tag{4.2}$$

A minimal requirement is that $d_{E}>=d_{A}$. How to choose d_E, p and Δ is given in [29] and has also been discussed in chapter 5.

For prediction in time-series domain, data is arranged in the following matrix form (initially our data is given as x(1), x(2),…x(T)):

$$[x(t),\ x(t+\Delta)\ x(t+2\Delta),\ldots,x(t+\overline{m-1}\Delta);\ x(t+m\Delta)]$$
$$[x(t+1),\ x(t+1+\Delta),\ x(t+1+2\Delta),\ldots,x(t+1+\overline{m-1}\Delta);\ x(t+1+m\Delta)]$$
$$\ldots$$

In particular, m=3 and Δ=6 is taken for experimental purposes. This gives rise to four input variable and one output variable. Thereafter, data is split into training and testing sets. Training part is used to construct the model whereas testing part is used for verifying the accuracy of the model.

4.2 Designing Fuzzy Inference System (FIS)

Fuzzy inference system is directly developed from the data. Approach followed here is as given in [30]. It is basically two stage development method. In the first stage, the most relevant variables are selected. In the second stage, we apply system identification and tuning algorithm called Wang and Mendel [31]. A general framework for stepwise development of FIS is given in fig 4.1.

Stage 1: Variable Selection

There are number of variables accounting for influencing unknown future value in the time series prediction. However, employing all known values as input to the autoregressor (FIS) does not necessarily improve its accuracy. As the number of inputs increases, and the known data become sparser in a high-dimensional space, building a model gets more and more complex. This is the well known “curse of dimensionality” problem. From the implementation point of view, an embedding

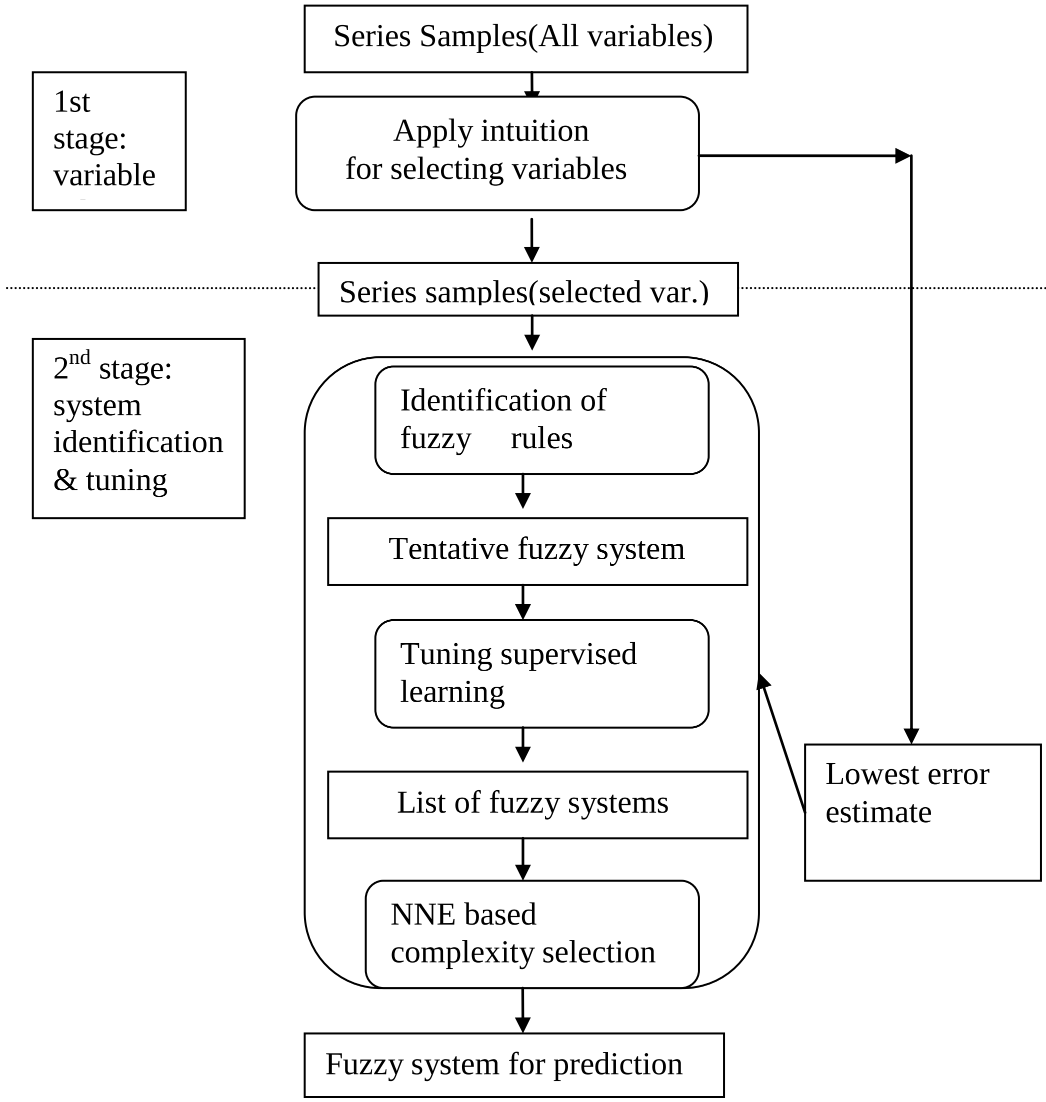


Fig. 4.1 Fuzzy inference system framework for time series prediction

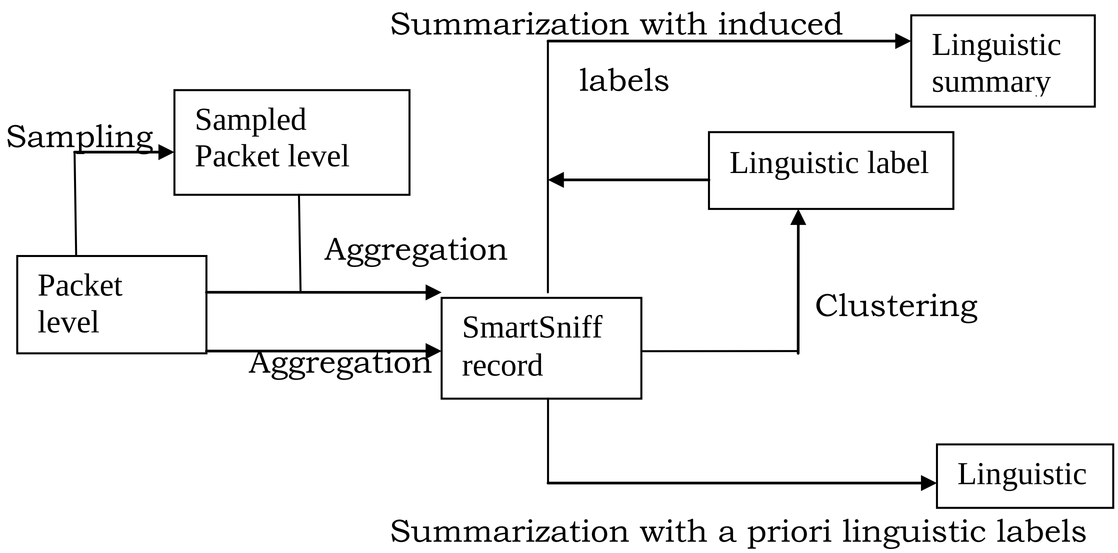


Fig. 4.2 Procedure of extracting linguistic summary

Dimension d_E equal to 4 is chosen. This gives from Eqn. (1) m=3. That is, there are four input variables x_1, x_2, x_3, x_4.

Stage 2: System Identification and Tuning

System identification is the core part of the FIS design process. It comprises of determining linguistic labels, choosing membership function and generating fuzzy rules. Linguistic summaries as proposed by Yager [32] are a data mining technique for summarizing data collections using linguistically quantified propositions [33], such as "Most traffic flows are short lived". The present work considers the extended definition by Kacprzyk and Zadro˙zny [34] that leverages on the concept of protoform or prototypical form.

Diagram in the figure 4.2 represents two kinds of linguistics summarization of traffic flow:

One uses the prior knowledge of linguistic labels deduced from the intuition and second one is derived from the clustering techniques.

Linguistic summarization using priori labels

Based upon our experience and intuition, we can assign some linguistics labels as follows:

Table 4.1 Linguistic summary for network traffic data

Attribute	Linguistics Labels
Duration	Short-lived, Long-lived
Average Packet Size	Small, medium, large
Bytes	Mice, Bulk, Elephants
Throughput	Low, medium, high
Packets	pk-mice, pk-bulk, pk-elephants

Below is the membership function diagram for above linguistic variables.

