DEVELOPMENT OF AN IMPROVED HIDDEN MARKOV MODEL BASED FUZZY TIME SERIES FORECASTING MODEL USING GENETIC ALGORITHM

BY

Bashir Baba ABBA B.Eng. (University of Maiduguri, 2007)

P14EGCM8013

A Dissertation Submitted to the Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria, in Partial Fulfillment of the Requirements for the Award of Master of Science (M.Sc) Degree in Computer Engineering

October, 2016

DECLARATION

I ABBA Bashir Baba, hereby declare that this Dissertation titled "DEVELOPMENT OF AN IMPROVED HIDDEN MARKOV MODEL BASED FUZZY TIME SERIES FORECASTING MODEL USING GENETIC ALGORITHM" was carried out by me under the supervision of Professor M. B. Mu'azu and Dr. E. A. Adedokun as part of the requirements for the award of degree of Master of Science in Computer Engineering. To the best of my knowledge, this research work has never been submitted anywhere for the award of any certificate. All literatures used and cited in this research, are duly acknowledged in the list of reference pages.

Bashir Baba ABBA _____ _____

Signature Date

CERTIFICATION

This Dissertation titled "DEVELOPMENT OF AN IMPROVED HIDDEN MARKOV MODEL BASED FUZZY TIME SERIES FORECASTING MODEL USING GENETIC ALGORITHM" meets the requirements for the award of degree of Master of Science (MSc) in Computer Engineering by Ahmadu Bello University, Zaria and it is approved for its contribution to knowledge and literary presentation.

Chairman, Supervisory Committee	_____	_____
(Professor M. B. Mu'azu)	**Signature**	**Date**
Member, Supervisory Committee		
(Dr. E. A. Adedokun)	_____	_____
	Signature	**Date**
Head of Department		
(Dr. Y. Jibril)	_____	_____
	Signature	**Date**
Dean, School of Postgraduate Studies		
(Professor S. Z. Abubakar)	_____	_____
	Signature	**Date**

DEDICATION

This research work is dedicated to my Parents.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the help of GOD, the author of knowledge and wisdom. I am highly indebted to you.

I wish to express my deepest gratitude to my supervisor Professor M. B. Mu'azu, for his tireless efforts, valuable guidance and constant supervision towards the success of this work. The completion of this work could not have been possible without your constant participation and assistance. Thank you very much sir. My thanks also go to my co-supervisor Dr. E. A. Adedokun for his constant encouragement and expertise. My deep appreciation goes to the Computer & Control Research Group for their keen interest, suggestion and constructive criticism during all those stages of my presentations.

I acknowledge with thanks all the lecturers of Electrical and Computer Engineering, Ahmadu Bello University. Professor B. G. Bajoga, Professor B. Jimoh, Dr. A.M.S. Tekanyi, Dr. S. M. Sani, Dr. A. D. Usman, Dr. S. Garba, Dr. Y. Jibril, Dr. Abu-Bilal, Dr. T. H. Sikiru Engr. Zanna, Engr. Mu'azu, Engr. A. I. Abdullahi, Engr. Bashir, Engr. Josiah and most especially, those whose names could not be mentioned. My deepest appreciation also goes to Abdullahi Tukur for all the administrative support towards me and the entire students of Department of Electrical and Computer Engineering.

I am extremely thankful to my special friend Salawudeen A. Tijani for his tireless support during the course of the program. Special thanks to Engr. Francis, Engr. Hussein, Engr. Muktar, Engr. Atuman, Engr. Umar, Engr. Busayo and the entire control group members. Their continuous support and contributions toward the success of this work would never be forgotten, may God

reward them and strengthen our friendship. I am very much thankful to all my course mates, Hafiz Abdulrahman, Lami, Mohammed, Dikko, may God reward you all.

Above all, I am overly indebted to my parents Alhaji Baba Abba and Hajja Hauwa Baba, your countless love, endless support, kind and understanding spirit will always be appreciated. Thank you very much. To my brothers Alhaji Babagana, Alhaji Shettima, Alhaji Mohammed, Babakura, Ibrahim, Shettima, Ahmed, Mahiru and Mustapha Abba. To my lovely sisters Hajja Yagana, Hajja Yachilla, Yagana, Falmata and Fatima Abba I owe you all a big thank you. To the family of Dr. Shettima Bukar Abba and Engr. Ahmed Ibrahim, I am highly grateful for your constant support all those that have directly and indirectly contributed to the success of this work. May God bless you real good.

Bashir Baba ABBA

October, 2016.

TABLE OF CONTENTS

Page

Title Page i

Declaration ii

Certification iii

Dedication iv

Acknowledgement v

Table of Contents vii

List of Tables xi

List of Figures xii

List of Abbreviations xiii

List of Appendices xiv

Abstract xvi

CHAPTER ONE: INTRODUCTION

1.1 Background 1

1.2 Motivation 3

1.3 Statement of Problem. 3

1.4 Significance of Research. 4

1.5 Aim and Objective. 4

1.6 Methodology. 5

1.7 Dissertation Organization. 5

CHAPTER TWO: LITERATURE REVIEW

2.1INTRODUCTION. 7

2.2 Review of Fundamentals Concepts. 7

2.2.1 Times Series Models and Analysis. 7

2.2.1.1 Classification of Time Series. 7

2.2.1.2 Components of time series. 8

2.2.2 Basic Concepts of Fuzzy Time Series. 9

2.2.3 Basic Steps of Fuzzy Time Series Forecasting. 10

2.2.4 Hidden Markov Model (HMM) and its approach to Time Series Data. . . . 11

*2.2.*4.1 Hidden Markov model. 11

2.2.4.2 Components of HMM. 14

2.2.5 The Standard HMM based FTS Forecasting Approach. 15

2.2.6 Parameter Estimation Methods of HMM. 20

2.2.7 Smoothing. 22

2.2.8 Performance Metrics. 25

2.3 Review of Similar Works.. 26

CHAPTER THREE: MATERIAL AND METHODS

3.1 Introduction. 32

3.2 Development of the Improved HMM based FTS Forecasting Model using GA for the Re-estimation of the Inner Fuzzy Relations 32

3.2.1 Obtaining Historical Time Series Data.. 33

3.2.2 Determination of the Universe of Discourse 34

3.2.3 Partition the Universe of Discourse into Several Even-Length Intervals. . . 34

3.2.4 Defining the Fuzzy Sets on the Universe of Discourse. . . . 35

3.2.5 Fuzzifying the Time Series Data. 35

3.2.6 Building the HMM Model to Estimate the Fuzzy Relations. . . . 36

3.2.7 Developing the GA-HMM Model to Re-estimate Parameters. . . 36

3.2.8 Smoothing HMM Model Parameters. 38

3.2.9 Calculating Forecast Outputs. 38

3.2.10 Defuzzifying the Forecasting Outputs. 41

3.3 Performance Evaluation. 41

3.4 Application of the Developed GA-HMM based FTS Model to Forecast Short-term Internet Traffic Data of ABU, Zaria . . . 42

3.4.1 Obtaining Historical Time Series Data. 42

3.4.2 Determination of the Universe of Discourse 42

3.4.3 Partition the Universe of Discourse into Several Even-Length Intervals. 43

3.4.4 Defining the Fuzzy Sets on the Universe of Discourse. . . 43

3.4.5 Fuzzifying the Time Series Data. 44

3.4.6 Building the HMM Model to Estimate the Fuzzy Relations. . . 44

3.4.7 Developing the GA-HMM Model to Re-estimate Parameters. . 45

3.4.8 Smoothing HMM Model Parameters. 46

3.4.9 Calculating Forecast Outputs. 46

3.4.10 Defuzzifying the Forecasting Outputs. 46

CHAPTER FOUR: RESULT AND DISCUSSION

4.1 Introduction. 47

4.2 Obtained Results for the Developed GA-HMM based FTS Forecasting Model . . 47

4.2.1 Forecast Outputs of the Developed Model. 47

4.2.2 Results of the Defuzzified the Forecasting Outputs. 47

4.3 Performance Evaluation on the Bivariate HMM-FTS data. 50

4.4 Application of the GA-HMM FTS Model on the Internet Traffic of ABU, Zaria.. . 51

4.4.1 Forecast outputs of the developed model. 51

4.4.2 Results of the Defuzzified the Forecasting Outputs 51

4.4.3 Performance Evaluation on the Bivariate HMM-FTS data. 52

4.5 Computer Specification. 53

4.6 Genetic Algorithm Parameter Specification. 53

CHAPTER FIVE: CONCLUSION SUMMARY AND RECOMMENDATION

5.1 Summary 54

5.2 Conclusion. 54

5.3 Significant Contributions. 55

5.4 Limitations. 55

5.5 Recommendations for Further Work 55

References. 56

LIST OF TABLES

Table 3.1: Universes of Discourse for Hidden and Observation Variables. . . 34

Table 3.2: Interval Lengths for the Hidden and Observation Variables. . . 35

Table 3.3: Linguistic Variables for Temperature and Cloud Density. . . 35

Table 3.4: Internet Traffic Universes of Discourse for Hidden and Observation Variables 43

Table 3.5: Internet Traffic Interval Lengths for the Hidden and Observation Variables. . 43

Table 3.6: Internet Traffic Linguistic Variables for Temperature and Cloud Density. . 44

Table 4.1: Performance Evaluation of the Proposed Model and Cheng & Sheng (2012). . 50

Table 4.2 Performance Evaluation of the Forecasted ABU, Zaria Internet Traffic 52

Table 4.3: Computer Specification. 54

Table 4.4: Parameter Values for Genetic Algorithm 53

LIST OF FIGURES

Figure 2.1: Block Diagram Showing the Steps Generic FTS Model. . . . 11

Figure 2.2: Markov Model for Tossing One Coin 13

Figure 2.3: Hidden Markov Model 13

Figure 2.4: Block Diagram Showing the Steps Standard HMM based FTS Model. . 20

Figure 2.5: Taxonomy of Parameter Estimation Methods 22

Figure 3.1: Bloc Diagram of the Proposed Model. 33

Figure 4.1: Comparison between Actual and Forecasted Values for June, 1996 . . 48

Figure 4.2: Comparison between Actual and Forecasted Values for July, 1996. . 48

Figure 4.3: Comparison between Actual and Forecasted Values for August, 1996. . 49

Figure 4.4: Comparison between Actual and Forecasted Values for September, 1996. . 49

Figure 4.5: Comparison of Actual and Predicted Internet Traffic for 21st March, 2016. . 51

Figure 4.6: Comparison of Actual and Predicted Internet Traffic for 22nd March, 2016. . 52

LIST OF ABBREVIATIONS

Acronym	Meaning
ABU . .	Ahmadu Bello University
AFEP . .	Average Forecasting Error Percentage
ARMA .	Autoregressive Moving Average
ASR . .	Automatic Signal Recognition
FLRG . .	Fuzzy Logic Relationship Group
FTS . .	Fuzzy Time Series
GA . .	Genetic Algorithm
HMM . .	Hidden Markov Model
MAE . .	Mean Absolute Error
MM . .	Markov Model
MSE . .	Mean Square Error
PMAD. .	Percent Mean Absolute Deviation
RMSE . .	Root Mean Square Error

LIST OF APPENDICES

Appendix A1: Training Data of Average Temperature and Cloud Density for 1993. . 59

Appendix A2: Training Data of Average Temperature and Cloud Density for 1994. . 60

Appendix A3: Training Data of Average Temperature and Cloud Density for 1995. . 61

Appendix B: Testing Data of Average Temperature and Cloud Density for 1996. . 62

Appendix C1: Fuzzified Training Data of Average Temperature and Cloud Density for June & July, 1993. 63

Appendix C2: Fuzzified Training Data of Average Temperature and Cloud Density for August and September 1993. 64

Appendix C3: Fuzzified Training Data of Average Temperature and Cloud Density for June and July 1994. 65

Appendix C4: Fuzzified Training Data of Average Temperature and Cloud Density for August and September 1994. 66

Appendix C5: Fuzzified Training Data of Average Temperature and Cloud Density for June and July, 1995. 67

Appendix C6: Fuzzified Training Data of Average Temperature and Cloud Density for August and September 1995. 68

Appendix D1: Fuzzified Testing Data of Average Temperature and Cloud Density for June and July, 1996. 69

Appendix D2: Fuzzified Testing Data of Average Temperature and Cloud Density for August and September. 1996. 70

Appendix E: Temperature Forecast Matlab Program. 71

Appendix F1: Forecasted Values of the Average Temperature for June and July 1996. . 97

Appendix F2: Forecasted Values of the Average Temperature August and September, 1996. 98

Appendix G: ABU Internet Traffic Training Data Set. 99

Appendix H: ABU Fuzzified Internet Traffic Training Data. 102

Appendix I: ABU Internet Traffic Testing Data Set. 107

Appendix J: ABU Fuzzified Internet Traffic Testing Data. 108

Appendix K: Internet Traffic Data Forecast Matlab Program. 109

Appendix L: Abu Internet Traffic Forecasted Values. 125

ABSTRACT

This research is aimed at the development of an improved Hidden Markov Model (HMM) based fuzzy time series (FTS) forecasting model using Genetic Algorithm (GA). In order to improve forecasting performance, a GA and HMM is developed to optimize and properly estimate membership values in the fuzzy relationship matrix in the fuzzy inference stage. Monte Carlo simulation was applied to estimate the stochastic outcome of the data and further improve the model reflection of real data and randomness. The developed model was implemented in MATLAB R2015a and tested with the Cheng and Sheng's data of the daily average temperature and cloud density of Taipei as a benchmark data for bivariate FTS. The performance of the proposed GA-HMM based FTS was evaluated using Mean Square Error (MSE) and the Average Forecasting Error Percentage (AFEP) as metrics. The results showed that the developed model had an MSE of 0.5976 and AFEP of 1.8673 for the bivariate benchmark HMM-FTS data of the daily average temperature and cloud density of Taipei, Taiwan as against 0.933 and 2.7464 respectively obtained from (Li and Cheng, 2012). This amounts to an improvement of 35% and 32% for the MSE and AFEP respectively. The model was also applied to forecast the short term Internet traffic data of ABU, Zaria. Simulation result shows an MSE and AFEP values of 68.32392 and 0.08904 respectively, indicating a good forecasting performance considering the large size of these traffics and their randomness. Thus, these results demonstrate both the superiority of the proposed GA-HMM based FTS model at making good forecasts considering the large sizes of these traffics and their randomness and also its robustness in adaptation to time series of different structural and statistical characteristics.

CHAPTER ONE

INTRODUCTION

1.1 Background

Time series is simply a collection of quantitative variables at regular intervals of time. Either discrete or continuous, time series are always both nonlinear and non-stationary since they are sample functions realized from processes that are always stochastic (Subanar & Abadi, 2011). Time series forecasting plays an important role in a great variety of applications, such as predicting university enrollments, stock prices, rainfall, blood pressure, and so on. Such forecasting usually uses a sequence of past data points which are typically measured successively for forecasting future outcomes (Sheng *et al*., 2009). Various techniques for time series forecasting have been evolved in recent decades. Compared with other models, Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average ARIMA-based models are prominent and highly useful. However, they cannot deal with time series vagueness and linguistic terms (Song & Chissom, 1993). In addition, these statistical methods could not perform appropriately on time series with a small amount of data (Tsaur, *et al*., 2005). Furthermore, the necessary conditions for applying the conventional time series with probabilistic models which requires some assumptions such as number of observations, normal distribution, and linearity (Egrioglu, 2014). Thus, these approaches results in misleading forecasting results when these assumptions are not satisfied. Therefore, non-probabilistic approaches have been put forward as an alternative to probabilistic time series forecasting models (Egrioglu, 2015).

To deal with such deficiencies, fuzzy time series (FTS) has been developed and widely applied (Radmehr & Gharneh, 2012). In recent years, FTS models have attracted the attention of many researchers because of their advantages: better performance in some real forecasting

problems (Song & Chissom, 1993), dealing with data in linguistic terms (Song & Chissom, 1993), and their ability to integrate with heuristic knowledge and models (Huarng, 2001).

One of the most important issues in FTS models is the determination of the fuzzy relations (Egrioglu *et.al.*, 2013). In the literature, many methods have been used for determining fuzzy relations. These include fuzzy logic relationship group (FLRG), artificial neural networks, fuzzy relation matrices obtained from some fuzzy set operations, particle swarm optimization and genetic algorithms (Egrioglu, 2014). The most commonly used method is the fuzzy logic relationship group as it does not need to perform complex matrix operations when the FLRG tables are formed. However, when the FLRG tables are exploited, membership values of fuzzy sets are ignored as only fuzzy sets' elements with the highest membership value are considered (Aladag *et al*., 2012). This situation causes information loss and it may affect the forecasting performance, negatively. Since the fuzzy relationships can be nonlinear and complex, an intelligent method is needed to calculate these relationships.

To deal with such deficiencies, Hidden Markov Model (HMM) have been developed and applied in formulating the fuzzy relationship, where the model parameters are estimated using a conventional search technique, known as the Baum–Welch algorithm (Li and Cheng, 2010). Since parameter learning in Hidden Markov Model using the Bawm-Welch algorithm is prone to be trapped in the local optima, it has become imperative that a technique for finding enhanced estimates of the fuzzy relations and also avoiding the local optima is required.

In recent years, artificial intelligence techniques have been used in different stages of fuzzy time series methods (Egrioglu, 2014). In this study, a Genetic Algorithm (GA) method was applied to obtain the optimal estimate of the inner fuzzy relations. GA is a well-known search heuristic that mimics the process of natural evolution. This heuristic is widely used to generate useful solutions to optimization and search problems including the partition problem in fuzzy time series (Cai et al., 2013). In general, GA consists of populations, chromosome,

fitness function and genetic operations. The population represents a set of proper solutions. And each individual in the population represents a potential solution to a specified object problem. The search space for the problem solution is defined in this population representation. Each of the variables that compose an individual is known as chromosome. The chromosomes are commonly coded into a string to form the individual. Each individual in the population is evaluated by a fitness function in order to determine how fit is the solution. The GA maintains a population of *n* possible solutions, i.e., individuals, with associated fitness values evaluated according to the fitness function (Koo *et. al*., 1990).

1.2 Motivation

A lot of research works has been done on improving the accuracy of fuzzy time series forecasting models (Uslu et al, 2013; Bas et al, 2014; Zhang et al, 2013). Many researches have also been conducted on improving on the accuracy of FTS models using artificial intelligence optimization algorithms (Yolcu, 2014; Aladag et al, 2013, Haneen *et. al*., 2014). However, addressing the challenge of effectively capturing the relations properly and consequently improving the forecasting accuracy of the model still remains.

1.3 Statement of Problem

Fuzzy time series methods are effective techniques to forecast time series. Since its emergence, the study of fuzzy time series (FTS) has attracted more attention because of its ability to deal with the uncertainty and vagueness that are often inherent in real world data resulting from inaccuracies in measurements, incomplete sets of observations, or difficulties in obtaining measurements under uncertain circumstances.

The representation of fuzzy relations that are obtained from a fuzzy time series plays a key role in forecasting. In the analysis of time invariant fuzzy time series, fuzzy logic group relationships tables have been generally preferred for determination of fuzzy logic relationships. The reason of this is that it does not need to perform complex matrix operations when these tables are used. On the other hand, when fuzzy logic group relationships tables

are exploited, membership values of fuzzy sets are ignored. Thus, in defiance of fuzzy set theory, fuzzy sets' elements with the highest membership value are only considered. This situation causes information loss, thus decreasing the forecasting accuracy of the model. Secondly, it is also apt to encounter the problem of rule redundancy and computational overhead. Consequently, there is the need for a technique that which can capture the relationships more properly regardless of the non-linear nature of the fuzzy time series data. Furthermore, the inherent uncertainty involving time evolution usually makes the transition of states in a system probabilistic.

As a means of addressing these limitations that are inherent in the existing FTS models, a forecasting model based on Hidden Markov Model (HMM) for fuzzy time series was employed to realize the probabilistic state transition. Typically, relationship (parameter) estimation for a HMM is performed using iterative scheme that are well-defined but is prone to being trapped into a local minima. Genetic Algorithm (GA) has been popular due to their capabilities in handling nonlinear relationships. To improve on the relationship representation, a GA-HMM based model was applied to effectively capture the relations properly and therefore, improve the forecasting accuracy of the model.

1.4 Significance of Research

The significance of the research is to develop an improved hidden markov model based fuzzy time series that can improve the forecasting accuracy through effective estimation of the fuzzy relations existing amongst the states of the historical time series data.

1.5 Aim and Objectives

The aim of this research is the development of an improved HMM based FTS forecasting model using Genetic Algorithm.

To accomplish the above aim, the following objectives were employed:

a) Development of the HMM based FTS model using Bauw-Welch estimation procedure.

b) Development of the improved HMM-based FTS model by optimizing the model parameters using GA.
c) Model validation using the bivariate benchmark FTS data of the daily average temperature and cloud density of Taipei and comparing results with those obtained by (Li & Cheng, 2012), using the performance measures of MSE and AFEP.
d) Application of the developed model on the Internet traffic data of ABU Zaria.

1.6 Methodology

The methodology adopted for this research towards developing an improved hidden Markov model based fuzzy time series forecasting model using genetic algorithm is highlighted as follows.

a) Development of the standard HMM based FTS forecasting model using the relative frequency Bawm-Welch estimation procedure.
b) Improvement of the developed HMM based FTS model through re-estimation of the model parameters using GA.
c) Model validation using the bivariate benchmark FTS data of the daily average temperature and cloud density of Taipei and comparing results with those obtained by (Li & Cheng, 2012), using the performance measures of MSE and AFEP .
d) Application of the developed model to forecast short-term Internet traffic data of ABU Zaria using Internet traffic data from 29th February to 31st March, 2016, obtained from ABU, Zaria data center and evaluate its performance.

1.7 Dissertation Organization

The general introduction has been presented in Chapter One. The rest of the chapters are structured as follows: First, detail review of related literature and relevant fundamental concepts about Time series, fuzzy time series forecasting, Markov Model (MM), Hidden Markov Models (HMM) and Genetic Algorithms (GA) are carried out in Chapter Two.

Second, an in-depth approach and relevant mathematical models describing the development of the improved hidden markov model based fuzzy time series forecasting model using genetic algorithm are presented in Chapter Three. Third, the analysis, performance and discussion of the result are shown in Chapter Four. Finally, conclusion and recommendations of further work makes up the Chapter Five. The complete MATLAB codes are provided in the appendices.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter consists of the review of the fundamental concepts and the review of similar works which are relevant to the scope of this research.

2.2 Review of Fundamental Concepts

In this section, fundamental concepts such as fuzzy time series forecasting, Markov process, Hidden Markov models and Genetic algorithms amongst others are reviewed.

2.2.1 Time Series Models and Analysis

A time series, conventionally represented as a set of discrete values $x_1, x_2, x_3, ..., x_n$, is a sequence of observations of a physical quantity or variable ordered sequentially usually at regular intervals of time or space. Whichever phenomenon that is observed or measured is indexed by time as the only parameter (independent variable); therefore, the name time series is used (Shrivastav & Ekata, 2012). An observed time series can be assumed as the realization of a stochastic process. Once we understand how the process operates, a can develop a mathematical model to predict the future values of the time series. Thus, there are two main objectives of time series analysis:

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

2.2.1.1 Classification of time series

There are two types of time series data:

a) Continuous: where there is a collection of observation at every instance of time. In this time series, the observation at time t is denoted as $X(t)$.

b) Discrete: where there is an observation at usually equi-spaced intervals

There are two types of time series (Adeola, 2008).

a) **Univariate:** Univariate time series are those where only one variable is measured over time.

b) **Multivariate:** Multiple time series are those, where more than one variable are measured simultaneously (Yi *et al.*,2015).

2.2.1.2 Components of time series

Time series is represented mathematically as:

$\{X_t : 1,2,3,...,N\}$

Where t is the time series index, N is the total number of observation and $X(t)$ is the function of components represented as:

$$X_t = f(T_t, S_t, C_t, I_t) \tag{2.1}$$

Where:

T_t, S_t, C_t, I_t represent the trend, seasonal, cyclic and irregular components respectively (Subanar & Abadi, 2011). These components are the features or characteristics that define a time series. They are briefly defined as:

a) Trend components: Trend is a long term movement of a time series in a particular direction. It is the general tendency of a time series to move upward or downward. A simple way of detecting trend in seasonal data is to take averages over a certain period. If these averages change with time, then there is evidence of the trend in the series (Adeola, 2008).

b) Cyclical Components: Cyclical variations of non-seasonal nature, whose periodicity is un-known.

c) Seasonal components: Seasonal fluctuations of known periodicity. It is the component of variation in a time series which is dependent on the time of the year. It describes any

regular fluctuations with a period of less than one year. For example, the costs of various types of fruits and vegetables, and average daily rainfall, all show marked seasonal variation (Box & Jenkins, 1976).

d) Irregular components: The irregular component is the left over when other components of the series (trend, seasonal and cyclical) have been accounted for. (Hui-Kuarng, 2005).

2.2.2 Basic Concepts of Fuzzy Time Series

Fuzzy time series (FTS) is the mathematical amalgamation of fuzzy set theory and time series for the purpose of analyses and forecasting of time series data. Among other factors, the motivation for introducing this new framework was the need to model time series problems when historical data are defined as linguistic values (Song & Chissom, 1993). In order to understand fuzzy time series, the following foundational concepts of the FTS framework as they were originally conceived in pioneering papers (Song & Chissom, 1993; Song & Chissom, 1994) are given as follows:

a) Fuzzy Time Series: Let $Y(t)(t=...,0,1,2,....)$, a subset of R, be the universe of discourse on which fuzzy sets $fi(t)(i=1,2,...)$ are defined and let $F(t)$ be a collection of $fi(t)$. Then, $F(t)$ is called a fuzzy time series on $Y(t)(t=...,0,1,2,...)$ (Song & Chissom, 1994).

b) Fuzzy Relation: If $F(t)$ is caused by $F(t-1)$, denoted by $F(t)F(t-1)$, then this relationship can be represented by:

$$F(t)=F(t-1)*R(t,t-1) \tag{2.2}$$

Where:

* denotes the composition operator and $R(t,t-1)$ is a fuzzy relation between $F(t)$ and $F(t-1)$, where they represents the present and previous time observations respectively (Song & Chissom, 1994).

c) First Order Model: The model in which the relation $R(t,t-1)$ is a fuzzy relation between $F(t)$ and $F(t-1)$ is called the first order model of $F(t)$ (Song & Chissom, 1994).

d) N[th] Order Model: Let $F(t)$ be a fuzzy time series. if $F(t)$ is caused by $F(t-1)$, $F(t-2),...,F(t-n)$, then this fuzzy relationship is represented by $F(t-n),...,F(t-2),F(t-1) \rightarrow F(t)$ and is called n[th] –order fuzzy time series.

e) Time Invariant Fuzzy Time Series: If in first order model of $F(t)$ relation $R(t,t-1) = R(t=1,t-2)$ for any time t, then $F(t)$ is called time invariant fuzzy time series (Song & Chissom, 1994).

f) Time Variant Fuzzy Time Series: If in first order model of $F(t)$ relation $R(t,t-1) \neq R(t-1,t-2)$ for any time t, then $F(t)$ is called time invariant fuzzy time series (Song & Chissom, 1994).

2.2.3 Basic Steps of Fuzzy Time Series Forecasting

In order to understand the fuzzy time series forecasting procedure, the following steps are outlined (Song & Chissom, 1993):

a) Defining the universe of discourse
b) Partitioning the universe of discourse into several intervals of equal length
c) Defining the fuzzy sets on the universe of discourse and fuzzfying the time series
d) Modeling fuzzy logical relationships
e) Calculate the forecast output
f) Defuzzifying (interpret) the forecasting outputs
g) Evaluate forecast performance

Figure 2.1 shows the steps of the generic FTS model.

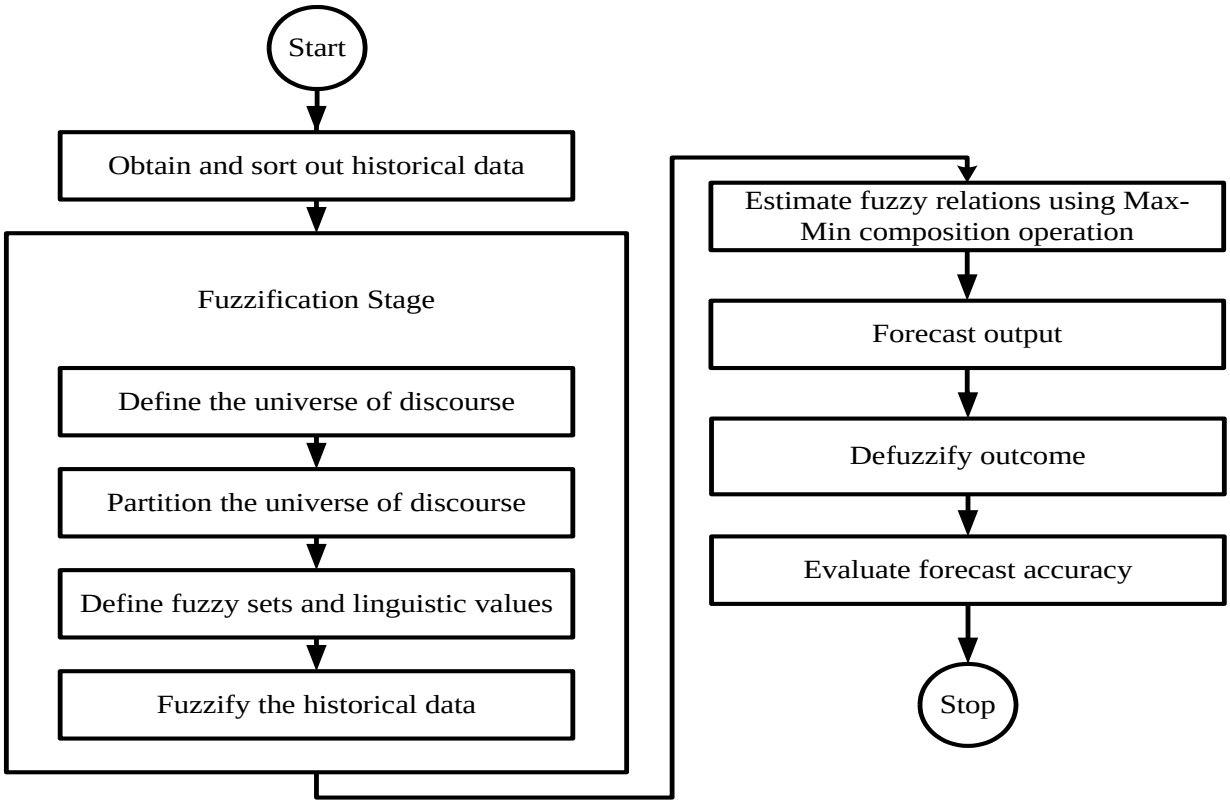


Figure 2.1: Block Diagram Showing the Steps Generic FTS Model (Song & Chissom, 1993)

2.2.4 Hidden Markov Model (HMM) and its Approach to Time Series Data

This section introduces the HMM and its approach to time series data. The HMM due to the characteristics it possess such as probabilistic, statistical, as well as robustness and scalability, it has attracted the knowledge of many research in diversified areas of study.

2.2.4.1 Hidden Markov model

The Markov model proposed in the early 1900s is often called an observable (or visible) Markov model (Markov, 1913), so as to be distinguished from the HMM. Since the HMM is derived from traditional Markov theory, this section begin with a brief description of a Markov Model. A Markov model is a statistical model, and can be viewed as a typical

transitional diagram composed of the following components: a set of different states; transition between states; and transition probabilities, which are probabilities linked to transitions. Beginning from a start state, a transitional process continues until it reaches an end state. The outcome of this sequence of states (observations) within a Markov model is called Markov chain. Formally, a Markov chain can be represented by a series of random variables, $X^{(0)}, X^{(1)}, X^{(2)}, ..., X^{(m+1)}$ each of which takes on a value from the state space, $S = \{S_1, S_2, ..., S_N\}$ (Babakura *et al*, 2014).

The main characteristic of a Markov model is that it predicts the future based on the present rather than the past: this is termed Markov condition. When the probability of a sequence of random variables, $P(X^{(0)}, ..., X^{(m+1)})$ is calculated, all the previous random variables are considered according to (Babakura *et al*, 2014). This is defined as:

$$P(X^{(0)}, ..., X^{(m+1)}) = P(X^{(0)})P(X^{(1)} / X^{(0)} ... P(X^{(m)} / X^{(0)}, ..., X^{(m-1)})P(X^{(m+1)} / X^{(0)}, ..., X^{(m)}) \quad (2.3)$$

By assuming the Markov condition, however, Markov theory only takes into consideration the current random variable by:

$$P(X^{(0)}, ..., X^{(m+1)}) = P(X^{(0)})P(X^{(1)} / X^{(0)} ... P(X^{(m)} / X^{(m-1)})P(X^{(m+1)} / X^{(m)}) \quad (2.4)$$

To present the Markov model's features, a simple example is provided. Figure 2.2 shows an example of a simple Markov model for tossing a coin. Each state of the model corresponds to the outcome of an observation (also called an event), either Heads (H) or Tails (T) of a coin.

