REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

RECURRENT TYPE-1 FUZZY FUNCTIONS APPROACH FOR TIME SERIES FORECASTING

NİHAT TAK

PHD THESIS
DEPARTMENT OF STATISTICS
PROGRAM OF STATISTICS

ADVISOR
ASSOC. PROF. DR. ATIF A. EVREN

CO-ADVISOR
PROF. DR. MÜJGAN TEZ

İSTANBUL, 2016

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

RECURRENT TYPE-1 FUZZY FUNCTIONS APPROACH FOR TIME SERIES FORECASTING

A thesis submitted by Nihat Tak in partial fulfillment of the requirements for the degree of **DOCTORATE OF PHILOSPHY** is approved by the committee on 25.12.2016 in Department of Statistics, Statistics Program.

Thesis Advisor
Assoc. Prof. Dr. Atıf A. Evren
Yıldız Technical University

Co- Advisor
Prof. Dr. Müjgan Tez
Marmara University

Approved By the Examining Committee
Assoc. Prof. Dr. Atıf A. Evren
Yıldız Technical University ______________________

Assist. Prof. Dr. Doğan Yıldız, Member
Yıldız Technical University ______________________

Prof. Dr. Ali Hakan Büyüklü, Member
Yıldız Technical University ______________________

Prof. Dr. Erol Eğrioğlu, Member
Giresun University ______________________

Assoc. Prof. Dr. Barış Aşıkgil, Member
Mimar Sinan Fine Arts University ______________________

ACKNOWLEDGEMENTS

It is pleasure to express my deep sense of thanks and gratitude to my advisor and guide Assoc. Prof. Dr. Atıf A. Evren and co-advisor Prof. Dr. Müjgan Tez. It is also pleasure to deliver my deep thanks to another my co-advisor Prof. Dr. Erol Eğrioğlu. Their dedication and keen interest, above all their overwhelming attitude to help their students have been mainly responsible completing my thesis. I feel so lucky to be a student of these three amazing friendly people. It has been great and unique experience for me to work with them.

I owe a deep sense of gratitude to my friend and colleague Erdinç Karakullukçu for helping me to organize my thesis. I also owe a deep thank to my friend Ali Z. Dalar for helping me during me running the applications of the proposed method. I thank profusely Ibrahim Karabayır and all the staffs of the department of Econometry at Kırklareli University for their kind help and co-operation throughout my study period.

It is my privilege to thank to my family, for their encourangment throughout my research period.

November, 2016

Nihat TAK

TABLE OF CONTENTS

Page

LIST OF SYMBOLS

A,B	Fuzzy sets
C	Constant
c	Number of clusters
$C_{1,}$ C_2	Acceleration coefficients
d	Distance function
f_2,f_2	Consequent parameters
f_i	Fuzzy index value
F_{t+1}	Forecast value for period t+1
gbest	Global best value
i	Number of iterations
p,q,r	Consequent parameters
pbest	Personel best value
r_1, r_2	Random numbers
x,y	Input values
V	Velocities
v	Cluster centers
w_i	Firing Strength
$\hat{Y}$	Forecasts
Y_{t-p}	Input values of a time series
z	Consequence distribution or consequence parameter
Z	Input matrix
α	Smoothing constant
β	Coefficients
ε_{t-q}	Disturbance term values of a time series
μ	The degrees of memberships
σ^2	Variance
δ, φ, θ	Coefficients of an ARIMA model
γ	Covariance
$\wedge$	Intersection for fuzzy sets
$\vee$	Union for fuzzy sets

LIST OF ABBREVIATIONS

ABC	Australian Beer Consumption
ACF	Autocorrelation Function
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Networks
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
BIST	Istanbul Stock Exchange
COV	Covariance
ES	Exponential Smoothing
FCM	Fuzzy C-Means
FF	Fuzzy Functions
FIS	Fuzzy Inference Systems
FTS-N	Fuzzy Time Series Network
MA	Moving Average
MANFIS	Modified Adaptive Neuro Fuzzy Inference System
MAPE	Mean Absolute Percentage Error
MLP-ANN	Multilayer Perceptron Artifical Neural Network
PACF	Partial Autocorrelation Function
PSO	Particle Swarm Optimization
RMSE	Root Mean Squared Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
SSE	Sum of Squared Error
T1FF	Type-1 Fuzzy Functions
TAIEX	Taiwan Stock Exchange
VAR	Variance

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

RECURRENT TYPE-1 FUZZY FUNCTIONS APPROACH FOR TIME SERIES FORECASTING

Nihat TAK

Department of Statistics

PHD Thesis

Advisor: Assoc. Prof. Dr. Atıf A. EVREN

Co-advisor: Prof. Dr. Müjgan TEZ

Forecasting future values of a time series is a widespread problem for researchers. There are a lot of methods for these kinds of problems. While some of these are probabilistic, some of them are non-probabilistic methods. For probabilistic methods, autoregressive integrated moving average and exponential smoothing methods are commonly used. For non-probabilistic methods, Artificial Neural Networks (ANN) and fuzzy systems have been commonly used. There are numerous fuzzy systems methods. While most of these methods are rule-based, there are a few methods which do not require rules, such as type-1 fuzzy functions approach. While it is possible to encounter with a model such as AR model integrated to T1FF, there has not been proposed any model including type-1 fuzzy functions and moving average model in one algorithm. Our intuition is to get better forecasting results taking into account the disturbance terms.

The input data set is organized with the following variables. First, lagged values of the time series are used for the AR(p) part. Second, FCM algorithm is used to cluster the inputs. The degree of memberships and centers are stored. Third, for the MA(q) part, fuzzy functions' residuals are used. So, AR(p), MA(q), and degree of memberships of the objects are restored in the input data set. Since the function we have is not a derivative function, particle swarm optimization algorithm is preferred to obtain estimations of the coefficients. Australian beer consumption (ABC) data set, Istanbul stock exchange (BIST100) data sets from 2009 to 2013, and Taiwan stock exchange (TAIEX) data sets from 1999 to 2004 are used to evaluate the performance of the proposed method. The outcomes show that the proposed method outperforms the other methods for 12 real-world time series data sets.

Keywords: Autoregressive model, forecasting, moving average model, nonlinear time series, particle swarm optimization, type-1 fuzzy functions.

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ÖZET

ZAMAN SERİLERİ ANALİZİNDE GERİ BESLEMELİ 1. TİP BULANIK FONKSİYON YAKLAŞIMI

Nihat TAK

Istatistik Anabilim Dalı

Doktora Tezi

Adviser: Assoc. Prof. Dr. Atıf A. EVREN

Co-adviser: Prof. Dr. Müjgan TEZ

Zaman serileri analizinde öngörü yapmak araştırmacılar tarafından incelenilen yaygın bir problemdir. Daha iyi öngörü elde edebilmek için önerilmiş birçok yöntem bulunmaktadır. Bunlardan bazıları olasılık tabanlı iken, bazıları olasılıksal olmayan yöntemlerdir. Otoregresif hareketli ortalamalar ve üstel düzleştirme yöntemleri olasılıksal, yapay sinir ağları ve bulanık çıkarım sistemi yöntemleri ise olasılıksal olmayan yöntemlerden en yaygın olarak kullanılanlardır. Bulanık çıkarım sistemlerinin bir çoğu kural tabanlı yöntemlerken, 1. tip bulanık fonksiyon kural tabanlı bir yöntem değildir. Literatürde 1. tip bulanık fonksiyon yöntemini otoregresif modeller ile birleştiren yöntemlere rastlamak mümkün olmala beraber, 1. tip bulanık fonksiyonu hareketli ortalamalar ile birleştiren bir yöntem henüz önerilmemiştir. Bu tez çalışmasının amacı 1. tip bulanık fonksiyon yaklaşımını otoregresif hareketli ortalamalar modeli ile birleştirerek zaman serilerine ilişkin daha iyi öngörüler elde etmektir.

Çalışmada, girdi matrisinin oluşturulması şu şekilde olmuştur: İlk olarak zaman serisinin gecikmeli değerleri modelin otoregresif kısmı için bağımsız değişkenler olarak alınmıştır. İkinci olarak, bulanık c-kümeleme yöntemi kullanılarak girdiler kümelenmiştir ve üyelik dereceleri ile küme merkezleri hafızada tutulmuştur. Son olarak önerilen yöntemin hareketli ortalamalar kısmı için bulanık fonksiyonun hataları kullanılarak kalıntı terimleri elde edilmiştir. Böylece, AR(p), MA(q) ve üyelik dereceleri girdi matrisinde toplanmıştır. Amaç fonksiyonumuz türevlenebilir bir fonksiyon olmadığından bu değeri minimum yapma amacı ile parçacık sürü optimizasyon yöntemi kullanılmıştır. Önerilen yöntemin performansını test etmek için

Avusturalya bira tüketim verisi (ABC), 1999-2004 yılları arası Taiwan borsası verisi ve 2009-2013 yılları arası İstanbul Borsası günlük verileri uygulama veri seti olarak kullanılmıştır. Elde edilen sonuçlara göre, önerilen yöntemin diğer kıyaslanan yöntemlere göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Anahtar Kelimeler: Otoregresif modeli, öngörü, hareketli ortalamalar modeli, lineer olmayan zaman serileri, parçacık sürü optimizasyonu, 1. tip bulanık fonksiyon.

CHAPTER 1

INTRODUCTION

1.1 Literature Review

A time series is defined as a variable whose observations are determined in a time interval. This time interval can be hourly, daily, weekly, monthly, seasonally, yearly, or etc. In order to do forecasts for these kinds of data sets, a lot of methods have been studied by researchers in the recent decades. These methods are called as time series forecasting methods. Time series forecasting methods are collected in two categories: probabilistic models and non-probabilistic models. Probabilistic or stochastic methods, also called traditional methods, put some assumptions on time series. Stationarity is an important assumption in probabilistic time series forecasting methods. This assumption requires that the time series has constant mean, variance, and covariance function. One of the most widely used model for probabilistic time series forecasting is the autoregressive integrated moving average (ARIMA) model. This model is organized systematically to get the best ARIMA (p,d,q) model by Box- Jenkins [1]. In other words, Box-Jenkins made contribution to the model organization process using ARIMA models. However, ARIMA models assume a linear structure among the time series values. So, if a time series has a non-linear structure, ARIMA models are not capable of dealing with it. Nevertheless, in applications, most of the data sets do not satisfy the assumption of stationarity or linearity of the time series. Therefore, a lot of researchers, in recent years, have been studied alternative methods. For example, artificial neural networks (ANN) and fuzzy inference systems (FIS) have been widely used by researchers for forecasting problems. Some of the fuzzy inference systems are given in details in Chapter 2. After the paper on fuzzy set theory was published by Zadeh [2], Zadeh [3] has introduced another paper on linguistic variables and fuzzy systems. Later, several researchers also combined fuzzy set theory with inference systems and proposed fuzzy inference systems. Some well-known fuzzy inference systems are Mamdani and Assilian [4] FIS, Takagi and Sugeno [5] FIS, adaptive-network FIS proposed by Jang [6], and type-1 fuzzy functions proposed by Turksen [7].

While the systems, proposed by Mamdani and Assilian [4], and Takagi and Sugeno [5] are rule based systems, the system, proposed by Turksen [7] is not a rule based system. Since the detection of rules is an important problem, T1FF approach has a big advantage. At the beginning, fuzzy inference systems have been designed for classification problems. They were not used in time series forecasting problems. Fuzzy time series forecasting methods were first proposed by Song and Chisom [8] in 1993. Song and Chisom [9, 10] have given the definition of fuzzy time series forecasting process that is we encounter with three stages: fuzzification, determination of the rules, and deffuzification. After Song and Chisom [9, 10] gave the definition of fuzzy-time-series forecasting process, the expansion of time series forecasting methods with the help of FIS is conducted by the following researchers. Chen [11] has proposed a high-order fuzzy time series model for forecasting problems in 2002. ANN was used to forecast fuzzy time series by Huarng and Yu [12]. These two studies and many more studies have been conducted by researchers for the determination of the fuzzy relations of fuzzy time series and to obtain better forecasts. Some of the studies using artificial intelligence techniques and the fuzzy set theory are listed below. Genetic algorithm has been used for time series forecasting problems and numerous purposes for their forecasting problems by Kuo et al. [13], Chen and Chung [14], Kim et al. [15], Egrioglu [16], and Bas et al. [17]. Multivariate fuzzy time series forecasting methods have been studied by Egrioglu et al. [18], Chen and Tanuwijaya [19], Jilani at al. [20], and Huarng [21]. Particle swarm optimization algorithm in fuzzy time series methods have been used by Chau [22], Park et al. [23], Kuo et al. [24], Aladag et al. [25], and Huang et al. [26]. Time series forecasting methods based on fuzzy inference systems have been studied by Catalao and et al. [27], Chabaa et al. [28], Chang [29], Chen and Ma [30], Chen and Zhang [31], Egrioglu et al. [32].

The fuzzy functions approach was first used in time series forecasting by Beyhan and Alici [33]. Aladag et al. [34] studied type-1 fuzzy functions approach for time series forecasting as well. On one hand probabilistic or linear models can deal with time series when the linear part of the time series overcomes the non-linear part, on the other hand when the non-linear part of the time series overcomes the linear part, non-linear models produces acceptable outcomes. However, since the most of the real world time series contain both parts in their nature, hybrid models are proposed by researchers. These models are invented in order to deal with both linear and non-linear part of the time series. Seasonal autoregressive integrated moving average model and multilayer perceptron ANN (MLP-ANN) are hybridized by Tseng et al. [35]. Some of methods in which adapt both the linear part and the non-linear part have been introduced by Bas et al. [36], Lee and Tong [37], Chen and Wang [38], Pai and Lin [39], Zhang [40], BuHamra et al. [41], Jain and Kumar [42], Yolcu et al. [43].

1.2 Objective of the Thesis

The methods that have been proposed so far, in terms of type-1 fuzzy functions approach, has not included disturbance terms as inputs in their nature. The objective of this thesis is to propose a new method taking into account the disturbance terms.

1.3 Hypothesis

Type-1 fuzzy functions approach is preferred in this study rather than classic inference systems. The proposed method in this paper is a model that combines an autoregressive model, a moving- average model, and type-1 fuzzy functions approach in one algorithm. Disturbance terms are determined by using the residuals of type-1 fuzzy function. In order to minimize the objective function (SSE), Particle Swarm Optimization (PSO) is used. In literature, there is no proposed method that includes disturbance terms and type-1 fuzzy functions in one algorithm. Our hypothesis is that including the disturbance terms into the model will give more accurate forecasting results.

CHAPTER 2

FUZZY SET THEORY AND FUZZY INFERENCE SYSTEMS

There are a lot of uncertain and imprecise concepts in the world. Because of the lack of classic logic for modeling these kinds of concepts, the idea of fuzzy logic was first introduced in 1965 by Lotfi A. Zadeh [2], a professor at the University of California at Berkely. Fuzzy logic is not presented as a control or decision mechanism, but as a way of presenting data by allowing an object having partial set membership rather than crisp set membership. So, the very first contribution of Zadeh to the set theory is that he generalizes the classic set theory and calls it fuzzy set theory.

In classic set theory, either an element belongs to a set or it does not. For example, either an integer is even or it is odd, or either a person is male or female etc. How about an example of a person being short or tall, or good and bad etc.? The concepts introduced in the first example are precise so there is no confusion there. However, the concepts in the second example are fuzzy in nature. For some people tallness might mean different thing. For example, a person with 1.80 cm height might mean short for a basketball player. How we can formulate these kinds of concepts will be shown below starting with some basic definitions in fuzzy set theory.

2.1 Fuzzy Sets and Definitions

In the fuzzy set theory, the first step is defining the degree of membership function and the sets with the degree of memberships. Some definitions about fuzzy sets are given below [44].

Definition 1: Let X be a nonempty set. A fuzzy set A in X is characterized by its membership function

$$\mu_A : X \rightarrow [0,1] \qquad (2.1)$$

and $\mu_A(x)$ is interpreted as the degree of membership of element x in fuzzy set A for each x ϵ X. It is clear that A is completely determined by the set of tuples

$$A= \{(u, \mu_A(u)) \mid u \in X \}. \quad (2.2)$$

Definition 2: (Normal fuzzy set) A fuzzy subset A of a classical set X is called normal if there exists an x ϵ X such that A(x) = 1. Otherwise A is subnormal.

2.2 Operations in Fuzzy Sets

Definition 3: (Intersection) The intersection of A and B is defined as

$$(A \cap B)(x)= \min[\{A(x),B(x)\} = A(x) \wedge B(x), \quad (2.3)$$

for all x ϵ X.