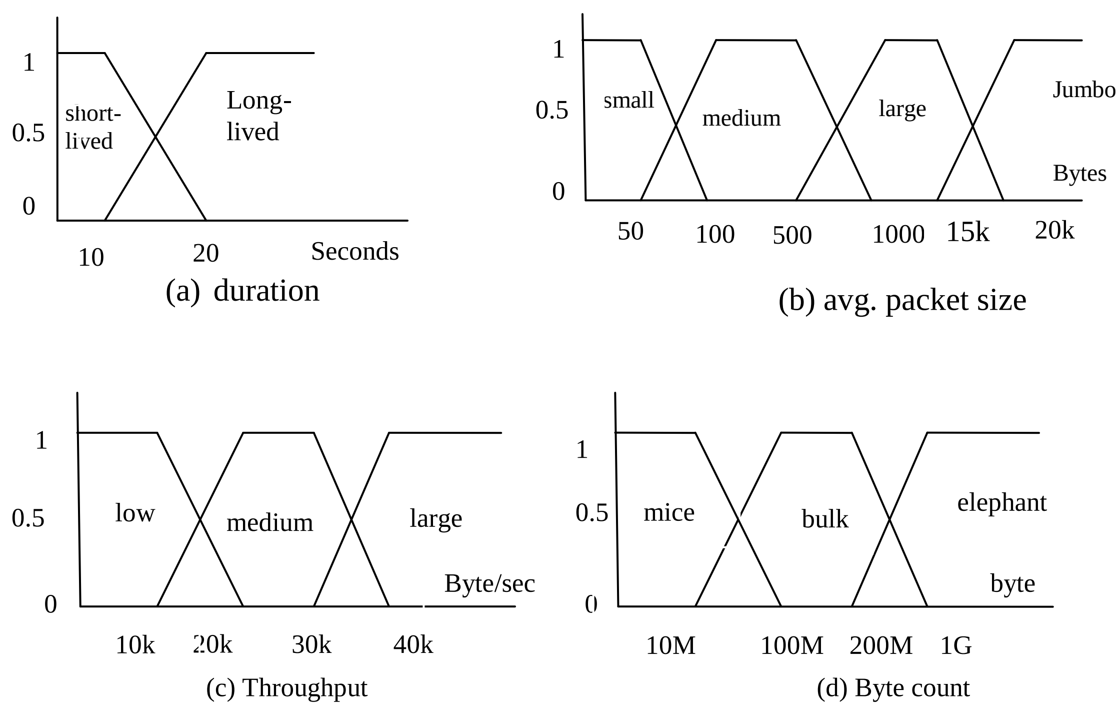


Fig 4.3 Membership functions for attributes given in Table 4.1

Linguistic summarization using clustering

Two methods for linguistic summarization based on clustering techniques are: k-mean (or hard c-mean) and fuzzy c-mean. The number of clusters is the most often a determining input variable to the algorithm. Normal approach is to check a) whether the number of optimal clusters matches the number of domain specific terms (current assumptions) and b) how summaries built with clustering derived labels compare with the summaries above?

The clustering processes could be run on the whole data sets. However, it has been found that it is not necessary to have a large number of flow samples in order to obtain consistent results that give a fairly good approximation to the clusters identified on the whole data sets. The approach followed here is as in [34] for fuzzy rule identification. It is basically five step procedure as outlined below.

Suppose that following input output pairs are available:

$$(x_1^1,\ x_{2,}^1\ ; y^1), (x_1^2,\ x_{2,}^2\ ; y^2), \ldots \tag{4.3}$$

Where x_1, x_2, are inputs and y is the singleton output. Our objective is to find Mamdani's type fuzzy rules of the form:

$$If\ x_1 is\ F_1\ and\ x_2 is\ F_2\ \ldots\ and\ x_m is\ F_m\ then\ y_1 is\ C_1\ and\ y_2 is\ C_2 \ldots\ and\ y_m is\ C_m \tag{4.4}$$

Where F_i and C_i are fuzzy sets defined on input and output space partitioning and finally determine a mapping of the form:

$$f: (x_1, x_2) \longrightarrow y.$$

An algorithmic procedure for such a mapping is described on the next page.

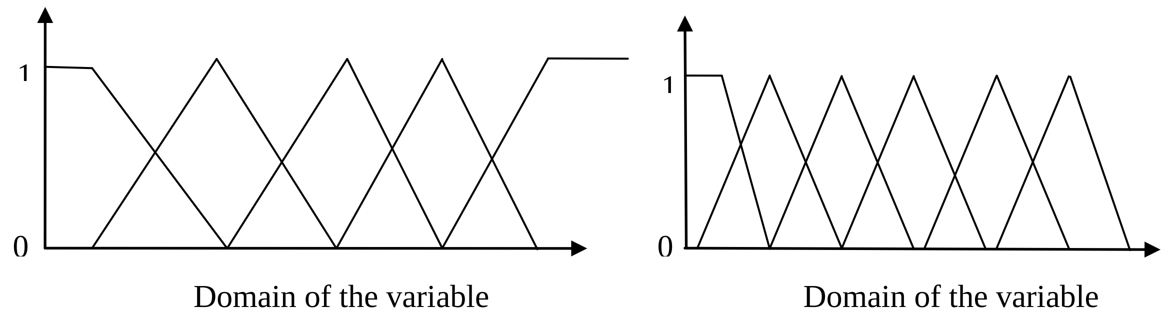


Fig. 4.4 Triangular Fuzzy membership functions

Development of FIS:

Step 1: *divide the input and output space into fuzzy regions*

Step 2: *Generate fuzzy rules from given data pairs*

Step 3: *Assign a degree of confidence to each rule*

Step 4: *Create a combined fuzzy rule base*

Step 5: *Determine a mapping based on combined fuzzy rule base*

Assume that the domain intervals of x1, x2 and y are [x_1^-, x_1^+], [x_2^-, x_2^+] and [y^-, y^+], respectively, where "domain interval" of a variable means that most probably this variable will lie in this interval (the values of a variable are allowed to lie outside its domain interval). Divide each domain into odd number of regions. Number and length of each region for each variable may be different. Assign a fuzzy membership function to each region as shown in the fig. 4.4. Note that any number and type of fuzzy membership function can be used, for the sake of simplicity, 5 to 7 triangular Membership function so as to avoid curse of dimensionality problem.

To generate fuzzy rules for each data pair, first determine degree of membership of each variable x_1^i, x_2^i *and* y^i in different regions just created above. Second, assign a given x_1^i, x_2^i *and* y^i to the region with highest degree of membership. Finally, get one rule for each pair of input-output data of the form:

$(x_1^i, x_2^i;\ y^i) \Longrightarrow [x_1^i(0.8\ in\ B1; max), x_2^i(0.6\ in\ B2; max); y^i(0.7\ in\ B2; max)] \Longrightarrow$

Rule i

The rules generated in this way are "and" rules, i.e., rules in which the conditions of the IF part must be met simultaneously in order for the result of the THEN part to occur. Due to large number of rules, one may have rules with the same IF part. Such rules are called 'conflicting rules'. One way to resolve this conflict is to assign a degree to each rule generated from data pairs, and accept only the rule from a conflict group that has maximum degree. In this way not only is the conflict problem resolved, but also the number of rules is greatly reduced.

The degree of a rule is calculated as the product of the membership values of each variable (input and output) as follows:

$$D(Rule)=M_A(x_1)M_B(x_2)M_C(y) \tag{4.5}$$

The form of a fuzzy rule base is illustrated in Fig. 4.5. Boxes of the rulebase are filled with fuzzy rules according to the following strategy: a combined fuzzy rule base is assigned rules from either those generated from numerical data or linguistic rules (assumption made here is that a linguistic rule also has a degree that is assigned by the human expert and reflects the expert's belief of the importance of the rule); if there is more than one rule in one box of the fuzzy rule base, use the rule that has maximum degree.
Defuzzification method used is centroid. First for given inputs (X_1,X_2), combine the antecedents of the i^{th} fuzzy rule using product operation to determine the degree, $m_{O_i}^i$, of the output control corresponding to (X_1,X_2), i.e.,

$$m_{O_i}^i = m_{I_1^i}(x1) m_{I_2^i}(x2) \tag{4.6}$$

where O_i denotes the output region of Rule i, and I_j^i denotes the input region of Rule i for the j^th component. Then use the formula given in Eqn.(6) for the defuzzification.