The numbers along the lines or curves specify transition probabilities linked to a directed path between the two states connected by the lines or curves. The arrows indicate the direction of the state transition. The S in Figure 2.2 shows the special state serving as the starting point in the model. This model must start from the special state and can move to other states by adding the probabilities corresponding to the direction that it takes.

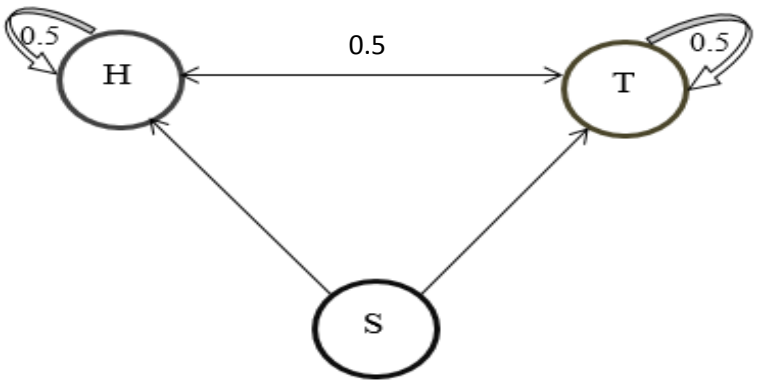


Figure 2.2: Markov Model for Tossing One Coin (Babakura *et al*, 2014)

An HMM is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols. At each discrete time instant, the process is assumed to be in a state, and an observation is generated by the probability distribution corresponding to the current state. The HMM is termed discrete if the output alphabet is finite, and continuous if the output alphabet is not necessarily finite, e.g., each state is governed by a parametric density function (Wael *et al*., 2012).

The hidden Markov process is based on two important and essential assumptions (Yi *et al*., 2015):

a) The next state is dependent only upon the current state, and

b) Each state-transition probability does not vary in time, i.e., it is a time-invariant model.

Figure 2.3 shows the graphical representation of the HMM.

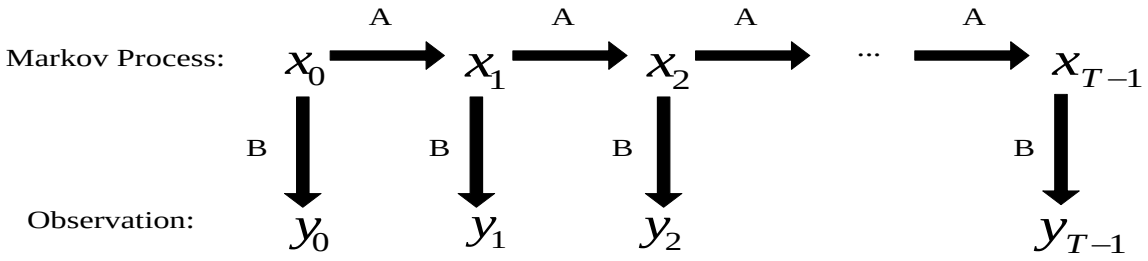


Figure 2.3: Hidden Markov Model (Emmanuel & Thierry, 2013)

It is different from the traditional markov model in that the system state is not directly observable and thus is termed “hidden states”. Each states emits an obse4vable symbol in such a way that when the system moves from one state to another state according to state-transition probabilities, a sequence of observable symbols are generated according to some emission probability distribution.

HMMs have been applied in various practical applications, where it has become a predominant methodology for design of automatic speech recognition systems (ASR). It has also been successfully applied to various other fields, such as signature verification communication and control, bioinformatics, computer vision, and computer and network security. The advantages of HMM lie on its established statistical framework and working well practically.

2.2.4.2 Components of HMM

In order to define a hidden Markov model, the following notations are defined (Babakura *et al*, 2014)

a) N, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to the set of states in the model. The set of states is denoted by:

$$S = \{S_1, S_2, ..., S_N\} \quad (2.5)$$

Where: S_i is state i, $i \in \{1, ..., N\}$ A fixed state sequence can be represented by

$$Q = \{q_1, q_2, ..., q_T\} \quad (2.6)$$

Where: the state at time t is q_t.

b) M, the number of distinct observation symbols per state. The observation symbol corresponds to the physical output of the system being modelled. Individual symbols are denoted as:

$$V = \{v_1, v_2, ..., v_M\} \quad (2.7)$$

c) π , the initial state distribution.

$$\pi = \{\pi_i\}, \pi_i = P_r(s_i, 1) \tag{2.8}$$

Where: π is a vector with the probability of the initial state

d) The state transition probability distribution given by:

$$A = \{a_{ij}\}, \tag{2.9}$$

Where: $a_{ij} = P_r(s_j, t / s_i, t-1)$

a_{ij} denotes the probability of being in state j at time t+1 given that it was in state i at time t. It is assumed that a_{ij} is independent of time.

e) The observation symbol probability distribution given by:

$$B = \{b_{ij}\} j = 1...N \tag{2.10}$$

Where:

$b_{ij} = P_r\{O_j = t / S_i = t-1\}$

$1 \le j \le N$

$1 \le k \le N$

2.2.5 The Standard HMM based FTS Forecasting Approach

The standard HMM based FTS forecasting model was introduced by (Li & Cheng, 2006). Its follows the methodology of the basic fuzzy time series forecasting outlined in section 2.2.3 to fuzzify the historical data and use the transition of HMM to forecast outcome. The main steps involved in this approach include:

Step 1: Determination of the Universe of Discourse

In general, U_s and U_o defined as:

$$U_s = [D_{s\min} - D_{s1}, D_{s\max} + D_{s2}] \tag{2.11}$$

$$U_o = [D_{o\min} - D_{o1}, D_{o\max} + D_{o2}] \tag{2.12}$$

Where:

U_s and U_o are the universe of discourses for the hidden variables and observation variables respectively. $D_{s\min}$, $D_{s\max}$, $D_{o\min}$, and $D_{o\max}$ are the respective minimal and maximal values of the historical data of hidden and observation variables while D_{s1}, D_{s2}, D_{o1}, and D_{o2} are the corresponding proper positive numbers.

Step 2: Partition the Universe of Discourse into Several Even-Length Intervals

Then, U_s is partitioned into n equal intervals i.e. $u_1, u_2, ..., u_n$, with the interval length l_s defined as:

$$l_s = \left(1/n\right) \times \left[(D_{s\max} - D_{s2}), (D_{s\min} + D_{s1})\right] \tag{2.13}$$

Where: u_1, u_2 and u_n represents the first, second and nth partitions respectively, while l_s is its interval length.

Similarly, U_o is partitioned into m equal intervals i.e. $v_1, v_2, ..., v_n$, with length l_o defined as:

$$l_o = \left(1/m\right) \times \left[(D_{o\max} - D_{o2}), (D_{o\min} + D_{o1})\right] \tag{2.14}$$

Where: v_1, v_2 and v_n represents the first, second and nth partitions respectively, while l_o is the interval length.

Step 3: Defining the Fuzzy Sets on the Universe of Discourse

Next, fuzzy sets of choice are defined for the intervals obtained in step 2. There is no restriction on the number of the fuzzy sets defined.

Step 4: Fuzzfying the Time Series Data

For hidden states, n fuzzy sets $u_1, u_2, ..., u_n$, can be defined on U_s using general membership functions, expressed as follows:

$$S_i = \sum_{j=1}^{n} \mu_{ij} / u_j \tag{2.15}$$

Where:

μ_{ij} is the membership degree of S_i belonging to u_j which is defined by:

$$\mu_{ij}=\begin{cases}1, \ if \ \ j=i\\ 0.5, \ if \ \ j=i-1 \ or \ i+1\\ 0, \ \ Otherwise\end{cases}$$

Then, for a given historical datum Y_t, its membership degree belonging to interval u_i is determined by the following heuristic rules:

Rule 1) If Y_t is located at u_1, the membership degrees are 1 for u_1, 0.5 for u_2, and 0 otherwise.

Rule 2) If Y_t belongs to u_i, $1 < i < n$, then the degrees are 1, 0.5, and 0.5 for u_i, $u_i - 1$, and $u_i + 1$, respectively, and 0 otherwise.

Rule 3) If Y_t is located at u_n, the membership degrees are 1 for u_n, 0.5 for $u_n - 1$, and 0 otherwise. Then, Y_t is fuzzified as S_j, where the membership degree in interval j is maximal.

For observable states, m fuzzy sets $v_1, v_2, ..., v_n$, can be defined on U_o expressed as follows:

$$O_i = \sum_{j=1}^{m} \mu_{ij} \Big/ u_j \tag{2.16}$$

Where:

μ_{ij} is the membership degree of O_i belonging to v_j and is defined by

$$\mu_{ij}=\begin{cases}1, \ if \ \ j=i\\ 0.5, \ if \ \ j=i-1 \ or \ i+1\\ 0, \ \ Otherwise\end{cases}$$

The observation variable can then be fuzzified in the same way as the hidden variable.

Step 5: Building the HMM Model to Estimate the Fuzzy Relations

As discussed in section 2.2.4, the HMM was characterized by the three matrices. That is,

$$A = \{a_{ij}\}, \tag{2.17}$$

Where:

$$a_{ij} = P_r\left(S_{j,}t / S_i, t-1\right)$$

$$\pi = \{\pi_i\}, \tag{2.18}$$

Where: $\pi_i = P_r(S_i,1)$

$$B = \{b_{ij}\} \tag{2.19}$$

Where:

$$b_{ij} = P_r\{O_j = t / S_i = t-1\}$$

Constructing an HMM, i.e., $\lambda = \pi, A, B$ for $F(t) = (F(t-1), G(t))^* R((t, t-1))$ and estimating its parameter set. To solve the forecasting problem of $f(t)$), an HMM can be constructed, where $F(t-1)$ and $G(t)$ t) are the hidden-state sequence and the observed state sequence, respectively.

The parameter set are then estimated using the Bawm-Welch estimation procedure expressed as:

a) The initial state vector π is set to be a $1 \times n$ matrix and is defined as:

$$\pi_i = P_r(S_i,1) = \frac{n(S_i 1)}{N_1} \tag{2.20}$$

Where:

$n(S_i 1)$ is the number of the initial state S_i in the data set and

$$N_1 = \sum_{i=1}^{n} n(S_i,1)$$

b) The state-transition matrix $A = \{a_{ij}\}$, is an $n \times n$ matrix defined as:

$$a_{ij} = P_r\left(S_{j,}t / S_i, t-1\right) = \frac{n(S_{j,}t - S_i, t-1)}{n(S_i, t-1)} \tag{2.21}$$

Where:

$n(S_j, t-S_i, t-1)$ is the number of transitions from state i at time $t-1$ to state j at time t, $n(S_i, t-1)$ is the total number of transitions to state i at time $t-1$, $\forall a_{ij} \geq 0$ and $\sum_{j=1}^{n} a_{ij} = 1$, $1 \leq i \leq n$

c) The confusion matrix $B = \{b_{ij}\}$ is an $n \times m$ matrix represented as:

$$b_{ij} = P_r\{O_j = t / S_i = t-1\} = \frac{n(O_j, t / S_i, t)}{n(S_i, t)} \quad (2.22)$$

Where:

$n(O_j, t / S_i, t)$ is the number of observing the observable symbol j and state i at the same time t, while $n(S_i, t)$ is total number of transitions to the S_i at time t, $\forall b_{ij} \geq 0$ and $\sum_{j=1}^{m} b_{ij} = 1,\ 1 \leq i \leq n$.

Step 6: Calculating Forecast Outputs

In general, the forecasting result is located at the state with the highest probability. The technique uses the initial state probability and confusion matrix values for the state with the highest probability to obtained the forecast. This is represented as:

$$\mu_t(k) = \pi^T * B(:[k]) = [\Pr(o_k, s_1) \Pr(o_k, s_2) ... \Pr(o_k, s_{n1})]^T \quad (2.23)$$

Where:

k is the $k-th$ column of the confusion matrix B

$B(:[k])$ is the fuzzy relationship matrix for the observation state O_k.

Step 7: Defuzzifying the Forecasting Outputs

The last step is defuzzifying the forecasting outputs in order to compare the performance represented as:

$$F(t) = m_k + F(t-1) \quad (2.24)$$

Where:

m_k is the midpoint of the interval for the state with the highest probability.

Figure 2.4 shows the block diagram for the steps of the Standard HMM based FTS model.

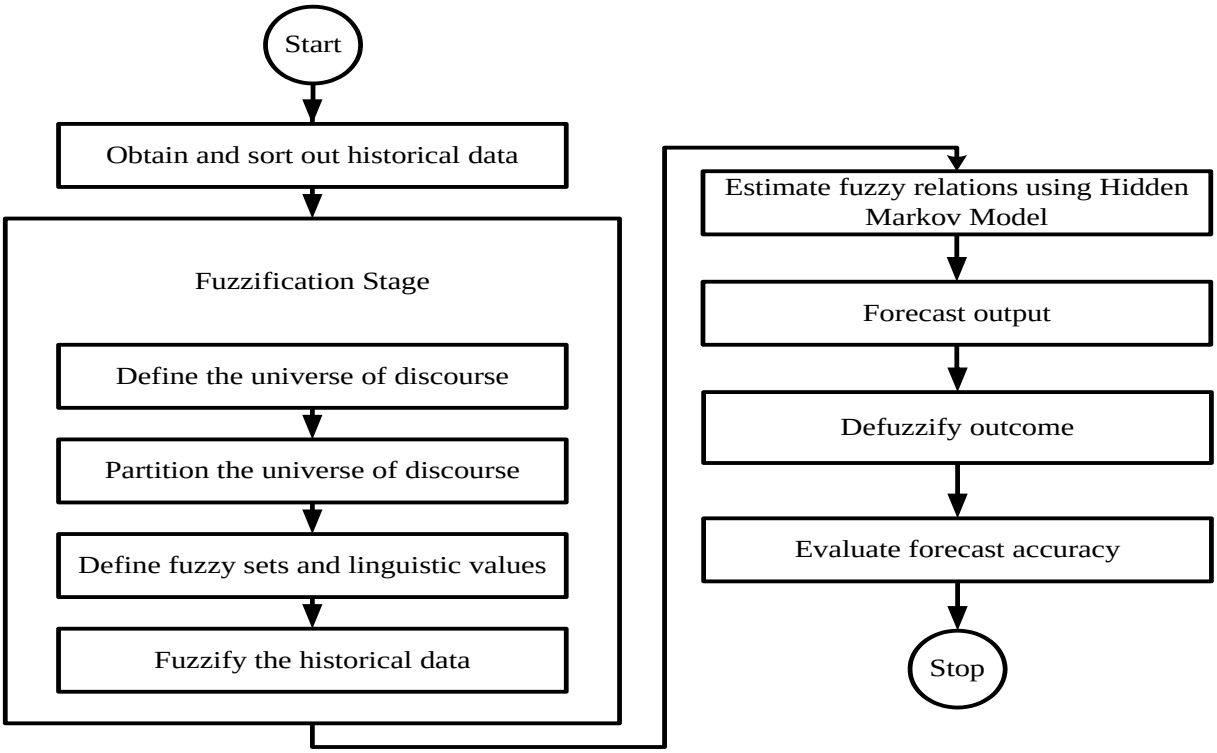


Figure 2.4: Block Diagram Showing the Steps of Standard HMM based FTS Model (Li & Cheng, 2006).

2.2.6 Parameter Estimation Methods of HMM.

In the following section, few popular methods used for estimating the parameters of the HMM model are briefly discussed.

a) Gradient Based Methods

Because the gradient provides information about how fast the function is increasing or decreasing in any given direction at a particular location on the landscape, gradients can inform the hill-climbing process by indicating the most promising direction at any given point i.e. the direction in which we can expect to most rapidly approach the optimum. Without gradient information a hill-climber typically needs to query a number of points surrounding the current point in order to identify a promising direction for further search. Thus, gradient methods (Steepest descent, steepest ascent etc.) can reduce the number of function evaluations needed during hill-climbing. Unfortunately, these methods require the

evaluation of the function's derivative, which for some functions is difficult to do analytically (Majoros, 2008). Baum-Welch algorithm is an iterative Expectation Maximization (EM) algorithm that, given an initial parameter configuration, adjusts model parameters to locally maximize the likelihood of unlabeled data. The EM algorithm is the most popular training method of HMMs because it converges quite fast and shows good results in recognition with careful initial estimates of the HMM parameters. A limitation of the EM algorithm is that it only performs local search in the parameter space and therefore may be trapped in local optima depending on the initial values of the parameters (Park & Lee, 2006).

b) Non-Gradient Based Methods:

The use of gradients during optimization involves an unfortunate tradeoff-namely, that while the gradients can inform the optimization process and thereby (in many cases) accelerate that process by reducing the number of steps needed to converge (as compared to blind hill climbing), evaluating the gradients can be very time-consuming, especially for functions taking many parameters (since a partial derivative must be evaluated for each parameter). A number of non-gradient-based optimization techniques have been shown to be effective for many applications, and in some cases these alternative methods can be somewhat less susceptible to the problem of converging to a non-global optimum. Some of these methods are briefly stated as follows:

i) Genetic Algorithm: Genetic algorithm (GA) is a stochastic search method that can perform global search within the defined searching space and hopefully obtains the global optimal solution. This method maintains a population $Q = \{\theta_1, ..., \theta_n\}$ of model parameterizations and attempts to "evolve" that population (via appropriately-chosen mutation and recombination operators) so as to improve the average population "fitness" (i.e., the average value of the objective function) (Majoros, 2008).

ii) Simulated Annealing: Simulated annealing instead maintains a single θ which is successively perturbed so as to improve the value of the objective function (i.e., climbing uphill), while occasionally accepting a perturbation which results in a somewhat worse evaluation under the objective function (i.e., occasionally stepping downhill) (Majoros, 2008).

Both methods differ from strict hill-climbing and gradient ascent by sometimes allowing movements over the optimization landscape which are not strictly uphill, in hopes that these occasional movements may allow the optimizer to escape a local optimum so as to find more promising regions of the optimization landscape. Figure 2.5 shows some parameter estimation methods commonly used in the field of HMM.

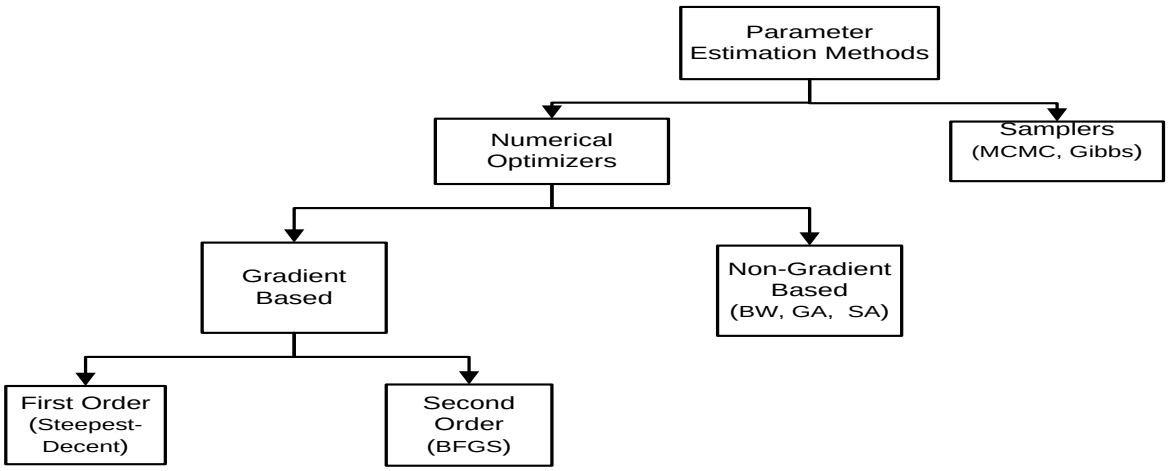


Figure 2.5: Taxonomy of Parameter Estimation Methods (Majoros, 2008)

2.2.7 Smoothing

There exists an important issue when estimating the parameters of the HMM, and herein conditional probabilities, in that one often encounters the zero-frequency problem for the event observed. This could deteriorate the forecasting accuracy and, especially, when insufficient training data is available (Li & Cheng, 2012). To deal with this issue, a number of smoothing techniques have been proposed and have mostly been applied in the area of

speech recognition (Chen & Goodman, 1996). The term smoothing describes techniques for adjusting the maximum likelihood estimate of probabilities to produce more accurate probabilities. The name smoothing comes from the fact that these techniques tend to make distributions more uniform, by adjusting low probabilities such as zero probabilities upward, and high probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of the model as a whole. Whenever a probability is estimated from few counts, smoothing has the potential to significantly improve estimation (Chen & Goodman, 1996).

An enormous number of techniques have been proposed for smoothing temporal pattern recognition models. The most commonly used ones are:

a) Laplace Smoothing

The Laplace smoothing (also known as add-1 smoothing), is not only the most conventional smoothing technique, but the simplest. The Laplace smoothing simply adds 1 to all frequency counts to avoid the zero probability (Li & Cheng, 2012). For example, if there is a data sequence that is constructed by elements x_i, the probability that the ith element occurs is denoted as:

$$\Pr(x_i) = \frac{n(x_i)}{N} \tag{2.25}$$

Where:

$n(x_i)$ is the frequency that x_i occurs, and N is the total number of the data sequence. After performing the Laplace smoothing, the probability that x_i, occurs will change to

$$\Pr_{Lap}(x_i) = \frac{1 + n(x_i)}{V + N} \tag{2.34}$$

Where:

V is the total number of times that 1 has been added.

Yet, although the Laplace smoothing is very simple to implement, it could also bring the problem that it may give too much probability to unseen states. The Laplace smoothing is unreasonable in the sense that it cannot explain why it adds 1 to all elements. For example, if $n(x_i)$ is zero, it will be replaced with 1, even though the dataset is very small; this seems unreasonable, since the amount of probability added to the zero element is overestimated.

b) Backoff Smoothing

To remove this flaw in Laplace smoothing, backoff smoothing which discounts some probability of the nonzero elements, and for the zero-probability element, it replaces higher order probability with a lower one was proposed (Li & Cheng, 2012). The equation of back off smoothing is defined as:

$$\Pr_{katz}(x_i / x_j) = \begin{cases} \Pr(x_i / x_j), \textit{ if } n(x_i / x_j) > 0 \\ \alpha \Pr(x_i), \textit{ Otherwise.} \end{cases} \quad (2.26)$$

Where:

α is used to normalize the probability mass so that it still sums to 1.

c) Jelinek-Mercer Smoothing

In addition to Katz's backoff smoothing, another widely used method is the so-called Jelinek–Mercer (JM) smoothing. The equation of this smoothing operation is defined as following:

$$\Pr_{JM}(x_i / x_j) = \gamma \Pr(x_i / x_j) + (1-\gamma)\Pr(x_i) \quad (2.27)$$

Where:

γ is an adjusting parameter so that $\gamma \in (0,1)$. Backoff smoothing only replaces the zero-probability element with a lower order one, whereas the JM smoothing interpolates the probabilities of all outcomes with a lower order one, and thus it can simply find a better result by tuning up the value of γ.

The traditional smoothing methods reviewed earlier are commonly applied in the field of speech recognition, in which the states are crisp in nature, and thus, they are inappropriate for the case of fuzzy time series (Koo & Un, 1990). Thus, the use of new smoothing method (fuzzy smoothing) which can deal with the fuzziness existing in the HMM-based FTS becomes imperative. Fuzzy smoothing is a general smoothing technique incorporating with fuzzy set theory, which can be applied in different areas, such as image processing, robot control, and speech recognition (Dai, 1995).

2.2.8 Performance Metrics

The history of the evaluation of forecast accuracy goes along with that of time-series analysis (Elizabeth & Jana, 2004). Over time, many researchers have come to view forecast evaluation as a vital component of empirical time series work (Hyndman & Khandakar, 2008). Forecast evaluation has become an important metric for evaluating models. If one model is superior to another, it ought to forecast more accurately. Forecasts need to be good to be useful for decision making. Thus, determining if forecasts are good involves formal evaluation of the forecasts. It is important to evaluate forecast accuracy using genuine forecasts. That is, it is invalid to look at how well a model fits the historical data; the accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when estimating the model. When choosing models, it is common to use a portion of the available data for testing, and use the rest of the data for estimating (or "training") the model. Then the testing data can be used to measure how well the model is likely to forecast on new data (Hyndman & Khandakar, 2008). There were reported performance measures used to evaluate the forecast accuracy of fuzzy time series models and they include amongst others; Mean Square Error and Average Forecast Error Percentage (Elizabeth & Jana, 2004).

a) Mean Square Error $= \dfrac{\sum_{i=1}^{n} \left(Forcasting_Value - Actual_Value\right)^2}{n}$ (2.28)

b) Average Forecast Error Percentage $= \dfrac{1}{n} \dfrac{\sum_{i=1}^{N} \left|Forcasting_Value - Actual_Value\right|}{Actual_Value}$ (2.29)

Where: n is the total number of data forecasted.

2.3 Review of Similar Works

The reviews of related works in the field of fuzzy time-series are discussed as follows:

Song & Chissom (1993) developed a forecasting model called fuzzy time series, which provided a theoretic framework to model a special dynamic process whose observations are linguistic values. They applied max-min composition operation on the data of historical enrollments of the University of Alabama to evaluate their forecasting model. The results obtained were better than those obtained using linear regression method. However, the technique involved high computational overheads due to complex matrix operations. Secondly, during the determination of fuzzy relations, only the fuzzy set having the highest membership value was considered and membership values were ignored. This led to loss of information and improper estimates of the fuzzy relations, which consequently affected the forecasting accuracy of their model.

Sullivan & Woodall (1994) employed a Markov based model by using conventional matrix multiplication in order to alleviate the computational overhead in deriving the fuzzy relationship rules existing in (Song & Chissom, 1993). In their model, the fuzzy relationships were represented by a time-invariant Markov process, in which the relationship matrix is determined by the probability distributions of linguistic values and each row of the matrix is unit normalized. However, the model did not take into consideration the frequency of state transitions on the estimation of the relationship matrix, resulting in missing information and

causing the forecasting to be unsatisfactory. In addition, only the fuzzy set having the highest membership value was considered when determining the fuzzy relations representing the internal relation of fuzzy time series and membership values were ignored. The model is only applicable to one-factor forecasting problems.

Li & Cheng (2006) developed a novel forecasting model based on the Hidden Markov Model by enhancing (Sullivan & Woodall, 1994) work to allow handling of two-factor forecasting problems. They considered the fact that the forecast value is usually related to the output of the recent day and the probability of the state caused by earlier states was also considered while forecasting. The performance evaluation on forecasting the temperature in Taiwan confirmed that the proposed model outperformed all previously proposed models in terms of Mean Squared Error (MSE). However, the effect of the state transition matrix on the forecast output was not considered. Secondly, the technique still employed a conventional optimizer at estimating the model parameters (fuzzy relations) which was not effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

Li & Cheng (2010) proposed a novel forecasting model based on the Hidden Markov Model to enhance the work of (Li & Cheng, 2006) by considering the state transition matrix at the defuzzification phase. The proposed model was established on the basis that the probability distribution of the observation at any time is always dependent on the previous states. The performance evaluation on forecasting the temperature of Taiwan confirmed that the model outperformed all previously proposed models. However, the model does not consider the optimal estimation of the model parameters, which might cause the model to be in-effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

Hongxing, *et al*. (2011) developed first order multivariate markov chain model for generating annual weather data for Hong Kong. The forecasting model was developed under the assumption that observation that the next state is dependent only on the present state not upon the accumulated results of the observations at previous times. The relationship matrix was constructed by calculating the transition frequencies occurring in the historical temporal data and linear programming was employed to obtain the optimal model parameters. Experimental results showed that the proposed model generated more suitable data for sizing and evaluating different renewable energy systems, when compared with the current existing TMY (Typical Meteorological Year) and TRY (Test Reference Year) data. However, the technique employed a conventional optimizer at estimating the model parameters which was not effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

Tsaur (2012) proposed a fuzzy time series-Markov chain model with an application to forecast the exchange rate between the Taiwanese and US dollar. The model exploited the advantage of the fuzzy logic relationship to group the collected time series data and used the obtained fuzzy logic group to obtain a Markov chain transition matrix to derive the outcomes with the largest probability. The experimental results showed that the model was effective at predicting the trend of the exchange rate when compared with some revised fuzzy time series methods (ARIMA-GARCH and the grey forecasting model). However, the method relied on constructing the fuzzy logic relationship groups (FLRG) represented as IF-THEN rules that are embedded in the fuzzy time series leading to improper estimate of the fuzzy relation leading to rule redundancy and computational overhead. Its applicability is limited to handling one-factor problems and the membership values are ignored when formulating the fuzzy relations.

Li & Cheng (2012) proposed a fuzzy time series forecasting with a probabilistic smoothing hidden Markov model in which the formulation of the fuzzy relations were established based

on the hidden Markov model framework. The model employed a novel fuzzy smoothing method to address the performance deterioration of (Li & Cheng, 2010) owing to fewer fuzzy relationships that existed in the historical temporal data. To test the effectiveness of the resulting stochastic model, two experiments were conducted and the results compared with those from other models. The first experiment consisted of forecasting the daily average temperature and cloud density in Taipei, Taiwan, and the second experiment was based on the Taiwan Weighted Stock Index by forecasting the exchange rate of the New Taiwan dollar against the U.S. dollar. Performance evaluation on forecasting the temperature of Taiwan confirmed that the model outperformed some of the proposed models in the literature in terms of the performance measures of the mean square error and the average forecasting error percentage However, the technique employed a conventional optimizer in estimating the model parameters which was not effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

Vamitha *et al.* (2012) developed a model for temperature prediction using fuzzy time series and multivariate Markov chain. The forecasting model was developed under the assumption that the observation at the next state is dependent only on the present state not upon the accumulated results of the observations at previous times and the relationship matrices were obtained by counting the transition frequencies occurring amongst the states of the categorical data sequences. Experimental results showed that the method got higher average forecasting accuracy rate than some of the existing methods on temperature prediction. However, the technique employed a conventional optimizer at estimating the model parameters which was not effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

Yi *et al.* (2015) developed an enhanced HMM-based for fuzzy time series forecasting model. The method expanded (Li & Cheng, 2010) probabilistic HMM model to forecasting multiple-factor problems. The method utilized more factors to obtain better forecasting accuracy rate

and used one actual datum to verify the differences between one observable variable and multi-observable variables. When applied to forecast the weather data of Alishan in Taiwan, the proposed model outperformed some of the revised FTS models in terms of the four evaluation indices of MAE (Mean Absolute Error), PMAD (Percent Mean Absolute Deviation), MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Squared Error However, the model does not consider the optimal estimation of the model parameters, which might cause the model to be in-effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model. Secondly, the model complexity and computational cost was significantly high especially when the order was more than one.

Suresh *et al.* (2016) developed a higher order multivariate Markov chain (HOMM) model for fuzzy time series to forecast global surface temperature with presence of CO_2 emission. The forecasting model was developed under the assumption the observation that the next state is dependent only on the present state not upon the accumulated results of the observations at previous times and also adopts simulation to estimate the forecast results. The relationship matrix was constructed by calculating the transition frequencies occurring in the historical temporal data and linear programming was employed to obtain the optimal model parameters. Experimental results showed that the HOMM - FTS methods gives better forecasting results compared to existing well known fuzzy time series approaches. However, the technique employed a conventional optimizer at estimating the model parameters which may not be capable of capturing the fuzzy relations continuous data properly and hence, deteriorating the forecasting performance of model.

Wang *et al.* (2016) developed a steady-state probabilities model for fuzzy time series forecasting. The forecasting model was developed under the assumption that observation at the next state is dependent only on the present state not upon the accumulated results of the observations at previous times. To test the effectiveness of the resulting stochastic model, two experiments were conducted and the results compared with those from other models. The first

experiment consisted of forecasting the enrollment data of the University of Alabama while second experiment was based on the weather data of Alishan in Taiwan. Experimental results showed that the proposed model obtained improved performance compared to some previously existing fuzzy time models in terms of the performance indices of Percent Mean Absolute Deviation and Root Mean Squared Error. However, the model does not consider the optimal estimation of the model parameters, which might cause the model to be in-effective at capturing the fuzzy relations properly and hence, deteriorating the forecasting performance of model.

From the similar works reviewed, it is evident that fuzzy time series forecasting has captured a wide area of applications. Efforts have been made to address these challenges using the standard and some hybridized fuzzy time series forecasting models. However, the challenge of improper estimate of the fuzzy relations using standard approaches to optimization still remains. Therefore this research employed the use of GA-HMM to formulate the fuzzy relations occurring amongst the states of the time series data.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

In this chapter, the methods, materials and procedures employed for the successful implementation of this research work are discussed and the improved hidden Markov model based fuzzy time series forecasting model using genetic algorithm was developed.