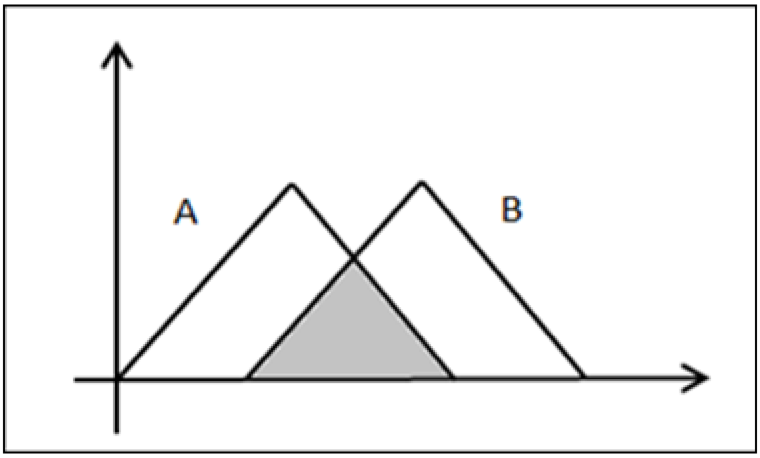

Figure 2.1 Intersection of A and B

Definition 4: (Union) The union of A and B is defined as

$$(AUB)(x)= \max\{A(x), B(x)\} = A(x) \vee B(x), \quad (2.4)$$

for all x ϵ X.

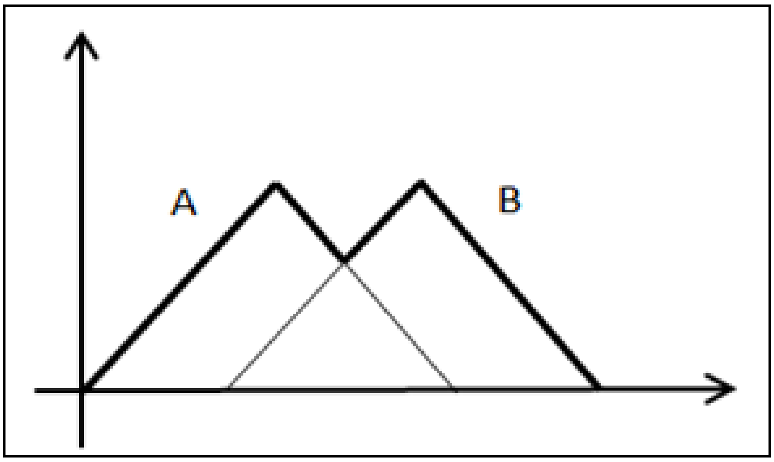

Figure 2.2 Union of A and B

Definition 5: (Complement) The complement of a fuzzy set A is defined as

$\bar{A}(x) = 1 - A(x).$ (2.5)

Definition 6: (Empty set) A fuzzy set is empty if and only if its membership function identically zero on X.

Definition 7: (Equal sets) Two fuzzy sets A and B are equal, written as A = B, if and only if $\mu_A(x) = \mu_B(x)$ for all x in X.

Definition 8: (Containment) A is contained in B (or, equivalently, A is a subset of B) if and only if $\mu_A < \mu_B$.

2.3 Fuzzy C-Means Algorithm

Clustering is widely used techniques in the field of data mining. Hathaway and Bezdek [45] have given the definition of clustering as "the objective of cluster analysis is the classification of objects according to similarities among them, and organizing of data into groups." The motivation of clustering is to detect the underlying structure in data.

Fuzzy c-means algorithm was introduced by Bezdek [46] in 1981. The main difference between hard clustering and fuzzy clustering is that in fuzzy clustering, we release the requirement that each data object belongs to one cluster. So, now, each data object can belong to more than one cluster with a certain degree, which is called the degree of membership in the concept of fuzzy sets. Fuzzy clustering methods are very useful for imprecise concepts such as good, bad, short, tall, young, old, and so on so forth; because these kinds of concepts differ one to another. While, the fuzzy c-means algorithm has exactly same steps with the k-means algorithm, the equations are slightly different. The steps of fuzzy c- means algorithm are given below.

Step 1 Initialize $\mu = [\mu_{ij}]$ matrix, determine the number of clusters and initial cluster centers.

Step 2 Calculate the membership value μ with the formula

$$\mu_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d(z_k, v_i)}{d(z_k, v_j)}\right)^{\frac{2}{f_i - 1}}\right]^{-1}, \quad (2.6)$$

under the constraint; $\sum_i^c \mu_{ik} = 1$.

where Z the data matrix, v the cluster centers, d(.) stands for Euclidean distance function, c is the number of clusters, and f_i fuzzy index value.

Step 3 Calculate the new cluster centers.

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^{fi} z_k}{\sum_{k=1}^{n} (\mu_{ik})^{fi}} \tag{2.7}$$

Step 4 Repeat Step 2 and Step 3 until the difference of clusters between two iterations drops under some threshold or the number of iterations that the researcher wants is achieved.

2.3.1 Distance Functions

Distance function d: X x Y → R calculates the distances or differences between two solutions x, y into real numbers R. The distance function must follow the following axioms of the metric space,

$$d(x,y) >= 0 \tag{2.8}$$

$$d(x,y) = 0 \text{ if } x=y\text{, otherwise } d(x,y)>0 \tag{2.9}$$

$$d(x,y) = d(y,x) \tag{2.10}$$

$$d(y,z) =< d(x,y) + d(y,z) \text{ (triangle inequality)} \tag{2.11}$$

The Euclidean distance function is given in the form of $d(z_k, v_i) = \|z_k - v_i\|$.

2.4 Fuzzy Inference Systems

Fuzzy inference system (FIS) is a widely used computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning [47]. Researchers use fuzzy inference systems in a wide variety of fields, such as automatic control, data classification, pattern recognition, clustering analysis, time series prediction, decision analysis, and so on so forth [48]. Because of its multidisciplinary nature, the fuzzy inference systems are also known as fuzzy-rule based systems, fuzzy expert system, and simply fuzzy systems [49].

Before Zadeh [2] published his first paper on fuzzy set, he was working in the field of systems. While he was studying on system analysis, he came to the realization that methods and techniques that were developed were not that suitable for humanistic systems. So, he came up with the idea of fuzzy logic. In 1965, he published his first paper on Fuzzy Logic, called Fuzzy sets. Eight years later in 1973, he introduced another paper [3], called "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", introducing linguistic variables and characterization of simple relations between variables by fuzzy conditional statements.

Zadeh [3], in his latter paper, defines that "a linguistic variable is defined as a variable whose values are sentences in a natural or artificial language. Thus, if *tall, not tall, very tall, very very tall,* etc. are values of height, then height is a linguistic variable". He also introduces the concept of fuzzy conditional statements, which is "expression of the form IF *A* THEN *B*, where *A* and *B* have fuzzy meaning, e.g., IF x is small THEN y is large" [4]. After these definitions were given, many researchers started studying on fuzzy inference systems.

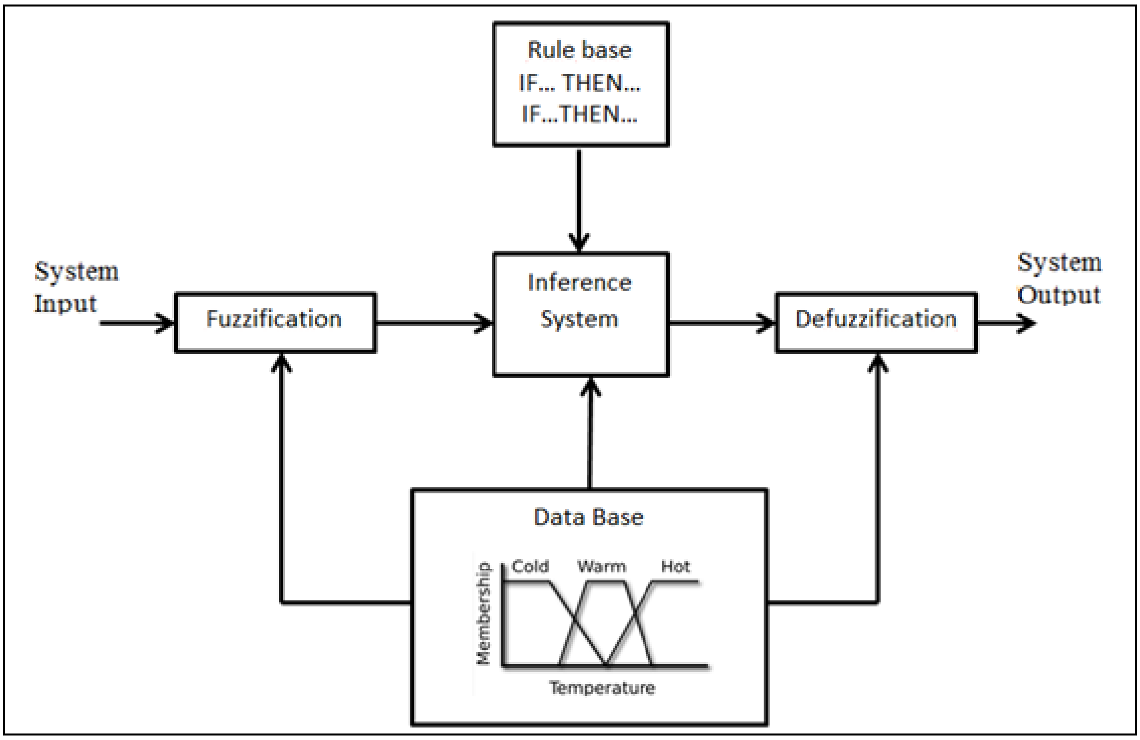

Figure 2.3 Fuzzy Inference Systems

Using linguistic terms, the aim of fuzzy inference systems is to define relationships between input and output variables of a system [50]. The basic structure of a fuzzy inference system consists of three components:

- Fuzzification: Each crisp input variable is transformed into a membership grade.
- Inference System: The inference system conducts the fuzzy reasoning process by applying the appropriate fuzzy operators in order to obtain the fuzzy set to be accumulated in the output variable.

- Defuzzification: The fuzzy output is transformed into a crisp output by applying a specific defuzzification method.

Widely used three inference systems are Mamdani and Assilian [4] fuzzy inference system, Takagi and Sugeno [5] fuzzy inference system, and Adaptive Neuro Fuzzy Inference System (ANFIS). The main difference between Mamdani FIS and Sugeno FIS is the specification of the consequent part. In the Mamdani [4] method, the consequents are fuzzy sets, and if we want to have a crisp output, we can use one of the several defuzzification methods to get the final crisp output. In contrast, in the Sugeno [5] method, consequents are real numbers, which can be either constant or linear. The final output is the weighted average of each rule’s output [51].

Sugeno, in his paper, has pointed out that “multidimensional fuzzy reasoning where we can reduce the number of implications.” So his main focus was on this and as well as “identification of a system using its input-output data. Identification is divided into two parts: structure identification and parameter identification” [5].

Jyh-Shing Roger Jang [6] has introduced in his paper Adaptive-Neuro Fuzzy Inference System, having a better and systematic approach to Sugeno’s fuzzy inference system in 1993. He has proposed that there are some basic aspects of Sugeno’s approach which are in need of better understanding. These aspects:

- There is no such method that converts human experience into the rule base and data base of a fuzzy inference system.
- There is need for effective methods for tuning the membership functions so as to minimize the output error measure or maximize performance index.

In this perspective, his aim was to build a fuzzy inference system, “which can serve as a basis for constructing a set of fuzzy if-then rules with appropriate membership functions to generate the stipulated input-output pairs” [6].

2.4.1 Mamdani Fuzzy Inference System

Mamdani fuzzy inference system, which is proposed by Mamdani and Assilian [4] in 1975, is the most commonly used methodology in literature. Mamdani type fuzzy inference system takes crisp data and fuzzifies it, and goes through the rule structure step, and finally gets a fuzzy output. A Mamdani fuzzy inference system example is given in Figure 2.4 [52].

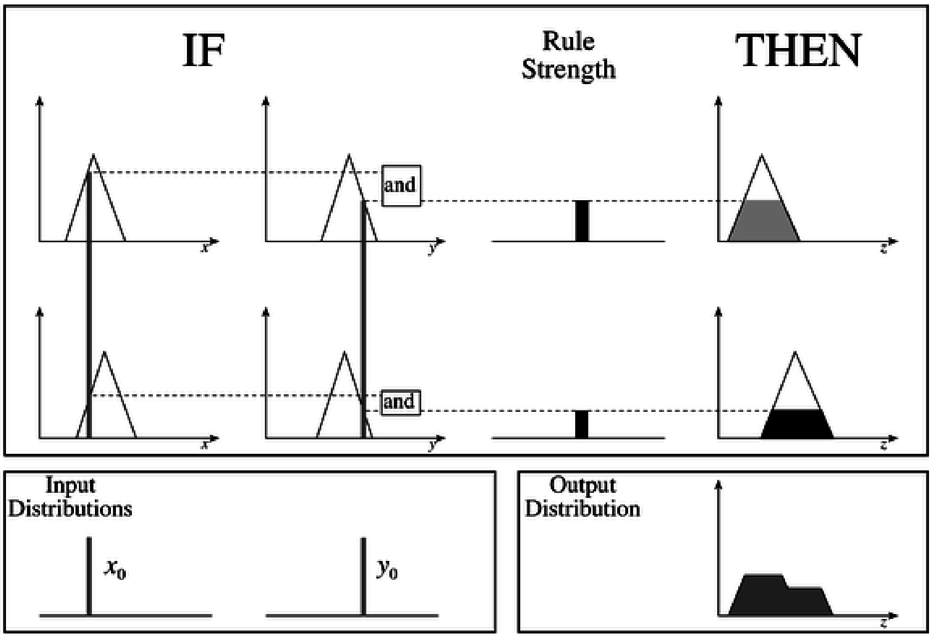

Figure 2.4 Mamdani Fuzzy Inference Sytem

In order to obtain the output of Mamdani fuzzy inference system, one must follow the steps given below.

- Determining fuzzy rules,
- Fuzzifying the inputs using membership functions,
- Establishing a rule strength with combining the fuzzified inputs according to the fuzzy rules,
- Finding the consequence of the rule,
- Combining the consequences to get an output distribution, and
- Deffuzifying the output distribution in a case that a crisp output is needed.

2.4.1.1 Creating Fuzzy Rules

Fuzzy rules are basically collection of linguistic terms which are defined by experts. They describe how the fuzzy inference system should make a decision. Fuzzy rules are written in the form given below.

IF (input1 is MembershipFunction1) AND/OR (input2 is MembershipFunction2) AND/OR ……. , THEN $output_n$ is $outputMembershipFunction_n$

2.4.1.2 Fuzzification

Input values are fuzzified using specified membership function.

These input functions can represent fuzzy concepts such as “large” or “small”, “old” or “young”, “hot” or “cold”, etc.

2.4.1.3 Fuzzy Combinations

In order to define a fuzzy rule, the concept of “and”, “or”, sometimes “not” are used. The most common definitions of these fuzzy combination operators are given below. Fuzzy combinations are also called as “T-norms”.

The fuzzy “and” is written as $U_{A \cap B} = T(U_A(x) ,\ U_B(x))$

There are many ways to compute “and”. The most common two ones;

- Zadeh’s - $\min(\mu_A(x) ,\ \mu_B(x))$
- Product - $\mu_A(x) \ * \ \mu_B(x)$

The fuzzy “or” is written as $\mu_{AUB} = T(\mu_A(x) ,\ \mu_B(x))$

- Zadeh’s - $\max(\mu_A(x) ,\ \mu_B(x))$
- Product - $\mu_A(x) \ + \ \mu_B(x) – (\mu_A(x) \ * \ \mu_B(x))$

2.4.1.4 Consequence

Using two steps given below, the consequence of a fuzzy rule is computed:

- First, by combining the fuzzified inputs, we compute the rule strength.
- Clipping the output membership function at the rule strength.

2.4.1.5 Combining Outputs into an Output Distribution

In order to obtain one fuzzy output distribution, the outputs of all of the fuzzy rules must be combined. This is usually done by using the fuzzy “or”. In order to get the output distribution, the output membership functions are combined using the fuzzy “or”.

2.4.1.6 Deffuzzification of Output Distribution

In many cases, a single crisp output is desired to come up with. Two common ways for deffuzziying is given below.

- Center of Mass takes the output distribution obtained in 2.4.1.5 and it takes its center for a crisp number.

- Mean of Maximum takes the output distribution obtained in 2.4.1.5 and it finds its mean of maxima for a crisp number.