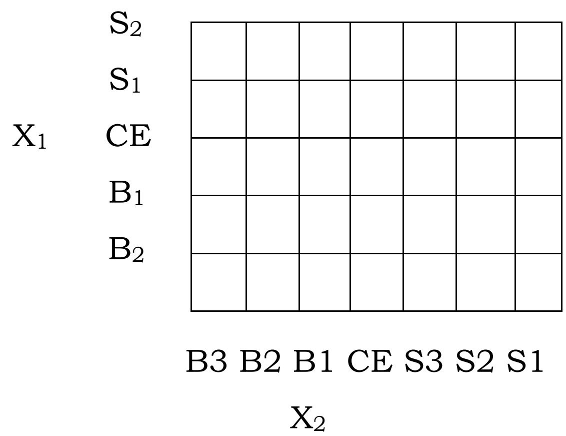


Fig. 4.5 the form of a fuzzy rule base

In this dissertation as a concrete implementation and for the sake of simplicity, direct prediction methodology has been used. Minimum as T-norm for conjunction operations and implications, triangular membership functions for inputs, singleton outputs and fuzzy mean as defuzzification method have been used. Therefore, in this particular case a fuzzy autoregressor for prediction horizon h can be formulated as follows:

$$F_h(\bar{y}) = \frac{\sum_{l=1}^{N_h} min(\mu_{R_l^h}, \underset{1 \le v \le 1}{min} \mu_{L_l^{i,h}}(y_v))}{\sum_{l=1}^{N_h} \underset{1 \le v \le 1}{min} \mu_{L_l^{i,h}}(y_v)} \tag{4.7}$$

Where N_h is the number of rules in the rulebase for horizon h, $\mu_{L_l^{i,h}}$, is Triangular membership functions and $\mu_{R_l^h}$ is singleton membership functions. A proof of Eqn.(4.7) is given in appendix.

The methodology described in the previous section builds a fuzzy regressor. But for prediction purposes, we need to take care of under fitting of test data. Therefore, Wang method is slightly modified to incorporate cross-validation of the data. The combined method is called Wang with cross-validation (WCV).

The problem of building a regressor can be precisely stated as that of defining a proper number and configuration of membership functions and building a fuzzy rulebase from a data set of t sample data from a time series such that the fuzzy systems $F_h(\bar{y})$ predict the h^{th} next values of the time series. The error metric to be minimized is the root mean squared error (MSE).

4.3 Self-Similarity

The self-similarity means that the statistical properties (all moments) of a stochastic process do not change for all aggregation levels. The main properties of self-similar processes include:

- Slowly decaying variance – the variance of the sample is decreased more slowly than the reciprocal of the sample size.
- Long-range dependence - the process is called a stationary process with long-range dependence if its autocorrelation function is non-summable. The speed of decay of autocorrelations is more hyperbolic than exponential.
- Hurst parameter (H) – it expresses the degree of self-similarity (0<=H<=1). The closer to 1 is H; the more similarity will be present in the series. And vice-versa.

Hurst parameter can be calculated using variance-time plot.

Variance-time plot

For a self-similar time-series:

$$\{X\}=\{X_1, X_2, \ldots, X_k\} \tag{4.8}$$

The m-aggregate $\{X_k^{(m)}\}$ with its k^{th} term:

$$X_k^{(m)} = \frac{X_{km-m+1}+\cdots+X_k}{m} \quad \text{Where } k=1, 2, 3\ldots \tag{4.9}$$

The degree of self-similarity and long-range dependence increases as H→1. In the present work, self-similarity is estimated using variance-time plot. In this process, variance of aggregate of a self-similar process is defined as:

$$\mathrm{Var}(X^m)=\mathrm{Var}(X)/m^{\beta} \tag{4.10}$$

This can be rewritten as:

$$\log\{\mathrm{VAR}(X^{(m)})\}=\log\{\mathrm{VAR}(X)\}- \beta \log\{m\} \tag{4.11}$$

If VAR(X) and m are plotted on a log-log graph then by fitting a least square line through the resulting points one can obtain a straight line with the slope of β and using which in Eqn. (5.6) yields Hurst parameter

$$H=1-\beta/2 \tag{4.12}$$

The Mackey-Glass series as well as JNU LAN traffic data exhibit self-similarity. The above theory about self-similarity is verified in section 5.4 of chapter 5.

Chapter 5

Implementation, Experimental Results and Analysis

5.1 Implementation

5.1.1 System Specification

Software requirements:

For the concrete implementation, platform can be any Operating system running MATLAB. Software packages required is MATLAB/Octave and Packet capturing software like SmartSniff. Often, Microsoft office package is used in data preprocessing.

Hardware requirements:

The WCV and W&M methods are implemented on Intel Core I3 Processor running on 3GB RAM (512MB minimum). Hard disk space required is 1GB minimum free space.

5.1.2 Implementation Details

Initially, prediction methodology as discussed in chapter 4 is implemented in the environment as specified in section 5.1.1 to develop a Fuzzy inference system (FIS). The prediction methodology WCV as represented in chapter 4 is compared with the Wang method [34, 35]. However, there is difference in the way the Wang method make prediction and WCV implemented in this dissertation. Instead of implementing “extrapolation of rules”, the present work employs “Holdout” and “LeaveMout” sampling methods in order to see the performance of FIS prediction on the network traffic data.
Fuzzy rule extrapolation is a technique to handle prediction of those test samples that fall outside the range of training samples. In Wang method

[35], two rules are neighbors to each other if they share the same IF part except in one variable and the fuzzy sets for this variable in the two rules are neighbor of each other. Finally, the rules that have already been generated from the data are called data-generated rules, and the rules to be extrapolated based on the data-generated rules are called extrapolating rules. Data mining Sampling methods like “Holdout” and “LeaveMout” are based on random sampling of the data and therefore add negligible complexity as compared to loosing accuracy due to “Fuzzy rule extrapolation” based on nearest neighbor fuzzy-sets.
The FIS prediction method is first evaluated on the data numerically generated by Mackey-glass delay differential equation given by (5.1) followed by JNU LAN preprocessed traffic data as given in section 3.3.

5.2 Data Description

In this work WCV is compared with WM using two time series data: a synthetic data generated numerically given Eq. (5.1) and JNU LAN traffic data as presented in section 3.3 of chapter 3. Although, the objective in the dissertation is to predict “data network” traffic, the reason behind choosing first dataset is that it is representative of “data network” traffic in the sense that for delay parameter τ>17, its exhibits chaotic behavior. In some way, we will have idea of whether the FIS developed is performing as expected (measured in terms of mean absolute error).

Synthetic data is generated using Mackey-Glass delay differential equation (5.1) as shown in figure 5.1.

$$\frac{dy(t)}{dt} = \frac{0.2y(t-\tau)}{1+y(t-\tau)^{10}} - 0.1y(t) \tag{5.1}$$

To obtain the time series value at Integer points, the fourth-order Runge-Kutta method is applied to find the numerical solution to the previous Mackey-Glass eqn. (5.1). The series consists of 765 samples. This time

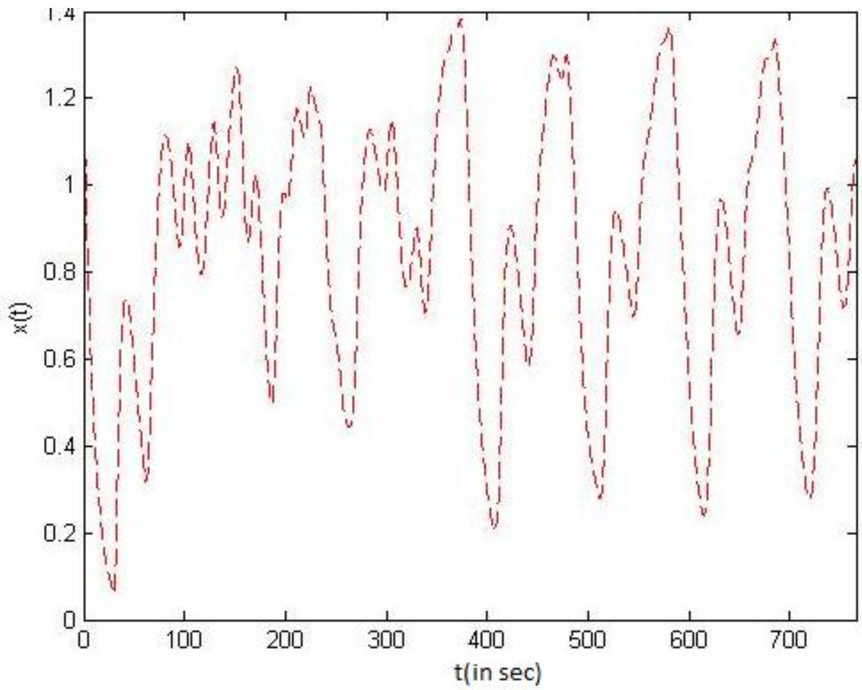
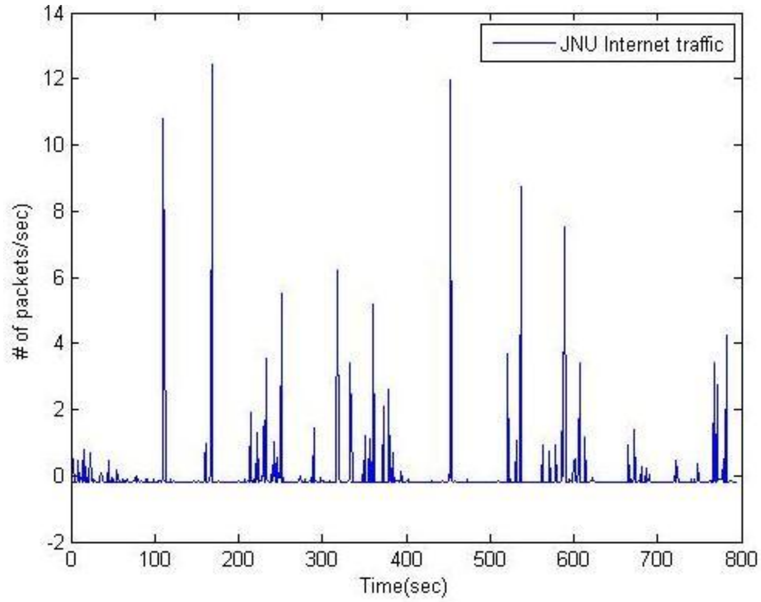