3.2 Development of the Improved HMM based FTS Forecasting Model using GA for the Re-estimation of the Inner Fuzzy Relations.

The data will be preprocessed with fuzzification and use the transition of hidden Markov model to forecast outcome. The Genetic Algorithm technique discussed earlier in section 2.2.6 is one of the possible approaches to addressing the challenges of parameter estimation in Hidden Markov Models. This helps in the best possible model parameters and consequently improving its estimates. In developing the improved HMM based FTS model, there are some very important steps (sorting, fuzzification, relationship representation, optimization and defuzzification) that make up the model. Figure 3.1 shows the stepwise block diagram implementation of the improved HMM-GA based FTS forecasting model.

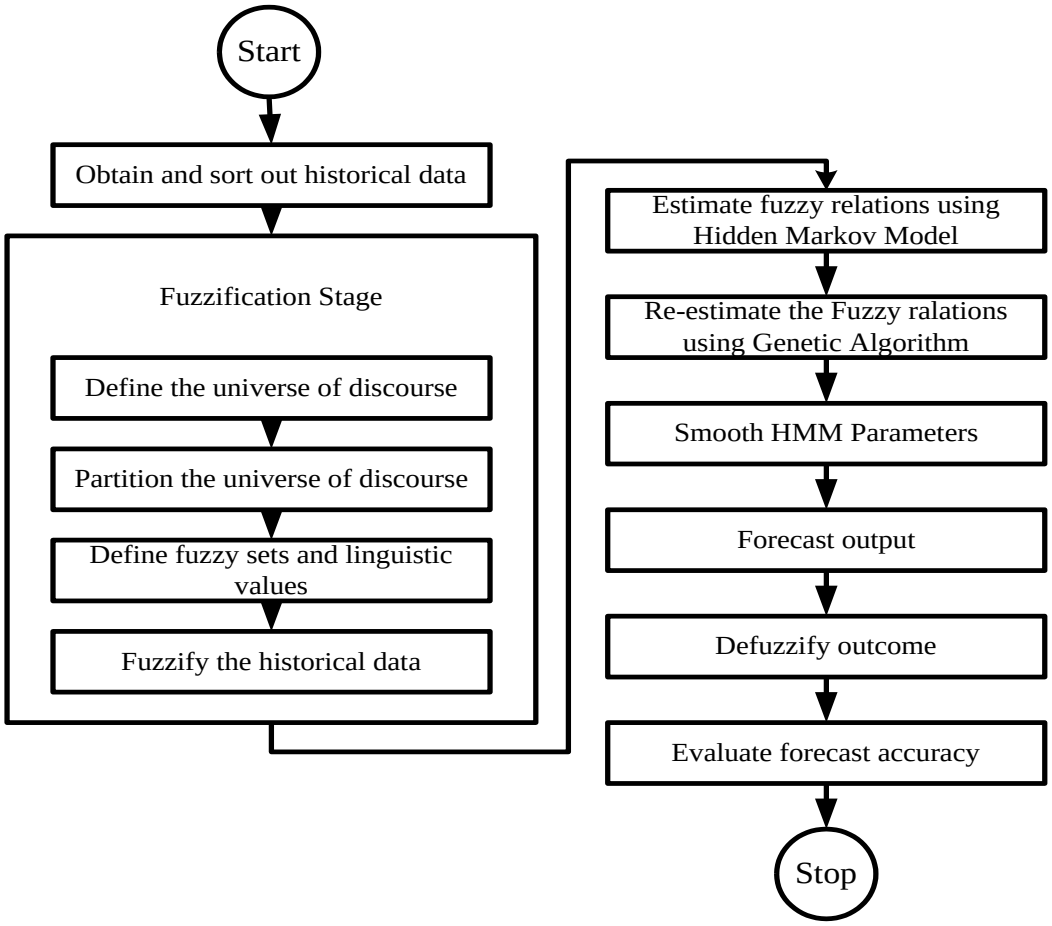


Fig. 3.1: Block Diagram of the Developed Model

The following equations and discussion describe the steps involved in each processes in the flowchart.

3.2.1 Obtaining Historical Time Series Data

The time series data considered in this research was the bivariate benchmark FTS data of the daily average temperature and cloud density of Taipei, Taiwan, obtained from the work of (Li & Cheng, 2010). The time span of the data used ranges from June to September of the years 1993, 1994, 1995, and 1996. The data was divided into training data set (i.e., that for 1993, 1994, and 1995) and testing data set (i.e., 1996). The data was then divided into training data set to estimate the model and testing data set to evaluate the model's performance. The training data are given in appendices (A1-A3) while the testing data can be found in appendix B.

3.2.2 Determination of the Universe of Discourse

In defining the universe of discourse, the minimum value $D_{\min}$ and the maximum value $D_{\max}$ of the historical data sets were first determined. Then the universes of discourse will be determined using the procedure outlined in section 2.2.5. From the data of appendices A1-A3, defined universes of discourse for the hidden variables and observation variables are as shown in Table 3.1.

Table 3.1: Universes of Discourse for Hidden and Observation Variables

SN	Temperature (ºC)	Cloud Density (okta)
1	$D_{s\max} = 32.2$	$D_{o\max} = 100$
2	$D_{s\min} = 21.6$	$D_{o\min} = 0$
3	$D_{s1} = 0.6$	$D_{o1} = 0$
4	$D_{s2} = 0.8$	$D_{o2} = 0$
5	$U_s = (21,32)$	$U_o = (0,100)$

3.2.3 Partition the Universe of Discourse into Several Even-Length Intervals

Having obtained the interval length as outlined in section 2.2.5, the universes of discourse U_s and U_o are then partitioned into n and m equal intervals respectively, using the following simplified expressions:

$$I_1 = [D_{\min} \rightarrow (D_{\min} + l)] \tag{3.1}$$

$$I_2 = [(D_{\min} + l) \rightarrow (D_{\min} + 2l)] \tag{3.2}$$

$$I_n = [(D_{\min} + (n-1)l) \rightarrow (D_{\min} + nl)] \tag{3.3}$$

Where:

I_1, I_2 and I_n represents the first, second and nth partitions respectively. In this research, the universes of discourse for the hidden and observation variables were partitioned into 6 and 5 intervals respectively, as shown in Table 3.2.

Table 3.2: Interval Lengths for the Hidden and Observation Variables

SN	Temperature (ºC)	Cloud Density (Okta)
1	$u_1 = [21,23]$	$v_1 = [0,20]$
2	$u_2 = [23,25]$	$v_2 = [20,40]$
3	$u_3 = [25,27]$	$v_3 = [40,60]$
4	$u_4 = [27,29]$	$v_4 = [0,20]$
5	$u_5 = [29,31]$	$v_5 = [80,100]$
6	$u_6 = [31,33]$	
7	Interval Length= 2	Interval Length= 20

3.2.4 Defining Fuzzy Sets on the Universe of Discourse

As discussed in section 2.2.5, fuzzy sets of choice are defined for the obtained intervals section 3.2.3. There is no restriction on the number of the fuzzy sets defined. In this research n and m linguistic fuzzy sets were defined for the hidden and observation variables' intervals, respectively. In this research, for the linguistic variables 'Temperature' and 'cloud density', six and five linguistic terms were defined respectively, as shown in Table 3.3.

Table 3.3: Linguistic Variables for Temperature and Cloud Density

SN	Temperature (ºC)	Cloud Density (Okta)
1	$S_1 = (\text{Freezing})$	$O_1 = (\text{Very low})$
2	$S_2 = (\text{Cold})$	$O_2 = (\text{Low})$
3	$S_3 = (\text{Cool})$	$O_3 = (\text{Medium})$
4	$S_4 = (\text{Mild})$	$O_4 = (\text{High})$
5	$S_5 = (\text{Warm})$	$O_5 = (\text{Very high})$
6	$S_6 = (\text{Hot})$	

3.2.5 Fuzzfying the Time Series Data

Given a traditional crisp time series, one needs a fuzzification procedure to obtain the corresponding fuzzy one. Thus, historical time series data is fuzzified using the procedure outlined in section 2.2.5 as shown in appendices C1-C6.

3.2.6 Building the HMM Model to Estimate the Fuzzy Relations

The HMM model parameters are estimated using the method outlined in section 2.2.5 expressed as:

$$A = \begin{bmatrix} 0.2500 & 0.2500 & 0.5000 & 0 & 0 & 0 \\ 0.14286 & 0.47619 & 0.33333 & 0.04762 & 0 & 0 \\ 0 & 0.13793 & 0.53448 & 0.31034 & 0.01724 & 0 \\ 0 & 0.01724 & 0.13793 & 0.51724 & 0.31034 & 0 \\ 0 & 0 & 0 & 0.22297 & 0.70946 & 0.06757 \\ 0 & 0 & 0.05263 & 0.15789 & 0.31579 & 0.47368 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0000 \\ 0.1428 & 0.09524 & 0.09524 & 0.04762 & 0.61905 \\ 0.06897 & 0.06897 & 0.06897 & 0.17241 & 0.62069 \\ 0 & 0.02586 & 0.10345 & 0.41379 & 0.45690 \\ 0.02703 & 0.10135 & 0.35135 & 0.36486 & 0.15541 \\ 0.10526 & 0.15789 & 0.42105 & 0.21053 & 0.10526 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.01093 & 0.05738 & 0.15847 & 0.31694 & 0.40437 & 0.05191 \end{bmatrix}$$

The procedural program for the estimation of the HMM model parameters can be found in Appendix E.

3.2.7 Developing the GA-HMM Model to Re-estimate Parameters

As discussed in section 2.2.6, genetic algorithm (GA) is a stochastic search method that can perform global search within the defined searching space and hopefully obtains the global optimal solution. GA was employed to increase the diversification of transition between one state and another. Assuming X is the fuzzified state matrices obtained from the set of fuzzy time series data, the genetic chromosome P formulation is obtained using the following formulation:

$$P = \begin{cases} X(:)' \ \ if \ \ X \in (m \times m) \\ \\ X(\ ) \ \ if \ \ X \in (1 \times m) \end{cases} \tag{3.4}$$

In GA, individuals are usually represented as a string of chromosomes (genes). In this research, an individual represent a HMM directly with a real-coded representation. Thus, one gene correspond to one coefficient of the HMM, and the whole individual encoded as a string or vector of real numbers by using the mechanism expressed in equation (3.4). Equation (3.4) is to ensure that all the HMM parameter matrices discussed in equations (2.24), (2.26) and (2.27) are converted to strings of chromosomes and a random genetic process conducted on the chromosome in order to reduce the presences of zero state and eventually increase transition between states. Of course, one must remember that several constraints will have to be satisfied in this string representation i.e. the sum of coefficients of each row must equal 1.In this, the re-estimate model parameters is shown as:

$$A_{GA-HMM} = \begin{bmatrix} 0.2500 & 0.2600 & 0.4800 & 0.0100 & 0 & 0 \\ 0.1619 & 0.4572 & 0.3428 & 0.0460 & 0 & 0 \\ 0 & 0.1486 & 0.5131 & 0.3210 & 0.01720 & 0 \\ 0 & 0.0172 & 0.1482 & 0.4965 & 0.3206 & 0 \\ 0 & 0 & 0 & 0.2372 & 0.6811 & 0.096 \\ 0 & 0 & 0.0526 & 0.1579 & 0.3253 & 0.4547 \end{bmatrix}$$

$$B_{GA-HMM} = \begin{bmatrix} 0 & 0 & 0 & 0.0200 & 0.9600 \\ 0.1410 & 0.0962 & 0.0942 & 0.0600 & 0.5952 \\ 0.0690 & 0.0690 & 0.0711 & 0.1793 & 0.5993 \\ 0.0010 & 0.0269 & 0.1081 & 0.4085 & 0.4469 \\ 0.1074 & 0.1621 & 0.4115 & 0.2126 & 0.1053 \\ 0.1053 & 0.1579 & 0.4211 & 0.2105 & 0.1053 \end{bmatrix}$$

$$\pi_{GA-HMM} = \begin{bmatrix} 0.0109 & 0.0574 & 0.1612 & 0.3197 & 0.4044 & 0.0519 \end{bmatrix}$$

Careful observation of the re-estimated probability matrices shows that the model parameters were slightly modified when compared to the original estimated model parameters. These include the proper distribution of the element values in the matrices and also the missing elements that were not initially captured in both matrices A and B.

The procedural program for the GA-based model can be found in Appendix E.

3.2.8 Smoothing HMM Model Parameters

As discussed in section 2.2.7, the traditional smoothing methods reviewed in the literature were commonly applied in the field of speech recognition, in which the states are crisp in nature, and thus, they are inappropriate for the case of fuzzy time series. Hence, the need for a smoothing technique that can deal with the fuzziness existing in the HMM- based FTS. Thus, the following smoothing techniques were adopted in this research:

$$A^s = A + A^L \tag{3.5}$$

Where:

A^s is the smoothed transition matrix, A is the original estimated transition matrix and A^L is the Adjusted transition matrix defined as:

$$A^L(:,[i]) = \frac{1}{2}k \times A^P(:,[i-1]) - k \times A^P(:,[i]) + \frac{1}{2}k \times A^P(:,[i+1]), i = 2,3,4,...,n-1 \tag{3.6}$$

Where:

A is the matrix to be smoothed

k is the smoothing factor that represents the degree of smoothness and A^P is the peak matrix of A determined as:

$$a_{ij}^P = \begin{cases} a_{ij}, \ if \ \ a_{ij} \ is \ a \ peak \ in \ row \\ \\ 0, \ Otherwise. \end{cases}$$

The confusion matrix *B* was smoothed similarly.

The procedural program for smoothing the model parameters can be found in Appendix E.

3.2.9 Calculating Forecast Outputs

As discussed in section 2.2.5, in the standard HMM FTS framework, the forecasting result is located at the state with the highest probability. However, according to the probability theorem, an event with a higher probability only has a greater chance of occurring, but may not necessarily occur. Due to the nature of conjecture and randomness, a stochastic

simulation, based on Monte Carlo method was adopted to estimate the true outcome of each forecast. The Monte Carlo method provides approximate solutions by stochastic sampling experiments and solves problems based on random numbers and probability statistics. The forecasting process consists of three subsequent tasks: probabilistic forecast, normalization and Monte Carlo simulation expressed as follows:

Given two fuzzy time series,

$$F(t) = \{f(t) = (S_i, t) / t = 1,2,...,T, i = 1,2,...,n\} \quad (3.7)$$

$$G(t) = \{g(t) = (O_i, t) / t = 1,2,...,T, j = 1,2,...,m\} \quad (3.8)$$

The fuzzy relation among $F(t)$, $G(t)$, and $F(t-1)$ can be formulated as a fuzzy relational equation:

$$F(t) = (F(t-1), G(t)) \circ R(t, t-1) \quad (3.9)$$

Where:

$F(t)$ and $G(t)$ are the hidden and observed state sequences, respectively. To solve the forecasting problem of $f_i(t)$, which is dependent on $f_i(t-1)$ and $g_i(t)$, the theory of HMM is applied. Thus, the probability of $S_i t$ is determined by $P_r(S_i, t / S_x, t-1)$ and $P_r(O_y, t / S_i, t)$.

The probabilities of all possible hidden states occurring at time $t(t \geq 2)$ with the transition influence of the previous hidden state $S_x, t-1$ and the observation state O_k, t are then computed. This relation is represented by the function $\alpha_t(S_x, t-1, O_y, t)$ which is defined as follows:

$$\alpha_t(S_x, t-1, O_y, t) = B^S(:, [y])^T .* A^S([x], :) \quad (3.10)$$

Where:

$A^S([x], :)$ is the xth row of state-transition matrix A^S and $B^S(:, [y])$ is the yth column of confusion matrix B^S. The symbol of operator "$.*$" is an array multiplication, and thus, $A \cdot * B$ means the element-by-element vector multiplication of A^S and B^S.

For the special case when $t = 1$, where the observation state is the only information available, the previous hidden state does not exist, and therefore the probability of the hidden state S_i at $t = 1$ was determined by a function $\beta_1((O_y,1))$, which is defined as:

$$\beta_1((O_y,1)) = \pi \cdot * B^S(:,[y]^T \tag{3.11}$$

Where:

".*" is defined as element by element vector multiplication.

$\alpha_t(S_x, t-1, O_y, t)$ and $\beta_1((O_y,1))$ result in a probability matrix presenting the probabilities of hidden states that may occur.

For the model to obey the row stochasticity constraint, the normalization was performed for the probability vectors of functions $\alpha_t(S_x, t-1, O_y, t)$ and $\beta_1((O_y,1))$, which are expressed, respectively, as:

$$N\alpha_t(S_x, t-1, O_y, t) = [\alpha_t^{S_1}, \alpha_t^{S_2}, ..., \alpha_t^{S_n}] \tag{3.12}$$

$$N\beta_1((O_y,1)) = [\beta_1^{S_1}, \beta_1^{S_2}, ..., \beta_1^{S_n}] \tag{3.13}$$

Where:

$$\alpha_t^{S_i} = \frac{P_r(O_y, t/S_i, t) P_r(S_i, t/S_x, t-1)}{\sum_{i=1}^{n} P_r(O_y, t/S_i, t) P_r(S_i, t/S_x, t-1)}$$

$$\beta_1^{S_i} = \alpha_t^{S_i} = \frac{P_r(O_y, 1/S_i, 1)}{\sum_{i=1}^{n} P_r(O_y, 1/S_i, 1)}$$

A stochastic experiment consisting of *l* Monte Carlo simulations is conducted to determine the forecasting result S_i from $\alpha_t(S_x, t-1, O_y, t)$ or $\beta_1((O_y,1))$. These forecasting results were represented with a vector C as:

$$C = [c_1, c_2, ..., c_n] \tag{3.14}$$

Where:

c_i is the number of forecasting hidden states belonging to S_i and

$$l = \sum_{i=1}^{n} c_i$$

The procedural program for carrying out these processes can be found in appendix E.

3.2.10 Defuzzifying the Forecasting Outputs

For simplicity, the defuzzification method based on center of gravity method was adopted. This is expressed as follows:

$$Deffuzified\ Output = \frac{\sum_{i=1}^{n} c_i \times t_i}{\sum_{i=1}^{n} c_i} \tag{3.15}$$

Where:

$$t_i = \begin{cases} \frac{1}{1.5}(m_i + 0.5 \times m_i),\ i = 1 \\ \frac{1}{2}(0.5 \times m_{i-1} + m_i + 0.5 \times m_{i+1}),\ i = 2,3,...,n-1 \\ \frac{1}{1.5}(0.5 \times m_{n-1} + m_n),\ i = n \end{cases}$$

and m_i is the middle point of interval u_i

The procedural program for carrying out these processes can be found in appendix E.

3.3 Performance Evaluation

In this section, the performance of the developed HMM based FTS forecasting model when applied on the bivariate benchmark FTS data of the daily average temperature and cloud density of Taipei is evaluated using the performance measures of MSE and AFEP as outlined in section 2.2.8 and the results are compared with those obtained by (Li & Cheng, 2012).

3.4 Application of the Developed GA-HMM based FTS Model to Forecast Short-term Internet Traffic Data of ABU, Zaria

In this section, the application of the developed model in section 3.2 to forecast short-term Internet traffic data of ABU, Zaria was demonstrated. However, it should be noted that although the same forecasting steps are followed, the values for the universes of discourse, defined fuzzy sets and the partition intervals would all be different, as they depend solely on the historical time series data used for the analysis.

3.4.1 Obtaining Historical Time Series Data

The time series data considered for the application demonstration was the short-term Internet traffic data of ABU, Zaria. The Internet traffic data was obtained for the period between 2/29/2016 12:00:00 AM - 3/27/2016 12:00:00 AM using PAESSLER PRTG Network Monitor. The data was divided into training and testing data set as was carried out in section 3.2.1. The training data are given in appendix G while the testing data can be found in appendix I.

3.4.2 Determination of the Universe of Discourse

In defining the universe of discourse, the minimum value $D_{\min}$ and the maximum value $D_{\max}$ of the historical data sets were first determined. Then the universes of discourse will be determined using the procedure outlined in section 2.2.5. From the data of appendix G, the values for the hidden variables and observation variables were obtained as shown in Table 3.4.

Table 3.4: Internet Traffic Universes of Discourse for Hidden and Observation Variables

SN	Total Download and Upload Traffic (MB)	Total Download Traffic (MB)
1	$D_{TT\max} = 8773$	$D_{IT\max} = 7779$
2	$D_{TT\min} = 73$	$D_{IT\min} = 33$
3	$D_{TT1} = 1$	$D_{IT1} = 3$
4	$D_{TT2} = 5$	$D_{IT2} = 1$
5	$U_{TT} = (72,8778)$	$U_{IT} = (30,7780)$

3.4.3 Partition the Universe of Discourse into Several Even-Length Intervals

Having obtained the interval length as outlined in section 2.2.5, the universes of discourse U_s and U_o are then partitioned into n and m equal intervals respectively using the procedure in section 3.2.3. In this application demonstration, the universes of discourse for the hidden and observation variables were partitioned into 6 and 5 intervals respectively, as shown in Table 3.5.

Table 3.5: Interval Lengths for the Hidden and Observation Variables

SN	Total Download and Upload Traffic (MB)	Total Download Traffic (MB)
1	$u_1 = [72,1523]$	$v_1 = [30,1580]$
2	$u_2 = [1523,2974]$	$v_2 = [1580,3130]$
3	$u_3 = [2974,4225]$	$v_3 = [3130,4680]$
4	$u_4 = [4425,5876]$	$v_4 = [4680,6230]$
5	$u_5 = [5876,7327]$	$v_5 = [6230,7780]$
6	$u_6 = [7327,8778]$	
7	Interval Length= 1451	Interval Length= 1550

3.4.4 Defining Fuzzy Sets on the Universe of Discourse

Fuzzy sets of choice that properly suites the time series data are defined for the obtained intervals as discussed in section 2.2.5. There is no restriction on the number of the fuzzy sets

defined. In this research n and m linguistic fuzzy sets were defined for the hidden and observation variables' intervals, respectively. In this application demonstration, for the linguistic variables 'Total Download and Upload' and 'Total Download', six and five linguistic terms were defined respectively, as shown in Table 3.6.

Table 3.6: Internet Traffic Linguistic Variables

SN	Total Download and Upload Traffic (MB)	Total Download Traffic (MB)
1	$S_1 = (\text{Very Low Traffic})$	$O_1 = (\text{Very low Traffic})$
2	$S_2 = (\text{Low Traffic})$	$O_2 = (\text{Low Traffic})$
3	$S_3 = (\text{Medium Traffic})$	$O_3 = (\text{Medium Traffic})$
4	$S_4 = (\text{High Traffic})$	$O_4 = (\text{High Traffic})$
5	$S_5 = (\text{Very High Traffic})$	$O_5 = (\text{Very high Traffic})$
6	$S_6 = (\text{Extremely High Traffic})$	

3.4.5 Fuzzfying the Time Series Data

Given a traditional crisp time series, one needs a fuzzification procedure to obtain the corresponding fuzzy type. Thus, historical time series data is fuzzified using the procedure outlined in section 2.2.5 as shown in appendices H.

3.4.6 Building the HMM Model to Estimate the Fuzzy Relations

The HMM model parameters are estimated using the method outlined in section 2.2.5 expressed as:

$$A = \begin{bmatrix} 0.7777 & 0 & 0 & 0.1111 & 0 & 0.1111 \\ 0 & 0 & 0.6666 & 0.3333 & 0 & 0 \\ 0.0833 & 0.0833 & 0 & 0.5833 & 0.25 & 0 \\ 0.0196 & 0.03921 & 0.0980 & 0.6470 & 0.1764 & 0.0196 \\ 0 & 0 & 0.02 & 0.035 & 0.885 & 0.06 \\ 0 & 0 & 0.01098 & 0.02197 & 0.12087 & 0.8461 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.3333 & 0.6666 & 0 & 0 & 0 \\ 0 & 0.16667 & 0.83333 & 0 & 0 \\ 0 & 0 & 0.2352 & 0.7647 & 0 \\ 0 & 0 & 0 & 0.975 & 0.025 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.00819 & 0.02459 & 0.03278 & 0.13934 & 0.5464 & 0.24863 \end{bmatrix}$$

The procedural program for the estimation of the HMM model parameters can be found in Appendix K.

3.4.7 Developing the GA-HMM Model to Re-estimate Parameters

The procedure described in section 3.2.7 was applied to obtain the re-estimated values for the HMM parameters. These are expressed as:

$$A_{GA-HMM} = \begin{bmatrix} 0.2500 & 0.2500 & 0.5000 & 0 & 0 & 0 \\ 0.1429 & 0.5238 & 0.3333 & 0.0476 & 0 & 0 \\ 0 & 0.1356 & 0.5254 & 0.3220 & 0.0169 & 0 \\ 0 & 0.0171 & 0.1453 & 0.5128 & 0.3077 & 0.0085 \\ 0 & 0 & 0 & 0.2162 & 0.7095 & 0.0676 \\ 0 & 0 & 0.0526 & 0.1579 & 0.0405 & 0.4737 \end{bmatrix}$$

$$B_{GA-HMM} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.000 \\ 0.1429 & 0.0952 & 0.0339 & 0.0476 & 0.6190 \\ 0.0678 & 0.0678 & 0.0847 & 0.1695 & 0.6271 \\ 0 & 0.0256 & 0.1026 & 0.4188 & 0.4188 \\ 0.0270 & 0.09046 & 0.3378 & 0.3649 & 0.1554 \\ 0.1053 & 0.1579 & 0.4211 & 0.2105 & 0.1053 \end{bmatrix}$$

$$\pi_{GA-HMM} = \begin{bmatrix} 0.0109 & 0.0574 & 0.1612 & 0.3197 & 0.4044 & 0.0519 \end{bmatrix}$$

Careful observation of the re-estimated probability matrices shows that the model parameters were slightly modified when compared to the original estimated model parameters. These

include the proper distribution of the element values in the matrices and also the missing elements that were not initially captured in both matrices A and B. The procedural program for the process can be found in k.

3.4.8 Smoothing HMM Model Parameters

The model parameters were smoothed using the procedure described in section 3.2.8 and the procedural program for the process can be found Appendix K.

3.4.9 Calculating Forecast Outputs

The procedure discussed in section 3.2.9 was also applied to the Internet data traffic and the procedural program for carrying out these processes can be found in appendix K.

3.4.10 Defuzzifying the Forecasting Outputs

Having obtained the forecast outputs in a fuzzy form, it was converted back in to crisp form using the method adopted for section 3.2.10. The procedural program for the defuzzification process can be found in appendix K.

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Introduction

In this section, the performance of the improved GA-HMM based FTS model on the obtained bivariate FTS data and its application on the Internet traffic data of ABU Zaria are discussed and relevant results reported.

4.2 Obtained Results for the Developed GA-HMM based FTS Forecasting Model

In this section, the obtained forecast results obtained for the bivariate HMM FTS data are presented and its performance evaluated.

4.2.1 Forecast Outputs of the Developed Model

Finally, the developed GA-HMM based FTS model was used to forecast the 1996 testing data set given in appendix B. The testing data was first fuzzified as shown in appendices D1-D2 and the forecasts was carried out using the procedure discussed in section 3.2.9.

4.2.2 Results of the Defuzzified the Forecasting Outputs.

The technique discussed in section 3.2.10 was employed and the forecasted values are shown in appendices F1 and F2. The experimental results are plotted in graphs for all the months as shown in Figures 4.1- 4.4.