2.4.2 Sugeno Fuzzy Inference System

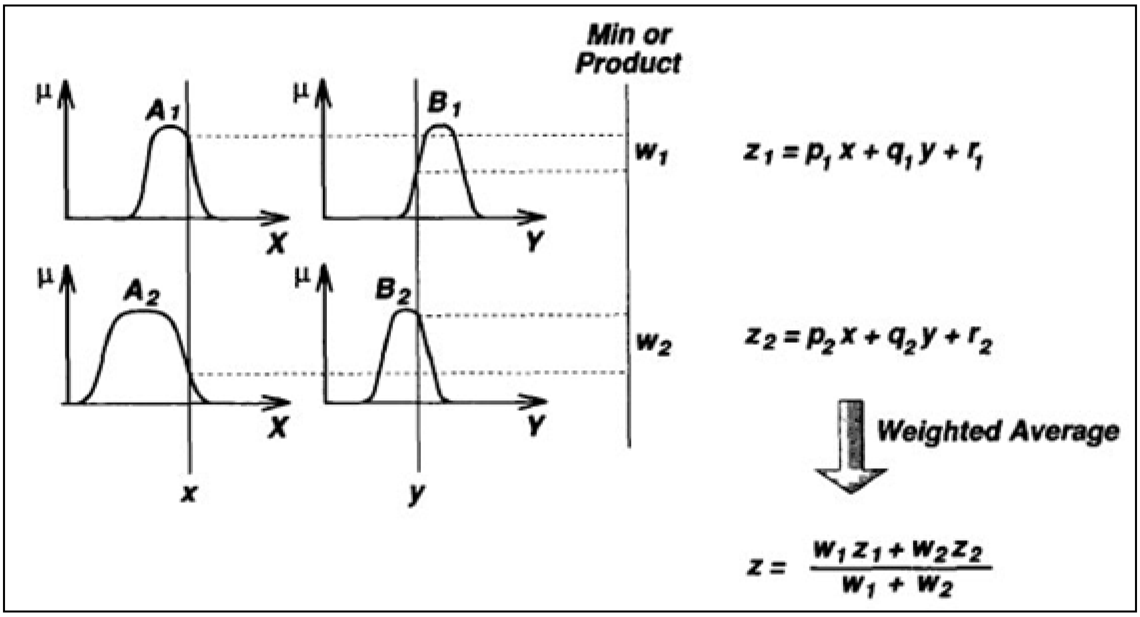

Figure 2.5 A two input, two rule Sugeno FIS with crisp inputs

Sugeno fuzzy inference system was proposed by Takagi and Sugeno [5] in 1985. Sugeno inference system is quite similar to Mamdani fuzzy inference system. The main difference is that the output consequence is in the form of constant or linear equation instead of in the form of output distribution. In fact, in Sugeno fuzzy inference system there is no output function. Instead, the output is a crisp number computed by multiplying each input by a constant and then summing up results. A typical form of a fuzzy rule in a Sugeno fuzzy inference system is given below.

IF x is A AND y is B THEN z= f(x,y) ,

where A and B are fuzzy sets, z = f(x, y), is a crisp function. An example of Sugeno fuzzy inference system is given in Figure 2.5 [53].

2.4.3 Adaptive-Neuro Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System (ANFIS) was proposed by Jang [6] in 1993. Because it has been one of the most commonly used and studied algorithm, it is very important tool for fuzzy learning society. Celikyilmaz and Turksen [51] points out that

“ANFIS is a neuro-fuzzy technique that brings learning capabilities of neural networks to fuzzy inference systems.”

An adaptive network consists of nodes and directional links.

Let us assume that we have the fuzzy system with two inputs x_1 and x_2 and one output variable y and two fuzzy rules. The graphical representation of this fuzzy system is given in Figure 2.6. The most used structure of ANFIS is Sugeno type fuzzy inference system, which is defined as follows:

Rule 1: IF x is A_1 AND y is A_3 THEN $f_1 = p_1 x + q_1 y + r_1$

Rule 2: IF x is A_2 AND y is A_4 THEN $f_2 = p_2 x + q_2 y + r_2$

As we can see from the Figure 3 that ANFIS consists of 5 layers.

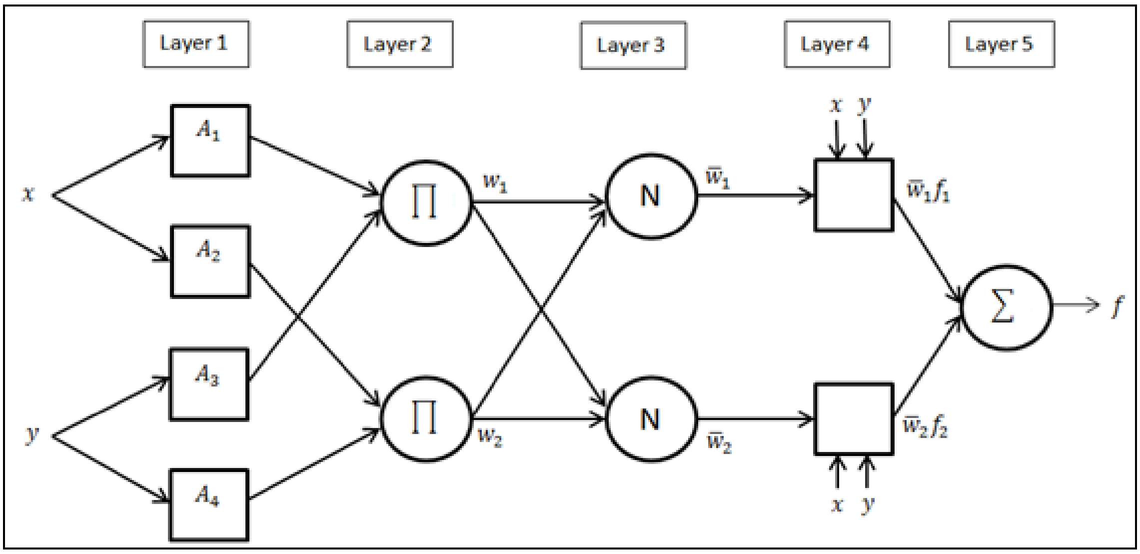

Figure 2.6 A two input, two rule ANFIS

Layer 1: (Fuzzification) We fuzzify the inputs in this layers, as it is described in Sugeno FIS or we can use fuzzy-C means (FCM) algorithm to convert the data into linguistic terms.

Layer 2: (Aggregation of Antecedents) Also called as product layer, every node in this layer is a fixed node. The output is the product of the all the incoming signals. Output can ben formalized as in Equation 2.12.

$$w_i = \mu_{Ai}(x) * \mu_{Bi}(y), i=1,2 \quad (2.12)$$

In this layer, each node represents the fire strength of the rule.

Layer 3: (Normalization of Degrees of Fire) Every node in this layer is a fixed node. The i[th] node calculates the ratio of i[th] rulet's firing strength to the sum of all rulet's firing strengths. Outputs are called normalized firing strengths.

$$\bar{w}i = \frac{wi}{w2+w1}, \quad i = 1,2 \tag{2.13}$$

Layer 4: (Implication) Every node in this layer is an adaptive node with a node function given in Equation 2.14.

$$\bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{2.14}$$

Where $\bar{w}_i$ is the normalized firing strength from layer 3, $\{p_i, q_i, r_i\}$is the parameter set of this node. These are called to as consequent parameters.

Layer 5: (Aggregation of Consequents) The single node in this layer is a fixed node labeled sum, which computes the overall output with the equation given below.

$$\sum_i \bar{w}ifi = \frac{\sum_i wi\, fi}{\sum_i wi} \tag{2.15}$$

2.4.4 Type-1 Fuzzy Functions Approach

Many methods for modelling uncertainty with fuzzy logic have been introduced in literature. Some of them are given above, such as Mamdani Fuzzy Inference System, Sugeno Inference System, Adaptive Neuro Fuzzy Inference System, which are rule based systems. Turksen [51] has introduced type-1 fuzzy functions, which is free of rule base, in 2008. This is an important advantage for modelling. Now, there is no need for an expert opinion to define the rules. The system can obtain the rules itself. Beside fuzzy functions method has been introduced for modelling regression and classification problems, it has been recently used for time series problems by many researchers. The steps of Type-1 Fuzzy Functions (T1FF) using fuzzy C-means (FCM) are given below [51].

Step 1 The number of fuzzy clusters, fuzzy index value (f_i), the number of iterations, and the center of clusters are determined. Inputs are lagged variables of time series.

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^{fi} z_k}{\sum_{k=1}^{n} (\mu_{ik})^{fi}}, \quad i=1,2,..,c \tag{2.16}$$

$$\mu_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d(z_k, v_i)}{d(z_k, v_j)} \right)^{\frac{2}{fi-1}} \right]^{-1}, \quad i=1,2,\ldots,c\, ;\ k=1,\ldots n \tag{2.17}$$

where matrix Z is composed of both inputs and output of the system, d(z,v) is the Euclidian distance and is calculated as below, and μ_{ik} is degree of belongingness of k^{th} observation to i^{th} cluster.

$$d(z_k, v_i) = \|z_k - v_i\| \tag{2.18}$$

Step 2 We can calculate the membership values of the input space as below.

$$\mu_{ik} = \left[\sum_{j=1}^{n}\left(\frac{d(x_k,v_i)}{d(x_k,v_j)}\right)^{\frac{2}{fi-1}}\right]^{-1}, \text{ i=1,2,…,c ; k=1,…n} \tag{2.19}$$

where x consists of only inputs which are generated for lagged variables.

Step 3 Original inputs and membership values of each input data sample (μ_{ik}) are combined for each cluster i. Then, we obtain the i^{th} fuzzy function from $Y^{(i)} = X^{(i)}\beta^{(i)} + \varepsilon^{(i)}$ multivariate regression model. For example, the number of inputs is p, $X^{(i)}$ and $Y^{(i)}$ matrices are as follows:

$$X^{(i)} = \begin{bmatrix} \mu_{i1} & x_{11} & . & . & . & x_{2p} \\ \mu_{i2} & x_{21} & . & . & . & x_{2p} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \mu_{in} & x_{n1} & . & . & . & x_{np} \end{bmatrix}, \quad Y^{(i)} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}$$

Celikyılmaz and Turksen [51] has proposed that mathematical transformations, such as exponential or logarithmic, of membership values might cause better results.

Step 4 By using the results obtained from fuzzy functions output values of system are calculated as follows:

$$\hat{y}_i = \frac{\sum_{i=1}^{c}\hat{y}_{ik}\mu_{ik}}{\sum_{i=1}^{c}\mu_{ik}} \quad \text{, k=1,2,…,n} \tag{2.20}$$

CHAPTER 3

FORECASTING

Sequence of data points is called as a time series. These data points are consisted of measurements of over a time interval. Some widely used time series examples are stock exchange data, passengers of an airplane firm, data of crisis, and etc. There are a lot of proposed techniques for modeling these kinds of concepts and the most commonly used models are Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing in literature. Time series are analyzed under two roofs, probabilistic and non-probabilistic approaches. Probabilistic approaches will be discussed in this chapter. However, some of the non-probabilistic approaches were given in chapter two; such as ANFIS, Mamdani FIS, and Sugeno FIS.

3.1 Autoregressive Integrated Moving Average Models

ARIMA models are regression models. They use lagged values of the dependent variable and/or random disturbance terms as explanatory variables. ARIMA models rely on the autocorrelation pattern in the data.

ARIMA models are combination of Autoregressive (AR) and Moving Average (MA) Models and "I" stands for integration. Three basic ARIMA models for a stationary time series [54];

- Autoregressive model of order p (AR(p))

$$y_t = \delta + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \tag{3.1}$$

i.e. y_t depends on its p previous values.

- Moving Average model of order q (MA(q))

$$y_t = \delta + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{3.2}$$

i.e. y_t depends on its q previous random error terms.

- Autoregressive- Moving Average model of order p and q (ARMA(p,q))

$$y_t = \delta + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad (3.3)$$

i.e. y_t depends on its p previous values and its q previous random error terms.

The random disturbance term ε_t is assumed to be white noise which means it is independently identically distributed with mean 0 and a common variance for all terms.

Box-Jenkins [1] has proposed an algorithm for modeling ARIMA models. The Box-Jenkins method is widely used method for time series. Researchers often prefer the Box-Jenkins approach because it combines set of procedures such as stationarity, identifying and estimating time series. The method consists of three steps .

3.1.1 A three-step Iterative Process

Step 1 Identification

Step 2 Estimation

Step 3 Diagnostic Checking and Forecasting

Step 1 Identification

The most important property of almost all time series models is stationarity. If a time series y_t satisfies the following conditions, then y_t is said to be stationary.

- E(y_t) = u_y for all t (3.4)
- Var(y_t) = E($(y_t - u_y)^2$) = σ_y^2 for all t (3.5)
- Cov(y_t, y_{t-k}) = γ_k for all t. (3.6)

If the time series y_t is not stationary, one must follow some procedures to make y_t stationary. These procedures may include taking log, differencing, integration, and etc. Once y_t is stationary, the next step is to identify whether the process follows AR, MA, or ARIMA. In order to do that we can look at autocorrelation function (ACF) and partial autocorrelation function (PACF).

Step 2 Estimation

After we identify the model, we estimate the parameters of the model. The most common two methods for estimating the parameters of the model are the least squares and maximum likelihood estimators. If a moving average process does not exist in the model, we can use the least squares method. A moving average model cannot be estimated with the least square estimators like autoregressive processes. Under the assumption of normality of the error term, the covariance matrix of the error term can be estimated using maximum likelihood estimators.

Step 3 Diagnostic Checking and Forecasting

Often, it is not possible to determine a single model that explains the data. Therefore, we should pursue a set of procedures to get best model that represents the data set. These procedures include coefficients significance, residual analysis, and model selection criteria. Also, in order to check whether the model can explain the series or not, we can test autocorrelation coefficients whether they are significant or not, and error terms whether they are normal or not.

3.2 Exponential Smoothing Methods

The idea of exponential smoothing method is to smooth the time series. If a time series has a deterministic or stochastic trend, then an exponential smoothing method is an alternative method in order to forecasting problems of a time series. Exponential smoothing methods are covered under three methods: simple exponential smoothing method, Holt's exponential smoothing method, and Winter's exponential smoothing method [55].

The simple exponential smoothing method is used when the time series is around its mean. Weights of the time series decline exponentially and most recent observations are weighted most in this method. In order to forecast the time series, the equation given below is used.

$$F_{t+1} = \alpha y_t + (1 - \alpha)F_t \tag{3.7}$$

where:

F_{t+1} = forecast value for period t+1

y_t = actual value for period t

α = alpha (smoothing constant)

Holt exponential smoothing method, also called double exponential smoothing method, is used when the time series has trend. In this case, second smoothing constant is accounted for trend.

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + b_{t-1}) \quad \text{(level)} \tag{3.8}$$

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \quad \text{(trend)} \tag{3.9}$$

$$F_{t+1} = L_t + b_t \quad \text{(forecast)} \tag{3.10}$$

for 0 $0 \le \alpha \le 1, 0 \le \beta \le 1$.

Winter exponential smoothing method, also called triple exponential smoothing method, takes into accounts both trend and seasonality.

$$L_t = \frac{Y_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + T_{t-1}) \quad \text{(level)} \tag{3.11}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad \text{(trend)} \tag{3.12}$$

$$S_t = \gamma\frac{Y_t}{L_t} + (1-\gamma)S_{t-s} \quad \text{(seasonal)} \tag{3.13}$$

$$F_{t+p} = (L_t + pT_t)S_{t-1+p} \quad \text{(forecast)} \tag{3.14}$$

where s is the length of the seasonal cycle, for 0 $0 \le \alpha \le 1, 0 \le \beta \le 1 \; and \; 0 \le \gamma \le 1$.

CHAPTER 4

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a method for optimization problems. The method was first introduced by Kennedy and Eberhart [56] in 1995. The purpose of this method was to optimize continuous nonlinear functions. The main idea behind PSO was animal social behavior in fish school or a bird flock. An individual`s experience can provide some information to the flock or, in another words, an individual in a flock can benefit from the rest of the flock`s experience during the search of food. So, the idea behind particle swarm optimization was to provide communication among members in the flock.

Kennedy and Eberhart [56] have introduced in their algorithm that each particle in a swarm has a random position and from that position they are searching for the optimum. Since each particle is moving to the optimum, they also have their own velocities. Each particle remembers its best position between iterations and it is called personal best (pbest). The best of personal best values is defined as the global best value (gbest). Using the personal best, the global best values and the velocities for each particle, Kennedy and Eberhart [56] have given the algorithm below.

Particle Swarm Optimization algorithm is an iterative algorithm. In each iteration, algorithm searches for the personal best value of each particle, and the global best value among all particles.