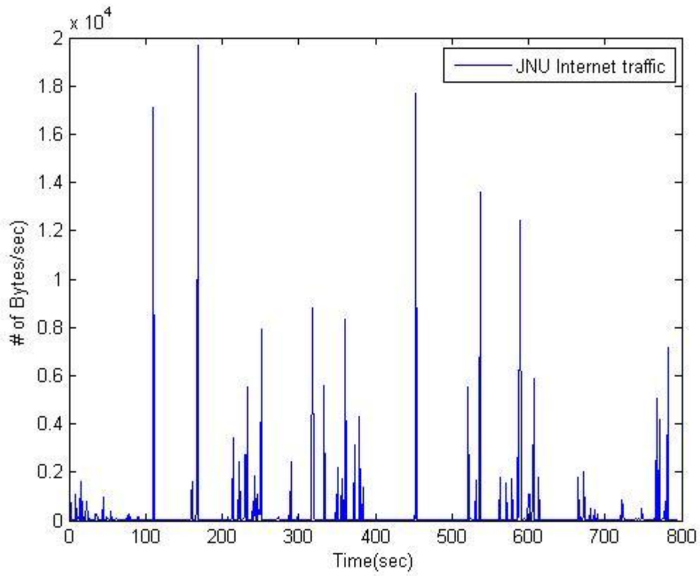
Fig. 5.1 Mackey-Glass series at τ=30

Series is chaotic for τ>17, and so there is no clearly defined period. The series does not converge or diverge, and the trajectory is highly sensitive to initial conditions. This benchmark problem is used in the neural network and fuzzy modeling research communities.

Another dataset representing network traffic that has been used in the present work consists of 770 samples as shown in the figures 5.2 (a) and (b). First one shows the internet traffic pattern in terms of Number of packets per second and second one number of bytes per second at JNU LAN Router. Packets captured at JNU LAN on May 11, 2012 for nearly 1 hour. However, traffic shows data up to 800 seconds (14 minutes) because of sampling data over every second.



(a)



(b)

Fig 5.2 (a) Number of packets/second (b) Number of Bytes/second

5.3 Experiment and Results

The experiments are carried out to compare the performances of WM and WCV in terms of prediction and self-similarity modeled by WCV by varying the following parameters.

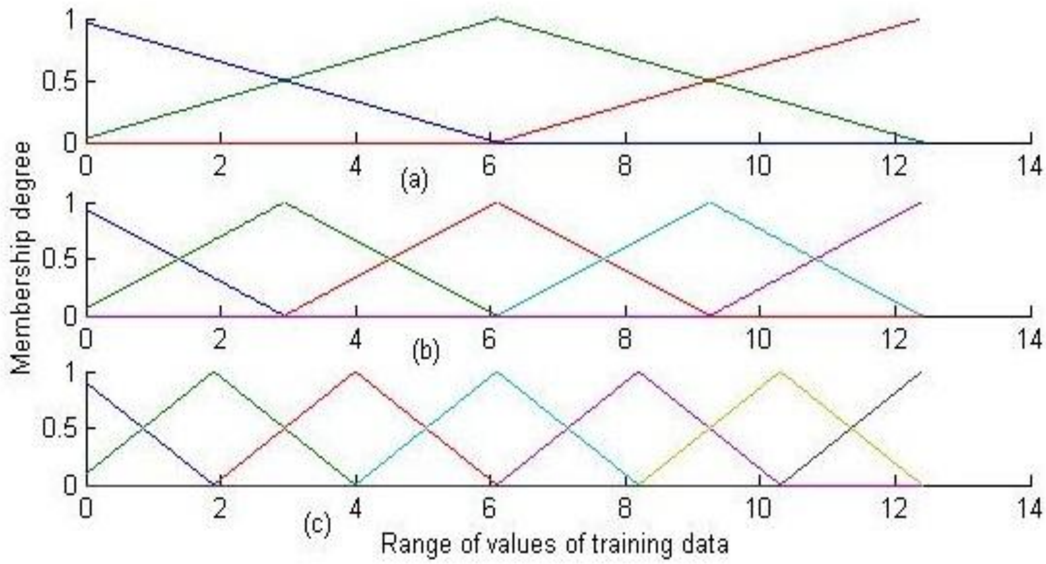


Fig. 5.3 Triangular membership functions for all inputs and outputs Three fuzzy sets (b) five fuzzy sets (c) seven fuzzy sets

I. Number of fuzzy sets keeping number of predictor variables constant.

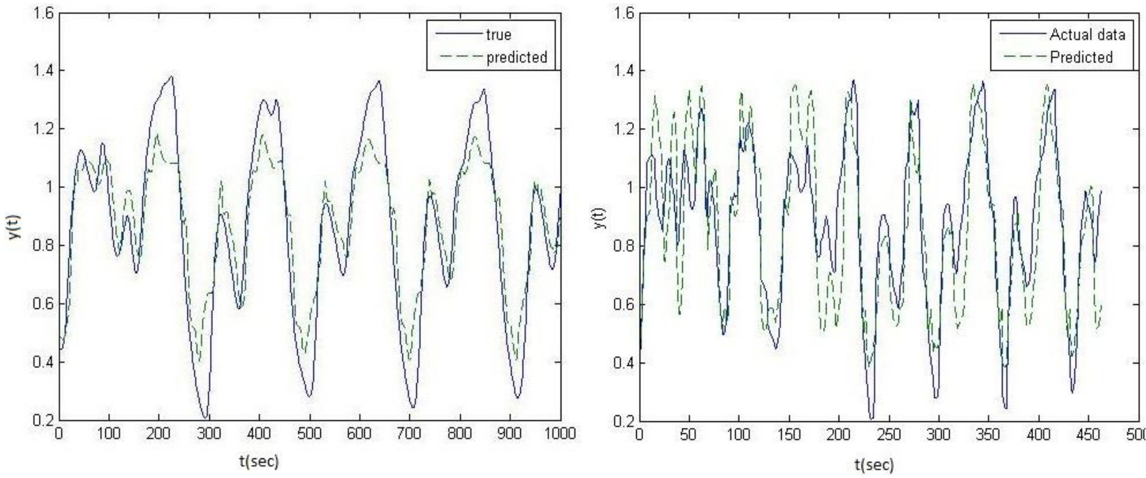
II. The Number of predictor variables keeping number of fuzzy sets constant.

The performance parameter to be measured is MSE (mean squared error) in the prediction. The experiment was carried out on two time series data namely (a) Mackey-Glass series (b) JNU LAN network traffic data as described in details in the beginning of this chapter. Comparison has been between the Wang method and the WCV method.

5.3.1 Prediction of Mackey-Glass Series

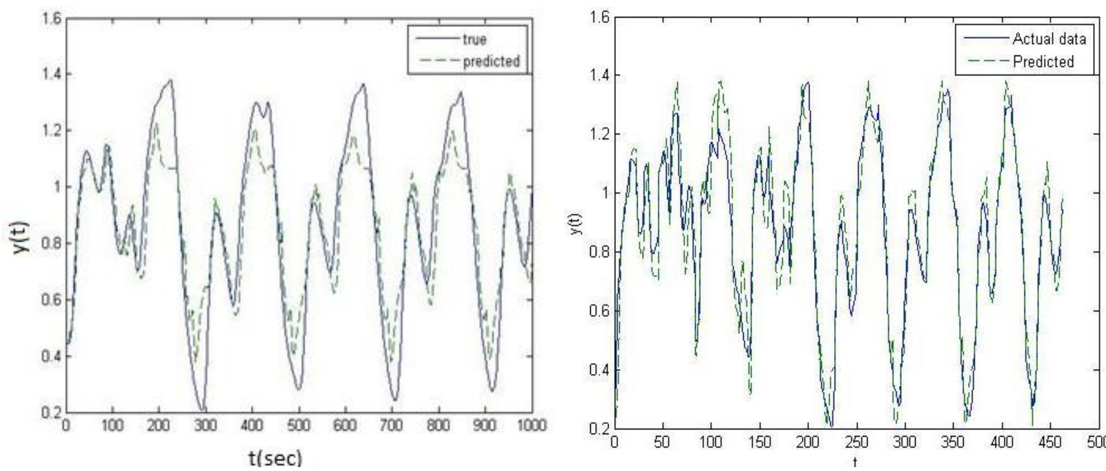
Varying Number of fuzzy sets keeping number of predictor variables constant

For the experiment on Mackey-Glass series prediction, number of fuzzy sets was varied from five to seven while keeping the number of predictors at four (as depicted in figure 5.3 (b) and (c)). The results are shown in figures 5.4 and 5.5 and table 5.1.