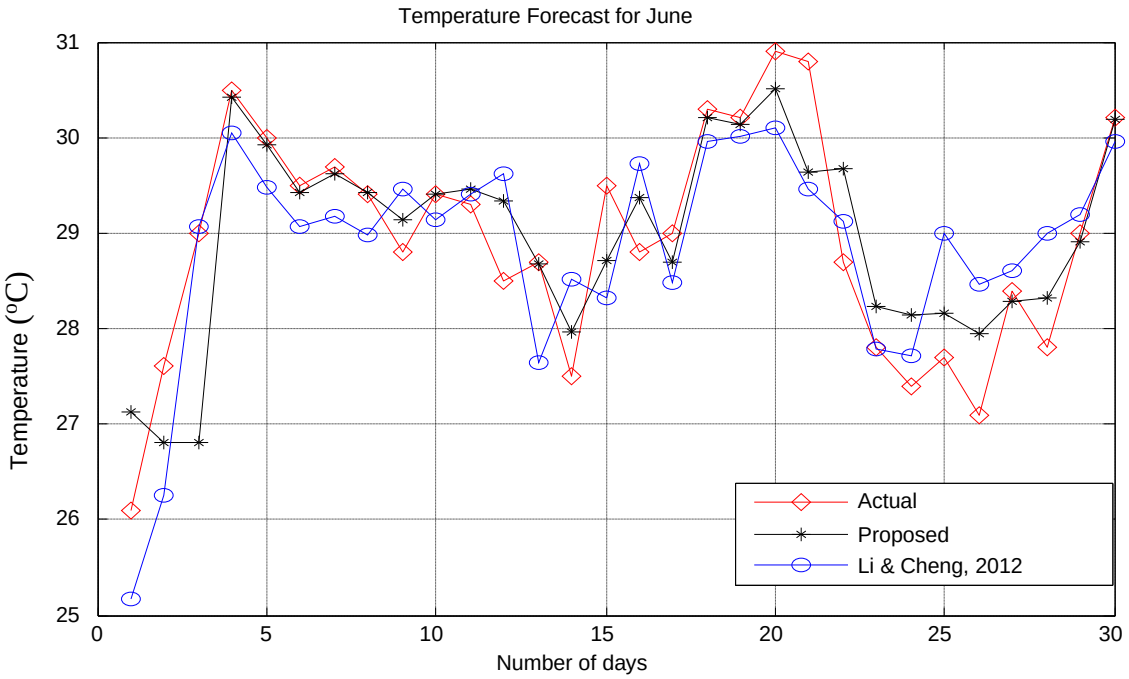


Figure 4.1: Comparison between Actual and Forecasted Values for June, 1996

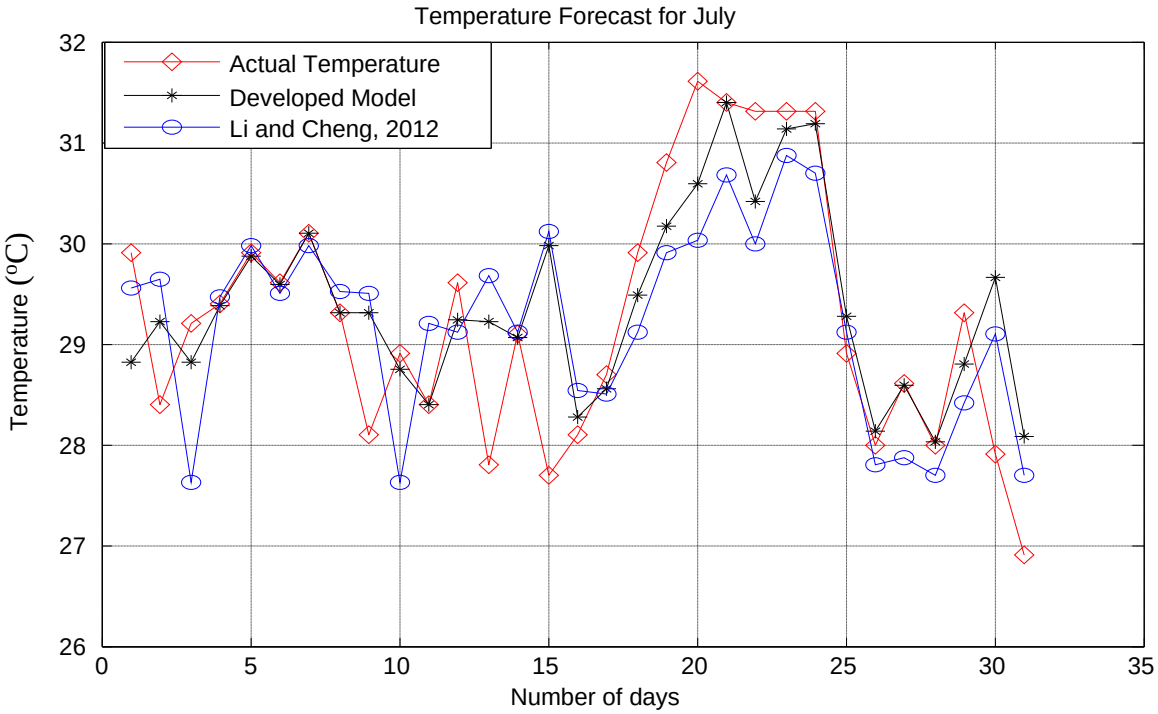


Figure 4.2: Comparison between Actual and Forecasted Values for July, 1996

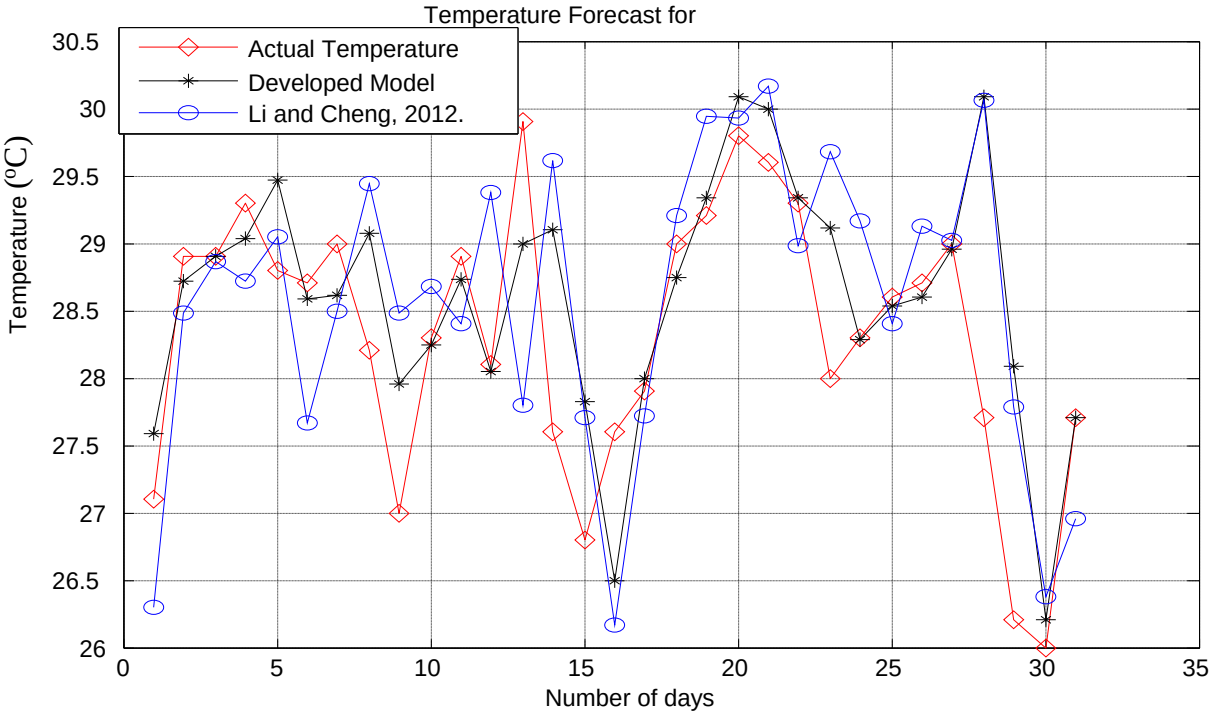


Figure 4.3: Comparison between Actual and Forecasted Values for August, 1996

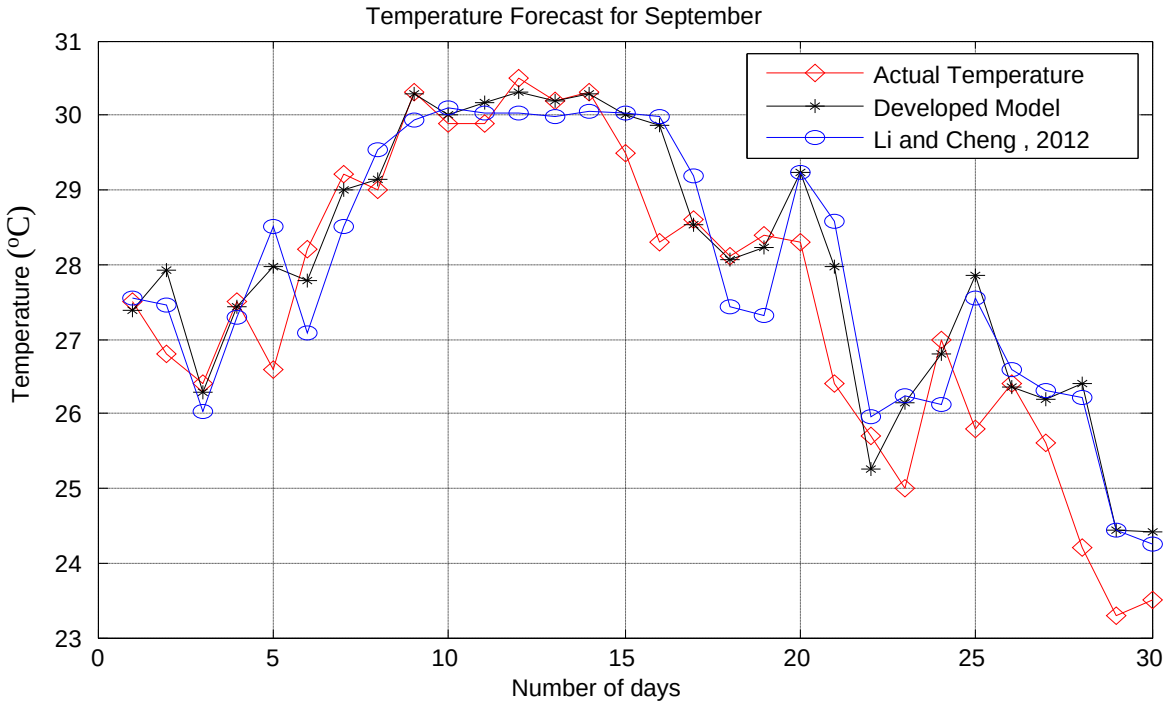


Figure 4.4: Comparison between Actual and Forecasted Values for September

Figures 4.1–4.4 shows the graphical representations for the actual temperature, the temperature forecast obtained using the developed model and the forecast obtained in (Li &

Cheng, 2012), when applied to forecast the bivariate benchmark HMM-FTS data for the months of June, July, August and September, 1996. From the graphs, it can be seen that the forecasted temperature using the developed model; more closely follows the trend of the actual temperature than when compared with the forecast obtained using model of (Li & Cheng, 2012). Thus, it can be concluded that forecasts obtained from the developed model are quite compatible with the observations that belong to the test set.

4.3 Performance Evaluation on the Bivariate HMM-FTS benchmark Data

The performance of the developed HMM based fuzzy time series forecasting model was evaluated using the performance measures of the Mean Square Error (MSE) and the Average Forecasting Error Percentage (AFEP). The forecast results of the developed model was compared with those obtained in the work of (Li & Cheng, 2012).To make the comparison fair, the same intervals and factors were used. The MSE and AFEP results are shown in Table 4.1.

Table 4.1: Performance Evaluation of the developed Model and (Li & Cheng, 2012)

Performance Measures	(Li & Cheng, 2012)	Proposed Model
	Model 2	
MSE	0.933	0.5976
AFEP	2.7464	1.8673

It can be seen from Table 4.1 that the results of the developed method have the smallest MSE and AFEP values of 0.5976 and 1.8673 respectively, when compared with the values obtained for both models in (Li & Cheng, 2012). Thus, according to the application results, the developed model obtained better forecasting performance in-terms of both the MSE and AFEP with improvements of 35% and 32%, respectively.

4.4 Application of the GA-HMM FTS Model on the Internet Traffic of ABU, Zaria.

In this section, the obtained forecast results for the ABU Internet traffic data are presented.

4.4.1 Forecast outputs of the developed model

The developed GA-HMM based FTS model was used to forecast the Internet traffic data of Monday, 21st and Tuesday 22nd March, 2016 given in appendix I. The testing data was first fuzzified as shown in appendix J after which forecast was performed using the Matlab program outlined in appendix K.

4.4.2 Results of the Defuzzified the Forecasting Outputs.

The technique discussed in section 3.2.10 was employed and the results are shown in appendix L. The empirical results were also shown graphically in Figures 4.5 and 4.6.

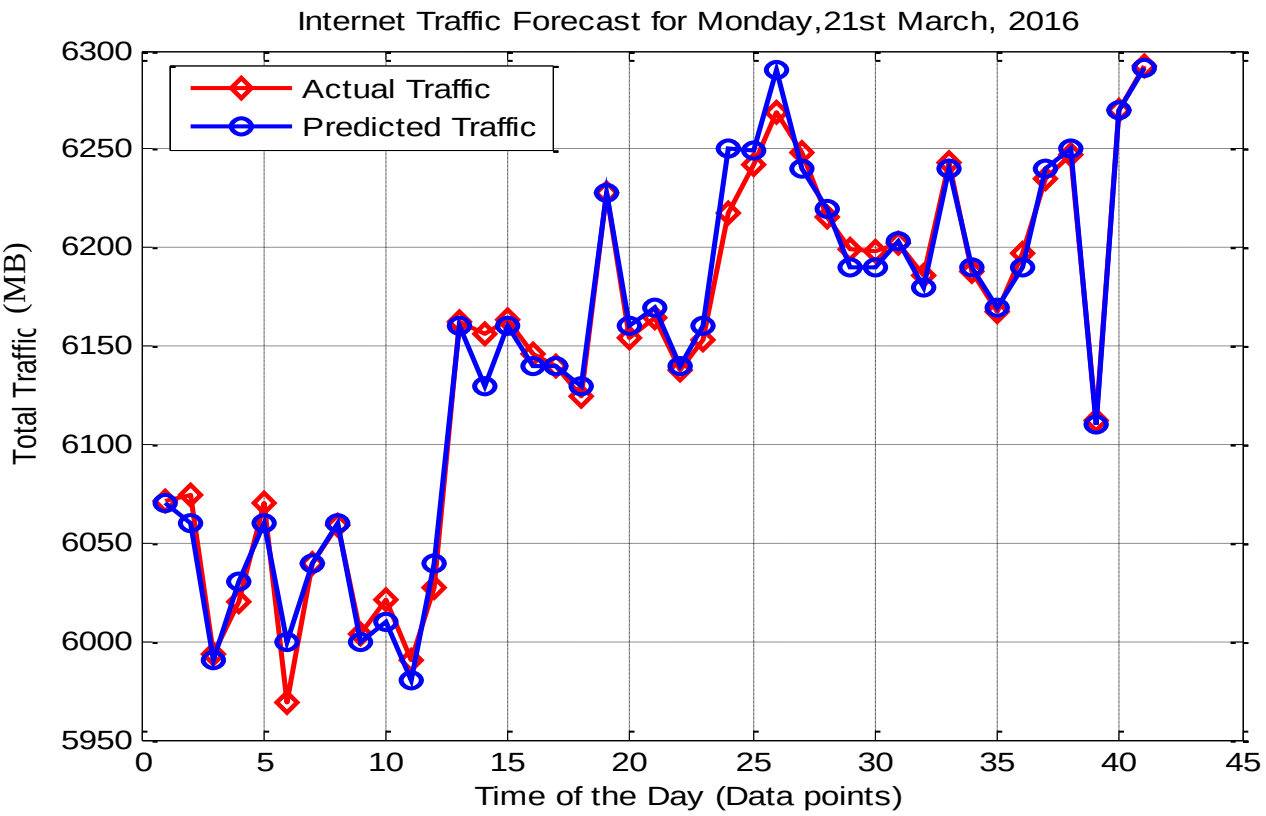


Figure 4.5: Comparison of the Actual and Forecasted Internet Data Traffic for 21st March, 2016.

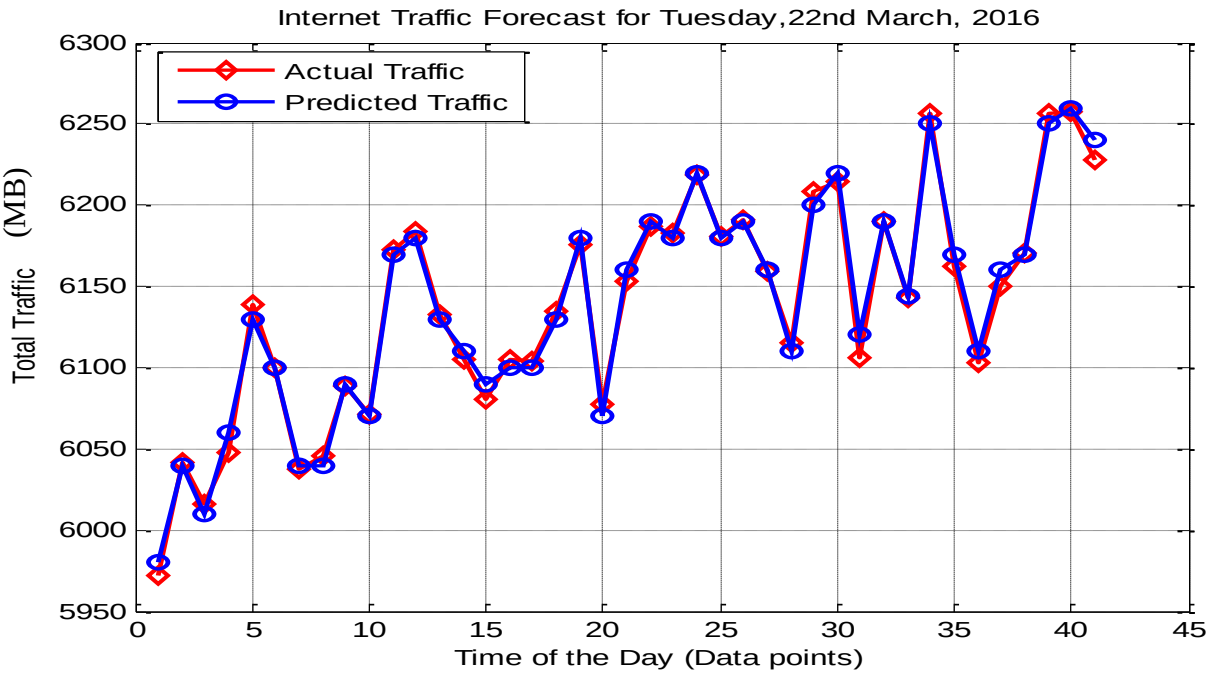


Figure 4.6: Comparison of the Actual and Forecasted Internet Data Traffic for, 22nd March, 2016.

Figures 4.5 and 4.6 shows the graphical representations for the forecast of Monday, 21st and Tuesday, 22nd March, 2016. From the figures, it can be seen that the forecasted Internet data traffic follows the trend of the actual traffic Internet data. Thus, the empirical results indicate good forecasting accuracy considering the large size of these traffics and their randomness.

4.4.3 Performance Evaluation of ABU, Zaria Internet Data Traffic

Performance of the developed HMM based fuzzy time series forecasting model on the Internet data traffic of ABU, Zaria was also evaluated using the performance measures of the Mean Square Error (MSE) and the Average Forecasting Error Percentage (AFEP). The MSE and AFEP results are shown in Table 4.2.

Table 4.2 Performance Evaluation of the forecasted ABU, Zaria Internet Traffic

Performance Measures	Predicted Traffic
MSE	68.32392
AFEP	0.08904

It can be seen from Table 4.2 that the results of the developed method have the MSE and AFEP values of 68.32392 and 0.08904, respectively. The empirical results indicates that the proposed method can still make good forecasts considering the large size of these traffics and their randomness.

4.5 Computer Specification

The experiments were performed on Toshiba Satellite C55-A with its specifications shown in Table 4.3.

Table 4.3: Computer System Specification

S/N	Items	Specification
1	Processor	Intel R Celeron R
2	Random Access Memory (RAM)	4.00 GB
3	Storage Capacity	500 GB
4	Operating System (OS)	64-bit Operating System, x64based processor
5	Rating	4.7

4.6 Genetic Algorithm Parameter Specification

Table 4.4: Parameter Values for Genetic Algorithm

S/N	Items	Specification
1	Population Size	50
2	Percentage Crossover	0.7
3	Percentage Mutation	0.2
4	Maximum Iteration	100

CHAPTER FIVE

CONCLUSION, SUMMARY, AND RECOMMENDATION

5.1 Summary

This research has developed an improved hidden markov based fuzzy time series forecasting model using genetic algorithm, which was aimed at improving the forecast accuracy of the existing hidden markov based models. The whole dissertation is summarized in this chapter. Conclusion, recommendation and limitation encountered during the course of this research are presented. Areas of future research have also been suggested.

5.2 Conclusion

The representation of fuzzy relations that are obtained from a fuzzy time series plays a key role in forecasting. Some of them use the rule based representation, which is apt to encounter the problem of missing information, rule redundancy and computational overhead, thus affects the forecasting performance. In order to overcome some of the problems (improper estimates of the fuzzy relations) associated with existing HMM-FTS based models, a new improved fuzzy time series model based on hidden markov model and genetic algorithm was developed. The developed model determines the membership values systematically by using the equal length technique in the fuzzification stage. These membership values were used in the stage of determination of fuzzy relations and so better forecasting results can be obtained by using more information. The GA-HMM based FTS model was developed in MATLAB R2014a. Accuracy analysis was carried out on the output of both the developed model and the most accurate model in the literature (Li & Cheng, 2012). The GA-HMM FTS model shows an accuracy improvement of over 30% on a bivariate FTS test data used in this research. Finally, the developed model was applied to forecast the short-term Internet data traffic of ABU, Zaria and its performance evaluated. The empirical results indicate a good accuracy of prediction considering the large size of these traffics and their randomness.

5.3 Significant Contributions

The significant contributions of this research work are as follows:

i) The developed GA-HMM model significantly improved the forecast accuracy of (Li & Cheng, 2012) FTS bivariate benchmark data of the daily average temperature and cloud density of Taipei by 35% and 32% using the performance measures of MSE and AFEP respectively.

ii) Finally, the developed model was applied to forecast the short-term Internet data traffic of ABU, Zaria and the empirical results indicates that the proposed method can still make good forecasts considering the large size of these traffics and their randomness

5.4 Limitations

The limitations of this research work are highlighted as follows:

a) In the development of the model, it was assumed that the relationships between the observed factors are independent; however some real data do not conform to this assumption.
b) The partitioning of the universe of discourse was obtained using equal length interval, thus limiting the robustness of the fuzzy time series model.

5.5 Recommendations for Further Work

The following possible areas of further work are recommended for consideration for future research:

a) Enhancing the developed model so that it is time variant to improve its forecasting accuracy.
b) The developed method can be modified for high order HMM based fuzzy time series forecasting models.
c) It is expected that in future studies researchers can concentrate on a different optimization technique for improving the estimate of the HMM parameters.

REFERENCES

Adeola, K. O. (2008). The effect of interval length and model basis on fuzzy time series: Unpublished master's dissertation. Ahmadu Bello University, Zaria, Nigeria.

Aladag, C. H., Yolcu, U., Egrioglu, E., & Dalar, A. Z. (2012). A new time invariant fuzzy time series forecasting method based on particle swarm optimization. *Applied Soft Computing, 12*(10), 3291-3299.

Aladag, C. H., Yolcu, U., & Egrioglu, E. (2013). Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. *Expert Systems with Applications, 4*0(3), 854–857.

Babakura, A., Sulaiman, M. N., Mustapha, N., & Perumal, T. (2014). HMM-based decision model for smart home environment. *International Journal of Smart Home, 8*(1), 129-138. doi: 10.14257/ijsh.2014.8.1.14

Box, G. E., & Jenkins, G. M. (1976). *Time series analysis: Forecasting and control.* San Francisco: Holden-Day, c1976.

Bas, E., Uslu, V. R., Yolcu, U., & Egrioglu, E. (2014). A modified genetic algorithm for forecasting fuzzy time series. *Applied Intelligence, 41*(2), 453-463.

Cai, Q. S., Zhang, D., Wu, B., & Leung, S. C. (2013). A novel stock forecasting model based on fuzzy time series and genetic algorithm. Paper presented at the Proceedings - International Conference on Computational Science. 2016.

Chen, S., & Goodman. J. (1996). *An empirical study of smoothing techniques for language modeling*. Paper presented at the 34th Annual Meeting of Association of Computer Linguistics, 1996. Santa Cruz.

Dai, J. (1995). Robust estimation of HMM parameters using fuzzy vector quantization and Parzen's window. *Pattern Recognition. 28*(1), 53–57.

Egrioglu, E., Bas, E., Yolcu, U., & Aladag, C. H. (2015). A fuzzy time series forecasting method based on operation of union and feed forward artificial neural network. *American Journal of Intelligent Systems, 5*(3), 81-91. doi: 10.5923/j.ajis.20150503.01

Egrioglu, E., Bas, E., Yolcu, U., & Uslu, V. R. (2013). A fuzzy time series analysis approach by using differential evolution algorithm based on the number of recurrences of fuzzy relations. *American Journal of Intelligent Systems, 3*(2), 75-82. doi: 10.5923/j.ajis.20130302.04

Egrioglu, E. (2014). A New Fuzzy Time Series Method Based On Artificial Bee Colony Algorithm. *An Official Journal of Turkish Fuzzy Systems Association*, 5(1), 59-77.

Egrioglu, E. (2015). PSO-based high order time invariant fuzzy time series method. *Economic Modelling 38*(4), 633–639.

Emmanuel, R., & Thierry, D. (2013). Making use of partial knowledge about hidden states in HMMs. *IEEE Transactions on Fuzzy Systems,* 11(1), 1-12.

Elizabeth, W., & Jana, C. (2004). *Evaluating forecast accuracy: Ökonometrische Prognose*. Vienna, Austria: University of Vienna Press.

Haneen, A., Ibraheem, K., & Salim, A. (2014). A hybrid algorithm to forecast enrolment based on genetic algorithms and fuzzy time series. *The International Arab Journal of Information Technology, 11(3).*250-257.

Hongxing, Y., Yutong, L., Lin L., & Ronghui, Q. (2011). First order multivariate markov chain model for generating annual weather data for Hong Kong. *Energy and Buildings*, 43, 2371–2377. doi:10.1016/j.enbuild.2011.05.035

Huarng, K. (2001). Effective lengths of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets and Systems*, 123(3), (pp. 387-394).

Hui-Kuarng, Y. (2005). A refined fuzzy time series model for forecasting. *Physica* (pp. 657 – 681).

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting. *Journal of Statistical Software, 26*(3), 1–22.

Koo, J. M., & Un, C. K. (1990). Fuzzy smoothing of HMM parameters in speech recognition, *Electron Lett* (pp. 743–744).

Li, S. T., & Cheng, Y. C. (2006). A hidden markov model-based forecasting model for fuzzy time series. *WSEAS Transactions on systems, 5*(8), 1255-1266.

Li, S. T., & Cheng, Y. C. (2010). A stochastic HMM-based forecasting model for fuzzy time series. *IEEE Transactions on Systems, Man, and Cybernetics, 40*(5), 1255-1266. doi: 10.1109/TSMCB.2009.2036860

Li, S. T., & Cheng, Y. C. (2012). Fuzzy time series forecasting with a probabilistic smoothing hidden markov model. *IEEE Transactions on Systems, Man, and Cybernetics, 20*(2), 291-304. doi: 10.1109/TFUZZ.2011.2173583

Majoros, W. H. (2008). Advanced methods for parameter estimation: Cambridge, United Kingdom: *Cambridge University Press.*

Markov, A. A. (1913). An example of statistical Investigation in the Text of 'Eugene Onyegin' Illustrating Coupling of 'tests' in chains. *Proceedings of the Academy of Sciences.*

Park, C. H., & Lee, J. S. (2006). Training hidden markov models by hybrid simulated annealing for visual speech recognition. Paper presented at the 2006 IEEE Conference on Systems, Man, and Cybernetics, 2006.

Radmehr, F., & Gharneh, N. S. (2012). Forecasting method based on high order fuzzy time series and simulated annealing technique. *South African Journal of Industrial Engineering*, 23(2), 76 -190.

Sheng, T. L., Shu, C. K., Yi, C. C., & Chih, C. C. (2009). Deterministic vector long-term forecasting for fuzzy time series. *Fuzzy Sets and Systems* (pp. 1852–1870).

Shrivastav, A, K., & Ekata**. (**2012). Applicability of soft computing technique for crime forecasting: A preliminary Investigation. *International Journal of Computer Science and Engineering Technology*. *3* (9), 2229-3345.

Song, Q., & Chissom, B. S**. (**1993). Forecasting enrollments with fuzzy time series. *Fuzzy Sets and Systems, 54* (1), 1-9.

Song, Q., & Chissom, B. S**. (**1994). Forecasting enrollments with fuzzy time series. *Fuzzy Sets and Systems, 62* (1), 1-8.

Subanar, A. and Abadi, M. (2011). Contribution of fuzzy systems for time series analysis. *In proceedings of the 6th SEAMS-UGM Conference*.

Sullivan, J., & Woodall, W. H. (1994). A comparison of fuzzy forecasting and Markov modeling*; Fuzzy Sets Systems, 64*(3), 279–293.

Suresh, S., Senthamarai, K. K., & Venkatesan, R. (2016). Higher order multivariate markov chain model for fuzzy time series. *Journal of Statistics and Management Systems*, 19(1), 21-35. doi: 10.1080/09720510.2014.894303

Tsaur, R. C., Yang, J. C., & Wang, H. F. (2005). Fuzzy relation analysis in fuzzy time series model, *Computers and Mathematics with Applications, 49* (4), 539-548.

Tsaur, R. C. (2012). A fuzzy time series-Markov chain model with an application to forecast the exchange rate between the Taiwan and US dollar. *International Journal of Innovative Computing, Information and Control, 8*(7), 493-4942.

Uslu, V. R., Bas, E., Yolcu, U., & Egrioglu, E. (2013). A new fuzzy time series analysis approach by using differential evolution algorithm and chronologically-determined weights. *Journal of Social and Economic Statistics, 2(*1), 18-30.

Vamitha, V., Jayanthi, M., Rajaram, S., & Revathi, T. (2012). Temperature prediction using fuzzy time series and multivariate markov chain. *International Journal of Fuzzy Mathematics and Systems, 2*(3), 217-230.

Wael, K., Eric, G., Ali M., & Robert, S. (2012). A survey of techniques for incremental learning of HMM parameters. *Information Sciences* (pp.105-130).

Wang, J. J., Sheng, T. L., Shu, C. K., Chih, C. C., & Hsuan, Y. C. (2016). *Steady-state probabilities model for fuzzy time series forecasting*. Paper presented at the Proceedings - 5th International Congress on Advanced Applied Informatics, IIAI 2016.

Yi, C. C., Pei, C. C., Chih, C. C., Hui, C. C., & Sheng, T. L. (2015). An enhanced HMM – based fuzzy time series forecasting model. *International Fuzzy Systems Association* (pp. 320-325).

Yolcu, O. C. (2014). The forecasting of Istanbul stock exchange by using a hybrid fuzzy time series approach. *Turkish Journal of Fuzzy*, 5(1), 10-26.

Zhang, D., Cai, Q., Wu, B., & Leung, C. H. (2013). *A novel stock forecasting model based on fuzzy time series and genetic algorithm*. Paper presented at the 2013 International Conference on Computational Science, 2013.

APPENDIX A1

TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE , JULY, AUGUST AND SEPTEMBER 1993 IN TAIPEI, TAIWAN (Li & Cheng, 2010)

Day	June		July		August		September	
	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)
1	27.5	86	30.2	90	29	58	26.3	98
2	25.3	96	29.3	98	29.7	80	25.4	91
3	26.2	85	29.4	89	29.9	43	25.9	43
4	25	89	30.1	81	29.8	65	26.4	5
5	23.7	95	30.7	68	29.9	58	26.7	8
6	24	96	29.8	66	30	58	26.6	29
7	24.5	96	30.5	51	29.2	69	25.6	25
8	25.8	89	29.8	75	29.7	40	25.1	36
9	24.5	100	29.3	70	28.6	93	25.5	11
10	25	95	29.6	84	28.7	89	26.4	18
11	25.7	100	28.9	68	28.6	64	28.2	35
12	28.1	76	28.5	63	28.3	60	29.6	86
13	28.7	83	29.3	84	29.1	69	28.7	98
14	29.2	63	29.4	79	28.3	91	29.2	65
15	29.6	65	29.6	73	28	86	30.1	21
16	27.7	63	30.6	46	27.8	70	31	59
17	27.4	71	31.2	63	28.8	80	30.1	48
18	26.9	79	31.4	81	30.7	55	30.2	55
19	28.5	83	31.3	71	29.9	51	30.1	23
20	28.6	89	30.9	76	28.3	70	30.1	23
21	29.4	96	27.6	94	29.2	74	29.2	83
22	30	76	27.9	65	29.1	65	28.7	83
23	29.3	74	28.9	66	29.5	84	26.7	98
24	28.3	93	30.4	85	29.2	71	25.8	99
25	28.6	81	29.9	76	29.5	58	25.1	91
26	30.2	41	28.4	71	29.8	64	26.3	74
27	31.6	53	28.6	61	29.1	71	27.1	40
28	28.2	66	28.5	85	27.5	68	27.7	44
29	28.9	59	28.9	61	27.4	53	27.8	83
30	29.8	70	29.5	50	27.3	20	24.2	98
31			28.4	79	27.3	43		

APPENDIX A2

TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE , JULY, AUGUST AND SEPTEMBER 1994 IN TAIPEI, TAIWAN (Li & Cheng, 2010)

Day	June		July		August		September	
	Temperature (^oC)	Cloud Density (Okta)	Temperature (^oC)	Cloud Density (Okta)	Temperature (^oC)	Cloud Density (Okta)	Temperature (^oC)	Cloud Density (Okta)
1	22.9	100	30.1	71	29.9	45	27	93
2	25.5	64	31.2	41	31.6	30	28.6	55
3	26.7	23	30.8	39	29.5	78	30.3	14
4	28.1	83	28.7	70	27.8	95	29.6	55
5	28.7	66	28.8	73	28.4	100	28.9	85
6	29.3	76	29.9	66	29.9	70	29.1	60
7	28.2	86	30.3	66	27.7	84	28.5	50
8	27.8	70	31.6	18	25.8	100	28.6	28
9	29.3	76	31.2	21	30.1	95	29.1	0
10	29.3	89	27	94	30.6	48	29.5	6
11	28.3	99	27.8	95	29.9	51	27.3	9
12	27.6	80	29.1	91	30.6	48	25.2	100
13	28.8	70	29.2	86	31	41	24.8	98
14	29.4	86	30.1	54	29.3	69	24.4	100
15	28.2	98	30.2	63	27.2	84	25.3	80
16	27.3	76	30.1	63	26.5	86	24.5	79
17	28.9	75	30	64	28.3	70	23.6	91
18	26.5	89	29.7	48	28.7	54	22.8	88
19	23.3	100	29.9	41	28.5	60	25.6	80
20	25.5	99	29.4	69	28.1	90	26.9	85
21	28.5	89	28.5	65	25.5	94	27.6	86
22	29	70	29.2	60	28.9	65	26.4	86
23	29.8	49	29.2	76	27.8	73	25.2	95
24	30.5	56	29.7	78	28.5	85	24.7	55
25	29.8	91	28.6	71	30	34	23.7	31
26	28.2	94	28.7	75	29.8	35	23.5	26
27	29	83	29.1	58	29.8	46	24.3	51
28	29.9	56	30.3	61	30	60	23.7	15
29	29.5	69	30.5	83	29.8	55	23.4	16
30	29.9	83	30	59	29.3	66	23.5	14
31			29.9	54	28.8	56		

APPENDIX A3

TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE , JULY,AUGUST AND SEPTEMBER 1995 IN TAIPEI, TAIWAN (Li & Cheng, 2010)

Day	June		July		August		September	
	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)
1	28.9	74	28.4	75	28.4	93	29.8	41
2	28.8	69	29.1	94	26.7	96	29.8	51
3	28.4	64	28.7	98	26.1	64	29.6	68
4	28.2	99	28.6	86	28.8	56	30.3	50
5	25.6	80	29	73	27.5	81	29.7	38
6	26	81	29.5	61	28.4	90	30	55
7	27.5	79	30.1	56	28.7	85	29.8	79
8	27.7	90	30.1	50	29.8	50	30.5	40
9	24.6	98	28.6	68	30.6	39	30.1	33
10	25	98	27.2	84	30.8	49	29.7	61
11	24.4	96	26.5	84	29.8	63	29.5	46
12	26.1	60	26.4	81	28.6	64	27.3	81
13	25.8	63	27.8	75	27.5	75	27.7	78
14	26.5	98	28.5	69	27.9	79	27.2	76
15	26	100	27.2	80	29.1	35	27.7	83
16	27.4	89	29.4	38	29.4	58	26.2	79
17	29	88	30	40	30.1	53	25.4	41
18	27.8	69	31.1	14	30	40	25.4	75
19	29.9	69	31.3	48	30.3	65	25.8	45
20	30	74	31.1	80	30.7	28	23.2	100
21	30.4	51	32.1	79	30.8	54	21.6	100
22	31.5	20	32.2	58	29.7	56	22.9	99
23	31.2	64	28.8	91	28.9	86	23.4	100
24	30.7	48	26.8	94	26.6	100	25.1	89
25	30.8	46	27	86	25.9	100	25.4	43
26	31.2	43	29	68	28.1	64	25.9	70
27	31.5	48	29	55	29.4	26	26.6	83
28	25.7	90	29.9	46	30.6	13	27.1	44
29	28	85	29.9	24	30.9	26	26.3	68
30	28	76	28.6	71	29.4	58	27.6	51
31			27.7	94	30.5	61		