Each particle`s value and velocity are stored in the vectors given below.

$$P_i = (p_{i1}, p_{i2}, \dots, p_{iM}) \tag{4.1}$$

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iM}) \tag{4.2}$$

In order to obtain the global best, fitness values corresponding to the particles are compared and the particle having the maximum or minimum fitness value is assigned to global best. Personal best values and global best value are stored in a vector given below.

$$pbest_i = (p_{i1}, p_{i2}, \dots, p_{iM}) \quad (4.3)$$

$$gbest = (p_1, p_2, \dots, p_M) \quad (4.4)$$

The equation in (4.5) is used to get velocities of particles.

$$velocity_{im}^{k+1} = velocity_{im}^{k} + c_1.r_1^k.\left(pbest_{im}^{k} - p_{im}^{k}\right) + \; c_2.r_1^k.(gbest^k - p_{im}^{k}) \quad (4.5)$$

where, m=1,2, …,M, M is dimension, k is the number of iterations, r_1^k and r_1^k are random numbers from uniform distribution, c_1 and c_2 are acceleration coefficients. Although there are a number of ways to specify acceleration coefficients, c_1 and c_2 are taken as 2 most of the time by researchers. We do not go into details for specifying acceleration coefficients because we also take $c_1 = \; c_2 = 2$ in our proposed method.

After updating the velocity of a particle the next step is to update the positions of the particle. In order to that the equation given below is used.

$$p = p_{im}^{k} + velocity_{im}^{k+1} \quad (4.6)$$

These updates are repeated until for the number of iteration times.

4.1 Algorithm

Step 1 Each particle's positions are randomly initialized.

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iM}) \quad (4.7)$$

Step 2 Velocities corresponding to each position are randomly initialized.

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iM}) \quad (4.8)$$

Step 3 Using the evaluation function, pbest and gbest values are updated.

$$pbest_i = (p_{i1}, p_{i2}, \dots, p_{iM}) \quad (4.9)$$

$$gbest = (p_1, p_2, \dots, p_M) \quad (4.10)$$

Step 4 Let c_1 and c_2 to be equal to two.

Step 5 Velocities and positions in the swarm are updated with the equations given in (4.5) and (4.6) respectively.

CHAPTER 5

PROPOSED METHOD

Type-1 fuzzy functions approach was designed as a fuzzy inference system in 2008 by Turksen [51]. Since the classic fuzzy inference systems are rule based, Turksen [7] proposed type-1 fuzzy functions approach in the need of a non-rule based system. This is a big advantage over classic fuzzy inference systems since they need an expert opinion for defining the rules. At the beginning, type-1 fuzzy functions approach was proposed for classification and regression problems. Therefore, T1FF is needed to be redesigned. T1FF approach was first adapted to time series by Beyhan and Alici [33] in 2010. Later, Aladag et al. [34] adapted T1FF to time series forecasting problems as well in 2015. Beyhan and Alici [33] used an ARX model structure in their study. This structure is not able to choose the best model. In other words, the structure was not capable of searching the best model. Therefore, Aladag et al. [34] proposed fuzzy time series function method to search for the best model. They adapted autoregressive model into their algorithm. Eventually, they have gotten better forecasting results than Beyhan and Alici. Among the conventional probabilistic time series models autoregressive models are the most important models as well as moving average models. Aladag et al. [34] used autoregressive model in their approach. In the proposed method, moving average model is also taken into consideration. The inputs are taken as the lagged values of the time series, the lagged values of the disturbance terms and the degree of memberships obtained from fuzzy c-means clustering method. Disturbance terms are obtained from residuals of the fuzzy functions. Since the objective function of the fuzzy functions is not derivative particle swarm optimization algorithm is adapted to the proposed method to obtain the estimation of coefficients that minimizes the objective function, and hence the sum of squared errors (SSE).

5.1 Flow Chart

The flow chart of the proposed method is given in Figure 5.1.

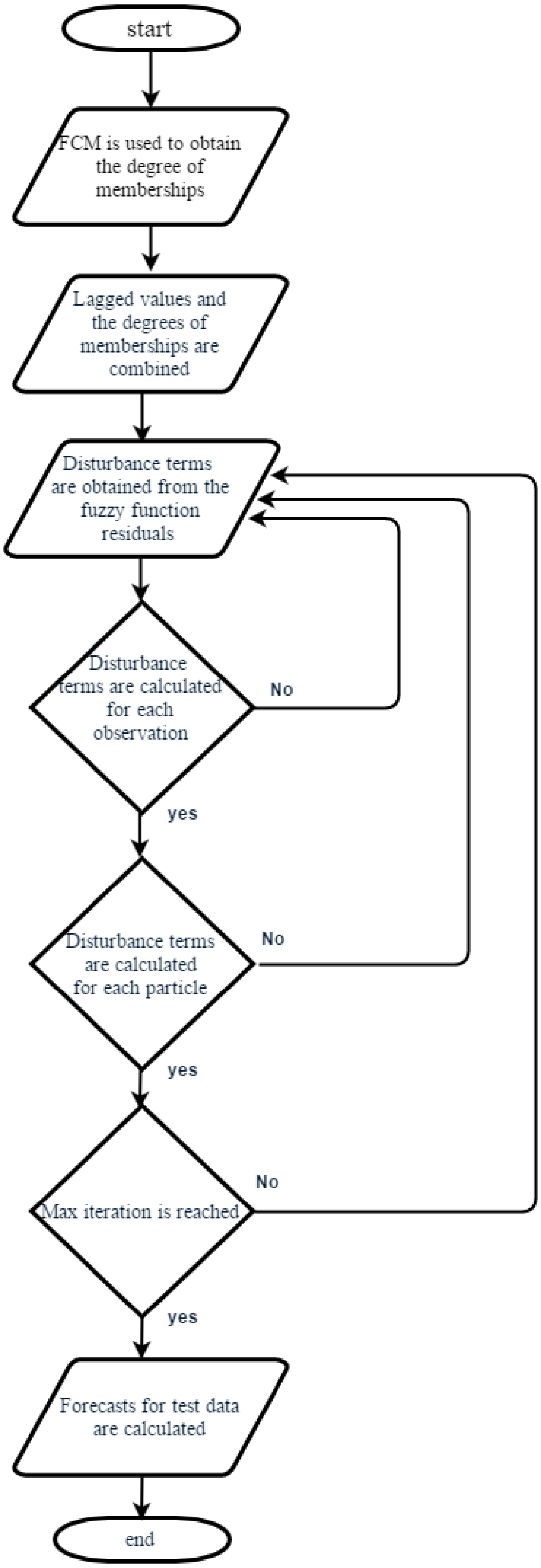

Figure 5.1 Flow chart of the proposed method

5.2 Algorithm

Step 1 The data set is divided into two groups: Test and training data sets.

Step 2 Model inputs are selected as the lagged variables of the data set and disturbances. Inputs are clustered using FCM algorithm.

Step 3 Lagged variables, degree of memberships, and the functions of degree of memberships are combined into the training data set. So that, the input matrix (X) is obtained. The dimension of the input matrix is $n * p * c * k$, n: number of observations, p: number of parameters, c: number of clusters, k: number of particles.

$$Y^{(i)(j)} = x^{(i)(j)} * \beta^{(i)(j)} + e^{(i)(j)} \quad , \quad \begin{matrix} i: 1,2,\dots,c \\ j: 1,2,\dots,k \end{matrix} \tag{5.1}$$

$$X^{(i)(j)} = \begin{bmatrix} \mu_{i1} & x_{11} & x_{21} & \cdots & x_{p1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mu_{in} & x_{1n} & x_{2n} & \cdots & x_{pN} \end{bmatrix}; \qquad Y^{(i)(j)} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad \beta^{(i)(j)} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$$

Step 4 The degrees of memberships are obtained for the training data set using fuzzy c-means algorithm. The centers of clusters are also obtained for the training data set. In this case, by using these centers, the degrees of memberships for the test data set are obtained for each observation using the formula below.

$$\mu_{test_{ik}} = \left[\sum_{j=1}^{c} \left(\frac{d(X_{test}, center_i)}{d(X_{test}, center_j)} \right)^{\frac{2}{f_i - 1}} \right]^{-1} \tag{5.2}$$

Step 5 μ_{test_i} and X_{test} are combined for each cluster and particle.

For example, for ith cluster and jth particle the input matrix;

$$X_{test} = \begin{bmatrix} C & \mu_{test} & log\mu_{test} & {\mu_{test}}^2 & Y_{t-1} & Y_{t-2} & \dots & Y_{t-p} & \epsilon_{t-1} & \epsilon_{t-2} \\ 1 & . & . & . & . & . & \dots & . & . & . \\ 1 & . & . & . & . & . & \dots & . & . & . \\ . & . & . & . & . & . & \dots & . & . & . \\ 1 & . & . & . & . & . & \dots & . & . & . \end{bmatrix}$$

Step 6 The parameters C_1 and C_2, the number of particles, and the number of iterations are specified for particle swarm optimization algorithm. The number of positions in each particle is $(p + q + 4) * c$. p: number of lags for AR, q: number of lags for MA. c: number of clusters, 4 stands for the first four columns in the input matrix, X_{test}.

Step 7 Initial positions for each particle are generated randomly from standard normal distribution. The coefficients (β), i,e, $Y^{(i)(j)} = x^{(i)(j)} * \beta^{(i)(j)} + \epsilon^{(i)(j)}$, represent the possible solutions in the regression model.

For example, for the first particle, assuming p=4, q=2, and c=2, the number of positions is (4+p+q)*2 = (4+4+2)*2=20.

Table 5.1 The positions for the first particle:

c	μ_1	$\log(\mu)_1$	${\mu^2}_1$	Y_{t-1}	Y_{t-2}	Y_{t-3}	Y_{t-4}	e_{t-1}	e_{t-2}
β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9

c	μ_2	$\log(\mu)_2$	${\mu^2}_2$	Y_{t-1}	Y_{t-2}	Y_{t-3}	Y_{t-4}	e_{t-1}	e_{t-2}
β_{10}	β_{11}	β_{12}	β_{13}	β_{14}	β_{15}	β_{16}	β_{17}	β_{18}	β_{19}

Initial velocities corresponding to the positions are generated randomly from $unif(0,1)$ distribution.

Step 8 Initial personal best (pbest) values are assigned to as initial positions. Initial global best (gbest) value is obtained by using fitness value.

Step 9 For each particle, e_{t-q} values are calculated using formulas below.

Step 9.1 $\hat{Y}_i^{(j)(k)} = X_{i,.}^{(j)(k)} . {\beta^T}_{i,.}^{(j)(k)}$ (5.3)

Step 9.2 $\widehat{Y*}_i^{(k)} = \hat{Y}_i^{(j)(k)} . {\mu^T}_{i,.}^{(j)}$ (5.4)

Step 9.3 $\epsilon_i^k = Y_i - \widehat{Y*}_i^{(k)}$ (5.5)

Step 9.4 $X_{i+1,p}^{(.)(k)} = e_i^k$ (5.6)

In Step 9.1, $\hat{Y}_i^{(k)}$ stands for the predicted value of the time series for i^{th} cluster, $X_{i,.}^{(j)(k)}$ stands for the input matrix and ${\beta^T}_{i,.}^{(j)(k)}$ stands for the position obtained via PSO for i^{th} cluster.

For example,

$X_{1,.}^{(1)(1)} = [C \quad \mu \quad \mu^2 \quad Y_{t-1} \quad Y_{t-2} \quad e_{t-1}]_{1x6}$,

$\beta_{1,.}^{(1)(1)} = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6]_{1x6}$,

$\widehat{Y^*}_i^{(k)} = [\ \ .\ \ ]_{1\times1}$

Step 9.1 is repeated for the number of clusters times (j) and the values of $\hat{Y}_i^{(j)(k)} = [\overbrace{\ldots\ldots}^{c}]$ are obtained for c clusters.

For i^th fuzzy function, $\hat{Y}_i^{*(k)}$ is calculated in Step 9.2.

For example, for the first object and k^th particle,

$\widehat{Y^*}_1^{(.)} = \hat{Y}_1^{(.)(k)} \,.\, \mu^{T}{}_{1,.}^{(.)}$

$$= [\mu_1^1 \quad \ldots \quad \mu_1^c]_{1\times c} \ \text{x} \begin{bmatrix} \hat{Y}_1^{(1)(k)} \\ . \\ \hat{Y}_1^{(c)(k)} \end{bmatrix}_{c\times1}$$

$\widehat{Y^*}_1^{(.)} = \qquad [\ \ .\ \ ]_{1\times1}$

ϵ_i^k value is obtained in Step 9.3. ϵ_i^k is the same value for each cluster. Substructing $\widehat{Y^*}_i^{(k)}$ value from Y_i , ϵ_i^k is obtained. The equation is given below.

$\epsilon_i^k = Y_i - \widehat{Y^*}_i^{(k)}$

For example,

$\epsilon_1^1 = Y_1 - \widehat{Y^*}_1^{(1)}$

In Step 9.4, ϵ_i^k value is assigned to X data set for each cluster.

For example,

$X_{2,p}^{(.)(1)} = \epsilon_1^1$, p: number of parameters.

Step 10 After repeating Step 9.1 for the number of cluster times, Step 9.2, Step 9.3 and Step 9.4 are calculated. These steps are proceeded for the number of observations (n) times one by one.

Step 11 Step 9 and Step 10 are repeated for the number of particle times.

Step 12 Gbest, which is obtained for the training data set, and the equations in Step 9 are used to get ϵ_t for the test data set. The equations are given below

Step 12.1 $\widehat{Y_{test_i}}^{(j)} = X_{test_{i,.}}^{(j)} . \beta_{test}{}^{T(j)}_{i,.}$ (5.7)

Step 12.2 $\widehat{Y_{test}}^{*}{}_{i} = \widehat{Y_{test_i}}^{(j)} . \mu_{test}{}^{T(j)}_{i,.}$ (5.8)

Step 12.3 $\epsilon_{test_i} = Y_{test_i} - \widehat{Y_{test}}^{*}{}_{i}$ (5.9)

Step 12.4 $X_{test_{i+1,p}}^{(.)} = e_{test_i}$ (5.10)

$\beta_{test}{}^{T(j)}_{i,.}$ are the coefficients that are obtained as gbest values in Step 9, Step 10, and Step 11 ; $\mu_{test}{}^{T(j)}_{i,.}$, are the the degree of memberships of the objects which are calculated in the Step 4; Y_{test_i} are the original time series values; $\widehat{Y_{test}}^{*}{}_{i}$ are the forecasts.

Step 12.1 is repeated for the number of cluster times, then Step 12.2, Step 12.3, and Step 12.4 are proceeded and these steps are repeated for each observation one by one.

Step 13 r_1 and r_2 are randomly generated from the standard normal distribution and new positions and velocities are updated using the formulas below.

$velocity_{id}^{k+1} = velocity_{id}^{k} + c_1 . r1_1^k . \left(pbest_{id}^k - p_{id}^k\right) + c_2 . r2_1^k . (gbest^k - p_{id}^k)$ (5.11)

$P_{id}^{k+1} = P_{id}^{k} + velocity_{id}^{k+1}$ (5.12)

Step 14 Pbest and gbest values are updated using the fitness value.

Step 15 The steps 9, 10, 11, 12, 13, and 14 are repeated for the number of iteration times.

Step 16 Lastly, by using the equations below, gbest value ($\beta_{test}{}^{T(j)}_{i,.}$) is used to forecast future values of the time series.

$\widehat{Y_{test_i}}^{(j)} = X_{test_{i,.}}^{(j)} . \beta_{test}{}^{T(j)}_{i,.}$ (5.13)

$\widehat{Y_{test}}^{*}{}_{i} = \widehat{Y_{test_i}}^{(j)} . \mu_{test}{}^{T(j)}_{i,.}$ (5.14)

As a result, $\widehat{Y_{test}}^{*}{}_{i}$ values are the predictions that we want to forecast.

CHAPTER 6

APPLICATIONS

To evaluate the performance of the proposed method, 12 real world time series data are analyzed by using R, statistical programming language [57]. Stock exchange forecasting problems have been commonly studied by researchers. Therefore, most of the data sets we used are stock exchange data sets. The first data set is Australian beer consumption data. The objects of this data set are quarterly observed from 1956 to 1994. The next 5 data sets are from Turkey's stock exchange market index (BIST100). The elements of BIST100 data sets are daily observed between 2009 and 2013. The last six data sets are from Taiwan's stock exchange market from 1999 to 2004. These data sets are chosen to be able to compare the performance of the proposed method with the other methods which used the same data sets previously. Methods are evaluated by using root mean squared error (RMSE) and mean absolute percentage error (MAPE) given in Equation (6.1) and Equation (6.2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_t - \hat{x}_t)^2} \tag{6.1}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{x_t - \hat{x}_t}{x_t}\right| \tag{6.2}$$

The number of observations of the original data sets, the number of observations of the test data sets, the number of lags for autoregressive model, the number of lags for moving average model and the number of clusters used are listed in Table 6.1.