(a) (b)

Fig. 5.4 True vs. predicted data plot (a) Wang method (b) WCV method.



(a) (b)

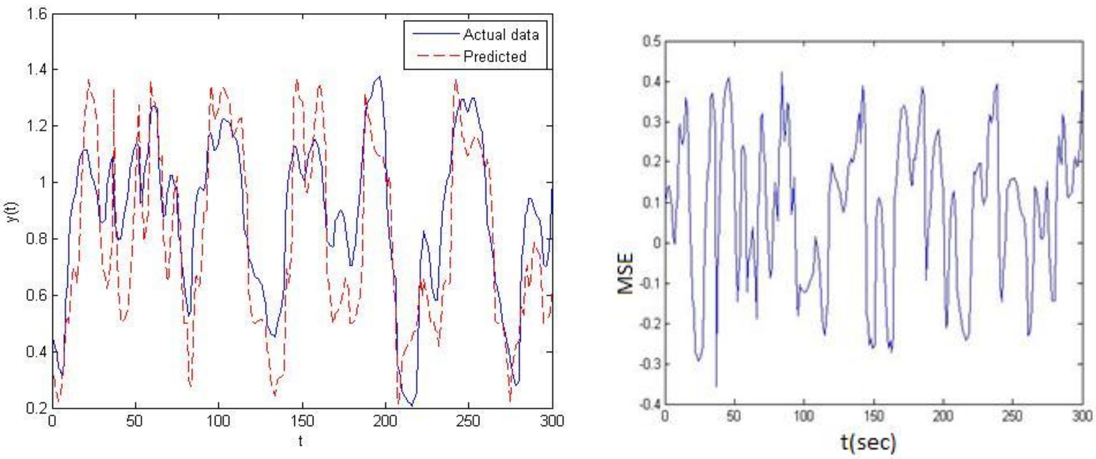
Fig. 5.5 True Vs. predicted data plot (a) Wang method (b) WCV method

Table 5.1 Comparision of WCV method with Wang method applied to Mackey-Glass series

No. of predictors	No. of fuzzy sets	Method	MSE in prediction
4	5	Wang	0.0740
		WCV	0.1178
	7	Wang	0.0586
		WCV	0.0760

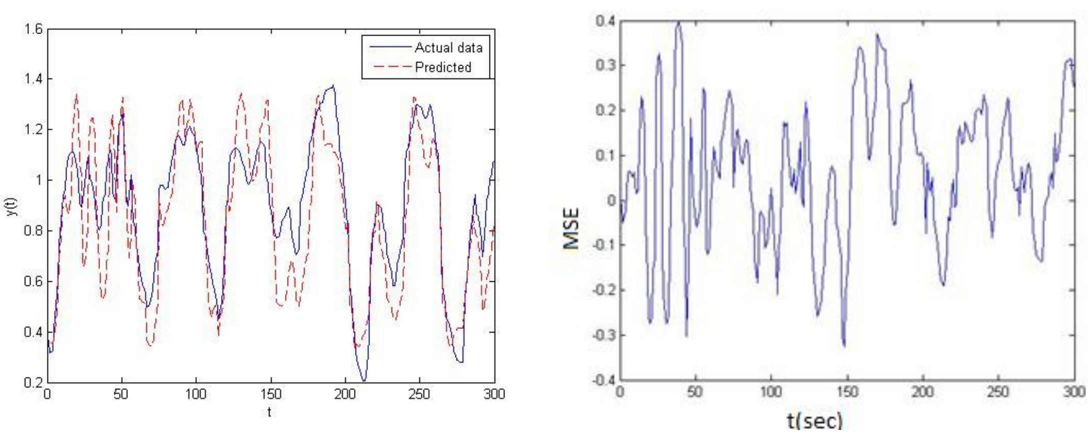
Varying number of predictor variable keeping number of fuzzy sets constant:

Now the number of fuzzy sets is kept at 5 as depicted in figure 5.3 (b) and number of predictor is varied from 3 to 4. The plots in figure 5.6 and 5.7 show the model prediction with different number of predictor variables used for prediction and corresponding MSE plot. The results are shown in table 5.2.



(a) (b)

Fig. 5.6 Prediction of Mackey-Glass series with two inputs and five fuzzy sets (a) True Vs. predicted (b) Absolute error plot.



(a) (b)

Fig. 5.7 Prediction of Mackey-Glass series with three inputs and five fuzzy sets (a) True Vs. predicted (b) Absolute error plot.

Table 5.2 Comparision of WCV method with Wang method applied to Mackey-Glass series.

No. of fuzzy sets	No. of predictors	Method	MSE in prediction
5	3	Wang	0.0708
		WCV	0.1258
	4	Wang	0.0740
		WCV	0.1178

5.3.2 Prediction of JNU LAN Traffic Load

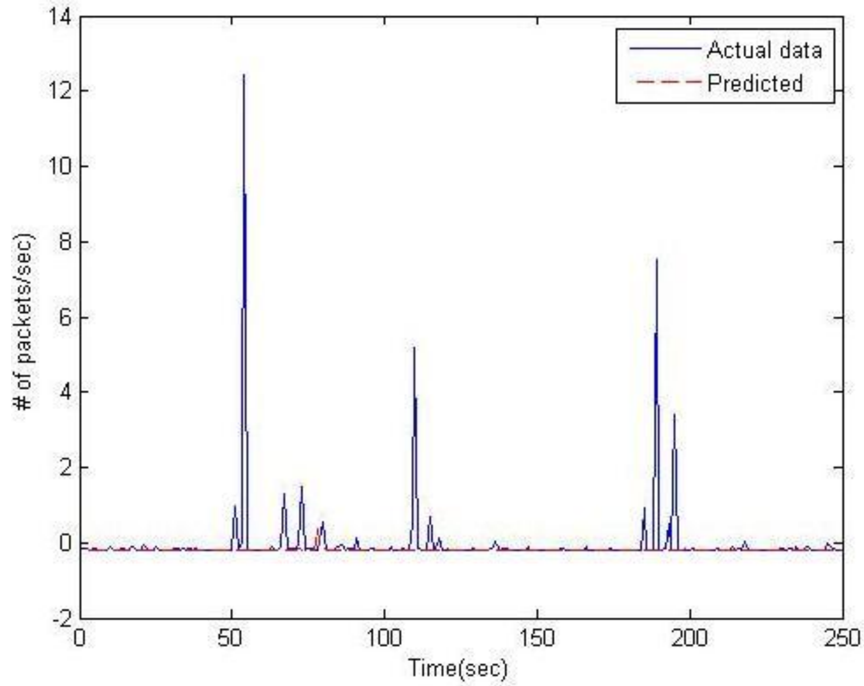
Network traffic data captured at JNU LAN consists of number of packets per second and number of bytes per second. In order to ensure fast convergence of the algorithm, mean normalization is used. That is, for each sample X(t) in the series normalized X'(t) is calculated is as follows:

$$X'(t)=(X(t)-\mu)/\sigma, \tag{5.7}$$

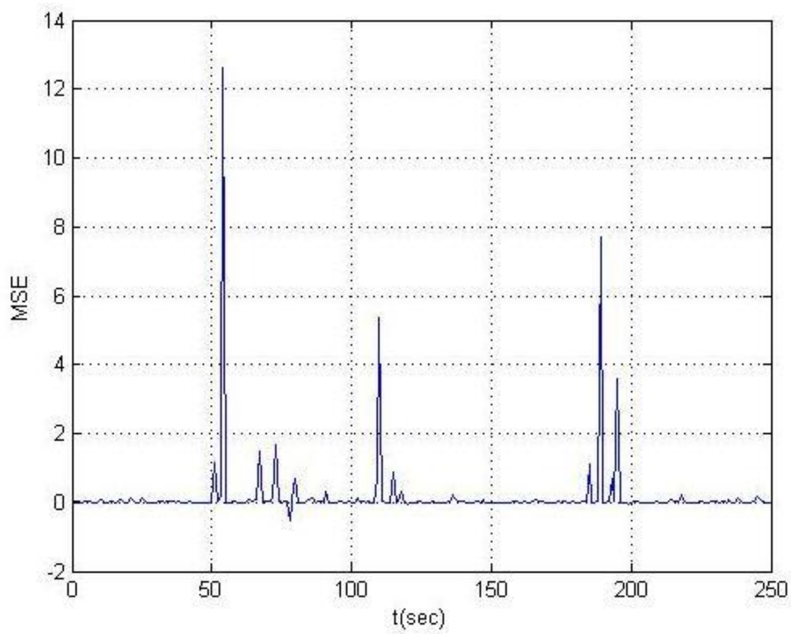
Where μ is mean value of the series sample and σ is standard deviation of the series sample.