APPENDIX B

TESTING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE, JULY, AUGUST AND SEPTEMBER 1996 IN TAIPEI, TAIWAN (Li & Cheng, 2010)

Day	June		July		August		September	
	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)	Temperature (°C)	Cloud Density (Okta)
1	26.1	98	29.9	70	27.1	96	27.5	89
2	27.6	95	28.4	79	28.9	76	26.8	88
3	29	40	29.2	80	28.9	59	26.4	81
4	30.5	39	29.4	66	29.3	23	27.5	69
5	30	64	29.9	53	28.8	83	26.6	71
6	29.5	84	29.6	65	28.7	80	28.2	69
7	29.7	83	30.1	56	29	63	29.2	26
8	29.4	96	29.3	66	28.2	79	29	79
9	28.8	74	28.1	63	27	64	30.3	24
10	29.4	43	28.9	85	28.3	48	29.9	24
11	29.3	66	28.4	49	28.9	63	29.9	34
12	28.5	70	29.6	49	28.1	68	30.5	35
13	28.7	91	27.8	63	29.9	84	30.2	30
14	27.5	76	29.1	59	27.6	75	30.3	34
15	29.5	68	27.7	38	26.8	83	29.5	33
16	28.8	73	28.1	29	27.6	84	28.3	43
17	29	71	28.7	73	27.9	81	28.6	41
18	30.3	55	29.9	54	29	49	28.1	18
19	30.2	36	30.8	58	29.2	23	28.4	11
20	30.9	36	31.6	28	29.8	41	28.3	48
21	30.8	71	31.4	36	29.6	34	26.4	75
22	28.7	86	31.3	66	29.3	85	25.7	38
23	27.8	95	31.3	43	28	66	25	98
24	27.4	81	31.3	25	28.3	45	27	93
25	27.7	51	28.9	88	28.6	65	25.8	84
26	27.1	69	28	90	28.7	43	26.4	55
27	28.4	73	28.6	94	29	45	25.6	95
28	27.8	54	28	88	27.7	51	24.2	98
29	29	50	29.3	63	26.2	85	23.3	100
30	30.2	43	27.9	89	26	85	23.5	94
31			26.9	100	27.7	74		

APPENDIX C1

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE AND JULY 1993 IN TAIPEI, TAIWAN

Day	June				July			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	27.5	S4	86	O5	30.2	S5	90	O5
2	25.3	S3	96	O5	29.3	S5	98	O5
3	26.2	S3	85	O5	29.4	S5	89	O5
4	25	S3	89	O5	30.1	S5	81	O5
5	23.7	S2	95	O5	30.7	S5	68	O4
6	24	S2	96	O5	29.8	S5	66	O4
7	24.5	S2	96	O5	30.5	S5	51	O3
8	25.8	S3	89	O5	29.8	S5	75	O4
9	24.5	S2	100	O5	29.3	S5	70	O4
10	25	S3	95	O5	29.6	S5	84	O5
11	25.7	S3	100	O5	28.9	S4	68	O4
12	28.1	S4	76	O4	28.5	S4	63	O4
13	28.7	S4	83	O5	29.3	S5	84	O5
14	29.2	S5	63	O4	29.4	S5	79	O4
15	29.6	S5	65	O4	29.6	S5	73	O4
16	27.7	S4	63	O4	30.6	S5	46	O3
17	27.4	S4	71	O4	31.2	S6	63	O4
18	26.9	S3	79	O4	31.4	S6	81	O5
19	28.5	S4	83	O5	31.3	S6	71	O4
20	28.6	S4	89	O5	30.9	S5	76	O4
21	29.4	S5	96	O5	27.6	S4	94	O5
22	30	S5	76	O4	27.9	S4	65	O4
23	29.3	S5	74	O4	28.9	S4	66	O4
24	28.3	S4	93	O5	30.4	S5	85	O5
25	28.6	S4	81	O5	29.9	S5	76	O4
26	30.2	S5	41	O3	28.4	S4	71	O4
27	31.6	S6	53	O3	28.6	S4	61	O4
28	28.2	S4	66	O4	28.5	S4	85	O5
29	28.9	S4	59	O3	28.9	S4	61	O4
30	29.8	S5	70	O4	29.9	S5	50	O3
31					28.4	S4	79	O4

APPENDIX C2

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR AUGUST AND SEPTEMBER, 1993 IN TAIPEI, TAIWAN

Day	August				September			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	29	S5	58	O3	26.3	S3	98	O5
2	29.7	S5	80	O5	25.4	S3	91	O5
3	29.9	S5	43	O3	25.9	S3	43	O3
4	29.8	S5	65	O4	26.4	S3	5	O1
5	29.9	S5	58	O3	26.7	S3	8	O1
6	30	S5	58	O3	26.6	S3	29	O2
7	29.2	S5	69	O4	25.6	S3	25	O2
8	29.7	S5	40	O3	25.1	S3	36	O2
9	28.6	S4	93	O5	25.5	S3	11	O1
10	28.7	S4	89	O5	26.4	S3	18	O1
11	28.6	S4	64	O4	28.2	S4	35	O2
12	28.3	S4	60	O4	29.6	S5	86	O5
13	29.1	S5	69	O4	28.7	S4	98	O5
14	28.3	S4	91	O5	29.2	S5	65	O4
15	28	S4	86	O5	30.1	S5	21	O2
16	27.8	S4	70	O4	31	S6	59	O3
17	28.8	S4	80	O5	30.1	S5	48	O3
18	30.7	S5	55	O3	30.2	S5	55	O3
19	29.9	S5	51	O3	30.1	S5	23	O2
20	28.3	S4	70	O4	30.1	S5	23	O2
21	29.2	S5	74	O4	29.2	S5	83	O5
22	29.1	S5	65	O4	28.7	S4	83	O5
23	29.5	S5	84	O5	26.7	S3	98	O5
24	29.2	S5	71	O4	25.8	S3	99	O5
25	29.5	S5	58	O3	25.1	S3	91	O5
26	29.8	S5	64	O4	26.3	S3	74	O4
27	29.1	S5	71	O4	27.1	S4	40	O3
28	27.5	S4	68	O4	27.7	S4	44	O3
29	27.4	S4	53	O3	27.8	S4	83	O5
30	27.3	S4	20	O2	24.2	S2	98	O5
31	27.3	S4	43	O3				

APPENDIX C3

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE AND JULY 1994 IN TAIPEI, TAIWAN

Day	June				July			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	22.9	S1	100	O5	30.1	S5	71	O4
2	25.5	S3	64	O4	31.2	S6	41	O3
3	26.7	S3	23	O2	30.8	S5	39	O2
4	28.1	S4	83	O5	28.7	S4	70	O4
5	28.7	S4	66	O4	28.8	S4	73	O4
6	29.3	S5	76	O4	29.9	S5	66	O4
7	28.2	S4	86	O5	30.3	S5	66	O4
8	27.8	S4	70	O4	31.6	S6	18	O1
9	29.3	S5	76	O4	31.2	S6	21	O2
10	29.3	S5	89	O5	27	S4	94	O5
11	28.3	S4	99	O5	27.8	S4	95	O5
12	27.6	S4	80	O5	29.1	S5	91	O5
13	28.8	S4	70	O4	29.2	S5	86	O5
14	29.4	S5	86	O5	30.1	S5	54	O3
15	28.2	S4	98	O5	30.2	S5	63	O4
16	27.3	S4	76	O4	30.1	S5	63	O4
17	28.9	S4	75	O4	30	S5	64	O4
18	26.5	S3	89	O5	29.7	S5	48	O3
19	23.3	S2	100	O5	29.9	S5	41	O3
20	25.5	S3	99	O5	29.4	S5	69	O4
21	28.5	S4	89	O5	28.5	S4	65	O4
22	29	S5	70	O4	29.2	S5	60	O4
23	29.8	S5	49	O3	29.2	S5	76	O4
24	30.5	S5	56	O3	29.7	S5	78	O4
25	29.8	S5	91	O5	28.6	S4	71	O4
26	28.2	S4	94	O5	28.7	S4	75	O4
27	29	S5	83	O5	29.1	S5	58	O3
28	29.9	S5	56	O3	30.3	S5	61	O4
29	29.5	S5	69	O4	30.5	S5	83	O5
30	29.9	S5	83	O5	30	S5	59	O3
31					29.9	S5	54	O3

APPENDIX C4

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR AUGUST AND SEPTEMBER 1994 IN TAIPEI, TAIWAN

Day	August				September			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	29.9	S5	45	O3	27	S4	93	O5
2	31.6	S6	30	O2	28.6	S4	55	O3
3	29.5	S5	78	O4	30.3	S5	14	O1
4	27.8	S4	95	O5	29.6	S5	55	O3
5	28.4	S4	100	O5	28.9	S4	85	O5
6	29.9	S5	70	O4	29.1	S5	60	O4
7	27.7	S4	84	O5	28.5	S4	50	O3
8	25.8	S3	100	O5	28.6	S4	28	O2
9	30.1	S5	95	O5	29.1	S5	0	O1
10	30.6	S5	48	O3	29.5	S5	6	O1
11	29.9	S5	51	O3	27.3	S4	9	O5
12	30.6	S5	48	O3	25.2	S3	100	O5
13	31	S6	41	O3	24.8	S2	98	O5
14	29.3	S5	69	O4	24.4	S2	100	O5
15	27.2	S4	84	O5	25.3	S3	80	O5
16	26.5	S3	86	O5	24.5	S2	79	O4
17	28.3	S4	70	O4	23.6	S2	91	O5
18	28.7	S4	54	O3	22.8	S1	88	O5
19	28.5	S4	60	O4	25.6	S3	80	O5
20	28.1	S4	90	O5	26.9	S3	85	O5
21	25.5	S3	94	O5	27.6	S4	86	O5
22	28.9	S4	65	O4	26.4	S3	86	O5
23	27.8	S4	73	O4	25.2	S3	95	O5
24	28.5	S4	85	O5	24.7	S2	55	O3
25	30	S5	34	O2	23.7	S2	31	O2
26	29.8	S5	35	O2	23.5	S2	26	O2
27	29.8	S5	46	O3	24.3	S2	51	O3
28	30	S5	60	O4	23.7	S2	15	O1
29	29.8	S5	55	O3	23.4	S2	16	O1
30	29.3	S5	66	O4	23.5	S2	14	O1
31	28.8	S4	56	O3				

APPENDIX C5

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE AND JULY, 1995 IN TAIPEI, TAIWAN

Day	June				July			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	28.9	S4	74	O4	28.4	S4	75	O4
2	28.8	S4	69	O4	29.1	S5	94	O5
3	28.4	S4	64	O4	28.7	S4	98	O5
4	28.2	S4	99	O5	28.6	S4	86	O5
5	25.6	S3	80	O5	29	S5	73	O4
6	26	S3	81	O5	29.5	S5	61	O4
7	27.5	S4	79	O4	30.1	S5	56	O3
8	27.7	S4	90	O5	30.1	S5	50	O3
9	24.6	S2	98	O5	28.6	S4	68	O4
10	25	S3	98	O5	27.2	S4	84	O5
11	24.4	S2	96	O5	26.5	S3	84	O5
12	26.1	S3	60	O4	26.4	S3	81	O5
13	25.8	S3	63	O4	27.8	S4	75	O4
14	26.5	S3	98	O5	28.5	S4	69	O4
15	26	S3	100	O5	27.2	S4	80	O5
16	27.4	S4	89	O5	29.4	S5	38	O2
17	29	S5	88	O5	30	S5	40	O3
18	27.8	S4	69	O4	31.1	S6	14	O1
19	29.9	S5	69	O4	31.3	S6	48	O3
20	30	S5	74	O4	31.1	S6	80	O5
21	30.4	S5	51	O3	32.1	S6	79	O4
22	31.5	S6	20	O2	32.2	S6	58	O3
23	31.2	S6	64	O4	28.8	S4	91	O5
24	30.7	S5	48	O3	26.8	S3	94	O5
25	30.8	S5	46	O3	27	S4	86	O5
26	31.2	S6	43	O3	29	S5	68	O4
27	31.5	S6	48	O3	29	S5	55	O3
28	25.7	S3	90	O5	29.9	S5	46	O3
29	28	S4	85	O5	29.9	S5	24	O2
30	28	S4	76	O4	28.6	S4	71	O4
31					27.7	S4	94	O5

APPENDIX C6

FUZZIFIED TRAINING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR AUGUST AND SEPTEMBER, 1995 IN TAIPEI, TAIWAN

Day	August				September			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	28.4	S4	93	O5	29.8	S5	41	O3
2	26.7	S3	96	O5	29.8	S5	51	O3
3	26.1	S3	64	O4	29.6	S5	68	O4
4	28.8	S4	56	O3	30.3	S5	50	O3
5	27.5	S4	81	O5	29.7	S5	38	O2
6	28.4	S4	90	O5	30	S5	55	O3
7	28.7	S4	85	O5	29.8	S5	79	O4
8	29.8	S5	50	O3	30.5	S5	40	O3
9	30.6	S5	39	O2	30.1	S5	33	O2
10	30.8	S5	49	O3	29.7	S5	61	O4
11	29.8	S5	63	O4	29.5	S5	46	O3
12	28.6	S4	64	O4	27.3	S4	81	O5
13	27.5	S4	75	O4	27.7	S4	78	O4
14	27.9	S4	79	O4	27.2	S4	76	O4
15	29.1	S5	35	O2	27.7	S4	83	O5
16	29.4	S5	58	O3	26.2	S3	79	O4
17	30.1	S5	53	O3	25.4	S3	41	O3
18	30	S5	40	O3	25.4	S3	75	O4
19	30.3	S5	65	O4	25.8	S3	45	O3
20	30.7	S5	28	O2	23.2	S2	100	O5
21	30.8	S5	54	O3	21.6	S1	100	O5
22	29.7	S5	56	O3	22.9	S1	99	O5
23	28.9	S4	86	O5	23.4	S2	100	O5
24	26.6	S3	100	O5	25.1	S3	89	O5
25	25.9	S3	100	O5	25.4	S3	43	O3
26	28.1	S4	64	O4	25.9	S3	70	O4
27	29.4	S5	26	O2	26.6	S3	83	O5
28	30.6	S5	13	O1	27.1	S4	44	O3
29	30.9	S5	26	O2	26.3	S3	68	O4
30	29.4	S5	58	O3	27.6	S4	51	O3
31	30.5	S5	61	O4				

APPENDIX D1

FUZZIFIED TESTING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR JUNE AND JULY, 1996 IN TAIPEI, TAIWAN

Day	June				July			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	26.1	S1	98	O5	29.9	S5	70	O4
2	27.6	S4	95	O5	28.4	S4	79	O4
3	29	S5	40	O3	29.2	S5	80	O5
4	30.5	S5	39	O2	29.4	S5	66	O4
5	30	S5	64	O4	29.9	S5	53	O3
6	29.5	S5	84	O5	29.6	S5	65	O4
7	29.7	S5	83	O5	30.1	S5	56	O3
8	29.4	S5	96	O5	29.3	S5	66	O4
9	28.8	S4	74	O4	28.1	S4	63	O4
10	29.4	S5	43	O3	28.9	S4	85	O5
11	29.3	S5	66	O4	28.4	S4	49	O3
12	28.5	S4	70	O4	29.6	S5	49	O3
13	28.7	S4	91	O5	27.8	S4	63	O4
14	27.5	S4	76	O4	29.1	S5	59	O3
15	29.5	S5	68	O4	27.7	S4	38	O2
16	28.8	S4	73	O4	28.1	S4	29	O2
17	29	S5	71	O4	28.7	S4	73	O4
18	30.3	S5	55	O3	29.9	S5	54	O3
19	30.2	S5	36	O2	30.8	S5	58	O3
20	30.9	S5	36	O2	31.6	S6	28	O2
21	30.8	S5	71	O4	31.4	S6	36	O2
22	28.7	S4	86	O5	31.3	S6	66	O4
23	27.8	S4	95	O5	31.3	S6	43	O2
24	27.4	S4	81	O5	31.3	S6	25	O2
25	27.7	S4	51	O3	28.9	S4	88	O5
26	27.1	S4	69	O4	28	S4	90	O5
27	28.4	S4	73	O4	28.6	S4	94	O5
28	27.8	S4	54	O3	28	S4	88	O5
29	29	S5	50	O3	29.3	S5	63	O4
30	30.2	S5	43	O3	27.9	S4	89	O5
31					26.9	S3	100	O5

APPENDIX D2

FUZZIFIED TESTING DATA OF DAILY AVERAGE TEMPERATURE AND CLOUD DENSITY FOR AUGUST AND SEPTEMBER, 1996 IN TAIPEI, TAIWAN

Day	August				September			
	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set	Temperature (°C)	Fuzzy Set	Cloud Density (Okta)	Fuzzy Set
1	27.1	S4	96	O5	27.5	S4	89	O5
2	28.9	S4	76	O4	26.8	S3	88	O5
3	28.9	S4	59	O3	26.4	S3	81	O5
4	29.3	S5	23	O2	27.5	S4	69	O4
5	28.8	S4	83	O5	26.6	S3	71	O4
6	28.7	S4	80	O5	28.2	S4	69	O4
7	29	S5	63	O4	29.2	S5	26	O2
8	28.2	S4	79	O4	29	S5	79	O4
9	27	S4	64	O4	30.3	S5	24	O2
10	28.3	S4	48	O3	29.9	S5	24	O2
11	28.9	S4	63	O4	29.9	S5	34	O2
12	28.1	S4	68	O4	30.5	S5	35	O2
13	29.9	S5	84	O5	30.2	S5	30	O2
14	27.6	S4	75	O4	30.3	S5	34	O2
15	26.8	S3	83	O5	29.5	S5	33	O2
16	27.6	S4	84	O5	28.3	S4	43	O3
17	27.9	S4	81	O5	28.6	S4	41	O3
18	29	S5	49	O3	28.1	S4	18	O1
19	29.2	S5	23	O2	28.4	S4	11	O1
20	29.8	S5	41	O3	28.3	S4	48	O3
21	29.6	S5	34	O2	26.4	S3	75	O4
22	29.3	S5	85	O5	25.7	S3	38	O2
23	28	S4	66	O4	25	S3	98	O5
24	28.3	S4	45	O3	27	S4	93	O5
25	28.6	S4	65	O4	25.8	S3	84	O5
26	28.7	S4	43	O3	26.4	S3	55	O3
27	29	S5	45	O3	25.6	S3	95	O5
28	27.7	S4	51	O3	24.2	S2	98	O5
29	26.2	S3	85	O5	23.3	S2	100	O5
30	26	S3	85	O5	23.5	S2	94	O5
31	27.7	S4	74	O4				