The applications will start with Australian Beer Consumption (ABC) data set. Second, Istanbul stock exchange data sets year by year will be performed. Lastly, Taiwan stock exchange data sets from 1999 to 2004 will be used to evaluate the performance of the proposed method.

Table 6.1 Summary of data sets and selections of the parameters for the simulation

No	Series/Year	Number of Observations	Order of AR	Order of MA	Number of Cluster	ntest
1	ABC	147	1-10	1-2	2-10	16
2	BIST100/2009	103	1-5	1-2	2-5	7,15
3	BIST100/2010	104	1-5	1-2	2-5	7,15
4	BIST100/2011	106	1-5	1-2	2-5	7,15
5	BIST100/2012	106	1-5	1-2	2-5	7,15
6	BIST100/2013	106	1-5	1-2	2-5	7,15
7	TAIEX/1999	266	1-4	1-2	2-5	45
8	TAIEX/2000	271	1-4	1-2	2-5	47
9	TAIEX/2001	244	1-4	1-2	2-5	43
10	TAIEX/2002	248	1-4	1-2	2-5	43
11	TAIEX/2003	249	1-4	1-2	2-5	43
12	TAIEX/2004	250	1-4	1-2	2-5	45

6.1 Application of Australian Beer Consumption (ABC) Data

In the first application, Australian beer consumption data set is taken into consideration. This data set consists of 148 observations, which are quarterly observed from 1956 to 1994. The line plot of the ABC data set is given in Figure 6.1.

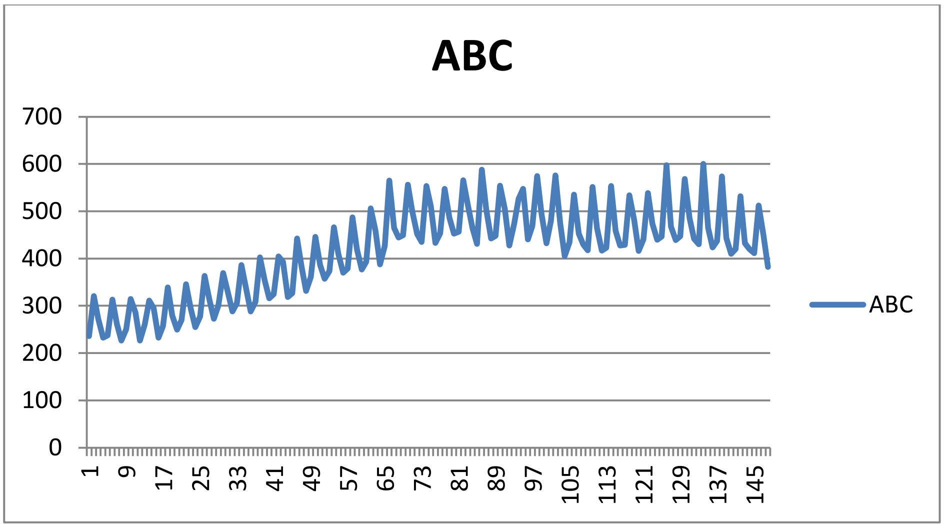

Figure 6.1 ABC data set

In order to compare the results of the proposed method, Seasonal Autoregressive Integrated Moving Average (SARIMA) model, Winter`s Exponential Smoothing method, Multilayer Perceptron Artifical Neural Network (MLP-ANN), Adaptive Neuro Fuzzy Inference System (ANFIS), and Modified Adaptive Neuro Fuzzy Inference

System (MANFIS) values are taken from the study of Egrioglu and et al. [58] and they are compared with the proposed method.

The algorithm searched for the best model when the number of clusters varied from two to ten, the number of lags for AR part varied one to ten, and the number of lags for MA part varied from one to two. The number of particles and the number of iterations are set to 35 and 100 respectively. Under these conditions the minimum RMSE and MAPE values are obtained when the number of clusters is five, the number of lags for AR part is seven, and the number of lags for MA part is one. Looking at the RMSE and MAPE values in Table 6.2, it is obvious that the minimum RMSE and MAPE values are obtained from the proposed method.

Table 6.2 Results obtained for ABC test data set when ntest=16

Test Data	**WMES**	**SARIMA**	**MLP-ANN**	**ANFIS**	**MANFIS**	**ARMA-T1FF**
430,50	453.91	452.72	453.88	446.71	445.23	442.82
600,00	575.22	578.29	557.81	553.73	575.63	554.44
464,50	502.32	487.71	497.52	482.07	494.07	477.33
423,60	444.73	446.28	437.39	434.19	434.56	443.47
437,00	459.66	456.77	449.01	438.55	444.69	422.46
574,00	582.48	583.51	569.01	559.01	575.42	571.21
443,00	508.64	492.13	471.08	472.52	481.28	463.85
410,00	450.31	450.36	424.33	427.57	414.44	410.47
420,00	465.4	461.01	448.87	445.01	430.31	420.02
532,00	589.74	588.96	560.04	562.94	565.18	551.34
432,00	514.96	496.77	447.01	459.14	452.05	436.37
420,00	455.89	454.64	408.64	416.16	392.14	390.61
411,00	471.15	465.46	428.11	431.71	419.33	398.69
512,00	597	594.71	537.69	544.98	536.88	517.62
449,00	521.28	501.67	438.43	444.31	446.32	430.76
382,00	461.46	459.17	420.58	426.01	406.64	408.27
RMSE	53.33	47.04	24.11	25.05	21.37	19.21*
MAPE	0.1072	0.0949	0.0476	0.0467	0.0401	0.0333*

The line plot of the forecasts of the proposed method and the original observations are given in Figure 6.2.

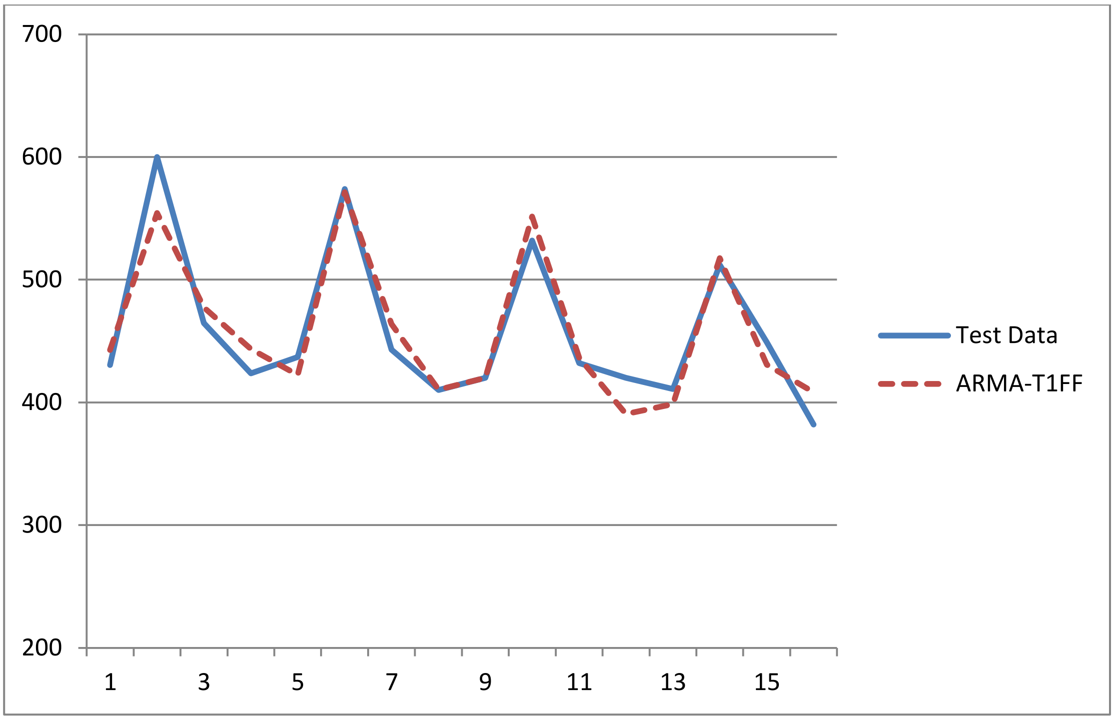

Figure 6.2 Line plot of forecasts and the test data for ABC data set (ntest=16)

6.2 Application of Istanbul Stock Exchange Data

In the applications of BIST100, the data sets between 2009 and 2013 are taken. The outcomes of ARIMA, Exponential Smoothing (ES), Multilayer Perceptron Artifical Neural Network (MLP-ANN), Type-1 Fuzzy Functions (FF), Fuzzy Time Series Network (FTS-N), and Type-1 Recurrent Fuzzy Functions (ARMA-T1FF) are compared and listed in the tables below. The outcomes of ARIMA, ES, MLP-ANN, FF, and FTS-N are taken from Bas and et al. [36].

The best results are obtained by using the Box-Jenkins procedure for the ARIMA procedure. Holt and Winter’s exponential smoothing methods are used and the best results are chosen. For the MLP-ANN method, hidden layer neurons and the number of inputs are specified from 1 to 5 and the best model is chosen. While performing FF, the number of clusters and the model order varies from 5 to 15 and from 1 to 5 respectively. The best outcomes are selected. For the results of FTS-N, the model order (p) varies from 1 to 5 and the number of clusters varies from 5 to 15. The best results are listed in the tables given in Chapter 6.2.

In order to obtain the best results for the proposed method, the number of clusters varies from 2 to 5, the number of lags for AR part varies from 1 to 5, and the number of lags for MA part varies from 1 to 2.

6.2.1 Istanbul Stock Exchange Data for 2009

Observations, for 2009 Istanbul stock exchange data, are collected daily from 1.2.2009 to 5.29.2009. The line plot of data is given in Figure 6.3.

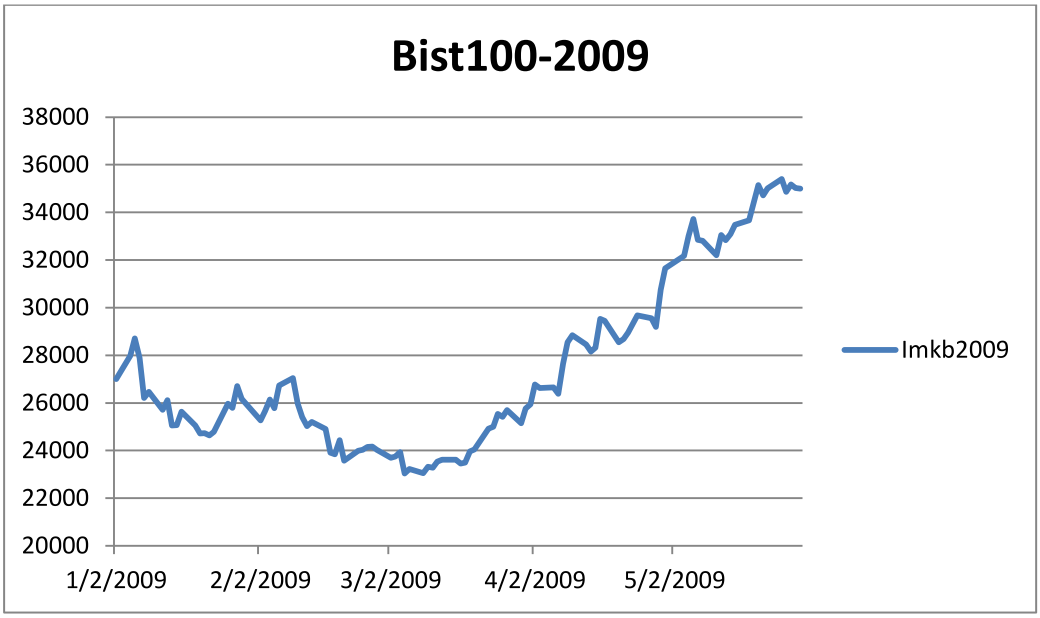

Figure 6.3 Line plot of Istanbul stock exchange data for 2009

The best results for the proposed method are obtained for 2 clusters, 2 lags for AR part, 2 lags for MA part, and the number of observations for ntest equals to 7. The results are given in Table 6.3. From Table 6.3, it can be seen that the proposed method outperforms the others in terms of RMSE and MAPE values.

Table 6.3 Results obtained for BIST2009 when ntest-7

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
34721	35140	35139.5968	35179.4149	35217.7252	34564.7192	34243.38
35015	34721	34720.5793	34803.4364	35038.9821	34835.5466	35080.42
35408	35015	35014.4794	35114.3825	35127.3924	34869.0244	35583.52
34861	35408	35407.6524	35406.9209	35733.4937	35216.5658	35150.24
35169	34861	34860.5983	34965.8835	35190.8515	35017.0478	35144.41
35021	35169	35168.7084	35245.7189	35401.2361	35014.0874	35172.88
35003	35021	35021.0203	35097.5295	35399.2696	35032.1007	35141.54
RMSE	344.91	344.93	325.1011	445.5147	266.6011	235.96*
MAPE	0.0087	0.0087	0.0083	0.0101	0.0058	0.00541*

The line plot of the forecasts of the proposed method and the original observations are given in Figure 6.4.

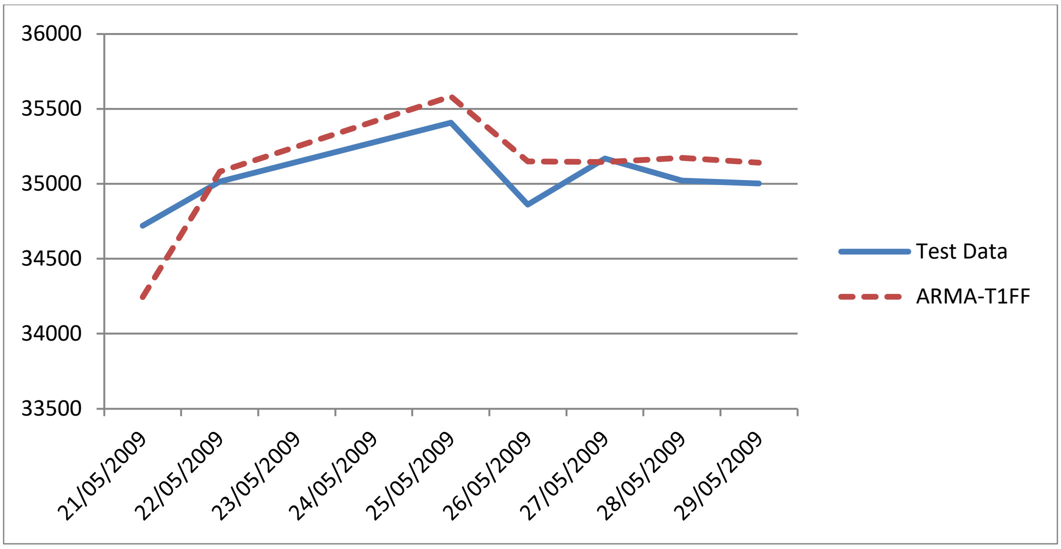

Figure 6.4 Line plot of forecasts of the proposed method and the test data for BIST100-2009 (ntest=7)

The proposed method reaches the best forecast values at 3 clusters, 2 lags for AR part, and 2 lags for MA part when ntest equals to 15. The results are listed in Table 6.4. It is again obvious that ARMA-T1FF method has the minimum RMSE and MAPE value. Thus, the proposed method shows the best forecasting performance.

Table 6.4 Results obtained for BIST2009 when ntest-15

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
32806	32843	32842.6108	33061.3111	33178.6227	33356.2365	33141.94
32203	32806	32805.7200	33159.0326	33102.0975	32907.7447	32324.43
33043	32203	32203.0606	32531.0429	32618.4834	32452.1009	31989.47
32829	33043	33043.2791	33338.9176	33314.4016	32699.2677	32932.85
33095	32829	32828.9002	33031.0147	33192.8211	32940.2821	33480.57
33485	33095	33094.9197	33356.0601	33350.4122	33084.7639	33603.53
33666	33485	33484.5196	33651.2007	33790.4828	33347.1154	33671.1
35140	33666	33666.0798	33794.6528	33960.6677	33661.6902	33851.14
34721	35140	35139.6985	34926.1540	35353.4672	34559.3495	34676.64
35015	34721	34720.5504	34547.2221	35065.6569	34999.7232	35140.3
35408	35015	35014.4997	34887.2350	35247.6082	35042.1766	35396.82
34861	35408	35407.6796	35108.2379	35720.9339	35257.9318	35186.02
35169	34861	34860.5605	34727.4115	35196.8809	35186.0090	34850.95
35021	35169	35168.7296	35002.2839	35444.5058	35103.9510	34719.4
35003	35021	35021.01015	34844.72488	35369.7569	35103.3650	34799.38
RMSE	540.21	540.2087	525.7264	534.1345	514.5627	478.1365*
MAPE	0.012	0.012	0.0114	0.0438	0.0112	0.0093*

The line plot of the forecasts of the proposed method and the original observations are given in Figure 6.5.