Varying the Number of fuzzy sets keeping number of predictor variables constant

For the experimental purpose, we used 520 samples for training and 250 samples for testing. Number of fuzzy sets used is 2, 3, 4, and 5 as depicted in figure 5.3. Number of predictors is kept at 4. The various results are shown in table 5.3 and figures 5.8, 5.9, and 5.10.

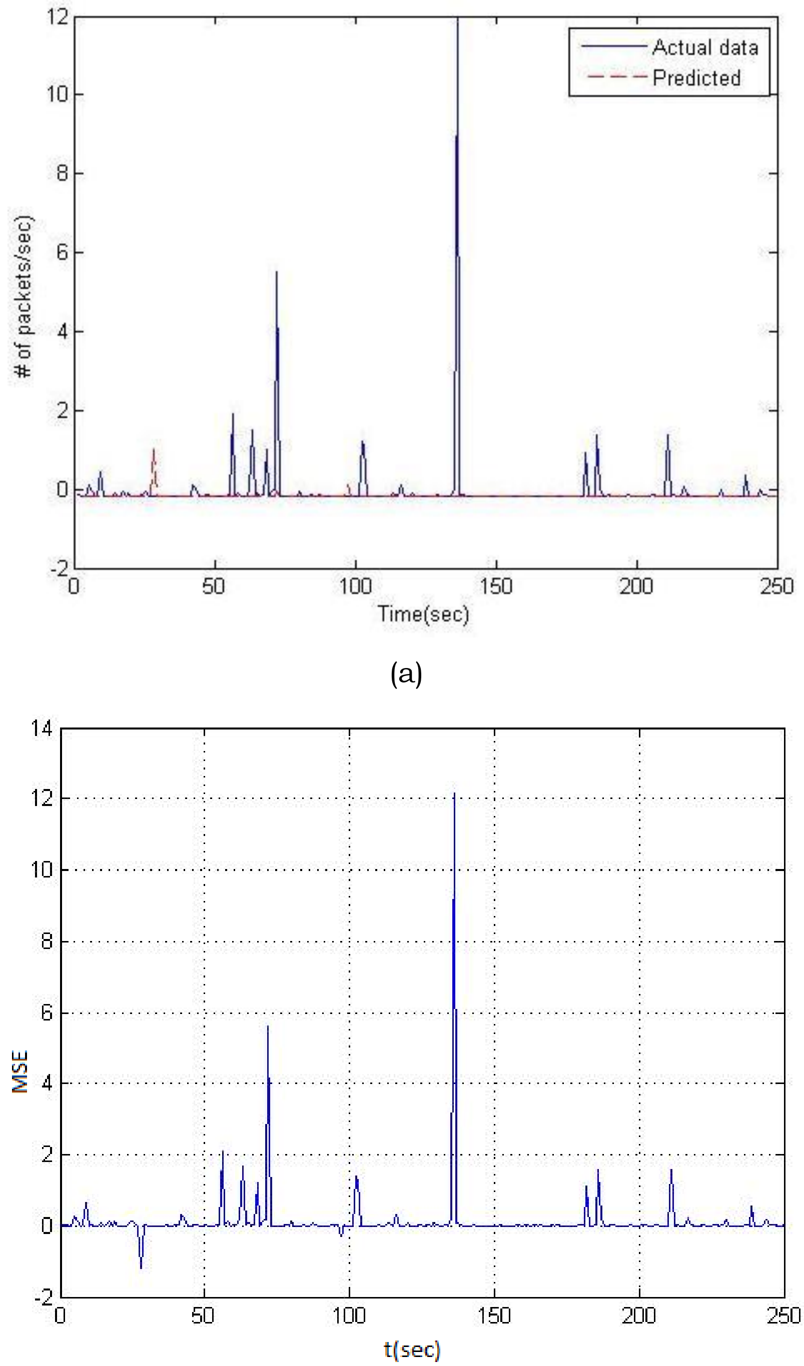


(a)



(b)

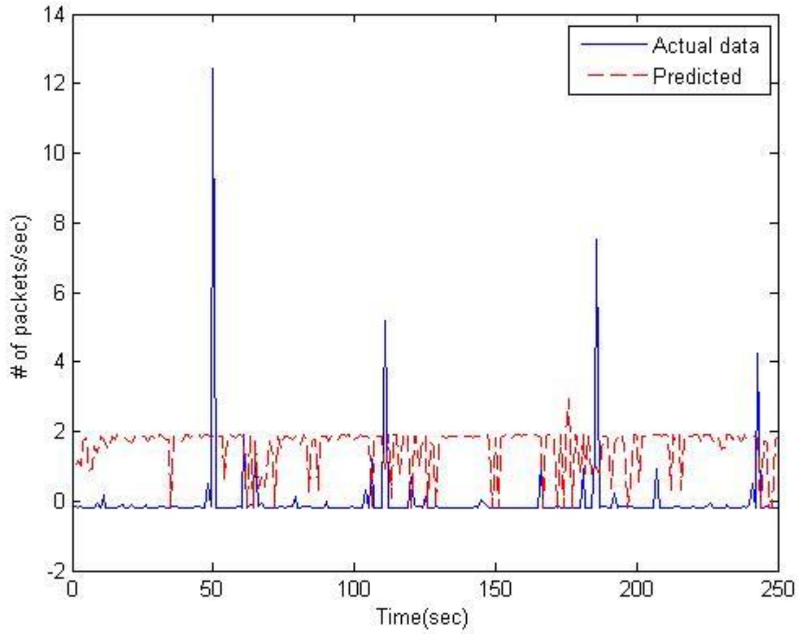
Fig. 5.8(a) JNU network traffic load prediction plot (b) MSE with 3 fuzzy sets and 4 predictor



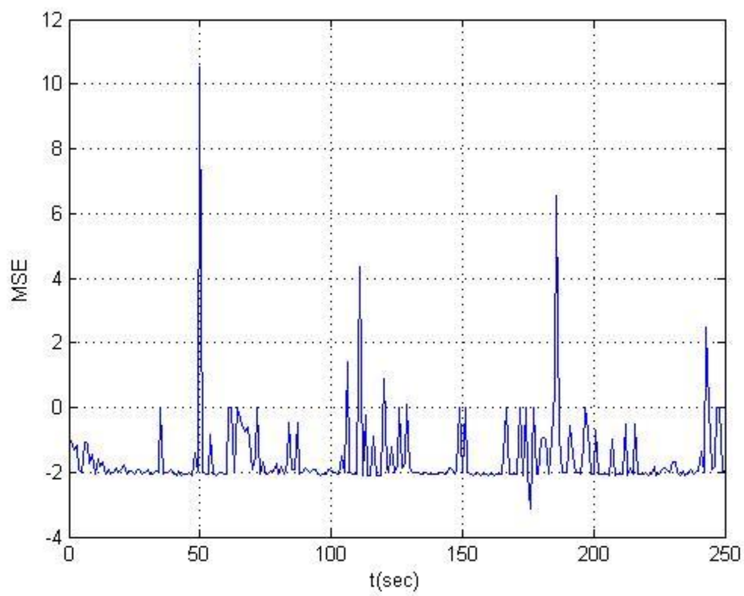
(a)

(b)

Fig 5.9(a) JNU network traffic load prediction plot (b) MSE with 5 fuzzy sets and 4 predictor



(a)



(b)

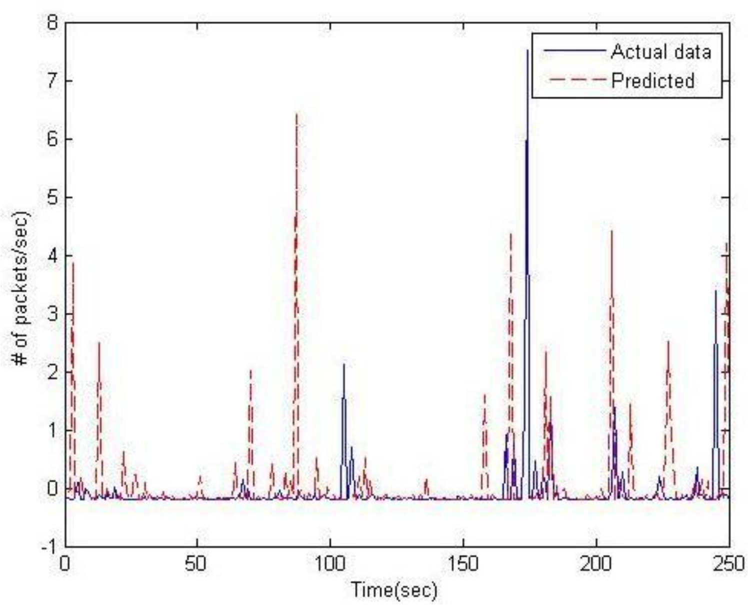
Fig 5.10 (a) JNU network traffic load prediction plot (b) MSE with 7 fuzzy sets and 4 predictor