APPENDIX E

TEMPERATURE FORECAST MATLAB PROGRAM

```
function gafuzzy
clc;
clear all;
close all;
%POSTGRADUATE DEPARTMENT OF CONTROL ENGINEERING AHMADU BELLO UNIVERSITY
%ZARIA...
% Author: Bashir Baba Abba
clc
tic
format short
Data=xlsread('Data_set.xlsx');
a_11=numel(Data(357,3));
S1=numel([Data(123,3) Data(232,3) Data(357,3),Data(358,3)]);
a11=a_11/S1;
a_12=numel(Data(358,3));
a12=a_12/S1;
a_13=numel([Data(123,3) Data(232,3)]);
a13=a_13/S1;
a_14=0;
a14=a_14/S1;
a_15=0;
a15=a_15/S1;
a_16=0;
a16=a_16/S1;
a_21=numel([Data(122,3) Data(231,3) Data(356,3)]);
S2=numel([Data(5,3) Data(6,3) Data(7,3) Data(9,3) Data(122,3) Data(141,3)...
    Data(227,3) Data(228,3) Data(230,3) Data(231,3) Data(238,3) Data(239,3)...
     Data(240,3) Data(241,3) Data(242,3) Data(243,3) Data(244,3) Data(253,3)...
     Data(255,3) Data(356,3) Data(359,3)]);
a21=a_21/S2;
a_22=numel([Data(5,3) Data(6,3) Data(227,3) Data(230,3) Data(238,3) Data(239,3)...
   Data(240,3) Data(241,3) Data(242,3) Data(243,3) Data(244,3)]);
a22=a_22/S2;
a_23=numel([Data(7,3) Data(9,3) Data(141,3) Data(228,3) Data(253,3)...
    Data(255,3) Data(359,3)]);
 a23=a_23/S2;
 a_24=numel(Data(244,3));
a24=a_24/S2;
```

```
a_25=0;
a25=a_25/S2;
a_26=0;
a26=a_26/S2;
S3=numel([Data(2,3) Data(3,3) Data(4,3) Data(8,3) Data(10,3) Data(11,3)...
   Data(18,3) Data(93,3) Data(94,3) Data(95,3) Data(96,3) Data(97,3)...
   Data(98,3) Data(99,3) Data(100,3) Data(101,3) Data(102,3) Data(115,3)...
   Data(116,3) Data(117,3) Data(118,3) Data(124,3) Data(125,3) Data(140,3)...
   Data(142,3) Data(191,3) Data(191,3) Data(204,3) Data(226,3) Data(229,3)...
   Data(233,3) Data(234,3) Data(236,3) Data(237,3) Data(249,3) Data(250,3)...
   Data(254,3) Data(256,3) Data(257,3) Data(258,3) Data(259,3) Data(272,3)...
   Data(285,3) Data(286,3) Data(298,3) Data(5,3) Data(307,3) Data(308,3)...
   Data(329,3) Data(330,3) Data(352,3) Data(353,3) Data(354,3) Data(355,3)...
   Data(360,3) Data(361,3) Data(362,3) Data(363,3) Data(365,3)]);
a_31=0;
a31=a_31/S3;
a_32=numel([Data(4,3) Data(8,3) Data(140,3) Data(226,3) Data(229,3) Data(237,3)...
    Data(254,3) Data(355,3)]);
 a32=a_32/S3;
 a_33=numel([Data(2,3), Data(3,3), Data(10,3) Data(93,3) Data(94,3) Data(95,3) Data(96,3) ...
    Data(97,3) Data(98,3) Data(99,3) Data(100,3) Data(101,3) Data(115,3) Data(116,3) ...
    Data(117,3) Data(124,3) Data(233,3) Data(236,3) Data(249,3) Data(256,3) ...
    Data(257,3) Data(258,3) Data(285,3) Data(307,3) Data(329,3) Data(352,3) ...
    Data(353,3) Data(354,3) Data(360,3) Data(361,3) Data(362,3)]);
 a33=a_33/S3;
 a_34=numel([Data(11,3) Data(18,3) Data(102,3) Data(118,3) Data(125,3) Data(142,3) ...
    Data(354,3) Data(199,3) Data(204,3) Data(234,3) Data(250,3) Data(259,3) ...
    Data(272,3) Data(286,3) Data(298,3) Data(308,3) Data(330,3) Data(363,3) ...
    Data(365,3)]);
 a34=a_34/S3;
 a_35=numel(Data(191,3));
 a35=a_35/S3;
 a_36=0/S3;
 a36=a_36/S3;
 S4=numel([Data(1,3) Data(12,3) Data(13,3) Data(16,3) Data(17,3) Data(19,3) ...
    Data(20,3) Data(24,3) Data(25,3) Data(28,3) Data(29,3) Data(41,3) Data(42,3)...
    Data(51,3) Data(52,3) Data(53,3) Data(56,3) Data(57,3) Data(58,3) ...
    Data(59,3) Data(61,3) Data(70,3) Data(71,3) Data(72,3) Data(73,3) Data(75,3) ...
    Data(76,3) Data(77,3) Data(78,3) Data(81,3) Data(89,3) Data(90,3) ...
    Data(91,3) Data(92,3) Data(103,3) Data(105,3) Data(13,3) Data(114,3) ...
    Data(119,3) Data(120,3) Data(121,3) Data(126,3) Data(127,3) Data(129,3) ...
```

```
    Data(130,3) Data(133,3) Data(134,3) Data(135,3) Data(137,3) Data(138,3) ...
    Data(139,3) Data(143,3) Data(148,3) Data(154,3) Data(157,3) Data(162,3) ...
    Data(163,3) Data(173,3) Data(177,3) Data(178,3) Data(187,3) Data(188,3) ...
    Data(190,3) Data(198,3) Data(200,3) Data(201,3) Data(202,3) Data(203,3) ...
    Data(205,3) Data(206,3) Data(207,3) Data(214,3) Data(215,3) Data(216,3) ...
    Data(219,3) Data(221,3) Data(222,3) Data(225,3) Data(235,3) Data(245,3) ...
    Data(246,3) Data(247,3) Data(248,3) Data(251,3) Data(252,3) Data(260,3) ...
    Data(262,3) Data(273,3) Data(274,3) Data(275,3) Data(277,3) Data(278,3) ...
    Data(283,3) Data(284,3) Data(287,3) Data(288,3) Data(289,3) Data(297,3) ...
    Data(299,3) Data(304,3) Data(305,3) Data(306,3) Data(309,3) Data(310,3) ...
    Data(311,3) Data(312,3) Data(317,3) Data(318,3) Data(319,3) Data(328,3) ...
    Data(331,3) Data(348,3) Data(349,3) Data(350,3) Data(351,3) Data(364,3) ...
    Data(366,3)]);
a_41=0;a41=a_41/S4;
a_42=numel([Data(121,3) Data(252,3)]);
a42=a_42/S4;
a_43=numel([Data(1,3) Data(17,3) Data(92,3) Data(114,3) Data(139,3) Data(190,3) ...
    Data(198,3) Data(203,3) Data(225,3) Data(253,3) Data(248,3) Data(284,3) ...
    Data(297,3) Data(306,3) Data(328,3) Data(351,3) Data(364,3)]);
a43=a_43/S4;
a_44=numel([Data(12,3) Data(17,3) Data(20,3) Data(25,3) Data(29,3) Data(42,3) ...
    Data(52:53,3)' Data(57:59,3)' Data(71:73,3)' Data(76:78,3)' Data(90:92,3)' ...
    Data(120:121,3)' Data(127,3) Data(130,3) Data(134:135,3)' Data(138:139,3)' ...
    Data(157,3) Data(163,3) Data(177,3) Data(187,3) Data(201:203,3)' ...
    Data(206:207,3)' Data(215:216,3)' Data(221,3) Data(246:248,3)' Data(251,3)' ...
    Data(274:275,3)' Data(277,3) Data(284,3) Data(288:289,3)' Data(305:306,3)' ...
    Data(310:312,3)' Data(318:319,3)' Data(349:351,3)']);
a44=a_44/S4;
a_45=numel([Data(13,3) Data(20,3) Data(25,3) Data(29,3) Data(42,3) Data(53,3) ...
    Data(59,3) Data(61,3) Data(73,3) Data(78,3) Data(81,3) Data(103,3) ...
    Data(105,3) Data(127,3) Data(130,3) Data(135,3) Data(143,3) Data(148,3) ...
    Data(157,3) Data(163,3) Data(173,3) Data(178,3) Data(188,3) Data(207,3) ...
    Data(217,3) Data(219,3) Data(222,3) Data(260,3) Data(262,3) Data(275,3) ...
    Data(278,3) Data(289,3) Data(299,3) Data(312,3) Data(319,3) Data(331,3)]);
a45=a_45/S4;
a_46=numel(0);
a46=a_46/S4;
S5=numel([Data(14:15,3)' Data(21:23,3)' Data(26,3) Data(30:40,3)' Data(43:46,3)' ...
   Data(50,3) Data(54:55,3)' Data(60,3) Data(62:69,3)' Data(74,3) Data(79:80,3)' ...
   Data(82:88,3)' Data(104,3) Data(106:107,3)' Data(109:113,3)' Data(128,3) Data(131:132,3)' ...
   Data(136,3) Data(144:147,3)' Data(149:153,3)' Data(155,3) Data(158:159,3)' Data(164:172,3)' ...
```

```
    Data(174:176,3)' Data(179:184,3)' Data(186,3) Data(189,3) Data(192:195,3)' Data(197,3) ...
    Data(208:213,3)' Data(217:218,3)' Data(220,3) Data(223:224,3)' Data(261,3) Data(263:265,3)' ...
    Data(268:269,3)' Data(276,3) Data(279:282,3)' Data(290:291,3)' Data(300:303,3)' ...
    Data(313:316,3)' Data(320:327,3)' Data(332:347,3)']);
a_51=0;
a51=a_51/S5;
a_52=0;
a52=a_52/S5;
a_53=0;
a53=a_53/S5;
a_54=numel([Data(15,3) Data(23,3) Data(40,3) Data(50,3) Data(55,3) Data(60,3) ...
    Data(69,3) Data(74,3) Data(80,3) Data(88,3) Data(104,3) Data(113,3) ...
    Data(128,3) Data(132,3) Data(136,3) Data(147,3) Data(155,3) Data(172,3) ...
    Data(176,3) Data(186,3) Data(189,3) Data(197,3) Data(213,3) Data(218,3) ...
    Data(221,3) Data(225,3) Data(261,3) Data(276,3) Data(282,3) Data(316,3) ...
    Data(327,3) Data(347,3)]);
a54=a_54/S5;
a_55=numel([Data(14,3)' Data(21:22,3)' Data(31:40,3)' Data(44:46,3)' Data(54,3)' ...
    Data(63:69,3)' Data(79,3) Data(83:88,3)' Data(106,3) Data(110:113,3)' ...
    Data(131,3) Data(145:147,3)' Data(150:153,3)' Data(159,3) Data(165:172,3)' ...
    Data(175:176,3)' Data(180:184,3)' Data(193:195,3)' Data(209:213,3)' Data(217,3) ...
    Data(224,3) Data(264:265,3)' Data(269,3) Data(280:282,3)' Data(291,3) Data(301:303,3)' ...
    Data(314:316,3)' Data(321:327,3)' Data(333:347,3)']);
a55=a_55/S5;
a_56=numel([Data(26,3) Data(47,3) Data(107,3) Data(153,3) Data(159,3) Data(184,3) ...
    Data(195,3) Data(265,3) Data(269,3) Data(291,3)]);
a56=a_56/S5;
S6=numel([Data(27,3) Data(47:49,3)' Data(108,3) Data(154,3) Data(160:161,3)' ...
    Data(185,3) Data(196,3) Data(266:267,3)' Data(270:271,3)' Data(292:296,3)' ...
    ]);
a_61=0;
a61=a_61/S6;
a_62=0;
a62=a_62/S6;
a_63=numel(Data(271,3));
a63=a_63/S6;
a_64=numel([Data(27,3) Data(161,3) Data(296,3)]);
a64=a_64/S6;
a_65=numel([Data(49,3) Data(108,3) Data(154,3) Data(185,3) Data(196,3) ...
    Data(267,3)]);
a65=a_65/S5;
```

```
a_66=numel([Data(48:49,3)' Data(161,3) Data(267,3) Data(271,3) Data(293:296,3)']);
a66=a_66/S6;
%Provide the data if you a different one.
m=input('PLEASE ENTER THE VALUE OF YOUR A MATRIX OR PRESS ZERO TO CONTINUE: ');
if m==0
   A=[a11 a12 a13 a14 a15 a16;
   a21 a22 a23 a24 a25 a26;
   a31 a32 a33 a34 a35 a36;
   a41 a42 a43 a44 a45 a46;
   a51 a51 a53 a54 a55 a56;
   a61 a62 a63 a64 a65 a66];
else
A=m;
end
b_11=0;b11=b_11/S1;
b_12=0;b12=b_12/S1;
b_13=0;b13=b_13/S1;
b_14=0;b14=b_14/S1;
b_15=numel([Data(123,3), Data(232,3) Data(257:258,3)']);
b15=b_15/S1;
b_21=numel(Data(242:244,3));
b21=b_21/S2;
b_22=numel(Data(239:240,3)');
b22=b_22/S2;
b_23=numel([Data(238,3), Data(241,3)]);
b23=b_23/S3;
b_24=numel(Data(230,3));
b24=b_24/S2;
b_25=numel([Data(5:7,3)' Data(9,3) Data(122,3) Data(141,3) Data(227:228,3)' ...
   Data(231,3) Data(253,3) Data(255,3) Data(356,3) Data(359,3)]);
b25=b_25/S2;
b_31=numel([Data(96:97,3)' Data(101:102,3)']);
b31=b_31/S3;
b_32=numel([Data(98:100,3)' Data(125,3)]);
b32=b_32/S3;
b_33=numel([Data(95,3) Data(353,3) Data(353,3) Data(355,3) Data(361,3)]);
b33=b_33/S3;
b_34=numel([Data(18,3) Data(118,3) Data(124,3) Data(256:257,3)' Data(308,3) ...
   Data(352,3) Data(354,3) Data(362,3) Data(365,3)]);
b34=b_34/S3;
b_35=numel([Data(2:5,3)' Data(8,3) Data(10:11,3)' Data(93:94,3)' Data(115:117,3)' ...
```

```
    Data(140,3) Data(142,3) Data(191,3) Data(199,3) Data(204,3) Data(226,3) ...
    Data(229,3) Data(233:234,3)' Data(236:237,3)' Data(249:250,3)' Data(254,3) ...
    Data(258:259,3)' Data(272,3) Data(285:286,3)' Data(298,3) Data(307,3)' ...
    Data(329:330,3)' Data(360,3) Data(363,3)]);
b35=b_35/S3;
b_41=0;
b41=b_41/S4;
b_42=numel([Data(91,3) Data(103,3) Data(222,3)]);
b42=b_42/S4;
b_43=numel([Data(29,3) Data(90,3) Data(92,3) Data(119:120,3)' Data(201,3) ...
    Data(214,3) Data(216,3) Data(221,3) Data(309,3) Data(364,3) Data(366,3)]);
b43=b_43/S4;
b_44=numel([Data(12,3) Data(16:17,3)' Data(28,3) Data(41:42,3)' Data(52:53,3)' ...
    Data(56:57,3)' Data(59,3) Data(61,3) Data(72:73,3)' Data(77,3) Data(81,3) ...
    Data(89,3) Data(127,3) Data(130,3) Data(135,3) Data(138:139,3)' Data(156:157,3)' ...
    Data(173,3) Data(177:178,3)' Data(200,3) Data(202,3) Data(205:206,3)' ...
    Data(245:247,3)' Data(251,3) Data(262,3) Data(274:275,3)' Data(283,3) ...
    Data(287:288,3)' Data(304,3) Data(309,3) Data(317:319,3)' Data(331,3) ...
    Data(349:350,3)']);
b44=b_44/S4;
b_45=numel([Data(1,3) Data(13,3) Data(19:20,3)' Data(24:25,3)' Data(51,3) ...
    Data(58,3) Data(70:71,3)' Data(75:76,3)' Data(78,3) Data(105,3) Data(114,3) ...
    Data(126,3) Data(129,3) Data(133:134,3)' Data(137,3) Data(143,3) Data(148,3) ...
    Data(162:163,3)' Data(187,3) Data(190,3) Data(198,3) Data(203,3) Data(207,3) ...
    Data(215,3) Data(219,3) Data(225,3) Data(235,3) Data(248,3) Data(252,3) ...
    Data(260,3) Data(273,3) Data(277:278,3)' Data(284,3) Data(289,3) Data(297,3)' ...
    Data(299,3) Data(305:206,3)' Data(310:312,3)' Data(328,3) Data(348,3) Data(351,3)]);
b45=b_45/S4;
b_51=numel([Data(217,3) Data(223:224,3)' Data(333,3)]);
b51=b_51/S5;
b_52=numel([Data(107,3) Data(111:112,3)' Data(155,3) Data(108:109,3)' Data(290,3) ...
    Data(303,3) Data(314,3) Data(320,3) Data(325,3) Data(334,3) Data(341,3) ...
    Data(345,3)]);
b52=b_52/S5;
b_53=numel([Data(26,3) Data(37,3) Data(46,3) Data(60,3) Data(62,3) Data(64,3) ...
    Data(66:67,3)' Data(69,3) Data(79:80,3)' Data(86,3) Data(109:110,3)' ...
    Data(145:146,3)' Data(150,3) Data(166,3) Data(170:171,3)' Data(179,3) ...
    Data(182:184,3)' Data(193:195,3)' Data(210,3) Data(212,3) Data(218,3) ...
    Data(265,3) Data(268:263,3)' Data(281:282,3)' Data(291,3) Data(301:302,3)' ...
    Data(313,3) Data(315,3) Data(321:323,3)' Data(326:327,3)' Data(335,3) ...
    Data(337:338,3)' Data(340,3) Data(342,3) Data(344,3) Data(347,3)]);
```

```
b53=b_53/S5;
b_54=numel([Data(14:15,3)' Data(22:23,3)' Data(30,3) Data(35:36,3)' Data(38:39,3)' ...
    Data(44:45,3)' Data(50,3) Data(55,3) Data(65,3) Data(68,3) Data(74,3) ...
    Data(82:83,3)' Data(85,3) Data(87:88,3)' Data(106,3) Data(128,3) ...
    Data(131,3) Data(144,3) Data(151,3) Data(153,3) Data(158:159,3)' ...
    Data(167:169,3)' Data(172,3) Data(174:176,3)' Data(180,3) Data(186,3) ...
    Data(189,3) Data(197,3) Data(211,3) Data(213,3) Data(220,3) Data(263:264,3)' ...
    Data(279:280,3)' Data(300,3) Data(316,3) Data(324,3) Data(336,3) ...
    Data(339,3) Data(343,3) Data(346,3)]);
b54=b_54/S5;
b_55=numel([Data(21,3) Data(31:34,3)' Data(40,3) Data(43,3) Data(54,3) ...
    Data(63,3) Data(84,3) Data(104,3) Data(113,3) Data(132,3) Data(136,3) ...
    Data(147,3) Data(149,3) Data(152,3) Data(164:165,3)' Data(181,3) Data(192,3) ...
    Data(261,3) Data(276,3)]);
b55=b_55/S5;
b_61=numel([Data(160,3) Data(292,3)]);
b61=b_61/S6;
b_62=numel([Data(161,3) Data(185,3) Data(266,3)]);
b62=b_62/S6;
b_63=numel([Data(27,3) Data(108,3) Data(154,3) Data(196,3) Data(270:271,3)' ...
    Data(293,3) Data(296,3)]);
b63=b_63/S6;
b_64=numel([Data(47,3) Data(49,3) Data(267,3) Data(295,3)]);
b64=b_64/S6;
b_65=numel([Data(48,3) Data(294,3)]);
b65=b_65/S6;
P=input('PLEASE ENTER THE VALUE OF YOUR MATRIX B OR PRESS ZERO TO CONTINUE: ');
if P==0
    B=[b11 b12 b13 b14 b15;
   b21 b22 b23 b24 b25;
   b31 b32 b33 b34 b35;
   b41 b42 b43 b44 b45;
   b51 b52 b53 b54 b55;
   b61 b62 b63 b64 b65];
else
    B=P;
end
%Provide the data if you a different one.
N=numel(Data(1:end,3));
%input the value of PAI if you another one
PAI=input('PLEASE ENTER THE VALUE OF YOUR PAI OR PRESS ZERO TO CONTINUE: ');
```

```
if PAI==0
theta=[S1/N S2/N S3/N S4/N S5/N S6/N];
else
   theta=PAI;
end
disp('PRESS ANY KEY TO DISPLAY THE VALUE OF A, B & PAI EXTRACTED FROM THE DATA SET:
')
pause
A
B
PAI=theta
disp('PLEASE PRESS ANY KEY TO CONTINUE: ')
pause
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
disp('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
%% Defined the Problem parameters and transfer function
% s=tf('s'); %Make 's' a tansfer function symbol
% G=input('Please Define the Transfer Function of the Plant:= ');% %please defined the plant transfer function
% CostFunction=@(x) objmse(x1,x2); % Objective Function Handle for the GA
nVar=3;                 % Number of Variables
VarSize=[1 nVar];          % Size of Variables Matrix
VarMin=-1;                % Lower Bound of Variables
VarMax= 1;                % Upper Bound of Variables
VarRange=[VarMin VarMax];   % Variation Range of Variables
%% GA Parameters
MaxIt=100;%input('Please Enter the maximum number of generation: ');%100;% Maximum Number of
Iterations
nPop=50;%input('Please Provide the population Size of Genes:= ');%50;% Population Size
pCrossover=0.7;%input('Please provide the Percentage of Cross Over Operator:= ');%0.7;% Crossover
Percentage
nCrossover=round(pCrossover*nPop/2)*2;  % Number of Parents (Offsprings)
pMutation=0.2;%input('Please Provide the Mutant percentage:= ');%0.2;% Mutation Percentage
nMutation=round(pMutation*nPop);% Number of Mutants
%% Initialization
% Empty Structure to Hold Individuals Data
ini_ind.Position=[];
ini_ind.Cost=[];
ini_ind.Out=[];
% Create Population Matrix
pop=repmat(ini_ind,nPop,1);
```

```
% Initialize Positions
for i=1:nPop
   pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
   pop(i).Cost=objmse(pop(i).Position);
end
% Sort Population
pop=SortPop(pop);
% Store Best Solution
BestSol=pop(1);
% Vector to Hold Best Cost Values
BestCost=zeros(MaxIt,1);
%% GA Main Loop
for it=1:MaxIt
   % Crossover
   popc=repmat(ini_ind,nCrossover/2,2);
   for k=1:nCrossover/2
      i1=randi([1 nPop]);
      i2=randi([1 nPop]);
      p1=pop(i1);
      p2=pop(i2);
      [popc(k,1).Position, popc(k,2).Position]=Crossover(p1.Position,p2.Position,VarRange);
      [popc(k,1).Cost]=objmse(popc(k,1).Position);
      [popc(k,2).Cost]=objmse(popc(k,2).Position);
   end
   popc=popc(:);
   % Mutation
   popm=repmat(ini_ind,nMutation,1);
   for k=1:nMutation
      i=randi([1 nPop]);
      p=pop(i);
      popm(k).Position=Mutate(p.Position,VarRange);
      [popm(k).Cost]=objmse(popm(k).Position);
   end
   % Merge Population
   pop=[pop
       popc
       popm];
   % Sort Population
   pop=SortPop(pop);
   % Delete Extra Individuals
   pop=pop(1:nPop);
```

```
    % Update Best Solution
    BestSol=pop(1);
    % Store Best Cost
    BestCost(it)=BestSol.Cost;
    % Show Iteration Information
    %disp(['TJ_Itr ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);
    % Plot Step Response
end
BCost=mean(BestCost);

%Randomize A as a form of optimization
[a,b]=size(A);
A=A+A+(eye(size(A)).*(BCost+rand).*ones(size(A)));
for i=1:a
    for j=1:b
        A(i,j)=A(i,j)/sum(A(i,1:end));
    end
end

%Randomize B as a form of optimization
[c,d]=size(B);
B=B+(eye(size(B)).*(BCost+rand).*ones(size(B)));
for i=1:c
    for j=1:d
        B(i,j)=B(i,j)/sum(B(i,1:end));
    end
end

[Y,I] = max(A, [], 2);
  ap = zeros(size(A));
  ap(sub2ind(size(A), 1:length(I), I')) = Y;
  disp('')
  disp('')
  %disp('The Peak Values of A is: ');
  Ap=ap;
  k=0.04;%input('Please Enter Then Smoothing Factor: ');
  disp('')
  disp('')
  AL=[-k*Ap(:,1)+k*Ap(:,2) (1/2)*k*Ap(:,1)-k*Ap(:,2)+(1/2)*k*Ap(:,3) ...
     (1/2)*k*Ap(:,2)-k*Ap(:,3)+(1/2)*k*Ap(:,4) (1/2)*k*Ap(:,3)-k*Ap(:,4)+(1/2)*k*Ap(:,5) ...
```

```
    (1/2)*k*Ap(:,4)-k*Ap(:,5)+(1/2)*k*Ap(:,6) k*Ap(:,5)-k*Ap(:,6)];
  %Smooted value of A is:
  As=A+AL;
  [X H]=max(B, [], 2);
  bp=zeros(size(B));
  bp(sub2ind(size(B), 1:length(H), H'))=X;
  disp('')
  disp('')
%   disp('The Peak Values of B is: ')
  Bp=bp;
  BL=[-k*Bp(:,1)+k*Bp(:,2) (1/2)*k*Bp(:,1)-k*Bp(:,1)+(1/2)*k*Bp(:,3) ...
    (1/2)*k*Bp(:,2)-k*Bp(:,3)+(1/2)*k*Bp(:,4) +(1/2)*k*Bp(:,3)-k*Bp(:,4)+(1/2)*k*Bp(:,5) ...
    k*Bp(:,4)-k*Bp(:,5)];
  %The Smooted value of B is:
  Bs=B+BL;
%->->->
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Forecast for June
disp('Forecasted Temperature for June 1 1996 is: ')
Jun_1_1996=theta.*B(:,5)';
%which is Normalized as follows.
disp('*****************************************')
June_1_1996=Jun_1_1996./sum(Jun_1_1996)
disp('Forecasted Temperature for June 2 1996 is: ')
Jun_2_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
June_2_1996=Jun_2_1996./sum(Jun_2_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 3 1996 is: ')
Jun_3_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
June_3_1996=Jun_3_1996./sum(Jun_3_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 4 1996 is: ')
Jun_4_1996=Bs(:,2)'.*As(5,:);
%which is Normalized as follows.
June_4_1996=Jun_4_1996./sum(Jun_4_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 5 1996 is: ')
Jun_5_1996=B(:,4)'.*A(5,:);
```

```
%which is Normalized as follows.
June_5_1996=Jun_5_1996./sum(Jun_5_1996)
disp('****************************************')
disp('Forecasted Temperature for June 6 1996 is: ')
Jun_6_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
June_6_1996=Jun_6_1996./sum(Jun_6_1996)
disp('****************************************')
disp('Forecasted Temperature for June 7 1996 is: ')
Jun_7_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
June_7_1996=Jun_7_1996./sum(Jun_7_1996)
disp('****************************************')
disp('Forecasted Temperature for June 8 1996 is: ')
Jun_8_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
June_8_1996=Jun_8_1996./sum(Jun_8_1996)
disp('****************************************')
disp('Forecasted Temperature for June 9 1996 is: ')
Jun_9_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
June_9_1996=Jun_9_1996./sum(Jun_9_1996)
disp('****************************************')
disp('Forecasted Temperature for June 10 1996 is: ')
Jun_10_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
June_10_1996=Jun_10_1996./sum(Jun_10_1996)
disp('****************************************')
disp('Forecasted Temperature for June 11 1996 is: ')
Jun_11_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
June_11_1996=Jun_11_1996./sum(Jun_11_1996)
disp('****************************************')
disp('Forecasted Temperature for June 12 1996 is: ')
Jun_12_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
June_12_1996=Jun_12_1996./sum(Jun_12_1996)
disp('****************************************')
disp('Forecasted Temperature for June 13 1996 is: ')
Jun_13_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
```

```
June_13_1996=Jun_13_1996./sum(Jun_13_1996)
disp('****************************************')
disp('Forecasted Temperature for June 14 1996 is: ')
Jun_14_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
June_14_1996=Jun_14_1996./sum(Jun_14_1996)
disp('****************************************')
disp('Forecasted Temperature for June 15 1996 is: ')
Jun_15_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
June_15_1996=Jun_15_1996./sum(Jun_15_1996)
disp('****************************************')
disp('Forecasted Temperature for June 16 1996 is: ')
Jun_16_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
June_16_1996=Jun_16_1996./sum(Jun_16_1996);
disp('****************************************')
disp('Forecasted Temperature for June 17 1996 is: ')
Jun_17_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
June_17_1996=Jun_17_1996./sum(Jun_17_1996)
disp('****************************************')
disp('Forecasted Temperature for June 18 1996 is: ')
Jun_18_1996=B(:,3)'.*A(5,:);
%which is Normalized as follows.
June_18_1996=Jun_18_1996./sum(Jun_18_1996)
disp('****************************************')
disp('Forecasted Temperature for June 19 1996 is: ')
Jun_19_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
June_19_1996=Jun_19_1996./sum(Jun_19_1996)
disp('****************************************')
disp('Forecasted Temperature for June 20 1996 is: ')
Jun_20_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
June_20_1996=Jun_20_1996./sum(Jun_20_1996)
disp('****************************************')
disp('Forecasted Temperature for June 21 1996 is: ')
Jun_21_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
June_21_1996=Jun_21_1996./sum(Jun_21_1996)
```

```
disp('*****************************************')
disp('Forecasted Temperature for June 19 1996 is: ')
Jun_22_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
June_22_1996=Jun_22_1996./sum(Jun_22_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 23 1996 is: ')
Jun_23_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
June_23_1996=Jun_23_1996./sum(Jun_23_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 24 1996 is: ')
Jun_24_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
June_24_1996=Jun_24_1996./sum(Jun_24_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 25 1996 is: ')
Jun_25_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
June_25_1996=Jun_25_1996./sum(Jun_25_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 26 1996 is: ')
Jun_26_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
June_26_1996=Jun_26_1996./sum(Jun_26_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 27 1996 is: ')
Jun_27_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
June_27_1996=Jun_27_1996./sum(Jun_27_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 28 1996 is: ')
Jun_28_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
June_28_1996=Jun_28_1996./sum(Jun_28_1996)
disp('*****************************************')
disp('Forecasted Temperature for June 29 1996 is: ')
Jun_29_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
June_29_1996=Jun_29_1996./sum(Jun_29_1996)
disp('*****************************************')
```

```
disp('Forecasted Temperature for June 30 1996 is: ')
Jun_30_1996=B(:,3)'.*A(5,:);
%which is Normalized as follows.
June_30_1996=Jun_30_1996./sum(Jun_30_1996)

%->->->
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Forecast for July 1996
disp('Forecasted Temperature for July 1 1996 is: ')
July_1_1996=theta.*B(:,4)';
July_1_1996=July_1_1996./sum(July_1_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 2 1996 is: ')
July_2_1996=B(:,4)'.*A(5,:);
July_2_1996=July_2_1996./sum(July_2_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 3 1996 is: ')
July_3_1996=B(:,5)'.*A(4,:);
July_3_1996=July_3_1996./sum(July_3_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 4 1996 is: ')
July_4_1996=B(:,4)'.*A(5,:);
July_4_1996=July_4_1996./sum(July_4_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 5 1996 is: ')
July_5_1996=B(:,3)'.*A(5,:);
July_5_1996=July_5_1996./sum(July_5_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 6 1996 is: ')
July_6_1996=B(:,4)'.*A(5,:);
July_6_1996=July_6_1996./sum(July_6_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 7 1996 is: ')
July_7_1996=B(:,3)'.*A(5,:);
```

```
July_7_1996=July_7_1996./sum(July_7_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 8 1996 is: ')
July_8_1996=B(:,4)'.*A(5,:);
July_8_1996=July_8_1996./sum(July_8_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 9 1996 is: ')
July_9_1996=B(:,4)'.*A(5,:);
July_9_1996=July_9_1996./sum(July_9_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 10 1996 is: ')
July_10_1996=B(:,5)'.*A(4,:);
July_10_1996=July_10_1996./sum(July_10_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 11 1996 is: ')
July_11_1996=B(:,3)'.*A(4,:);
July_11_1996=July_11_1996./sum(July_11_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 12 1996 is: ')
July_12_1996=B(:,3)'.*A(4,:);
July_12_1996=July_12_1996./sum(July_12_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 13 1996 is: ')
July_13_1996=B(:,4)'.*A(5,:);
July_13_1996=July_13_1996./sum(July_13_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 14 1996 is: ')
July_14_1996=B(:,3)'.*A(4,:);
July_14_1996=July_14_1996./sum(July_14_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 15 1996 is: ')
July_15_1996=B(:,2)'.*A(5,:);
July_15_1996=July_15_1996./sum(July_15_1996)
```

```
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 16 1996 is: ')
July_16_1996=B(:,2)'.*A(4,:);
July_16_1996=July_16_1996./sum(July_16_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 17 1996 is: ')
July_17_1996=B(:,4)'.*A(4,:);
July_17_1996=July_17_1996./sum(July_17_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 18 1996 is: ')
July_18_1996=B(:,3)'.*A(4,:);
July_18_1996=July_18_1996./sum(July_18_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 19 1996 is: ')
July_19_1996=B(:,3)'.*A(5,:);
July_19_1996=July_19_1996./sum(July_19_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 20 1996 is: ')
July_20_1996=B(:,2)'.*A(5,:);
July_20_1996=July_20_1996./sum(July_20_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 21 1996 is: ')
July_21_1996=B(:,2)'.*A(6,:);
July_21_1996=July_21_1996./sum(July_21_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 22 1996 is: ')
July_22_1996=B(:,4)'.*A(6,:);
July_22_1996=July_22_1996./sum(July_22_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 23 1996 is: ')
July_23_1996=B(:,3)'.*A(6,:);
July_23_1996=July_23_1996./sum(July_23_1996)
%which is Normalized as follows.
```

```
disp('****************************************')
disp('Forecasted Temperature for July 24 1996 is: ')
July_24_1996=B(:,2)'.*A(6,:);
July_24_1996=July_24_1996./sum(July_24_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 25 1996 is: ')
July_25_1996=B(:,5)'.*A(6,:);
July_25_1996=July_25_1996./sum(July_25_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 26 1996 is: ')
July_26_1996=B(:,5)'.*A(4,:);
July_26_1996=July_26_1996./sum(July_26_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 27 1996 is: ')
July_27_1996=B(:,5)'.*A(4,:);
July_27_1996=July_27_1996./sum(July_27_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 28 1996 is: ')
July_28_1996=B(:,5)'.*A(4,:);
July_28_1996=July_28_1996./sum(July_28_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 29 1996 is: ')
July_29_1996=B(:,4)'.*A(4,:);
July_29_1996=July_29_1996./sum(July_29_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 30 1996 is: ')
July_30_1996=B(:,5)'.*A(5,:);
July_30_1996=July_30_1996./sum(July_30_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for July 31 1996 is: ')
July_31_1996=B(:,5)'.*A(4,:);
July_31_1996=July_31_1996./sum(July_31_1996)
%which is Normalized as follows.
disp('****************************************')
```

```
%->->->
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Forecast for August 1996
disp('Forecasted Temperature for August 1 1996 is: ')
Aug_1_1996=theta.*B(:,5)';
August_1_1996=Aug_1_1996./sum(Aug_1_1996)
%which is Normalized as follows.
disp('****************************************')
disp('Forecasted Temperature for August 2 1996 is: ')
Aug_2_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_2_1996=Aug_2_1996./sum(Aug_2_1996)
disp('****************************************')
disp('Forecasted Temperature for August 3 1996 is: ')
Aug_3_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_3_1996=Aug_3_1996./sum(Aug_3_1996)
disp('****************************************')
disp('Forecasted Temperature for August 4 1996 is: ')
Aug_4_1996=B(:,2)'.*A(4,:);
%which is Normalized as follows.
August_4_1996=Aug_4_1996./sum(Aug_4_1996)
disp('****************************************')
disp('Forecasted Temperature for August 5 1996 is: ')
Aug_5_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
August_5_1996=Aug_5_1996./sum(Aug_5_1996)
disp('****************************************')
disp('Forecasted Temperature for August 6 1996 is: ')
Aug_6_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
August_6_1996=Aug_6_1996./sum(Aug_6_1996)
disp('****************************************')
disp('Forecasted Temperature for August 7 1996 is: ')
Aug_7_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_7_1996=Aug_7_1996./sum(Aug_7_1996)
disp('****************************************')
disp('Forecasted Temperature for August 8 1996 is: ')
Aug_8_1996=B(:,4)'.*A(5,:);
```

```
%which is Normalized as follows.
August_8_1996=Aug_8_1996./sum(Aug_8_1996)
disp('****************************************')
disp('Forecasted Temperature for August 9 1996 is: ')
Aug_9_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_9_1996=Aug_9_1996./sum(Aug_9_1996)
disp('****************************************')
disp('Forecasted Temperature for August 10 1996 is: ')
Aug_10_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_10_1996=Aug_10_1996./sum(Aug_10_1996)
disp('****************************************')
disp('Forecasted Temperature for August 11 1996 is: ')
Aug_11_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_11_1996=Aug_11_1996./sum(Aug_11_1996)
disp('****************************************')
disp('Forecasted Temperature for August 12 1996 is: ')
Aug_12_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_12_1996=Aug_12_1996./sum(Aug_12_1996)
disp('****************************************')
disp('Forecasted Temperature for August 13 1996 is: ')
Aug_13_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
August_13_1996=Aug_13_1996./sum(Aug_13_1996)
disp('****************************************')
disp('Forecasted Temperature for August 14 1996 is: ')
Aug_14_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
August_14_1996=Aug_14_1996./sum(Aug_14_1996)
disp('****************************************')
disp('Forecasted Temperature for August 15 1996 is: ')
Aug_15_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
August_15_1996=Aug_15_1996./sum(Aug_15_1996)
disp('****************************************')
disp('Forecasted Temperature for August 16 1996 is: ')
Aug_16_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
```

```
August_16_1996=Aug_16_1996./sum(Aug_16_1996)
disp('****************************************')
disp('Forecasted Temperature for August 17 1996 is: ')
Aug_17_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
August_17_1996=Aug_17_1996./sum(Aug_17_1996)
disp('****************************************')
disp('Forecasted Temperature for August 18 1996 is: ')
Aug_18_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_18_1996=Aug_18_1996./sum(Aug_18_1996)
disp('****************************************')
disp('Forecasted Temperature for August 19 1996 is: ')
Aug_19_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
August_19_1996=Aug_19_1996./sum(Aug_19_1996)
disp('****************************************')
disp('Forecasted Temperature for August 20 1996 is: ')
Aug_20_1996=B(:,3)'.*A(5,:);
%which is Normalized as follows.
August_20_1996=Aug_20_1996./sum(Aug_20_1996)
disp('****************************************')
disp('Forecasted Temperature for August 21 1996 is: ')
Aug_21_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
August_21_1996=Aug_21_1996./sum(Aug_21_1996)
disp('****************************************')
disp('Forecasted Temperature for August 22 1996 is: ')
Aug_22_1996=B(:,5)'.*A(5,:);
%which is Normalized as follows.
August_22_1996=Aug_22_1996./sum(Aug_22_1996)
disp('****************************************')
disp('Forecasted Temperature for August 23 1996 is: ')
Aug_23_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
August_23_1996=Aug_23_1996./sum(Aug_23_1996)
disp('****************************************')
disp('Forecasted Temperature for August 24 1996 is: ')
Aug_24_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_24_1996=Aug_24_1996./sum(Aug_24_1996)
```

```
disp('*****************************************')
disp('Forecasted Temperature for August 25 1996 is: ')
Aug_25_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
August_25_1996=Aug_25_1996./sum(Aug_25_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 26 1996 is: ')
Aug_26_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_26_1996=Aug_26_1996./sum(Aug_26_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 27 1996 is: ')
Aug_27_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
August_27_1996=Aug_27_1996./sum(Aug_27_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 28 1996 is: ')
Aug_28_1996=B(:,3)'.*A(5,:);
%which is Normalized as follows.
August_28_1996=Aug_28_1996./sum(Aug_28_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 29 1996 is: ')
Aug_29_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
August_29_1996=Aug_29_1996./sum(Aug_29_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 30 1996 is: ')
Aug_30_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
August_30_1996=Aug_30_1996./sum(Aug_30_1996)
disp('*****************************************')
disp('Forecasted Temperature for August 31 1996 is: ')
Aug_31_1996=B(:,4)'.*A(3,:);
%which is Normalized as follows.
August_31_1996=Aug_31_1996./sum(Aug_31_1996)
disp('*****************************************')
%->->->
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% forecas for September
disp('Forecasted Temperature for Setember 1 1996 is: ')
```

```
Sept_1_1996=theta.*B(:,5)';
September_1_1996=Sept_1_1996./sum(Sept_1_1996)
%which is Normalized as follows.
disp('*****************************************')
disp('Forecasted Temperature for September 2 1996 is: ')
Sept_2_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
September_2_1996=Sept_2_1996./sum(Sept_2_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 3 1996 is: ')
Sept_3_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
September_3_1996=Sept_3_1996./sum(Sept_3_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 4 1996 is: ')
Sept_4_1996=B(:,4)'.*A(3,:);
%which is Normalized as follows.
September_4_1996=Sept_4_1996./sum(Sept_4_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 5 1996 is: ')
Sept_5_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
September_5_1996=Sept_5_1996./sum(Sept_5_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 6 1996 is: ')
Sept_6_1996=B(:,4)'.*A(3,:);
%which is Normalized as follows.
September_6_1996=Sept_6_1996./sum(Sept_6_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 7 1996 is: ')
Sept_7_1996=B(:,2)'.*A(4,:);
%which is Normalized as follows.
September_7_1996=Sept_7_1996./sum(Sept_7_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 8 1996 is: ')
Sept_8_1996=B(:,4)'.*A(5,:);
%which is Normalized as follows.
September_8_1996=Sept_8_1996./sum(Sept_8_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 9 1996 is: ')
Sept_9_1996=B(:,2)'.*A(5,:);
```

```
%which is Normalized as follows.
September_9_1996=Sept_9_1996./sum(Sept_9_1996)
disp('****************************************')
disp('Forecasted Temperature for September 10 1996 is: ')
Sept_10_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_10_1996=Sept_10_1996./sum(Sept_10_1996)
disp('****************************************')
disp('Forecasted Temperature for September 11 1996 is: ')
Sept_11_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_11_1996=Sept_11_1996./sum(Sept_11_1996)
disp('****************************************')
disp('Forecasted Temperature for September 12 1996 is: ')
Sept_12_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_12_1996=Sept_12_1996./sum(Sept_12_1996)
disp('****************************************')
disp('Forecasted Temperature for September 13 1996 is: ')
Sept_13_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_13_1996=Sept_13_1996./sum(Sept_13_1996)
disp('****************************************')
disp('Forecasted Temperature for September 14 1996 is: ')
Sept_14_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_14_1996=Sept_14_1996./sum(Sept_14_1996)
disp('****************************************')
disp('Forecasted Temperature for September 15 1996 is: ')
Sept_15_1996=B(:,2)'.*A(5,:);
%which is Normalized as follows.
September_15_1996=Sept_15_1996./sum(Sept_15_1996)
disp('****************************************')
disp('Forecasted Temperature for September 16 1996 is: ')
Sept_16_1996=B(:,3)'.*A(5,:);
%which is Normalized as follows.
September_16_1996=Sept_16_1996./sum(Sept_16_1996)
disp('****************************************')
disp('Forecasted Temperature for September 17 1996 is: ')
Sept_17_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
```

```
September_17_1996=Sept_17_1996./sum(Sept_17_1996)
disp('****************************************')
disp('Forecasted Temperature for September 18 1996 is: ')
Sept_18_1996=B(:,1)'.*A(4,:);
%which is Normalized as follows.
September_18_1996=Sept_18_1996./sum(Sept_18_1996)
disp('****************************************')
disp('Forecasted Temperature for September 19 1996 is: ')
Sept_19_1996=B(:,1)'.*A(4,:);
%which is Normalized as follows.
September_19_1996=Sept_19_1996./sum(Sept_19_1996)
disp('****************************************')
disp('Forecasted Temperature for September 20 1996 is: ')
Sept_20_1996=B(:,3)'.*A(4,:);
%which is Normalized as follows.
September_20_1996=Sept_20_1996./sum(Sept_20_1996)
disp('****************************************')
disp('Forecasted Temperature for September 21 1996 is: ')
Sept_21_1996=B(:,4)'.*A(4,:);
%which is Normalized as follows.
September_21_1996=Sept_21_1996./sum(Sept_21_1996)
disp('****************************************')
disp('Forecasted Temperature for September 22 1996 is: ')
Sept_22_1996=B(:,2)'.*A(3,:);
%which is Normalized as follows.
September_22_1996=Sept_22_1996./sum(Sept_22_1996)
disp('****************************************')
disp('Forecasted Temperature for September 23 1996 is: ')
Sept_23_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
September_23_1996=Sept_23_1996./sum(Sept_23_1996)
disp('****************************************')
disp('Forecasted Temperature for September 24 1996 is: ')
Sept_24_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
September_24_1996=Sept_24_1996./sum(Sept_24_1996)
disp('****************************************')
disp('Forecasted Temperature for September 25 1996 is: ')
Sept_25_1996=B(:,5)'.*A(4,:);
%which is Normalized as follows.
September_25_1996=Sept_25_1996./sum(Sept_25_1996)
```

```
disp('*****************************************')
disp('Forecasted Temperature for September 26 1996 is: ')
Sept_26_1996=B(:,3)'.*A(3,:);
%which is Normalized as follows.
September_26_1996=Sept_26_1996./sum(Sept_26_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 27 1996 is: ')
Sept_27_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
September_27_1996=Sept_27_1996./sum(Sept_27_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 28 1996 is: ')
Sept_28_1996=B(:,5)'.*A(3,:);
%which is Normalized as follows.
September_28_1996=Sept_28_1996./sum(Sept_28_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 29 1996 is: ')
Sept_29_1996=B(:,5)'.*A(2,:);
%which is Normalized as follows.
September_29_1996=Sept_29_1996./sum(Sept_29_1996)
disp('*****************************************')
disp('Forecasted Temperature for September 30 1996 is: ')
Sept_30_1996=B(:,5)'.*A(2,:);
%which is Normalized as follows.
September_30_1996=Sept_30_1996./sum(Sept_30_1996)
disp('*****************************************')
n=input('PLEASE ENTER THE TOTAL NUMBER OF MONTE CARLO ITERATION: ');
V=input('Please Enter The Day of the Month you want: Note June_1_1996 is the format: ');%June_30_1996;
U=0;
for q=1:n
   U=U+V;
end
Vec=round(U)
u=[22.6667 24 26 28 30 31.3333];
%uu=[];
Z=sum(Vec.*u)/n
```

APPENDIX F1

FORECASTED VALUES OF THE DAILY AVERAGE TEMPERATURE FOR JUNE AND JULY, 1996 IN TAIPEI, TAIWAN.