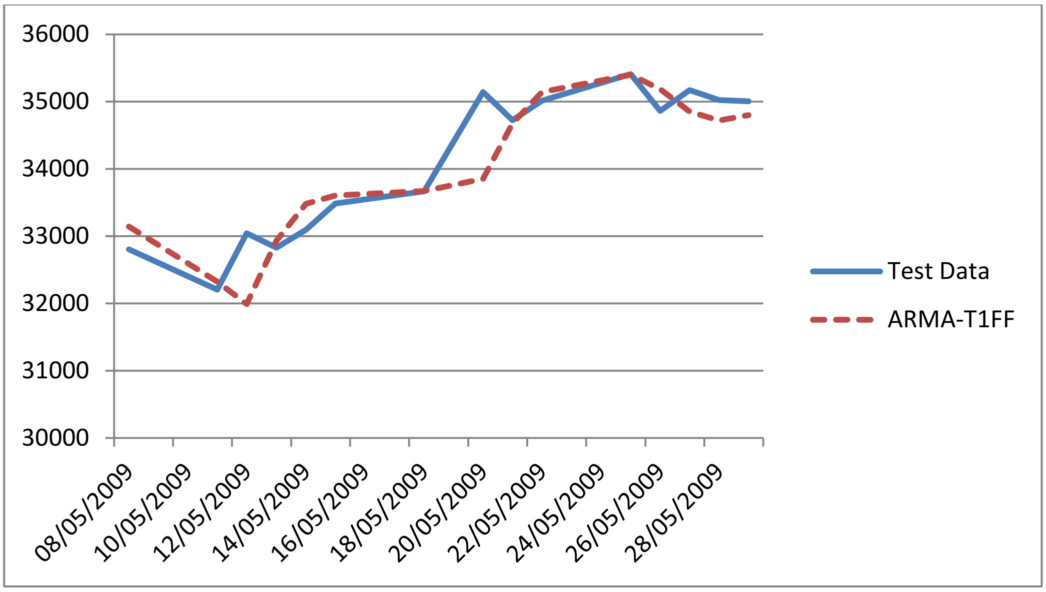

Figure 6.5 Line plot of forecasts of the proposed method and the test data for BIST100-2009 (ntest=15)

6.2.2 Istanbul Stock Exchange Data for 2010

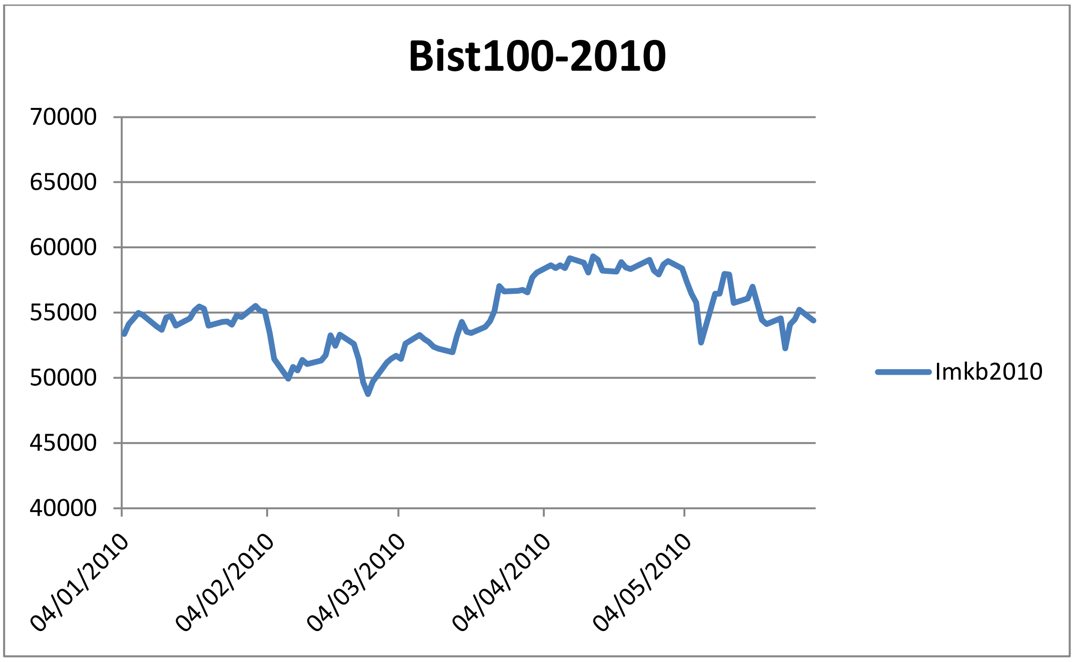

Figure 6.6 Line plot of Istanbul stock exchange data for 2010

Observations are daily determined between 4.1.2010 and 31.5.2010 for Istanbul stock exchange data set. The line plot of the observations is given in Figure 6.6.

For the BIST2010 data set, ARMA-T1FF with 3 clusters, 2 lags for AR part, 2 lags for MA part and the alternative methods are compared, the best result, in terms of RMSE value, is obtained from fuzzy time series network (FTS-N) when ntest equals to 7. However, in terms of both MAPE and RMSE values, the proposed method has the second best result. The line plot of the forecasts obtained by ARMA-T1FF and test data are given in Figure 6.7.

Table 6.5 Results obtained for BIST100-2010 when ntest=7

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
54112	54450	54519.9728	54249.1522	54349.0364	53724.46	54882.7
54558	54112	54123.0381	54460.1322	54031.7912	54316.22	54471.58
52257	54558	54545.9726	54750.9949	54601.6229	54515.04	54235.21
54104	52257	52321.1798	53127.2583	52430.3252	53164.05	52663.39
54498	54104	54053.5790	54013.4820	54110.8198	53774.49	54454.25
55234	54498	54485.5036	54714.6329	54561.3348	54475.41	54378.89
54385	55234	55212.7243	55031.5905	55099.6774	55032.93	55091.35
RMSE	1221	1208.1	1077.4	1179.9	1049.5*	1057.097
MAPE	0.0183	0.0185	0.0143*	0.0179	0.0159	0.0157

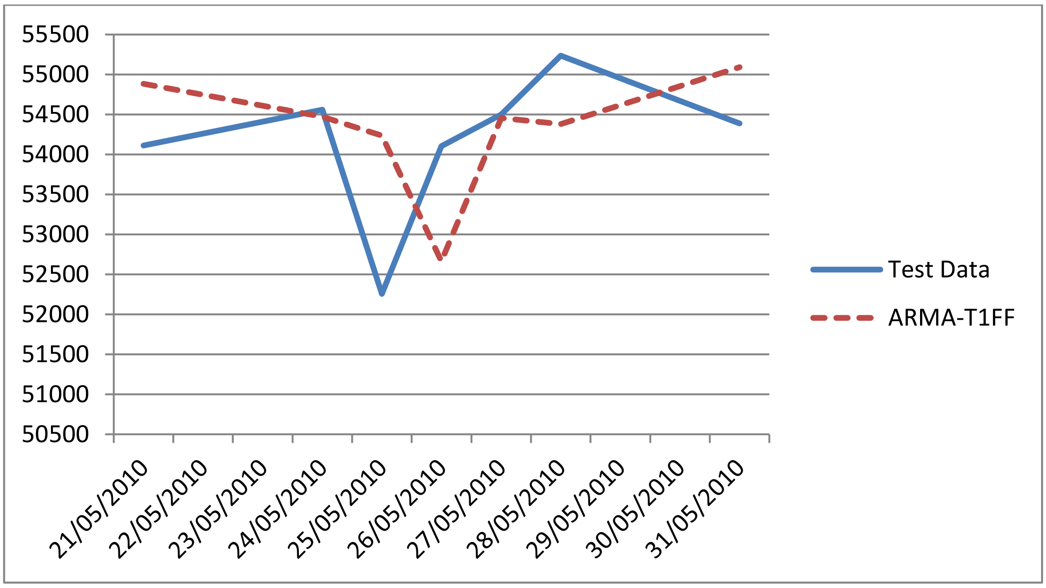

Figure 6.7 Line plot of forecasts of the proposed method and the test data for BIST100-2010 (ntest=7)

For BIST100-2010, ARMA-T1FF method achieved the best performance when the number of clusters is 5, the number of lags for AR part is 1, and the number of lags for MA part is 1, and ntest equals to 15. It is obvious from Table 6.6 that the minimum RMSE and MAPE values are obtained from the proposed method.

Table 6.6 Results obtained for BIST100-2010 when ntest=15

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
56448	52687	52687.0376	52671.4839	51770.5326	54619.4282	56147.56
56462	56448	56447.9172	56424.4092	56908.1789	55442.8972	56265.68
57976	56462	56461.7197	56438.4080	56108.9606	55794.0364	55005.73
57930	57976	57976.4266	57902.0523	57735.3283	56291.3273	56693.63
55748	57930	57929.6510	57859.4106	57827.3571	56251.8566	56421.07
56071	55748	55747.9280	55718.0677	54773.4417	56069.5756	56587.04
56978	56071	56071.0828	56045.3497	55595.0758	55922.8268	58431.59
54450	56978	56978.2300	56949.4656	56677.4691	56545.4899	55571.79
54112	54450	54449.9356	54397.9510	53543.0067	55431.5277	52984.81
54558	54112	54111.6074	54057.7081	53788.6813	54495.6420	54419
52257	54558	54558.1501	54507.2257	54475.0417	54629.5540	53166.65
54104	52257	52257.1206	52274.9258	51697.3757	53634.9310	51463.42
54498	54104	54103.4593	54049.6925	54377.8428	53584.4442	54183.15
55234	54498	54497.9413	54446.4890	54445.1352	53960.8837	54071.15
54385	55234	55233.6638	55194.7477	55221.3386	54360.6331	55612.92
RMSE	1611.5	1611.5	1603	1852	1357.4	1332.159*
MAPE	0.0220	0.022	0.0220	0.0264	0.0202	0.019*

The actual values and the forecasts are drawn in Figure 6.8.

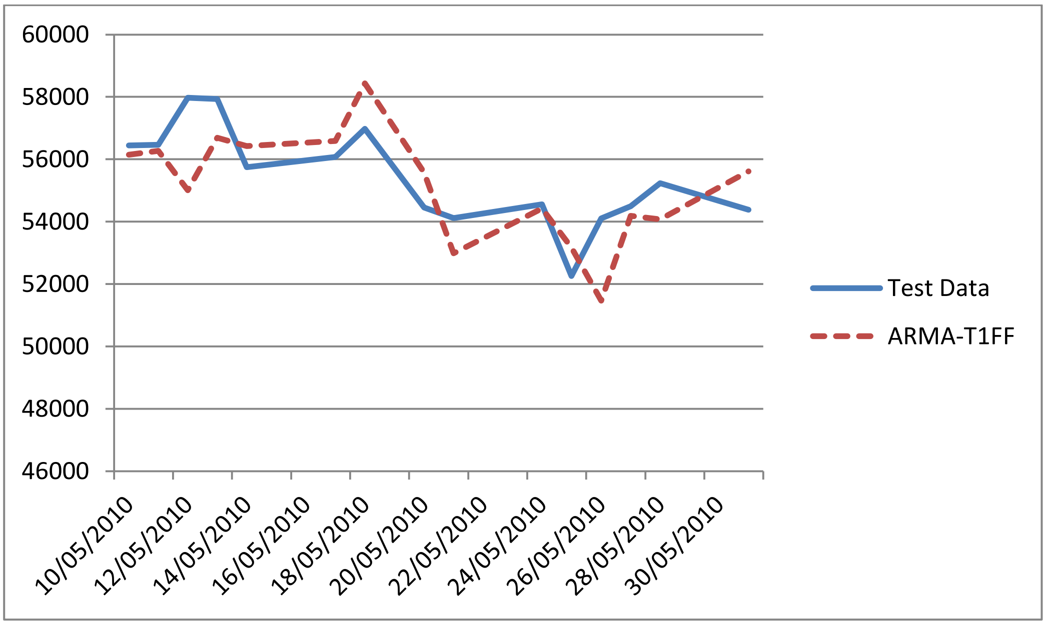

Figure 6.8 Line plot of forecasts of the proposed method and the test data for BIST100-2010 (ntest=15)

6.2.3 Istanbul Stock Exchange Data for 2011

Observations are daily determined between 3.1.2011 and 31.5.2011 for Istanbul stock exchange data set. The line plot of the observations is given in Figure 6.9.

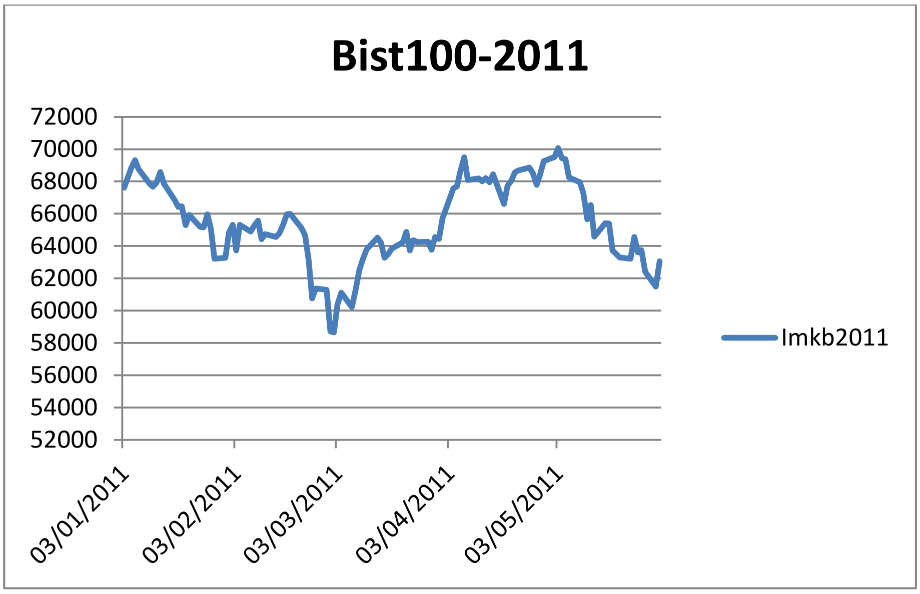

Figure 6.9 Line plot of Istanbul stock exchange data for 2011

In the application of BIST2011, ARMA-T1FF has the best forecasting results when the number of clusters is 5, the number of lags for AR is 3, the number of lags for MA is 1 and ntest equals to 7. The outcomes are listed in Table 6.7. When the outcomes are compared with the other methods, the proposed method has better results than the others.

Table 6.7 Results obtained for BIST2011 when ntest=7

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
63210	63299	63300.0961	63005.5729	63466.9715	63912.8711	64884.84
64561	63210	63210.1907	63518.9317	63119.9928	64013.4641	64650.51
63609	64561	64556.7450	64354.6129	64592.1226	64045.7269	63935.44
63755	63609	63612.5739	63558.6571	63495.1485	63348.6914	63559.22
62407	63755	63754.8581	63924.8197	64037.9655	63184.7404	62723.25
61492	62407	62412.0619	62532.2333	62458.0164	63006.9784	62175.9
63046	61492	61494.8806	62142.8242	61844.8620	62827.6759	62834.91
RMSE	1057.6	1057	919.9204	1083.2	765.07	714.1724*
MAPE	0.0144	0.0144	0.0128	0.0153	0.0105	0.0079*

The line plot of actual values and the forecasts are given in the Figure 6.10.