Table 5.3 Comparision of WCV method with Wang method

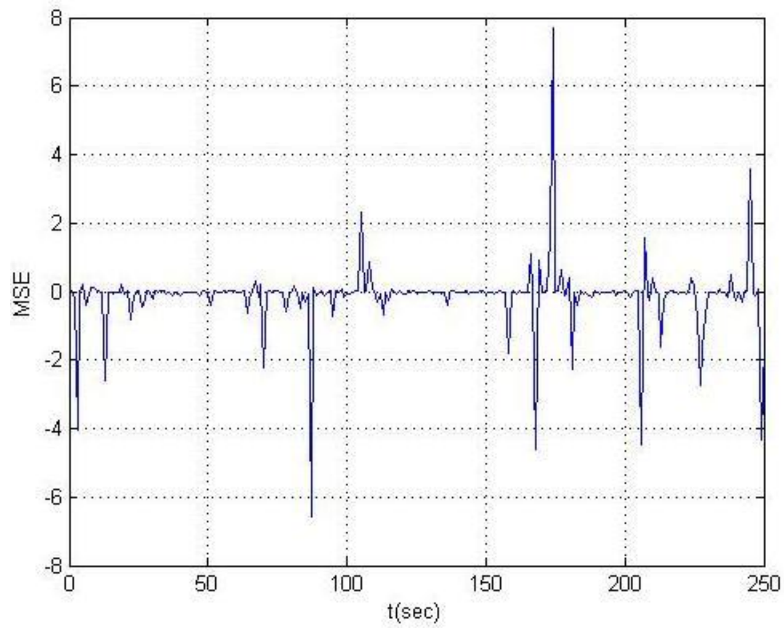
No. of predictor	No. of fuzzy sets	Method	MSE in prediction
4	2	Wang	0.3190
		WCV	0.2031
	3	Wang	0.3210
		WCV	0.1905
	4	Wang	0.2900
		WCV	5.0309
	5	Wang	0.3053
		WCV	2.6679

Varying the number of predictor variable keeping number of fuzzy set constant:

Number of fuzzy sets used is 5 as depicted in figure 5.3 and number of predictors is varied from 3 to 4. The corresponding results are shown in table 5.4 and figure 5.11 show the performance of FIS prediction with different number of predictor variables as shown against them.



(a)



(b)

Fig 5.11 (a) JNU network traffic load prediction plot (b) MSE with 5 fuzzy sets and 3 predictor

Table 5.4 Comparison of WCV method with Wang method

No. of fuzzy sets	No. of predictors	Method	MSE in prediction
5	3	Wang	0.3299
		WCV	2.0238
	4	Wang	0.3053
		WCV	2.6679

5.4 Self-Similarity in the Experimental Data

Variance-time plot following the approach discussed in chapter 4 is drawn and Hurst parameter is calculated. It is found that H (Hurst parameter) is about 0.5273 that is greater than 0.5 and hence degree of similarity is present in the series.

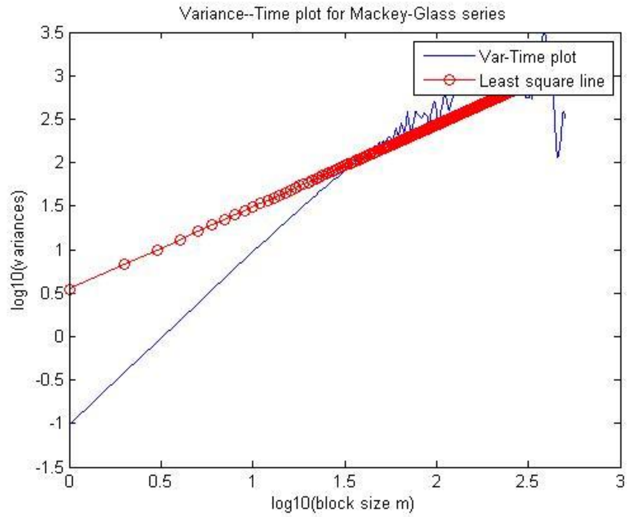
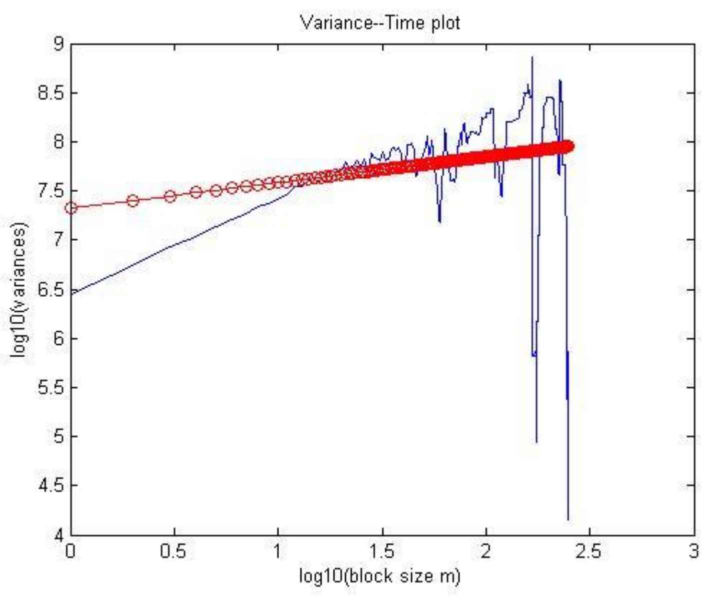


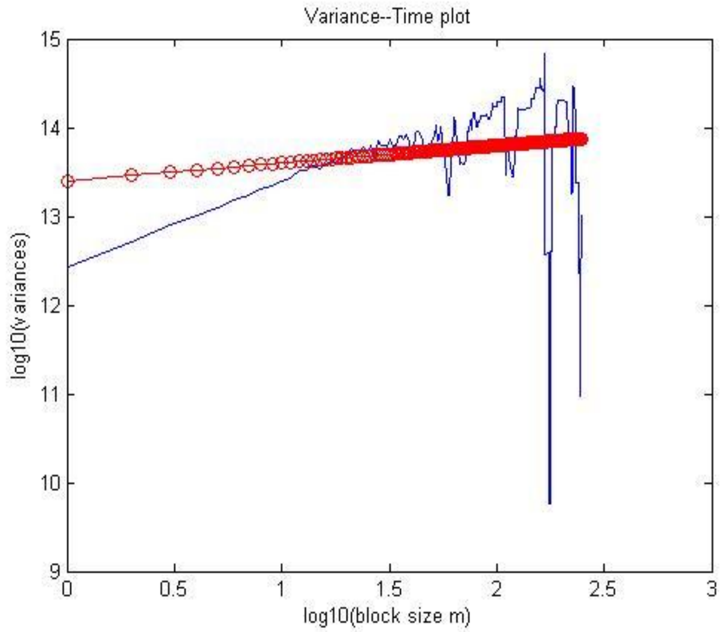
Fig. 5.12 Variance–time plot for Mackey-Glass series and calculation of Hurst parameter

Self-Similarity in Mackey-Glass series

Figure 5.12 shows the variance-time plot for Mackey-Glass series and value of Hurst parameter.



(a) H=0.8688



(b) H=0.9012

Fig 5.13 Estimation of Hurst parameter for JNU LAN traffic data (a) Variance-time plot for number of packets per second (b) Variance-time plot for number of Bytes per second

Self-Similarity in JNU LAN traffic data

The plots in figure 5.13 (a) and (b) depict the self-similar nature found in the JNU LAN traffic series which is indeed in accordance with statistical proof give in [36].

5.5 Analysis

we observe from table 5.1 that keeping the number of fuzzy sets at 5 and increasing the number of predictors from 3 to 4 in the FIS developed for Mackey-Glass series prediction give rise to increase in prediction error (MSE) by 0.32% in W&M method whereas there is decrease in prediction error by 0.80% in the WCV method. On the other hand, an observation in table 5.2 indicates that keeping the number of predictors at 4 and increasing the number of fuzzy sets from 5 to 7 bring about decrease in MSE by 1.54% in W&M method as compared to 4.18% by the WCV method.

The analysis of JNU LAN traffic prediction follows from tables 5.3 and 5.4. From table 5.3, we observe that with 4 predictors and 2 to 3 fuzzy sets, MSE in the prediction of JNU LAN traffic series by the WCV method is about 10% less than that given by W&M method. However, with 4 to 5 fuzzy sets and 4 predictors, W&M outperform the WCV method. An observation from table 5.4 implies that there is no advantage of increasing the number of predictors while keeping the number of fuzzy sets constant. From the figure 5.9, we observe that there is sudden spike in the MSE (mean squared error) plot. This is due to the fact that at this particular instant, there is burst of traffic flowing through the network.

In addition, as we increase number of fuzzy sets from 5 to 7, the MSE begins increasing instead of decreasing (fig. 5.10) which is against the fact Found in the prediction of Mackey-Glass series. This idiosyncratic behavior in the prediction can be explained on the grounds of the following point. LeaveMout sampling could not take care of test samples falling outside the rage of training sample and could not be properly fuzzified. Further, Self-similarity in JNU LAN agrees with the fact observed in the literature.