Day	June		July	
	Actual Temperature (^oC)	Forecasted Temperature (^oC)	Actual Temperature (^oC)	Forecasted Temperature (^oC)
1	26.1	**28.04135**	29.9	**28.642**
2	27.6	**26.556**	28.4	**29.2133**
3	29	**26.556**	29.2	**28.362**
4	30.5	**30.35333**	29.4	**29.3733**
5	30	**29.50732**	29.9	**29.8667**
6	29.5	**29.68535**	29.6	**29.58**
7	29.7	**29.738**	30.1	**30.1**
8	29.4	**29.78802**	29.3	**29.3133**
9	28.8	**29.46065**	28.1	**29.3133**
10	29.4	**28.65267**	28.9	**28.74**
11	29.3	**29.61932**	28.4	**28.4**
12	28.5	**29.56931**	29.6	**29.24**
13	28.7	**28.366**	27.8	**29.2133**
14	27.5	**28.19265**	29.1	**29.06**
15	29.5	**28.38931**	27.7	**29.98**
16	28.8	**29.52397**	28.1	**28.2667**
17	29	**28.49797**	28.7	**28.5533**
18	30.3	**30.13067**	29.9	**29.48**
19	30.2	**30.21066**	30.8	**30.16**
20	30.9	**30.22668**	31.6	**30.5933**
21	30.8	**29.47465**	31.4	**31.4**
22	28.7	**29.82201**	31.3	**30.42**
23	27.8	**28.464**	31.3	**31.1333**
24	27.4	**28.524**	31.3	**31.1866**
25	27.7	**28.796**	28.9	**29.28**
26	27.1	**28.30598**	28	**28.14**
27	28.4	**28.48397**	28.6	**28.58**
28	27.8	**28.70467**	28	**28.02**
29	29	**28.68**	29.3	**28.7933**
30	30.2	**30.15066**	27.9	**29.6533**
31			26.9	**28.08**

APPENDIX F2

FORECASTED VALUES OF THE DAILY AVERAGE TEMPERATURE FOR AUGUST AND SEPTEMBER, 1996 IN TAIPEI, TAIWAN.

Day	August		September	
	Actual Temperature (°C)	Forecasted Temperature (°C)	Actual Temperature (°C)	Forecasted Temperature (°C)
1	27.1	**28.15069**	27.3867	**27.94534**
2	28.9	**28.38365**	27.92	**28.434**
3	28.9	**28.615**	26.28	**26.5**
4	29.3	**28.41201**	27.44	**27.586**
5	28.8	**29.76868**	27.98	**28.50897**
6	28.7	**28.31**	27.78	**27.526**
7	29	**28.34731**	28.9933	**28.61132**
8	28.2	**29.46398**	29.1333	**29.53932**
9	27	**28.37931**	30.28	**30.32599**
10	28.3	**28.62067**	30	**30.25332**
11	28.9	**28.38065**	30.16	**30.29333**
12	28.1	**28.33665**	30.3133	**30.224**
13	29.9	**28.504**	30.1867	**30.22934**
14	27.6	**29.47665**	30.2933	**30.23666**
15	26.8	**28.354**	30.0133	**30.25067**
16	27.6	**26.374**	29.8667	**30.20399**
17	27.9	**28.218**	28.54	**28.576**
18	29	**28.152**	28.0733	**28.28066**
19	29.2	**30.26801**	28.2333	**28.32065**
20	29.8	**30.20467**	29.24	**28.636**
21	29.6	**30.23666**	27.96	**28.42331**
22	29.3	**29.72602**	25.26	**25.238**
23	28	**29.41998**	26.14	**26.51**
24	28.3	**28.642**	26.8	**26.398**
25	28.6	**28.50664**	27.86	**28.34889**
26	28.7	**28.516**	26.36	**26.40222**
27	29	**28.69**	26.2	**26.52**
28	27.7	**30.21333**	26.4	**26.522**
29	26.2	**28.314**	24.44	**24.6593**
30	26	**26.514**	24.4267	**24.68866**
31	27.7	**27.628**		

APPENDIX G

ABU INTERNET TRAFFIC TRAINING DATA SET (ABU Zaria Data Center PAESSLER PRTG Network Monitor)

Time	29th February, 2016		1st March, 2016		2nd March, 2016	
	Traffic Total Volume (MB)	Traffic In Volume (MB)	Traffic Total Volume (MB)	Traffic In Volume (MB)	Traffic Total Volume (MB)	Traffic In Volume (MB)
9:00	7308	6757	7923	7344	6055	5533
9:05	7629	6992	8095	7452	6107	5565
9:10	7691	6982	7996	7329	6056	5512
9:15	7681	6979	7933	7331	6031	5495
9:20	7797	7044	7812	7202	6102	5538
9:25	7644	6876	6850	6251	6095	5516
9:30	7668	6868	5903	5334	6097	5490
9:35	7680	6845	5856	5250	6150	5478
9:40	5525	4993	5820	5208	6092	5440
9:45	7646	6819	5897	5318	6109	5410
9:50	7650	6860	6000	5361	6151	5456
9:55	7778	6978	6009	5370	6135	5461
10:00	7731	6910	5878	5214	6153	5463
10:05	7840	6994	5991	5323	6127	5433
10:10	7711	6914	5963	5285	6008	5325
10:15	7640	6856	6074	5353	6052	5362
10:20	7738	6944	6033	5270	6007	5316
10:25	7669	6922	6152	5379	5973	5310
10:30	7671	6930	6199	5385	6084	5357
10:35	7839	7021	6117	5338	6163	5400
10:40	7777	6961	6204	5369	6089	5348
10:45	7907	7073	6106	5294	6240	5351
10:50	7790	6942	6066	5294	6138	5295
10:55	7748	6926	6002	5273	6193	5351
11:00	7830	6966	6058	5337	6126	5335
11:05	7778	6904	6810	6014	6181	5339
11:10	7691	6869	6063	5277	6151	5338
11:15	7672	6842	6059	5298	6151	5329
11:20	7735	6881	5863	5172	6130	5321
11:25	7707	6846	5916	5209	6210	5372
11:30	6485	5710	5851	5140	6153	5312
11:35	6011	5227	5936	5230	6211	5357
11:40	3731	3246	6007	5272	6403	5491
11:45	6063	5292	5984	5256	8000	6872
11:50	3463	3016	5898	5163	7824	6876
11:55	194	130	5914	5220	7559	6654
12:00	138	70	6105	5297	6215	5294

12:05	552	447	6075	5271	6154	5231
12:10	400	284	6076	5276	6184	5248
12:15	151	67	6016	5240	6171	5272
12:20	169	98	6016	5237	3621	3358

Time	7th March, 2016		8th March, 2016		9th March, 2016	
9:00	6117	5653	7766	7132	6028	5424
9:05	6963	6382	7925	7314	7491	6817
9:10	7268	6702	7947	7312	8015	7306
9:15	7397	6836	7824	7173	7903	7220
9:20	7847	7239	7741	7072	7997	7236
9:25	5322	4910	7836	7124	8074	7275
9:30	5136	4658	7843	7123	8033	7223
9:35	5673	5183	7836	7124	8022	7241
9:40	5673	5135	7822	7033	8079	7254
9:45	1632	1454	5960	5282	7990	7249
9:50	3908	3519	5933	5222	7984	7216
9:55	5622	5116	4171	3671	4363	4078
10:00	5640	5096	5846	5215	4565	4310
10:05	5615	5051	5702	5016	4956	4683
10:10	5685	5171	1463	1278	5625	5290
10:15	5638	5113	73	33	5657	5318
10:20	5563	5024	105	54	5835	5493
10:25	5543	5006	4653	3442	4736	4447
10:30	5157	4606	5862	4156	3740	3454
10:35	5767	5123	4381	3833	4600	4192
10:40	5684	5020	7253	5423	5856	5313
10:45	5745	5058	7076	5444	6052	5415
10:50	5834	5109	6926	5407	6006	5385
10:55	5809	5111	3557	3016	6042	5376
11:00	5807	5113	4601	4252	5949	5283
11:05	5918	5164	5802	5353	5963	5299
11:10	6674	5868	5810	5359	5949	5290
11:15	8102	7228	4425	4090	6006	5331
11:20	5980	5219	2186	1994	5990	5318
11:25	7515	6672	4109	3750	6011	5329
11:30	7051	6242	6806	6218	6099	5333
11:35	6319	5507	7911	7232	6200	5394
11:40	7609	6766	7947	7266	6181	5360
11:45	6004	5261	7493	6816	6197	5404
11:50	6077	5252	5991	5327	6145	5378
11:55	6090	5233	5952	5249	6185	5395
12:00	6024	5197	5877	5226	5999	5275
12:05	6081	5258	5967	5263	6087	5336
12:10	8206	7281	6719	5984	6218	5426
12:15	7765	6849	6126	5381	6233	5359
12:20	6085	5232	6176	5381	5835	5493

Time	14th March, 2016		15th March, 2016		16th March, 2016	
9:00	4569	4247	6052	5561	8069	7434
9:05	4271	3955	6045	5530	8059	7422
9:10	4409	4081	6110	5542	8036	7371

9:15	4674	4299	6096	5492	8124	7411
9:20	4707	4323	6054	5463	7505	6694
9:25	5145	4731	5908	5346	6013	5290
9:30	5038	4650	5876	5281	6052	5338
9:35	5310	4876	2510	2233	5996	5286
9:40	6030	5546	5834	5237	6044	5313
9:45	6160	5671	6113	5308	6073	5361
9:50	6056	5536	5918	5305	6003	5283
9:55	5796	5276	5982	5328	6016	5269
10:00	5641	5085	6063	5394	6061	5332
10:05	5628	5125	6038	5302	6041	5285
10:10	5599	5106	5970	5313	6015	5269
10:15	6410	5869	6015	5328	6052	5354
10:20	5723	5206	6068	5340	6084	5332
10:25	6341	5773	6080	5348	6134	5354
10:30	7393	6694	5952	5199	6120	5396
10:35	7392	6702	6137	5340	6190	5450
10:40	7427	6707	6187	5368	6218	5419
10:45	7425	6639	6174	5385	6041	5214
10:50	7457	6662	6066	5308	6196	5328
10:55	7873	7101	6250	5424	6237	5313
11:00	7968	7193	6125	5323	6154	5293
11:05	6319	5610	6083	5274	5999	5245
11:10	5914	5212	6270	5400	5989	5287
11:15	5975	5240	8340	7365	6073	5309
11:20	5937	5218	8773	7779	5995	5272
11:25	6000	5261	8453	7395	6017	5289
11:30	6068	5276	8395	7425	5971	5258
11:35	5993	5239	8393	7424	5991	5309
11:40	5953	5211	8438	7504	5979	5306
11:45	6066	5309	8450	7516	5935	5252
11:50	6011	5265	8382	7450	5963	5265
11:55	6144	5337	8385	7497	6013	5316
12:00	6077	5282	8423	7556	5938	5213
12:05	6017	5282	8440	7544	5819	5147
12:10	6041	5278	8386	7474	5868	5211
12:15	6076	5301	7748	6908	5873	5224
12:20	5933	5227	8367	7458	5903	5228

APPENDIX H

ABU FUZZIFIED INTERNET TRAFFIC TRAINING DATA

Time	29th February, 2016				1st March, 2016			
	Traffic Total Volume (MB)	Total Traffic Fuzzy Sets	Traffic In Volume (MB)	Traffic In Fuzzy Sets	Traffic Total Volume (MB)	Total Traffic Fuzzy Sets	Traffic In Volume (MB)	Traffic In Fuzzy Sets
9:00	7308	S5	6757	O5	7923	S6	7344	O5
9:05	7629	S6	6992	O5	8095	S6	7452	O5
9:10	7691	S6	6982	O5	7996	S6	7329	O5
9:15	7681	S6	6979	O5	7933	S6	7331	O5
9:20	7797	S6	7044	O5	7812	S6	7202	O5
9:25	7644	S6	6876	O5	6850	S5	6251	O4
9:30	7668	S6	6868	O5	5903	S4	5334	O4
9:35	7680	S6	6845	O5	5856	S4	5250	O3
9:40	5525	S4	4993	O3	5820	S4	5208	O3
9:45	7646	S6	6819	O5	5897	S4	5318	O4
9:50	7650	S6	6860	O5	6000	S4	5361	O4
9:55	7778	S6	6978	O5	6009	S4	5370	O4
10:00	7731	S6	6910	O5	5878	S4	5214	O3
10:05	7840	S6	6994	O5	5991	S4	5323	O4
10:10	7711	S6	6914	O5	5963	S4	5285	O4
10:15	7640	S6	6856	O5	6074	S4	5353	O4
10:20	7738	S6	6944	O5	6033	S4	5270	O4
10:25	7669	S6	6922	O5	6152	S4	5379	O4
10:30	7671	S6	6930	O5	6199	S4	5385	O4
10:35	7839	S6	7021	O5	6117	S4	5338	O4
10:40	7777	S6	6961	O5	6204	S4	5369	O4
10:45	7907	S6	7073	O5	6106	S4	5294	O4
10:50	7790	S6	6942	O5	6066	S4	5294	O4
10:55	7748	S6	6926	O5	6002	S4	5273	O4
11:00	7830	S6	6966	O5	6058	S4	5337	O4
11:05	7778	S6	6904	O5	6810	S5	6014	O4
11:10	7691	S6	6869	O5	6063	S4	5277	O4
11:15	7672	S6	6842	O5	6059	S4	5298	O4
11:20	7735	S6	6881	O5	5863	S4	5172	O3
11:25	7707	S6	6846	O5	5916	S4	5209	O3
11:30	6485	S5	5710	O4	5851	S4	5140	O3
11:35	6011	S4	5227	O3	5936	S4	5230	O3
11:40	3731	S4	3246	O4	6007	S4	5272	O4
11:45	6063	S4	5292	O4	5984	S4	5256	O4
11:50	3463	S4	3016	O4	5898	S4	5163	O3
11:55	194	S4	130	O4	5914	S4	5220	O3
12:00	138	S4	70	O4	6105	S4	5297	O4
12:05	552	S4	447	O4	6075	S4	5271	O4
12:10	400	S4	284	O4	6076	S4	5276	O4
12:15	151	S4	67	O4	6016	S4	5240	O3
12:20	169	S4	98	O4	6016	S4	5237	O3
Time	2nd March, 2016				7th March, 2016			

9:00	6055	S4	5533	O4	6117	S4	5653	O4
9:05	6107	S4	5565	O4	6963	S5	6382	O4
9:10	6056	S4	5512	O4	7268	S5	6702	O5
9:15	6031	S4	5495	O4	7397	S5	6836	O5
9:20	6102	S4	5538	O4	7847	S6	7239	O5
9:25	6095	S4	5516	O4	5322	S4	4910	O3
9:30	6097	S4	5490	O4	5136	S3	4658	O3
9:35	6150	S4	5478	O4	5673	S4	5183	O3
9:40	6092	S4	5440	O4	5673	S4	5135	O3
9:45	6109	S4	5410	O4	1632	S1	1454	O1
9:50	6151	S4	5456	O4	3908	S2	3519	O2
9:55	6135	S4	5461	O4	5622	S4	5116	O3
10:00	6153	S4	5463	O4	5640	S4	5096	O3
10:05	6127	S4	5433	O4	5615	S4	5051	O3
10:10	6008	S4	5325	O4	5685	S4	5171	O3
10:15	6052	S4	5362	O4	5638	S4	5113	O3
10:20	6007	S4	5316	O4	5563	S4	5024	O3
10:25	5973	S4	5310	O4	5543	S4	5006	O3
10:30	6084	S4	5357	O4	5157	S3	4606	O3
10:35	6163	S4	5400	O4	5767	S4	5123	O3
10:40	6089	S4	5348	O4	5684	S4	5020	O3
10:45	6240	S4	5351	O4	5745	S4	5058	O3
10:50	6138	S4	5295	O4	5834	S4	5109	O3
10:55	6193	S4	5351	O4	5809	S4	5111	O3
11:00	6126	S4	5335	O4	5807	S4	5113	O3
11:05	6181	S4	5339	O4	5918	S4	5164	O3
11:10	6151	S4	5338	O4	6674	S5	5868	O4
11:15	6151	S4	5329	O4	8102	S6	7228	O5
11:20	6130	S4	5321	O4	5980	S4	5219	O3
11:25	6210	S4	5372	O4	7515	S5	6672	O5
11:30	6153	S4	5312	O4	7051	S5	6242	O4
11:35	6211	S4	5357	O4	6319	S4	5507	O4
11:40	6403	S5	5491	O4	7609	S6	6766	O5
11:45	8000	S6	6872	O5	6004	S4	5261	O4
11:50	7824	S6	6876	O5	6077	S4	5252	O4
11:55	7559	S5	6654	O5	6090	S4	5233	O3
12:00	6215	S4	5294	O4	6024	S4	5197	O3
12:05	6154	S4	5231	O3	6081	S4	5258	O4
12:10	6184	S4	5248	O3	8206	S6	7281	O5
12:15	6171	S4	5272	O4	7765	S6	6849	O5
12:20	3621	S2	3358	O3	6085	S4	5232	O3

Time		8th March, 2016				9th March, 2016		
9:00	7766	S6	7132	O5	6028	S4	5424	O4
9:05	7925	S6	7314	O5	7491	S5	6817	O5
9:10	7947	S6	7312	O5	8015	S6	7306	O5
9:15	7824	S6	7173	O5	7903	S6	7220	O5
9:20	7741	S6	7072	O5	7997	S6	7236	O5
9:25	7836	S6	7124	O5	8074	S6	7275	O5
9:30	7843	S6	7123	O5	8033	S6	7223	O5
9:35	7836	S6	7124	O5	8022	S6	7241	O5
9:40	7822	S6	7033	O5	8079	S6	7254	O5

9:45	5960	S4	5282	O4	7990	S6	7249	O5
9:50	5933	S4	5222	O3	7984	S6	7216	O5
9:55	4171	S3	3671	O2	4363	S3	4078	O3
10:00	5846	S4	5215	O3	4565	S3	4310	O3
10:05	5702	S4	5016	O3	4956	S3	4683	O3
10:10	4662	S3	4312	O3	5625	S4	5290	O4
10:15	5029	S3	4610	O3	5657	S4	5318	O4
10:20	5877	S4	5378	O4	5835	S4	5493	O4
10:25	4653	S3	3442	O2	4736	S3	4447	O3
10:30	5862	S4	4156	O3	3740	S2	3454	O2
10:35	4381	S3	3833	O2	4600	S3	4192	O3
10:40	7253	S5	5423	O4	5856	S4	5313	O4
10:45	7076	S5	5444	O4	6052	S4	5415	O4
10:50	6926	S5	5407	O4	6006	S4	5385	O4
10:55	3557	S2	3016	O2	6042	S4	5376	O4
11:00	4601	S3	4252	O3	5949	S4	5283	O4
11:05	5802	S4	5353	O4	5963	S4	5299	O4
11:10	5810	S4	5359	O4	5949	S4	5290	O4
11:15	4425	S3	4090	O3	6006	S4	5331	O4
11:20	2186	S1	1994	O1	5990	S4	5318	O4
11:25	4109	S3	3750	O2	6011	S4	5329	O4
11:30	6806	S5	6218	O4	6099	S4	5333	O4
11:35	7911	S6	7232	O5	6200	S4	5394	O4
11:40	7947	S6	7266	O5	6181	S4	5360	O4
11:45	7493	S6	6816	O5	6197	S4	5404	O4
11:50	5991	S4	5327	O4	6145	S4	5378	O4
11:55	5952	S4	5249	O3	6185	S4	5395	O4
12:00	5877	S4	5226	O3	5999	S4	5275	O4
12:05	5967	S4	5263	O4	6087	S4	5336	O4
12:10	6719	S5	5984	O4	6218	S4	5426	O4
12:15	6126	S4	5381	O4	6233	S4	5359	O4
12:20	6176	S4	5381	O4	5835	S4	5493	O4

Time	14th March, 2016				15th March, 2016			
9:00	4569	S3	4247	O3	6052	S4	5561	O4
9:05	4271	S3	3955	O2	6045	S4	5530	O4
9:10	4409	S3	4081	O3	6110	S4	5542	O4
9:15	4674	S3	4299	O3	6096	S4	5492	O4
9:20	4707	S3	4323	O3	6054	S4	5463	O4
9:25	5145	S3	4731	O3	5908	S4	5346	O4
9:30	5038	S3	4650	O3	5876	S4	5281	O4
9:35	5310	S4	4876	O3	2510	S1	2233	O1
9:40	6030	S4	5546	O4	5834	S4	5237	O3
9:45	6160	S4	5671	O4	6113	S4	5308	O4
9:50	6056	S4	5536	O4	5918	S4	5305	O4
9:55	5796	S4	5276	O4	5982	S4	5328	O4
10:00	5641	S4	5085	O3	6063	S4	5394	O4
10:05	5628	S4	5125	O3	6038	S4	5302	O4
10:10	5599	S4	5106	O3	5970	S4	5313	O4
10:15	6410	S5	5869	O4	6015	S4	5328	O4
10:20	5723	S4	5206	O3	6068	S4	5340	O4
10:25	6341	S5	5773	O4	6080	S4	5348	O4

10:30	7393	S5	6694	O5	5952	S4	5199	O3
10:35	7392	S5	6702	O5	6137	S4	5340	O4
10:40	7427	S5	6707	O5	6187	S4	5368	O4
10:45	7425	S5	6639	O5	6174	S4	5385	O4
10:50	7457	S5	6662	O5	6066	S4	5308	O4
10:55	7873	S6	7101	O5	6250	S4	5424	O4
11:00	7968	S6	7193	O5	6125	S4	5323	O4
11:05	6319	S4	5610	O4	6083	S4	5274	O4
11:10	5914	S4	5212	O3	6270	S4	5400	O4
11:15	5975	S4	5240	O3	8340	S6	7365	O5
11:20	5937	S4	5218	O3	8773	S6	7779	O5
11:25	6000	S4	5261	O4	8453	S6	7395	O5
11:30	6068	S4	5276	O4	8395	S6	7425	O5
11:35	5993	S4	5239	O3	8393	S6	7424	O5
11:40	5953	S4	5211	O3	8438	S6	7504	O5
11:45	6066	S4	5309	O4	8450	S6	7516	O5
11:50	6011	S4	5265	O4	8382	S6	7450	O5
11:55	6144	S4	5337	O4	8385	S6	7497	O5
12:00	6077	S4	5282	O4	8423	S6	7556	O5
12:05	6017	S4	5282	O4	8440	S6	7544	O5
12:10	6041	S4	5278	O4	8386	S6	7474	O5
12:15	6076	S4	5301	O4	7748	S6	6908	O5
12:20	5933	S4	5227	O3	8367	S6	7458	O5

Time		16th March, 2016		
9:00	8069	S6	7434	O5
9:05	8059	S6	7422	O5
9:10	8036	S6	7371	O5
9:15	8124	S6	7411	O5
9:20	7505	S5	6694	O5
9:25	6013	S4	5290	O4
9:30	6052	S4	5338	O4
9:35	5996	S4	5286	O4
9:40	6044	S4	5313	O4
9:45	6073	S4	5361	O4
9:50	6003	S4	5283	O4
9:55	6016	S4	5269	O4
10:00	6061	S4	5332	O4
10:05	6041	S4	5285	O4
10:10	6015	S4	5269	O4
10:15	6052	S4	5354	O4
10:20	6084	S4	5332	O4
10:25	6134	S4	5354	O4
10:30	6120	S4	5396	O4
10:35	6190	S4	5450	O4
10:40	6218	S4	5419	O4
10:45	6041	S4	5214	O3
10:50	6196	S4	5328	O4

10:55	6237	S4	5313	O4
11:00	6154	S4	5293	O4
11:05	5999	S4	5245	O3
11:10	5989	S4	5287	O4
11:15	6073	S4	5309	O4
11:20	5995	S4	5272	O4
11:25	6017	S4	5289	O4
11:30	5971	S4	5258	O4
11:35	5991	S4	5309	O4
11:40	5979	S4	5306	O4
11:45	5935	S4	5252	O4
11:50	5963	S4	5265	O4
11:55	6013	S4	5316	O4
12:00	5938	S4	5213	O3
12:05	5819	S4	5147	O3
12:10	5868	S4	5211	O3
12:15	5873	S4	5224	O3
12:20	5933	S4	5227	O3

APPENDIX I

ABU INTERNET TRAFFIC TESTING DATA SET (ABU Zaria Data Center PAESSLER PRTG Network Monitor)

Time	21st March, 2016		22nd March, 2016	
	Traffic Total Volume (MB)	Traffic In Volume (MB)	Traffic Total Volume (MB)	Traffic In Volume (MB)
9:00	6071	5563	5972	5451
9:05	7629	5549	6042	5492
9:10	5993	5447	6016	5447
9:15	6020	5459	6048	5457
9:20	6070	5486	6139	5513
9:25	5969	5366	6100	5444
9:30	6040	5433	6037	5393
9:35	6059	5417	6046	5350
9:40	6004	5336	6089	5402
9:45	6021	5347	6071	5405
9:50	5990	5374	6173	5451
9:55	6027	5388	6184	5413
10:00	6162	5476	6133	5374
10:05	6156	5447	6105	5382
10:10	6163	5461	6080	5396
10:15	6146	5425	6105	5395
10:20	6140	5381	6104	5364
10:25	6125	5388	6135	5397
10:30	6228	5446	6176	5388
10:35	6154	5409	6077	5345
10:40	6164	5433	6153	5408
10:45	6138	5418	6187	5447
10:50	6153	5370	6183	5475
10:55	6218	5422	6219	5440
11:00	6242	5438	6181	5430
11:05	6269	5507	6191	5413
11:10	6248	5446	6159	5429
11:15	6216	5441	6115	5418
11:20	6199	5417	6208	5488
11:25	6198	5408	6215	5485
11:30	6202	5375	6106	5420
11:35	6186	5354	6190	5476
11:40	6243	5403	6143	5423
11:45	6188	5377	6256	5511
11:50	6167	5412	6162	5443
11:55	6197	5408	6103	5353
12:00	6235	5425	6150	5415
12:05	6247	5443	6171	5400
12:10	6112	5385	6257	5459
12:15	6270	5478	6258	5446
12:20	6292	5513	6228	5426

APPENDIX J
FUZZIFIED ABU INTERNET TRAFFIC TESTING DATA

Time	21st March, 2016				22nd March, 2016			
	Traffic Total Volume (MB)	Total Traffic Fuzzy Sets	Traffic In Volume (MB)	Traffic In Fuzzy Sets	Traffic Total Volume (MB)	Total Traffic Fuzzy Sets	Traffic In Volume (MB)	Traffic In Fuzzy Sets
9:00	6071	S5	5563	O4	5972	S5	5451	O4
9:05	6074	S5	5549	O4	6042	S5	5492	O4
9:10	5993	S5	5447	O4	6016	S5	5447	O4
9:15	6020	S5	5459	O4	6048	S5	5457	O4
9:20	6070	S5	5486	O4	6139	S5	5513	O4
9:25	5969	S5	5366	O4	6100	S5	5444	O4
9:30	6040	S5	5433	O4	6037	S5	5393	O4
9:35	6059	S5	5417	O4	6046	S5	5350	O4
9:40	6004	S5	5336	O4	6089	S5	5402	O4
9:45	6021	S5	5347	O4	6071	S5	5405	O4
9:50	5990	S5	5374	O4	6173	S5	5451	O4
9:55	6027	S5	5388	O4	6184	S5	5413	O4
10:00	6162	S5	5476	O4	6133	S5	5374	O4
10:05	6156	S5	5447	O4	6105	S5	5382	O4
10:10	6163	S5	5461	O4	6080	S5	5396	O4
10:15	6146	S5	5425	O4	6105	S5	5395	O4
10:20	6140	S5	5381	O4	6104	S5	5364	O4
10:25	6125	S5	5388	O4	6135	S5	5397	O4
10:30	6228	S5	5446	O4	6176	S5	5388	O4
10:35	6154	S5	5409	O4	6077	S5	5345	O4
10:40	6164	S5	5433	O4	6153	S5	5408	O4
10:45	6138	S5	5418	O4	6187	S5	5447	O4
10:50	6153	S5	5370	O4	6183	S5	5475	O4
10:55	6218	S5	5422	O4	6219	S5	5440	O4
11:00	6242	S5	5438	O4	6181	S5	5430	O4
11:05	6269	S5	5507	O4	6191	S5	5413	O4
11:10	6248	S5	5446	O4	6159	S5	5429	O4
11:15	6216	S5	5441	O4	6115	S5	5418	O4
11:20	6199	S5	5417	O4	6208	S5	5488	O4
11:25	6198	S5	5408	O4	6215	S5	5485	O4
11:30	6202	S5	5375	O4	6106	S5	5420	O4
11:35	6186	S5	5354	O4	6190	S5	5476	O4
11:40	6243	S5	5403	O4	6143	S5	5423	O4
11:45	6188	S5	5377	O4	6256	S5	5511	O4
11:50	6167	S5	5412	O4	6162	S5	5443	O4
11:55	6197	S5	5408	O4	6103	S5	5353	O4
12:00	6235	S5	5425	O4	6150	S5	5415	O4
12:05	6247	S5	5443	O4	6171	S5	5400	O4
12:10	6112	S5	5385	O4	6257	S5	5459	O4
12:15	6270	S5	5478	O4	6258	S5	5446	O4
12:20	6292	S5	5513	O4	6228	S5	5426	O4