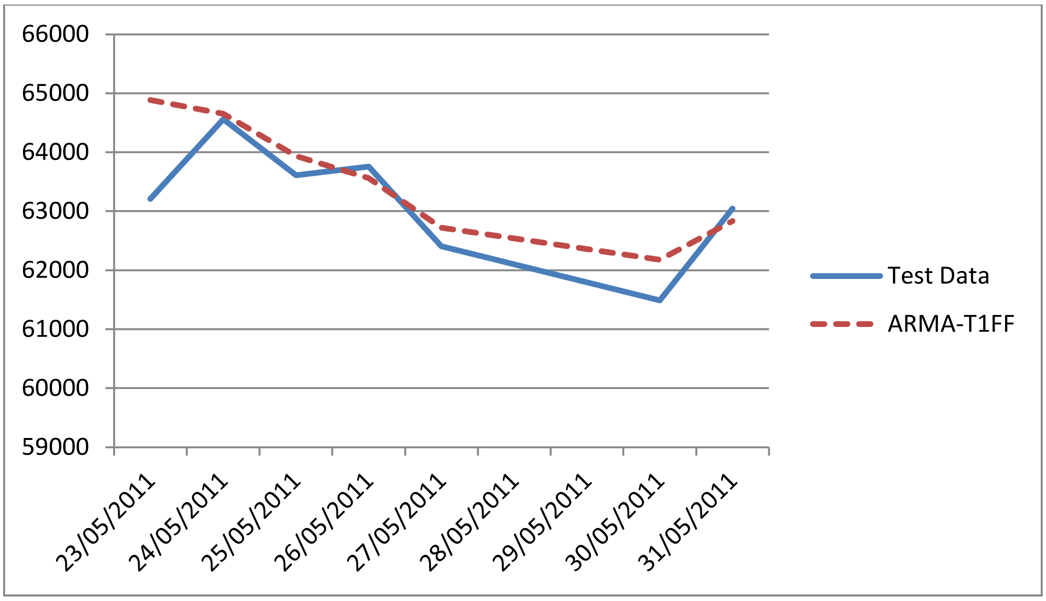

Figure 6.10 Line plot of forecasts of the proposed method and the test data for BIST100-2011 (ntest=7)

The outcomes for BIST100-2011 when ntest equals to 15 are shown in Table 6.8. In order to get the best performance from the proposed method, the number of clusters is taken as 3, the number of lags for AR is taken as 2, and the number of lags for MA is taken as 1. Looking at the RMSE and MAPE values, the best outcomes are obtained from FTS-N, whereas the second best outcomes are obtained from the proposed method.

Table 6.8 Results obtained for BIST2011 when ntest=15

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
67260	67956	67956.4910	67986.4979	67989.915	67256.2727	67130.36
65643	67260	67260.0415	67387.0789	67104.3432	66767.8572	66591.77
66535	65643	65643.4596	65827.6542	65662.2549	65618.5737	65451.57
64585	66535	66535.0640	66423.4340	66567.4146	65591.3056	65160.04
65418	64585	64585.3642	64643.7671	64011.6970	64776.3059	64523.64
65385	65418	65417.8583	65291.1844	65548.5084	64607.1340	64096.48
63733	65385	65385.3824	65105.5561	64751.7222	64865.8440	64412.71
63299	63733	63733.2822	63647.0510	63734.6339	63912.4009	63559.76
63210	63299	63298.6121	63615.3071	63500.6678	63120.6522	62597.67
64561	63210	63209.8865	63782.2304	63121.8594	62925.9354	62356.36
63609	64561	64561.2999	64968.5387	64690.3811	63661.8660	63079.91
63755	63609	63609.3804	63569.8332	63565.9730	63571.0872	63262.51
62407	63755	63755.3392	64074.4171	64245.5920	63339.1152	62956.39
61492	62407	62407.5197	62966.4493	62360.5299	62617.4991	62324.37
63046	61492	61491.7777	62465.2460	61782.2044	61648.5591	61212.24
RMSE	1129.6	1129.7	1095.7	1145.6	916.5411*	1017.41
MAPE	0.015	0.015	0.0146	0.0156	0.0121*	0.0134

The line plot which includes the actual values and the outcomes of the proposed method are given in Figure 6.11.

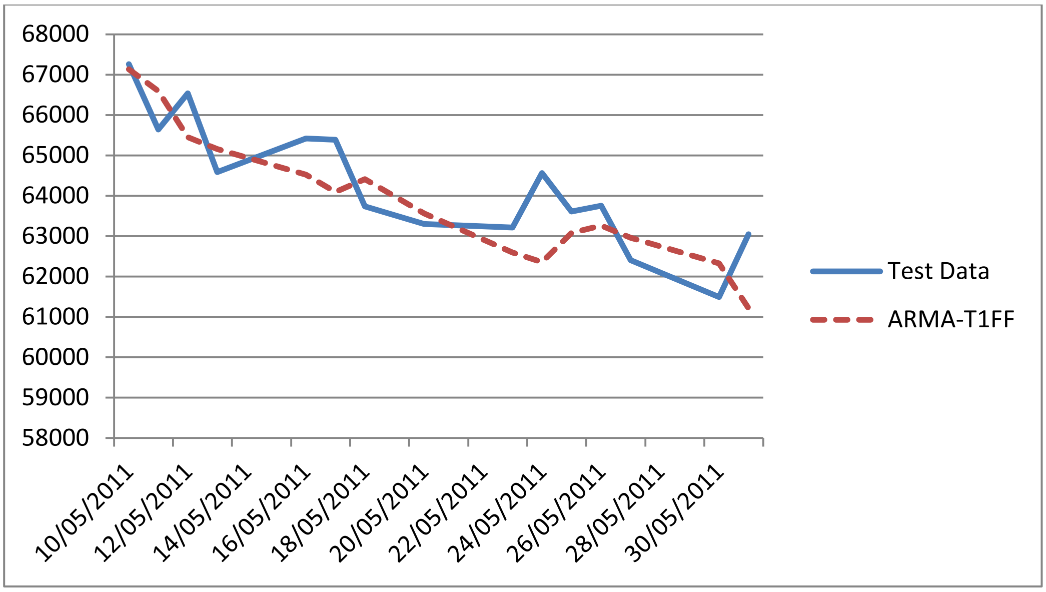

Figure 6.11 Line plot of forecasts of the proposed method and the test data for BIST100-2011 (ntest=15)

6.2.4 Istanbul Stock Exchange Data for 2012

Observations are daily collected from 2.1.2012 to 31.5.2012 for Istanbul stock exchange data set. The line plot of the observations is given in Figure 6.12.

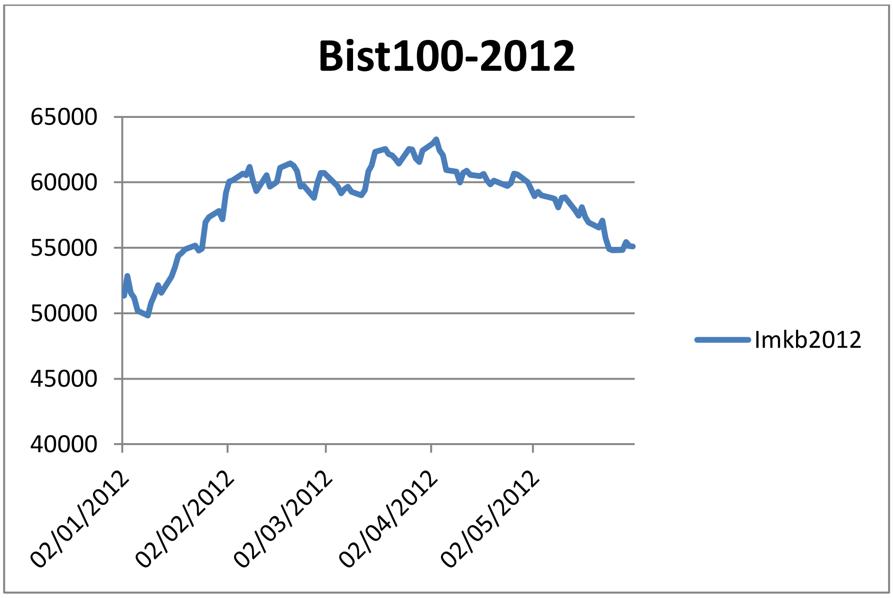

Figure 6.12 Line plot of Istanbul stock exchange data for 2012

For the proposed method, 4 clusters, 3 lags for the AR part, and 2 lags for the MA part are specified. For the BIST100-2012 data set, the optimum results are obtained when ntest equals to 7 is shown in Table 6.9. It can be seen that the best forecasts are obtained from the proposed method in terms of RMSE value and the best forecasts are obtained from ARIMA, ES, and FTS-N in terms of MAPE value.

Table 6.9 Results obtained for BIST2012 when ntest=7

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
55734	57079	57079.43	57327.82	57613.6733	56543.5124	56252.41
54917	55734	55734.45	55898.05	56208.7363	55914.2919	54934.57
54810	54917	54916.68	55115.84	55545.8026	54972.0851	54262.19
54844	54810	54809.95	55051.55	55577.9826	54633.4822	54414.4
55450	54844	54843.93	55094.09	55556.8610	54617.8677	54486.92
55125	55450	55449.46	55732.83	56044.2703	54993.1450	54954.94
55099	55125	55125.32	55353.34	55689.9261	55002.5204	54478.25
RMSE	650.56	650.7387	774.6103	1034.2	590.3545	547.13*
MAPE	0.0084*	0.0084*	0.0111	0.0162	0.0084*	0.0085

The line plot of the actual values and the forecasts of the proposed method are given in Figure 6.13.

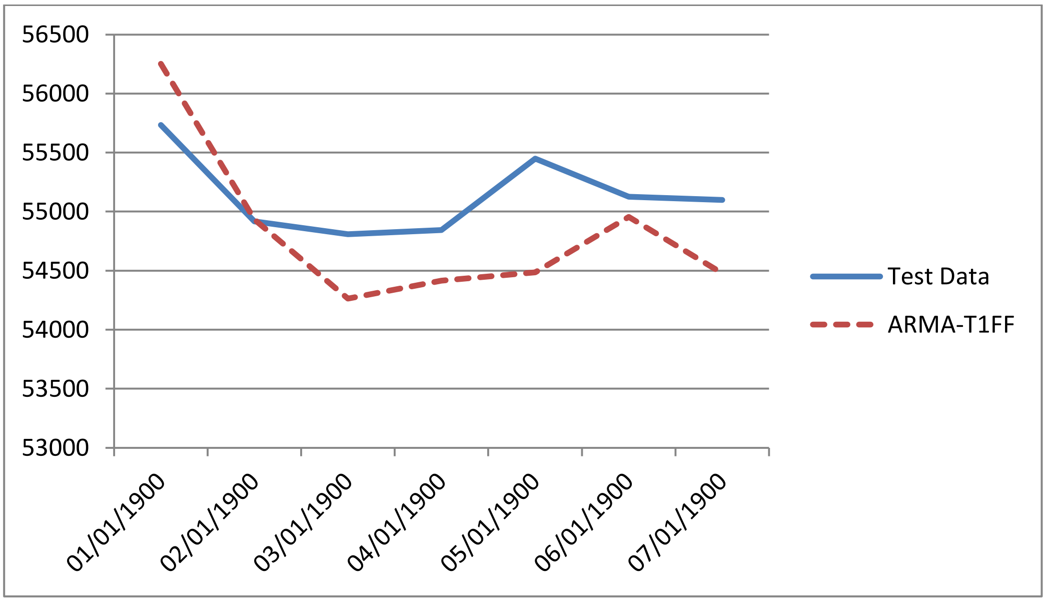

Figure 6.13 Line plot of forecasts of the proposed method and the test data for BIST100-2012 (ntest=7)

For the BIST100-2012 data set, the parameters of the proposed method, the number of clusters, the number of lags for the AR part, and the number of lags for the MA part are taken as 3,2, and 1 respectively when ntest equals to 15. The minimum RMSE and

MAPE values among the methods in the Table 6.10 are obtained from the proposed method.

Table 6.10 Results obtained for BIST2012 when ntest=15

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
58873	58821	58820.75	59066.53	59213.7731	58263.3548	58256.19
57854	58873	58872.62	59084.72	59269.9821	58520.0566	58303.12
57453	57854	57853.87	58090.3455	58184.8034	57910.4045	57465.81
58101	57453	57453.05	57737.36	57895.7200	57352.4403	57116.75
57331	58101	58100.84	58396.97	58637.9425	57625.5809	57618.03
56936	57331	57331.18	57603.89	57869.0021	57352.9884	57099.44
56540	56936	56935.76	57240.52	57463.1991	56874.2661	56555.45
57079	56540	56539.64	56855.13	57239.9156	56509.5018	56063.19
55734	57079	57079.46	57421.46	57781.0406	56717.8621	56461.83
54917	55734	55734.35	56014.45	56433.7639	56055.8249	55329.15
54810	54917	54916.62	55218.22	55638.0920	55140.4222	54546.99
54844	54810	54809.94	55143.91	55762.3794	54823.2672	54533.63
55450	54844	54843.93	55185.19	55733.9022	54811.1181	54731.61
55125	55450	55449.5	55823.22	56244.4720	55193.6584	55193.92
55099	55125	55125.29	55452.13	55899.0899	55181.5927	55460.04
RMSE	620.7892	620.829	783.3547	1037.6	581.71	529.69*
MAPE	0.0088	0.0088	0.0117	0.0161	0.0087	0.0076*

The line plot of the actual values and the forecasts are given in Figure 6.14.

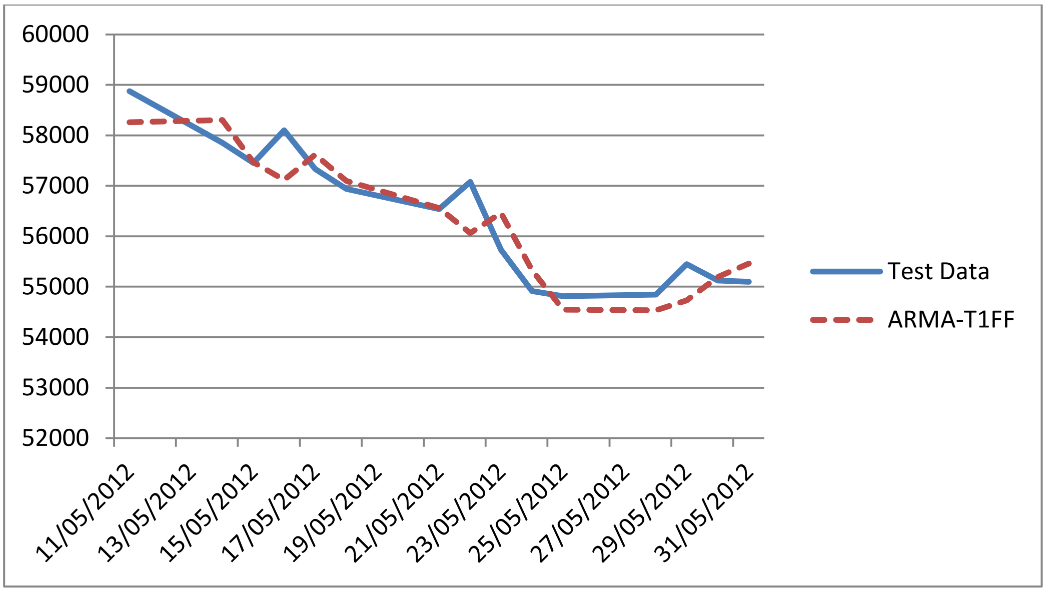

Figure 6.14 Line plot of forecasts of the proposed method and the test data for BIST100-2012 (ntest=15)

6.2.5 Istanbul Stock Exchange Data for 2013

Observations are daily collected from 2.1.2013 to 31.5.2013 for Istanbul stock exchange data set. The line plot of the observations is given in Figure 6.15.

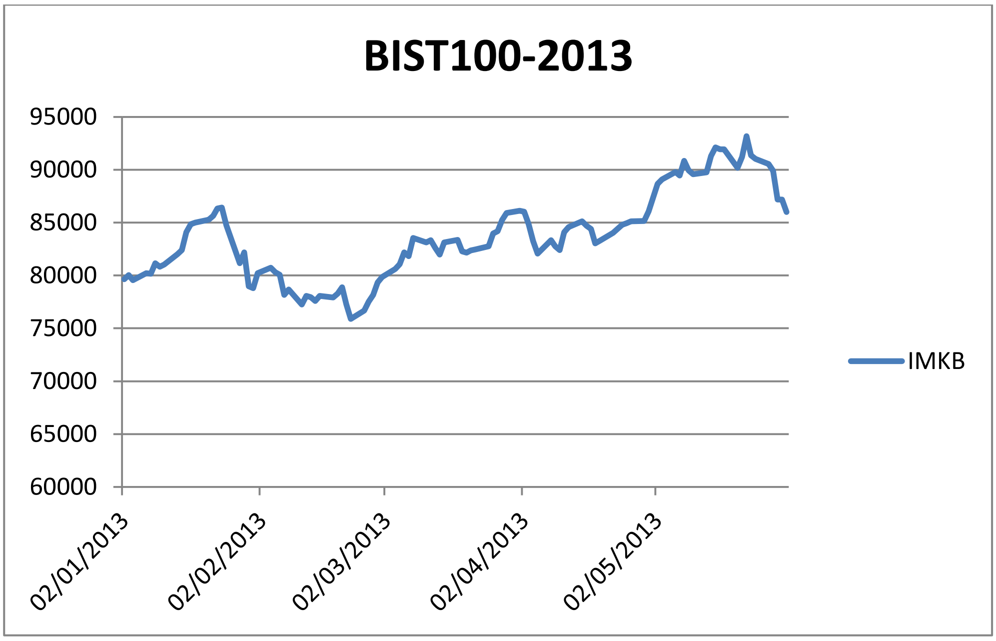

Figure 6.15 Line plot of Istanbul stock exchange data for 2013

For the BIST100-2013 data set, the best performance of the proposed method is observed when the number of clusters is 5, the number of lags for AR is 2, and the number of lags for MA is 1. When the proposed method is compared with the other methods, the minimum RMSE and MAPE values are obtained from the proposed method.