Chapter 6

Conclusion and Future work

6.1 Conclusion

Understanding the nature of traffic in high-speed, high-bandwidth communications systems is essential for engineering, operations, and performance evaluation of these networks. As a first step toward this juncture, it is important to know the traffic behavior of some of the expected major contributors to future high-speed network traffic. The fuzzy inference system (FIS) developed in this dissertation following the Wang method employs cross-validation. The FIS is then used for prediction of Mackey-Glass series. Prediction results show that FIS performs well for prediction of JNU LAN with two to three fuzzy sets. Secondly better accuracy is achieved by increasing the number of fuzzy sets for Mackey-Glass but not for Network traffic data. Thirdly, the same performance can be achieved by increasing the number of input variables and keeping the number of fuzzy sets constant. Fourthly, Mackey-Glass series is self-similar (this has been demonstrated by variance-time plot in fig 5.12). The same is found for JNU LAN traffic (fig 5.13). And finally, Predicted traffic resembles actual traffic for small and medium time-scale (i.e. in seconds and minutes) (fig 5.11 a).

6.2 Future Work

There still remain many issues to be tackled as future research. For example, how to use rule extrapolation for incorporating samples whose fuzzification yield values that fall outside the range of predictor variable. In addition, researcher can employ interval type-2 fuzzy inference system for handling random noise in the data. Using FIS as a network traffic controller is another direction of future research.

References

[1] DatCat, Datcat: Internet measurement data catalog (2008), http://www.datcat.org.

[2] Endace Limited, DAG Network Monitoring Cards (2008), http://www.endace.com/our-products/dag-network-monitoring-cards/

[3] R., Albert, H., Jeong, & A.L, Barab´asi,.: Diameter of the world wide web. Nature 401, 130–131 (1999)

[4] Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis, 2nd edn. Cambridge University Press, Cambridge (2004)

[5] Lowekamp, B.B.: Combining active and passive network measurements to build scalable monitoring systems on the grid. ACM SIGMETRICS Performance Evaluation Review Special issue on grid computing 30(4), 19–26 (2003)

[6] Weigend, A., Gershenfeld, N.: Times Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley Publishing Company, Reading (1994) ISBN: 0201626020

[7] Box, G., Jenkins, G.M., Reinsel, G.: Time Series Analysis: Forecasting & Control, 3rd edn. Prentice Hall, Englewood Cliffs (1994) ISBN: 0130607746

[8] Chatfield, C., The Analysis of Time Series. An Introduction, 6th edn. CRC Press, BocaRaton (2003) ISBN: 1-58488-317-0

[9] Sang, A. and Li, S., "A predictability analysis of network traffic", Computer networks, vol. 39, pp. 329-345, 2002.

[10] Corradi, M., Garroppo, R.G., Giordano, S., and Pagano, M., "Analysis of f-ARIMA processes in the modeling of broadband traffic", ICC'01, vol.3, pp.964-968, 2001.

[11] Rudolf H. Riedi, Vinay J. Ribeiro, Matthew S. Crouse, Richard G. Baraniuk, "Network Traffic.

[12] You, C. and Chandra, K., "Time series models for Interent data traffic", Proceedings of the 24th Conference on Local Computer Networks (LCN 99), pp. 164-171.

[13] Gao, Y., He, G., and Hou, J.C., "On Leveraging traffic predictability in active queue management", Proc. IEEE INFOCOM 2002, June 2002.

[14] Zhou, B., D. He, Z. Sun and W.H. Ng, 2006. Network traffic modeling and prediction with ARIMA/GARTH, HET-NETs Conference 11-13, University of Surrey Guildford, Surrey United Kingdom, pp: 1-10.ttp://www.comp.brad.ac.uk/het-net/HET-NETs05/ReadCamera05/P08.pdf

[15] Masugi, M. and T. Takuma, 2007. Using a Volterra system model to analyze nonlinear response in video-packet transmission over IP networks. Commun. Nonlinear Sci. Numer. Simulat., 12: 411-421. DOI: 0.1016/j.cnsns.2005.03.009

[16] Antari, J. and A. Zeroual, 2008. Blind identification of Diagonal of non linear process using cumulants: Application to modeling internet traffic. Int. J. Inform. Commun. Technol., 1: 37-42. http://www.serialspublications.com/contentnormal. asp?jid=354&jtype=1.

[17] Kim, S., December 2011 "Forecasting Internet Traffic by Using Seasonal GARCH Models", Journal Of Communications And Networks, Vol. 13, No. 6.

[18] B. Krithikaivasan, Y. Zeng, and D. Medhi, September 2008, "Generalized cost function based forecasting for periodically measured nonstationary traffic," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 38, pp. 1105–1117.

[19] Chabaa, S. and Zeroual, A., 2009, Predicting Packet Transmission Data over IP Networks Using Adaptive Neuro-Fuzzy Inference Systems, Journal of Computer Science 5 (2), pp. 123-130.

[20] Pouzols, F.M., Lopez, D.R. and Barros, A.B., 2011, Modeling Time Series by Means of Fuzzy Inference Systems, Kacprzyk, J., ed., *Modeling*

and control of network traffic using computational intelligence, Springer, Verlag Berlin Heidelberg, pp. 53-83.

[21] Jang, J.S.R. and C.T. Sun, 1995. Neuro-fuzzy modeling and control. Proc. IEEE, 3: 378-405. DOI: 10.1109/5.364486.

[22] Perendeci, A., S. Arslan, S.S. Celebi and A. Tanylaç, 2008. Prediction of effluent quality of an anaerobic treatment plant under unsteady state through ANFIS modeling with on-line input variables. Chem. Eng. J., 145: 78-85. DOI: 10.1016/j.cej.2008.03.008

[23] Han and Kamber, 2006.

[24] Allman, M.: Measuring End-to-End Bulk Transfer Capacity. In: ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, USA, pp. 139–143 (2001)

[25] Montesino-Pouzols, F., Barriga, A., Lopez, D.R., S´anchez-Solano, S.: Performance Analysis of Computer Networks. In: Encyclopedia of Networked and Virtual Organizations, Information Science Reference (an imprint of IGI Global), Hershey, New York, USA, pp. 1216–1222 (2008) ISBN: 978-1-59904-885-7

[26] Fowler, H.J., Leland, W.E.: Local Area Network Traffic Characteristics, with Impli-cations for Broadband Network Congestion Management. IEEE Journal on Selected Areas in Communication 9(7), 1139–1149 (1991).

[27] N. Packard et al. phys. Rev. let. 45, No. 9, p712 (1980).

[28] Taken, F. , “Detecting Strange Attractor In Turbulence,” Lecture Notes in Mathematics, O. Rand, L. Young (editors), Springer Berlin, P366 (1981).

[29] Lapedes, A. and Farber, R., 1987, Nonlinear signal processing using neural network: Prediction and System Modeling, LA-UR-87-2662.

[30] Guillaume, S., Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review, IEEE transactions on fuzzy systems, vol. 9, no. 3, June 2001.

[31] Yager, R.R.: A New Approach to the Summarization of Data. Information Sciences 28, 69–86 (1982)

[32] Zadeh, L.A.: A Computational Approach to Fuzzy Quantifiers in Natural Languages. Computers and Mathematics with Applications 9, 149–184 (1983)

[33] Kacprzyk, J., Zadrozny, S.: Linguistic Database Summaries and Their Protoforms: Towards Natural Language Based Knowledge Discovery Tools. Information Sciences 173(4), 281–304 (2005)

[34] Wang, Li. And Mendel, J.M., 1992, Generating fuzzy rules by learning from examples, IEEE Transactions on System, Man and cybernatics, Vol.12, No. 6.

[35] Wang, Li., The WM Method Completed: A Flexible Fuzzy System Approach to Data Mining, IEEE transactions on fuzzy systems, vol. 11, no. 6, december 2003.

[36] M., Ghaderi, (2003), On the Relevance of Self-Similarity in Network Traffic Prediction. Tech. Rep. CS-2003-28, School of Computer Science, University of Waterloo.

Appendix

We know that degree of rule is defined as

$$D(Rule)=M_A(x_1)M_B(x_2) \qquad \text{For 2 input case.}$$

M denotes membership value(sometimes denoted by μ). If there are n input-output pairs, then output control has membership value given

$$M^i = \prod_{1\leq j\leq n}[m_j^i(x_j)] \qquad (1)$$

Centroid method for defuzzification is given by:

$$y = \frac{\sum_{i=1}^{K} M^i * \bar{y}^i}{\sum_{i=1}^{K} M^i} \qquad (2)$$

Plugging the value of M^i from Eqn. (1) to Eqn. (2) we get the following required result.

$$y = \frac{\sum_{i=1}^{K} \prod_{1\leq j\leq n}[m_j^i(x_j)] * \bar{y}^i}{\sum_{i=1}^{K} \prod_{1\leq j\leq n}[m_j^i(x_j)]}$$

This is the same formula with little changes in notations. Hence proved