APPENDIX K

INTERNET TRAFFIC FORECAST MATLAB PROGRAM

```
function ABUUntitledInternetTraffic

clc;
clear all;
close all;
%POSTGRADUATE DEPARTMENT OF CONTROL ENGINEERING AHMADU BELLO UNIVERSITY
%ZARIA...
% Author: Bashir Baba Abba
clc
tic
format short
Data=xlsread('Data_Set.xlsx');
a_11=numel(Data(357,3));
S1=numel([Data(123,3) Data(232,3) Data(357,3),Data(358,3)]);
a11=a_11/S1;
a_12=numel(Data(358,3));
a12=a_12/S1;
a_13=numel([Data(123,3) Data(232,3)]);
a13=a_13/S1;
a_14=0;
a14=a_14/S1;
a_15=0;
a15=a_15/S1;
a_16=0;
a16=a_16/S1;
a_21=numel([Data(122,3) Data(231,3) Data(356,3)]);
S2=numel([Data(5,3) Data(6,3) Data(7,3) Data(9,3) Data(122,3)
Data(141,3)...
     Data(227,3) Data(228,3) Data(230,3) Data(231,3) Data(238,3)
Data(239,3)...
      Data(240,3) Data(241,3) Data(242,3) Data(243,3) Data(244,3)
Data(253,3)...
      Data(255,3) Data(356,3) Data(359,3)]);
a21=a_21/S2;
a_22=numel([Data(5,3) Data(6,3) Data(227,3) Data(230,3) Data(238,3)
Data(239,3)...
    Data(240,3) Data(241,3) Data(242,3) Data(243,3) Data(244,3)]);
a22=a_22/S2;
a_23=numel([Data(7,3) Data(9,3) Data(141,3) Data(228,3) Data(253,3)...
     Data(255,3) Data(359,3)]);
 a23=a_23/S2;
 a_24=numel(Data(244,3));
a24=a_24/S2;
a_25=0;
a25=a_25/S2;
a_26=0;
a26=a_26/S2;
S3=numel([Data(2,3) Data(3,3) Data(4,3) Data(8,3) Data(10,3) Data(11,3)...
    Data(18,3) Data(93,3) Data(94,3) Data(95,3) Data(96,3) Data(97,3)...
    Data(98,3) Data(99,3) Data(100,3) Data(101,3) Data(102,3)
Data(115,3)...
    Data(116,3) Data(117,3) Data(118,3) Data(124,3) Data(125,3)
Data(140,3)...
    Data(142,3) Data(191,3) Data(191,3) Data(204,3) Data(226,3)
Data(229,3)...
    Data(233,3) Data(234,3) Data(236,3) Data(237,3) Data(249,3)
Data(250,3)...
```

```
    Data(254,3) Data(256,3) Data(257,3) Data(258,3) Data(259,3)
Data(272,3)...
    Data(285,3) Data(286,3) Data(298,3) Data(5,3) Data(307,3)
Data(308,3)...
    Data(329,3) Data(330,3) Data(352,3) Data(353,3) Data(354,3)
Data(355,3)...
    Data(360,3) Data(361,3) Data(362,3) Data(363,3) Data(365,3)]);
a_31=0;
a31=a_31/S3;
a_32=numel([Data(4,3) Data(8,3) Data(140,3) Data(226,3) Data(229,3)
Data(237,3)...
     Data(254,3) Data(355,3)]);
 a32=a_32/S3;
 a_33=numel([Data(2,3), Data(3,3), Data(10,3) Data(93,3) Data(94,3)
Data(95,3) Data(96,3) ...
     Data(97,3) Data(98,3) Data(99,3) Data(100,3) Data(101,3) Data(115,3)
Data(116,3) ...
     Data(117,3) Data(124,3) Data(233,3) Data(236,3) Data(249,3)
Data(256,3) ...
     Data(257,3) Data(258,3) Data(285,3) Data(307,3) Data(329,3)
Data(352,3) ...
     Data(353,3) Data(354,3) Data(360,3) Data(361,3) Data(362,3)]);
 a33=a_33/S3;
 a_34=numel([Data(11,3) Data(18,3) Data(102,3) Data(118,3) Data(125,3)
Data(142,3) ...
     Data(354,3) Data(199,3) Data(204,3) Data(234,3) Data(250,3)
Data(259,3) ...
     Data(272,3) Data(286,3) Data(298,3) Data(308,3) Data(330,3)
Data(363,3) ...
     Data(365,3)]);
 a34=a_34/S3;
 a_35=numel(Data(191,3));
 a35=a_35/S3;
 a_36=0/S3;
 a36=a_36/S3;
 S4=numel([Data(1,3) Data(12,3) Data(13,3) Data(16,3) Data(17,3) Data(19,3)
...
     Data(20,3) Data(24,3) Data(25,3) Data(28,3) Data(29,3) Data(41,3)
Data(42,3)...
     Data(51,3) Data(52,3) Data(53,3) Data(56,3) Data(57,3) Data(58,3) ...
     Data(59,3) Data(61,3) Data(70,3) Data(71,3) Data(72,3) Data(73,3)
Data(75,3) ...
     Data(76,3) Data(77,3) Data(78,3) Data(81,3) Data(89,3) Data(90,3) ...
     Data(91,3) Data(92,3) Data(103,3) Data(105,3) Data(13,3) Data(114,3)
...
     Data(119,3) Data(120,3) Data(121,3) Data(126,3) Data(127,3)
Data(129,3) ...
     Data(130,3) Data(133,3) Data(134,3) Data(135,3) Data(137,3)
Data(138,3) ...
     Data(139,3) Data(143,3) Data(148,3) Data(154,3) Data(157,3)
Data(162,3) ...
     Data(163,3) Data(173,3) Data(177,3) Data(178,3) Data(187,3)
Data(188,3) ...
     Data(190,3) Data(198,3) Data(200,3) Data(201,3) Data(202,3)
Data(203,3) ...
     Data(205,3) Data(206,3) Data(207,3) Data(214,3) Data(215,3)
Data(216,3) ...
     Data(219,3) Data(221,3) Data(222,3) Data(225,3) Data(235,3)
Data(245,3) ...
     Data(246,3) Data(247,3) Data(248,3) Data(251,3) Data(252,3)
Data(260,3) ...
```

```
     Data(262,3) Data(273,3) Data(274,3) Data(275,3) Data(277,3)
Data(278,3) ...
     Data(283,3) Data(284,3) Data(287,3) Data(288,3) Data(289,3)
Data(297,3) ...
     Data(299,3) Data(304,3) Data(305,3) Data(306,3) Data(309,3)
Data(310,3) ...
     Data(311,3) Data(312,3) Data(317,3) Data(318,3) Data(319,3)
Data(328,3) ...
     Data(331,3) Data(348,3) Data(349,3) Data(350,3) Data(351,3)
Data(364,3) ...
     Data(366,3)]);
 a_41=0;a41=a_41/S4;
 a_42=numel([Data(121,3) Data(252,3)]);
 a42=a_42/S4;
 a_43=numel([Data(1,3) Data(17,3) Data(92,3) Data(114,3) Data(139,3)
Data(190,3) ...
     Data(198,3) Data(203,3) Data(225,3) Data(253,3) Data(248,3)
Data(284,3) ...
     Data(297,3) Data(306,3) Data(328,3) Data(351,3) Data(364,3)]);
 a43=a_43/S4;
 a_44=numel([Data(12,3) Data(17,3) Data(20,3) Data(25,3) Data(29,3)
Data(42,3) ...
     Data(52:53,3)' Data(57:59,3)' Data(71:73,3)' Data(76:78,3)'
Data(90:92,3)' ...
     Data(120:121,3)' Data(127,3) Data(130,3) Data(134:135,3)'
Data(138:139,3)' ...
     Data(157,3) Data(163,3) Data(177,3) Data(187,3) Data(201:203,3)' ...
     Data(206:207,3)' Data(215:216,3)' Data(221,3) Data(246:248,3)'
Data(251,3)' ...
     Data(274:275,3)' Data(277,3) Data(284,3) Data(288:289,3)'
Data(305:306,3)' ...
     Data(310:312,3)' Data(318:319,3)' Data(349:351,3)']);
 a44=a_44/S4;
 a_45=numel([Data(13,3) Data(20,3) Data(25,3) Data(29,3) Data(42,3)
Data(53,3) ...
     Data(59,3) Data(61,3) Data(73,3) Data(78,3) Data(81,3) Data(103,3) ...
     Data(105,3) Data(127,3) Data(130,3) Data(135,3) Data(143,3)
Data(148,3) ...
     Data(157,3) Data(163,3) Data(173,3) Data(178,3) Data(188,3)
Data(207,3) ...
     Data(217,3) Data(219,3) Data(222,3) Data(260,3) Data(262,3)
Data(275,3) ...
     Data(278,3) Data(289,3) Data(299,3) Data(312,3) Data(319,3)
Data(331,3)]);
 a45=a_45/S4;
 a_46=numel(0);
 a46=a_46/S4;
S5=numel([Data(14:15,3)' Data(21:23,3)' Data(26,3) Data(30:40,3)'
Data(43:46,3)' ...
    Data(50,3) Data(54:55,3)' Data(60,3) Data(62:69,3)' Data(74,3)
Data(79:80,3)' ...
    Data(82:88,3)' Data(104,3) Data(106:107,3)' Data(109:113,3)'
Data(128,3) Data(131:132,3)' ...
    Data(136,3) Data(144:147,3)' Data(149:153,3)' Data(155,3)
Data(158:159,3)' Data(164:172,3)' ...
    Data(174:176,3)' Data(179:184,3)' Data(186,3) Data(189,3)
Data(192:195,3)' Data(197,3) ...
    Data(208:213,3)' Data(217:218,3)' Data(220,3) Data(223:224,3)'
Data(261,3) Data(263:265,3)' ...
    Data(268:269,3)' Data(276,3) Data(279:282,3)' Data(290:291,3)'
Data(300:303,3)' ...
    Data(313:316,3)' Data(320:327,3)' Data(332:347,3)']);
```

```
a_51=0;
a51=a_51/S5;
a_52=0;
a52=a_52/S5;
a_53=0;
a53=a_53/S5;
a_54=numel([Data(15,3) Data(23,3) Data(40,3) Data(50,3) Data(55,3)
Data(60,3) ...
    Data(69,3) Data(74,3) Data(80,3) Data(88,3) Data(104,3) Data(113,3) ...
    Data(128,3) Data(132,3) Data(136,3) Data(147,3) Data(155,3) Data(172,3)
...
    Data(176,3) Data(186,3) Data(189,3) Data(197,3) Data(213,3) Data(218,3)
...
    Data(221,3) Data(225,3) Data(261,3) Data(276,3) Data(282,3) Data(316,3)
...
    Data(327,3) Data(347,3)]);
a54=a_54/S5;
a_55=numel([Data(14,3)' Data(21:22,3)' Data(31:40,3)' Data(44:46,3)'
Data(54,3)' ...
    Data(63:69,3)' Data(79,3) Data(83:88,3)' Data(106,3) Data(110:113,3)'
...
    Data(131,3) Data(145:147,3)' Data(150:153,3)' Data(159,3)
Data(165:172,3)' ...
    Data(175:176,3)' Data(180:184,3)' Data(193:195,3)' Data(209:213,3)'
Data(217,3) ...
    Data(224,3) Data(264:265,3)' Data(269,3) Data(280:282,3)' Data(291,3)
Data(301:303,3)' ...
    Data(314:316,3)' Data(321:327,3)' Data(333:347,3)']);
a55=a_55/S5;
a_56=numel([Data(26,3) Data(47,3) Data(107,3) Data(153,3) Data(159,3)
Data(184,3) ...
    Data(195,3) Data(265,3) Data(269,3) Data(291,3)]);
a56=a_56/S5;
S6=numel([Data(27,3) Data(47:49,3)' Data(108,3) Data(154,3)
Data(160:161,3)' ...
    Data(185,3) Data(196,3) Data(266:267,3)' Data(270:271,3)'
Data(292:296,3)' ...
    ]);
a_61=0;
a61=a_61/S6;
a_62=0;
a62=a_62/S6;
a_63=numel(Data(271,3));
a63=a_63/S6;
a_64=numel([Data(27,3) Data(161,3) Data(296,3)]);
a64=a_64/S6;
a_65=numel([Data(49,3) Data(108,3) Data(154,3) Data(185,3) Data(196,3) ...
    Data(267,3)]);
a65=a_65/S5;
a_66=numel([Data(48:49,3)' Data(161,3) Data(267,3) Data(271,3)
Data(293:296,3)']);
a66=a_66/S6;
%Provide the data if you a different one.
m=input('PLEASE ENTER THE VALUE OF YOUR A MATRIX OR PRESS ZERO TO CONTINUE:
');
if m==0
    A=[a11 a12 a13 a14 a15 a16;
    a21 a22 a23 a24 a25 a26;
    a31 a32 a33 a34 a35 a36;
    a41 a42 a43 a44 a45 a46;
    a51 a51 a53 a54 a55 a56;
    a61 a62 a63 a64 a65 a66];
```

```
else
A=m;
end
b_11=0;b11=b_11/S1;
b_12=0;b12=b_12/S1;
b_13=0;b13=b_13/S1;
b_14=0;b14=b_14/S1;
b_15=numel([Data(123,3), Data(232,3) Data(257:258,3)']);
b15=b_15/S1;
b_21=numel(Data(242:244,3));
b21=b_21/S2;
b_22=numel(Data(239:240,3)');
b22=b_22/S2;
b_23=numel([Data(238,3), Data(241,3)]);
b23=b_23/S3;
b_24=numel(Data(230,3));
b24=b_24/S2;
b_25=numel([Data(5:7,3)' Data(9,3) Data(122,3) Data(141,3) Data(227:228,3)'
...
    Data(231,3) Data(253,3) Data(255,3) Data(356,3) Data(359,3)]);
b25=b_25/S2;
b_31=numel([Data(96:97,3)' Data(101:102,3)']);
b31=b_31/S3;
b_32=numel([Data(98:100,3)' Data(125,3)]);
b32=b_32/S3;
b_33=numel([Data(95,3) Data(353,3) Data(353,3) Data(355,3) Data(361,3)]);
b33=b_33/S3;
b_34=numel([Data(18,3) Data(118,3) Data(124,3) Data(256:257,3)' Data(308,3)
...
    Data(352,3) Data(354,3) Data(362,3) Data(365,3)]);
b34=b_34/S3;
b_35=numel([Data(2:5,3)' Data(8,3) Data(10:11,3)' Data(93:94,3)'
Data(115:117,3)' ...
    Data(140,3) Data(142,3) Data(191,3) Data(199,3) Data(204,3) Data(226,3)
...
    Data(229,3) Data(233:234,3)' Data(236:237,3)' Data(249:250,3)'
Data(254,3) ...
    Data(258:259,3)' Data(272,3) Data(285:286,3)' Data(298,3) Data(307,3)'
...
    Data(329:330,3)' Data(360,3) Data(363,3)]);
b35=b_35/S3;
b_41=0;
b41=b_41/S4;
b_42=numel([Data(91,3) Data(103,3) Data(222,3)]);
b42=b_42/S4;
b_43=numel([Data(29,3) Data(90,3) Data(92,3) Data(119:120,3)' Data(201,3)
...
    Data(214,3) Data(216,3) Data(221,3) Data(309,3) Data(364,3)
Data(366,3)]);
b43=b_43/S4;
b_44=numel([Data(12,3) Data(16:17,3)' Data(28,3) Data(41:42,3)'
Data(52:53,3)' ...
    Data(56:57,3)' Data(59,3) Data(61,3) Data(72:73,3)' Data(77,3)
Data(81,3) ...
    Data(89,3) Data(127,3) Data(130,3) Data(135,3) Data(138:139,3)'
Data(156:157,3)' ...
    Data(173,3) Data(177:178,3)' Data(200,3) Data(202,3) Data(205:206,3)'
...
    Data(245:247,3)' Data(251,3) Data(262,3) Data(274:275,3)' Data(283,3)
...
    Data(287:288,3)' Data(304,3) Data(309,3) Data(317:319,3)' Data(331,3)
...
```

```
    Data(349:350,3)']);
b44=b_44/S4;
b_45=numel([Data(1,3) Data(13,3) Data(19:20,3)' Data(24:25,3)' Data(51,3)
...
    Data(58,3) Data(70:71,3)' Data(75:76,3)' Data(78,3) Data(105,3)
Data(114,3) ...
    Data(126,3) Data(129,3) Data(133:134,3)' Data(137,3) Data(143,3)
Data(148,3) ...
    Data(162:163,3)' Data(187,3) Data(190,3) Data(198,3) Data(203,3)
Data(207,3) ...
    Data(215,3) Data(219,3) Data(225,3) Data(235,3) Data(248,3) Data(252,3)
...
    Data(260,3) Data(273,3) Data(277:278,3)' Data(284,3) Data(289,3)
Data(297,3)' ...
    Data(299,3) Data(305:206,3)' Data(310:312,3)' Data(328,3) Data(348,3)
Data(351,3)]);
b45=b_45/S4;
b_51=numel([Data(217,3) Data(223:224,3)' Data(333,3)]);
b51=b_51/S5;
b_52=numel([Data(107,3) Data(111:112,3)' Data(155,3) Data(108:109,3)'
Data(290,3) ...
    Data(303,3) Data(314,3) Data(320,3) Data(325,3) Data(334,3) Data(341,3)
...
    Data(345,3)]);
b52=b_52/S5;
b_53=numel([Data(26,3) Data(37,3) Data(46,3) Data(60,3) Data(62,3)
Data(64,3) ...
    Data(66:67,3)' Data(69,3) Data(79:80,3)' Data(86,3) Data(109:110,3)'
...
    Data(145:146,3)' Data(150,3) Data(166,3) Data(170:171,3)' Data(179,3)
...
    Data(182:184,3)' Data(193:195,3)' Data(210,3) Data(212,3) Data(218,3)
...
    Data(265,3) Data(268:263,3)' Data(281:282,3)' Data(291,3)
Data(301:302,3)' ...
    Data(313,3) Data(315,3) Data(321:323,3)' Data(326:327,3)' Data(335,3)
...
    Data(337:338,3)' Data(340,3) Data(342,3) Data(344,3) Data(347,3)]);
b53=b_53/S5;
b_54=numel([Data(14:15,3)' Data(22:23,3)' Data(30,3) Data(35:36,3)'
Data(38:39,3)' ...
    Data(44:45,3)' Data(50,3) Data(55,3) Data(65,3) Data(68,3) Data(74,3)
...
    Data(82:83,3)' Data(85,3) Data(87:88,3)' Data(106,3) Data(128,3) ...
    Data(131,3) Data(144,3) Data(151,3) Data(153,3) Data(158:159,3)' ...
    Data(167:169,3)' Data(172,3) Data(174:176,3)' Data(180,3) Data(186,3)
...
    Data(189,3) Data(197,3) Data(211,3) Data(213,3) Data(220,3)
Data(263:264,3)' ...
    Data(279:280,3)' Data(300,3) Data(316,3) Data(324,3) Data(336,3) ...
    Data(339,3) Data(343,3) Data(346,3)]);
b54=b_54/S5;
b_55=numel([Data(21,3) Data(31:34,3)' Data(40,3) Data(43,3) Data(54,3) ...
    Data(63,3) Data(84,3) Data(104,3) Data(113,3) Data(132,3) Data(136,3)
...
    Data(147,3) Data(149,3) Data(152,3) Data(164:165,3)' Data(181,3)
Data(192,3) ...
    Data(261,3) Data(276,3)]);
b55=b_55/S5;
b_61=numel([Data(160,3) Data(292,3)]);
b61=b_61/S6;
b_62=numel([Data(161,3) Data(185,3) Data(266,3)]);
```

```
b62=b_62/S6;
b_63=numel([Data(27,3) Data(108,3) Data(154,3) Data(196,3) Data(270:271,3)'
...
    Data(293,3) Data(296,3)]);
b63=b_63/S6;
b_64=numel([Data(47,3) Data(49,3) Data(267,3) Data(295,3)]);
b64=b_64/S6;
b_65=numel([Data(48,3) Data(294,3)]);
b65=b_65/S6;
P=input('PLEASE ENTER THE VALUE OF YOUR MATRIX B OR PRESS ZERO TO CONTINUE:
');
if P==0
    B=[b11 b12 b13 b14 b15;
   b21 b22 b23 b24 b25;
   b31 b32 b33 b34 b35;
   b41 b42 b43 b44 b45;
   b51 b52 b53 b54 b55;
   b61 b62 b63 b64 b65];
else
    B=P;
end
%Provide the data if you a different one.
N=numel(Data(1:end,3));
%input the value of PAI if you another one
PAI=input('PLEASE ENTER THE VALUE OF YOUR PAI OR PRESS ZERO TO CONTINUE:
');
if PAI==0
theta=[S1/N S2/N S3/N S4/N S5/N S6/N];
else
    theta=PAI;
end
disp('PRESS ANY KEY TO DISPLAY THE VALUE OF A, B & PAI EXTRACTED FROM THE
DATA SET: ')
pause
A
B
PAI=theta
disp('PLEASE PRESS ANY KEY TO CONTINUE: ')
pause
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
disp('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
%% Defined the Problem parameters and transfer function
% s=tf('s'); %Make 's' a tansfer function symbol
% G=input('Please Define the Transfer Function of the Plant:= ');% %please
defined the plant transfer function
% CostFunction=@(x) objmse(x1,x2); % Objective Function Handle for the GA
nVar=3;                     % Number of Variables
VarSize=[1 nVar];           % Size of Variables Matrix
VarMin=-1;                  % Lower Bound of Variables
VarMax= 1;                  % Upper Bound of Variables
VarRange=[VarMin VarMax];   % Variation Range of Variables
%% GA Parameters
MaxIt=100;%input('Please Enter the maximum number of generation: ');%100;%
Maximum Number of Iterations
nPop=50;%input('Please Provide the population Size of Genes:= ');%50;%
Population Size
pCrossover=0.7;%input('Please provide the Percentage of Cross Over
Operator:= ');%0.7;% Crossover Percentage
nCrossover=round(pCrossover*nPop/2)*2;  % Number of Parents (Offsprings)
pMutation=0.2;%input('Please Provide the Mutant percentage:= ');%0.2;%
Mutation Percentage
```

```
nMutation=round(pMutation*nPop);% Number of Mutants
%% Initialization
% Empty Structure to Hold Individuals Data
ini_ind.Position=[];
ini_ind.Cost=[];
ini_ind.Out=[];
% Create Population Matrix
pop=repmat(ini_ind,nPop,1);
% Initialize Positions
for i=1:nPop
    pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
    pop(i).Cost=objmse(pop(i).Position);
end
% Sort Population
pop=SortPop(pop);
% Store Best Solution
BestSol=pop(1);
% Vector to Hold Best Cost Values
BestCost=zeros(MaxIt,1);
%% GA Main Loop
for it=1:MaxIt
    % Crossover
    popc=repmat(ini_ind,nCrossover/2,2);
    for k=1:nCrossover/2
        i1=randi([1 nPop]);
        i2=randi([1 nPop]);
        p1=pop(i1);
        p2=pop(i2);
        [popc(k,1).Position,
popc(k,2).Position]=Crossover(p1.Position,p2.Position,VarRange);
        [popc(k,1).Cost]=objmse(popc(k,1).Position);
        [popc(k,2).Cost]=objmse(popc(k,2).Position);
    end
    popc=popc(:);
    % Mutation
    popm=repmat(ini_ind,nMutation,1);
    for k=1:nMutation
        i=randi([1 nPop]);
        p=pop(i);
        popm(k).Position=Mutate(p.Position,VarRange);
        [popm(k).Cost]=objmse(popm(k).Position);
    end
    % Merge Population
    pop=[pop
         popc
         popm];
    % Sort Population
    pop=SortPop(pop);
    % Delete Extra Individuals
    pop=pop(1:nPop);
    % Update Best Solution
    BestSol=pop(1);
    % Store Best Cost
    BestCost(it)=BestSol.Cost;
    % Show Iteration Information
    %disp(['TJ_Itr ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);
    % Plot Step Response
end
BCost=mean(BestCost);

%Randomize A as a form of optimization
[a,b]=size(A);
```

```
A=A+A+(eye(size(A)).*(BCost+rand).*ones(size(A)));
for i=1:a
    for j=1:b
        A(i,j)=A(i,j)/sum(A(i,1:end));
    end
end

%Randomize B as a form of optimization
[c,d]=size(B);
B=B+(eye(size(B)).*(BCost+rand).*ones(size(B)));
for i=1:c
    for j=1:d
        B(i,j)=B(i,j)/sum(B(i,1:end));
    end
end

[Y,I] = max(A, [], 2);
  ap = zeros(size(A));
  ap(sub2ind(size(A), 1:length(I), I')) = Y;
  disp('')
  disp('')
  %disp('The Peak Values of A is: ');
  Ap=ap;
  k=0.04;%input('Please Enter Then Smoothing Factor: ');
  disp('')
  disp('')
  AL=[-k*Ap(:,1)+k*Ap(:,2) (1/2)*k*Ap(:,1)-k*Ap(:,2)+(1/2)*k*Ap(:,3) ...
      (1/2)*k*Ap(:,2)-k*Ap(:,3)+(1/2)*k*Ap(:,4) (1/2)*k*Ap(:,3)-
k*Ap(:,4)+(1/2)*k*Ap(:,5) ...
      (1/2)*k*Ap(:,4)-k*Ap(:,5)+(1/2)*k*Ap(:,6) k*Ap(:,5)-k*Ap(:,6)];
  %Smooted value of A is:
  As=A+AL;
  [X H]=max(B, [], 2);
  bp=zeros(size(B));
  bp(sub2ind(size(B), 1:length(H), H'))=X;
  disp('')
  disp('')
%   disp('The Peak Values of B is: ')
  Bp=bp;
  BL=[-k*Bp(:,1)+k*Bp(:,2) (1/2)*k*Bp(:,1)-k*Bp(:,1)+(1/2)*k*Bp(:,3) ...
      (1/2)*k*Bp(:,2)-k*Bp(:,3)+(1/2)*k*Bp(:,4) +(1/2)*k*Bp(:,3)-
k*Bp(:,4)+(1/2)*k*Bp(:,5) ...
     k*Bp(:,4)-k*Bp(:,5)];
  %The Smooted value of B is:
  Bs=B+BL;
%->->->
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Forecast for March
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_00=Mar_21_9_00./sum(Mar_21_9_00)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_05=B(:,4)'.*A(5,:);
%which is Normal0ized as follows.
march_21_9_05=Mar_21_9_05./sum(Mar_21_9_05)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
```

```
march_21_9_10=Mar_21_9_10./sum(Mar_21_9_10)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_15=Mar_21_9_15./sum(Mar_21_9_15)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_20=Mar_21_9_20./sum(Mar_21_9_20)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_25=Mar_21_9_25./sum(Mar_21_9_25)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_30=Mar_21_9_30./sum(Mar_21_9_30)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_35=Mar_21_9_35./sum(Mar_21_9_35)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_40=Mar_21_9_40./sum(Mar_21_9_40)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_45=Mar_21_9_45./sum(Mar_21_9_45)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_50=Mar_21_9_50./sum(Mar_21_9_50)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_9_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_9_55=Mar_21_9_55./sum(Mar_21_9_55)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_00=Mar_21_10_00./sum(Mar_21_10_00)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_05=Mar_21_10_05./sum(Mar_21_10_05)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_10=Mar_21_10_10./sum(Mar_21_10_10)
disp('****************************************')
```

```
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_15=Mar_21_10_15./sum(Mar_21_10_15)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_20=Mar_21_10_20./sum(Mar_21_10_20)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_25=Mar_21_10_25./sum(Mar_21_10_25)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_30=Mar_21_10_30./sum(Mar_21_10_30)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_35=Mar_21_10_35./sum(Mar_21_10_35)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_40=Mar_21_10_40./sum(Mar_21_10_40)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_45=Mar_21_10_45./sum(Mar_21_10_45)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_50=Mar_21_10_50./sum(Mar_21_10_50)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_10_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_10_55=Mar_21_10_55./sum(Mar_21_10_55)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_00=Mar_21_11_00./sum(Mar_21_11_00)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_05=Mar_21_11_05./sum(Mar_21_11_05)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_10=Mar_21_11_10./sum(Mar_21_11_10)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_15=B(:,4)'.*A(5,:);
```

```
%which is Normalized as follows.
march_21_11_15=Mar_21_11_15./sum(Mar_21_11_15)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_20=Mar_21_11_20./sum(Mar_21_11_20)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_25=Mar_21_11_25./sum(Mar_21_11_25)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_30=Mar_21_11_30./sum(Mar_21_11_30)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_35=Mar_21_11_35./sum(Mar_21_11_35)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_40=Mar_21_11_40./sum(Mar_21_11_40)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_45=Mar_21_11_45./sum(Mar_21_11_45)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_50=Mar_21_11_50./sum(Mar_21_11_50)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_11_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_11_55=Mar_21_11_55./sum(Mar_21_11_55)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_12_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_12_00=Mar_21_12_00./sum(Mar_21_12_00)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_12_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_12_05=Mar_21_12_05./sum(Mar_21_12_05)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_12_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_12_10=Mar_21_12_10./sum(Mar_21_12_10)
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_12_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_12_15=Mar_21_12_15./sum(Mar_21_12_15)
```

```
disp('****************************************')
disp('Forecasted Traffic for 21st March 2016: ')
Mar_21_12_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_21_12_20=Mar_21_12_20./sum(Mar_21_12_20)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_00=Mar_22_9_00./sum(Mar_22_9_00)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_05=B(:,4)'.*A(5,:);
%which is Normal0ized as follows.
march_22_9_05=Mar_22_9_05./sum(Mar_22_9_05)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_10=Mar_22_9_10./sum(Mar_22_9_10)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_15=Mar_22_9_15./sum(Mar_22_9_15)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_20=Mar_22_9_20./sum(Mar_22_9_20)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_25=Mar_22_9_25./sum(Mar_22_9_25)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_30=Mar_22_9_30./sum(Mar_22_9_30)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_35=Mar_22_9_35./sum(Mar_22_9_35)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_40=Mar_22_9_40./sum(Mar_22_9_40)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_45=Mar_22_9_45./sum(Mar_22_9_45)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_9_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_50=Mar_22_9_50./sum(Mar_22_9_50)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
```

```
Mar_22_9_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_9_55=Mar_22_9_55./sum(Mar_22_9_55)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_00=Mar_22_10_00./sum(Mar_22_10_00)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_05=Mar_22_10_05./sum(Mar_22_10_05)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_10=Mar_22_10_10./sum(Mar_22_10_10)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_15=Mar_22_10_15./sum(Mar_22_10_15)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_20=Mar_22_10_20./sum(Mar_22_10_20)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_25=Mar_22_10_25./sum(Mar_22_10_25)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_30=Mar_22_10_30./sum(Mar_22_10_30)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_35=Mar_22_10_35./sum(Mar_22_10_35)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_40=Mar_22_10_40./sum(Mar_22_10_40)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_45=Mar_22_10_45./sum(Mar_22_10_45)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_10_50=Mar_22_10_50./sum(Mar_22_10_50)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_10_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
```

```
march_22_10_55=Mar_22_10_55./sum(Mar_22_10_55)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_00=Mar_22_11_00./sum(Mar_22_11_00)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_05=Mar_22_11_05./sum(Mar_22_11_05)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_10=Mar_22_11_10./sum(Mar_22_11_10)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_15=Mar_22_11_15./sum(Mar_22_11_15)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_20=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_20=Mar_22_11_20./sum(Mar_22_11_20)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_25=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_25=Mar_22_11_25./sum(Mar_22_11_25)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_30=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_30=Mar_22_11_30./sum(Mar_22_11_30)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_35=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_35=Mar_22_11_35./sum(Mar_22_11_35)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_40=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_40=Mar_22_11_40./sum(Mar_22_11_40)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_45=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_45=Mar_22_11_45./sum(Mar_22_11_45)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_50=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_50=Mar_22_11_50./sum(Mar_22_11_50)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_11_55=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_11_55=Mar_22_11_55./sum(Mar_22_11_55)
disp('****************************************')
```

```
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_12_00=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_12_00=Mar_22_12_00./sum(Mar_22_12_00)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_12_05=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_12_05=Mar_22_12_05./sum(Mar_22_12_05)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_12_10=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_12_10=Mar_22_12_10./sum(Mar_22_12_10);
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_12_15=B(:,4)'.*A(5,:);
%which is Normalized as follows.
march_22_12_15=Mar_22_12_15./sum(Mar_22_12_15)
disp('****************************************')
disp('Forecasted Traffic for 22nd March 2016: ')
Mar_22_12_20=B(:,2)'.*A(3,:);
%which is Normalized as follows.
march_22_12_20=Mar_22_12_20./sum(Mar_22_12_20)
disp('****************************************')
n=input('PLEASE ENTER THE TOTAL NUMBER OF MONTE CARLO ITERATION: ');
V=input('Please Enter The Day of the Month you want: Note march_21_10_00 is
the format: ');%march_21_10_00;
U=0;
for q=1:n
   U=U+V;
end
Vec=round(U)
u=[797.5 2248.5 3699.5 5150.5 6615.5 8052.5];
%uu=[];
Z=sum(Vec.*u)/n
```

APPENDIX L

ABU INTERNET DATA TRAFFIC FORECASTED VALUES

Time	21st March, 2016		22nd March, 2016	
	Actual Traffic (MB)	Forecasted Traffic (MB)	Actual Traffic (MB)	Forecasted Traffic (MB)
9:00 AM	6071	**6073.2**	5972	**5976.2**
9:05 AM	6074	**6060**	6042	**6037.6**
9:10 AM	5993	**5990.2**	6016	**6013.9**
9:15 AM	6020	**6025.8**	6048	**6056.4**
9:20 AM	6070	**6056.4**	6139	**6132.5**
9:25 AM	5969	**6002**	6100	**6096.6**
9:30 AM	6040	**6037.6**	6037	**6044.5**
9:35 AM	6059	**6056.4**	6046	**6037.6**
9:40 AM	6004	**6002**	6089	**6085.1**
9:45 AM	6021	**6013.9**	6071	**6073.2**
9:50 AM	5990	**5978.3**	6173	**6173.1**
9:55 AM	6027	**6037.6**	6184	**6179.9**
10:00 AM	6162	**6156.2**	6133	**6132.5**
10:05 AM	6156	**6132**	6105	**6108.8**
10:10 AM	6163	**6156.2**	6080	**6085.1**
10:15 AM	6146	**6137.5**	6105	**6101.9**
10:20 AM	6140	**6144.4**	6104	**6103.8**
10:25 AM	6125	**6120.6**	6135	**6132.5**
10:30 AM	6228	**6227.4**	6176	**6179.9**
10:35 AM	6154	**6156.2**	6077	**6073.2**
10:40 AM	6164	**6168.1**	6153	**6156.2**
10:45 AM	6138	**6127.5**	6187	**6186.8**
10:50 AM	6153	**6156.2**	6183	**6179.9**
10:55 AM	6218	**6179.9**	6219	**6215.5**
11:00 AM	6242	**6249**	6181	**6179.9**
11:05 AM	6269	**6286.7**	6191	**6191.8**
11:10 AM	6248	**6244.2**	6159	**6156.2**
11:15 AM	6216	**6215.5**	6115	**6113.8**
11:20 AM	6199	**6191.8**	6208	**6203.7**
11:25 AM	6198	**6191.8**	6215	**6215.5**
11:30 AM	6202	**6203**	6106	**6103.8**
11:35 AM	6186	**6184.9**	6190	**6191.8**
11:40 AM	6243	**6239.2**	6143	**6144.4**
11:45 AM	6188	**6191.8**	6256	**6251.1**
11:50 AM	6167	**6168.1**	6162	**6168.1**
11:55 AM	6197	**6191.8**	6103	**6108.8**
12:00 PM	6235	**6239.2**	6150	**6156.2**
12:05 PM	6247	**6251.1**	6171	**6168.1**
12:10 PM	6112	**6108.8**	6257	**6251.1**
12:15 PM	6270	**6274.8**	6258	**6263**
12:20 PM	6292	**6291.7**	6228	**6239.2**