Table 6.11 Results obtained for BIST2013 when ntest=7

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
91351	93179	93179	91863.3837	93690.6620	91237.5541	91132.14
91016	91351	91351	91859.8407	91551.1456	90535.9969	91013.62
90547	91016	91016	91201.0054	91026.1888	89889.0609	89887.33
89916	90547	90547	90361.1494	90740.4976	89507.4975	89455.98
87175	89916	89916	90306.7637	90092.2830	88989.4408	89049.92
87170	87175	87175	86568.6308	86997.2777	87064.3989	87459.45
85990	87170	87170	86566.4256	86875.9782	86413.8639	86082.61
RMSE	1361.6	1361.6	1314.9	1511.6	786.13	783.9803*
MAPE	0.0116	0.0116	0.0109	0.0131	0.0065	0.0058*

The line plot of the forecasts and the actual values are given in Figure 6.16.

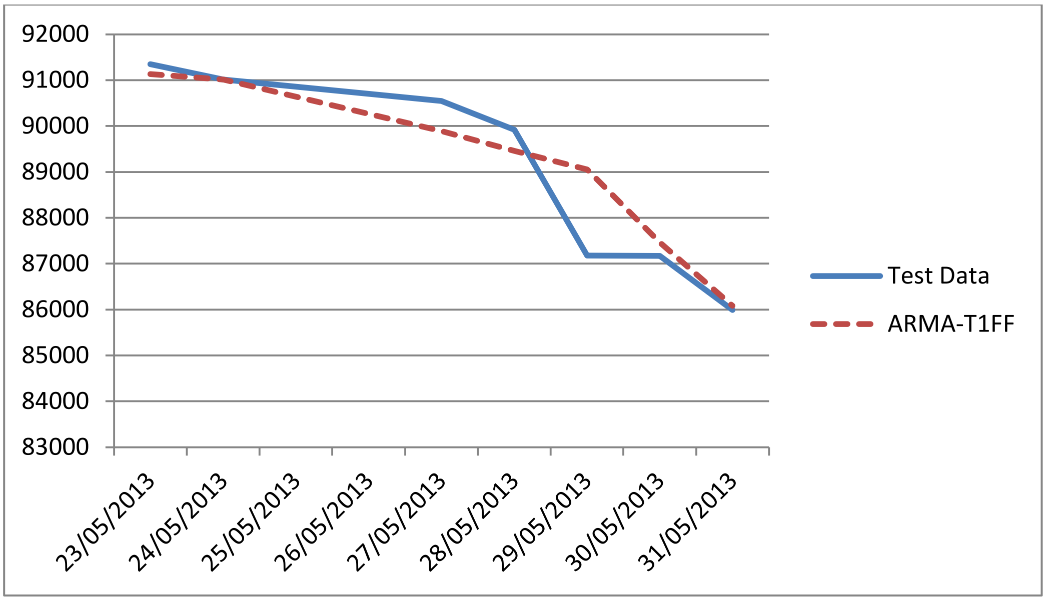

Figure 6.16 Line plot of forecasts of the proposed method and the test data for BIST100-2013 (ntest=7)

For the BIST100-2013 data set, the best performance of the proposed method is reached when the number of clusters is 2, the number of lags for AR is 1, the number of lags for MA is 1 and ntest equals to 15. When the results are compared, the best forecasts are observed by the proposed method in terms of RMSE. In terms of MAPE, however, the best forecasts are observed by both the proposed method and FTS-N.

Table 6.12 Results obtained for BIST2013 when ntest=15

Test Data	ARIMA	ES	MLP-ANN	FF	FTS-N	ARMA-T1FF
89765	89569	89568.93	89442.7721	89271.3321	89472.338	90698.97
91287	89765	89765.16	89691.9695	89372.3950	90876.544	90816.47
92112	91287	91286.96	91255.3193	91224.7881	90571.870	92388.22
91947	92112	92111.98	91926.1694	92010.3770	91438.080	93165.47
91925	91947	91946.76	91671.0437	91559.0460	91567.412	93011.24
90165	91925	91924.84	91666.6506	91268.3629	91575.096	92631.52
91191	90165	90165.35	89848.3890	89383.7758	90315.959	91031.24
93179	91191	91190.69	91113.1438	90378.4375	90726.661	91964.58
91351	93179	93178.63	92989.5045	92726.4317	92195.589	93229.75
91016	91351	91351.62	90946.0403	90939.8585	91328.657	91016.63
90547	91016	91016.35	90802.1309	90135.1729	90894.023	89626.58
89916	90547	90546.81	90351.8423	89724.7111	90388.258	89066.75
87175	89916	89916.48	89741.0422	89296.9503	89837.628	88445.05
87170	87175	87174.87	86848.7707	86321.8485	87750.943	85757.44
85990	87170	87170.23	87165.4927	86173.7953	87223.693	85778.08
RMSE	1268.7	1268.7	1232.5	1278.6	1207.9	1159.598*
MAPE	0.0109	0.0109	0.0107	0.0108	0.0106*	0.0106*

The line plot of the actual values and the forecasts are given in Figure 6.17.

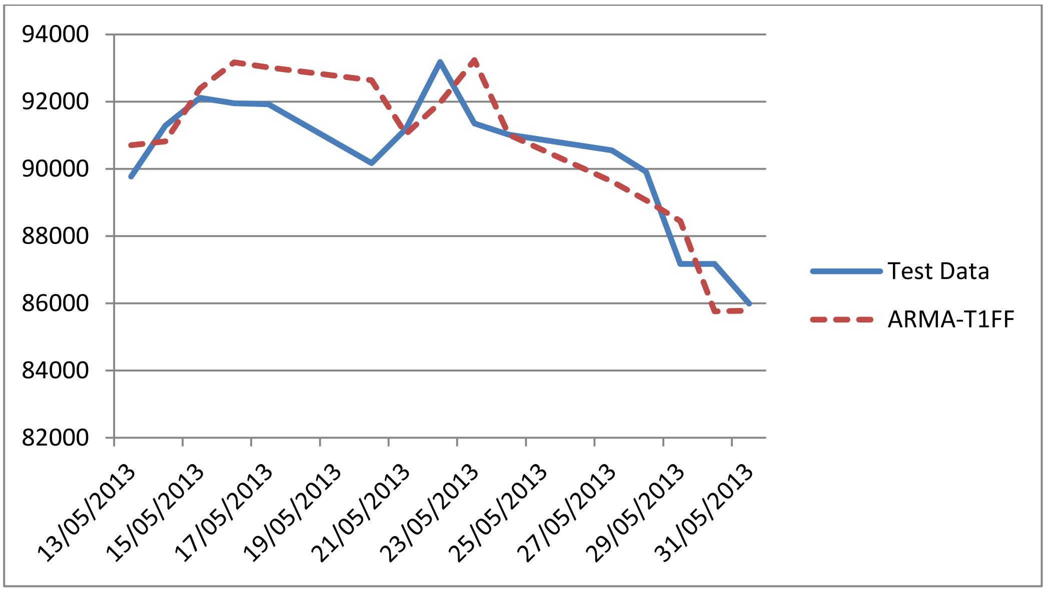

Figure 6.17 Line plot of forecasts of the proposed method and the test data for BIST100-2013 (ntest=15)

For Istanbul stock exchange data sets, the results given above show that the proposed method outperforms the other methods overall in terms of RMSE and MAPE values.

6.3 Application of Taiwan Stock Exchange Data Set

In order to evaluate the performance of the proposed method for Taiwan stock exchange data set the following methods are chosen: Chen(1996), Chen and Chang (2010), Chen and Chen(2011), and Chen et al. (2012). The results of the methods in Table 6.13 are taken from the paper of Bas and et al. [36].

Table 6.13 Results obtained for TAIEX

			RMSE				
Methods	**1999**	**2000**	**2001**	**2002**	**2003**	**2004**	**Mean**
Chen (1996)	120	176.32	147.84	101.18	74.46	84.28	117.34
Chen and Chang (2010)	101.97	129.42	113.33	66.82	53.51	60.48	87.58
Chen and Chen (2011)	112.47	123.62	115.33	71.01	58.06	57.73	89.7
Chen et al. (2012)	99.87	119.98*	114.47	67.17	52.49	52.27*	84.37
ANFIS (1993)	101.16	137.02	114.72	65.99	57.04	61.36	89.54
MANFIS (2014)	101.94	124.92	112.47	62.57*	52.33*	53.66	84.64
ARMA-T1FF	98.33*	128.18	106.48*	65.14	52.38	53.78	84.05*

For Taiwan stock exchange data sets from 1999 to 2004, the results obtained by using the proposed method are listed in Table 6.13. For 1999, the best forecasts are reached when the number of clusters is 3, the number of lags for AR part is 3, and the number of lags for MA part is 1. For 2000, the number of clusters is taken as 3, the model order for AR part is taken 1, and the model order for MA part is taken as 1. For 2001, the model reached the best forecasts when the number of clusters, lags for AR and MA part are 3,1, and 1 respectively. For 2002, the best model is obtained when the number of clusters is 2, the numbers of lags for AR and MA parts are 1 and 1, respectively. For 2003, the minimum RMSE value is obtained when the number of clusters is 2, the number of lags for AR part is 1, and the number of lags for MA part is 1. The best model for 2004 is reached when the number of clusters is 3, the number of lags for AR is 1, and the number of lags for MA part is 1. The number of iterations and the number of particles are set to 100 and 30 respectively for all the years. Under these conditions, the forecasting results from the proposed method and the other methods are compared in terms of the root mean squared errors. For 1999 and 2001, the best forecasting result is obtained from the proposed method. For 2002, 2003, and 2005, MANFIS outperforms the other methods. However, looking at the means of the years we see that the proposed method has the best forecasting result than others. The line plot of the proposed method and the other methods by years is given in Figure 6.18.

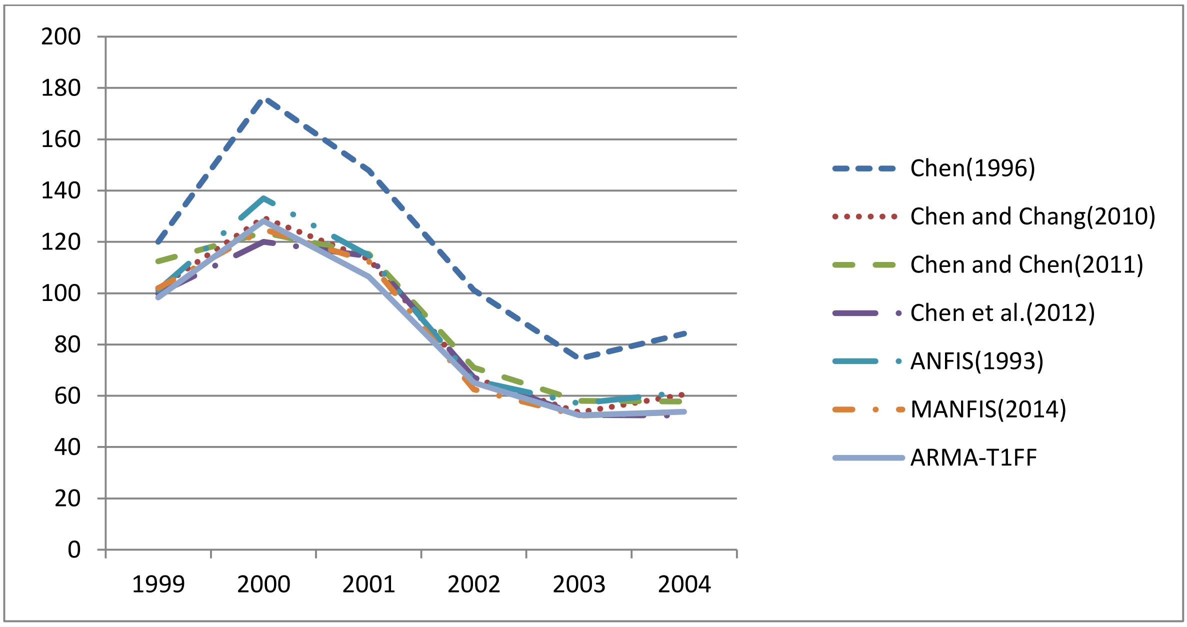

Figure 6.18 Line plot of forecasts of the proposed method and the other methods from 1999 to 2004.

CHAPTER 7

RESULTS AND DISCUSSION

Fuzzy inference systems are widely used for time series forecasting problems which mostly rely on uncertain data. These systems are established on data and fuzzy cluster theory. Most of the fuzzy inference systems are rule based. Since it is difficult to define rules, Turksen [51] has introduced a novel fuzzy inference system, type-1 fuzzy functions, which is free of rules. In this thesis, a new method which uses type-1 fuzzy functions as autoregressive and moving average model terms is proposed. The proposed method is the first recurrent fuzzy functions approach. In order to estimate the coefficients of the model, particle swarm optimization method is preferred. The proposed approach has the following advantages:

- Unlike most of the fuzzy inference systems, proposed method does not need rules to be defined.
- The assumptions of classical time series forecasting methods are not needed for the proposed method. In other words, for recurrent type-1 fuzzy functions, there is no assumption on time series.
- Since the function that is to be optimized is not a derivative, particle swarm optimization algorithm is preferred to estimate the coefficients of the model. The advantage of particle swarm optimization is that it is less likely to stick around the local optimum.
- Recurrent type-1 fuzzy functions approach is the first method that uses recurrent learning approach.
- The number of inputs is fewer than the other proposed methods because of the contribution of the moving average model.
- The proposed method gives better forecasting results than the most of the methods in literature.

When we look at the results of the applications, recurrent type-1 fuzzy functions has overall better forecasting results than other methods. For Australian Beer Consumption (ABC) data set, we see that the proposed method outperforms the other methods in terms of Root Mean Squared Errors (RMSE) and Mean Absolute Percentage Error (MAPE). For Istanbul Stock Exchange data set from 2009 to 2013, recurrent type-1 fuzzy functions approach, in terms of RMSE and MAPE, has better forecasting results most of the time. For TAIEX data set, the mean of the RMSE of the years is taken as the performance criteria of the models. Considering the mean of RMSE, the proposed method gives the best result. In summary, considering ABC, BIST100, and TAIEX data sets, recurrent type-1 fuzzy functions approach gives better forecasting results. Therefore, we are able to say that the proposed method gives satisfactory forecasting result for some time series.

REFERENCES

[1] Box, G.E.P and Jenkins, G.M. (1976). Time Series Analysis: Forecasting and Control, Holdan-Day, San Francisco, CA.

[2] Zadeh, L.A. (1965). “Fuzzy Sets”, Information and Control, 8:338-353.

[3] Zadeh, L.A., (1973). “Outline of a New Approach to the Analysis of Complex Systems and Decision Process”, IEEE Transactions on Systems, Man and Cybernetics, 3(1):28-44.

[4] Mamdani, E.H. and Assilian, S. (1975) “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller” , Int J Man Math Stat, 7(1):1-13

[5] Takagi, T. and Sugeno, M. (1985). “Fuzzy Identification of Systems and its Applications to modeling and control”, IEEE Transactions on Systems, Man and Cybernetics, 15(1):116-132.

[6] Jang, J.S.R (1993). “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, IEEE Transactions on Systems, Man and Cybernetics, 23(3):665-685

[7] Turksen, I.B., (2008). “Fuzzy Functions with LSE”, Applied Soft Computing, 8(3):1178-1188.

[8] Song, Q. and Chissom, B.S. , (1993a). “Fuzzy Time Series and its Models”, Fuzzy Sets Systems, 54:269-277.

[9] Song, Q. and Chissom, B.S. , (1993b). “Forecasting Enrollments with Fuzzy Time Series – Part I”, Fuzzy Sets Systems, 54:1-10.

[10] Song, Q. and Chissom, B.S. , (1994). “Forecasting Enrollments with Fuzzy Time Series – Part II”, Fuzzy Sets Systems, 62(1):1-8.

[11] Chen, S.M. (1996). “Forecasting Enrollments with Fuzzy Time Series”, Fuzzy Sets Systems, 81:311-319.

[12] Huarng, K. and Yu, H.K. (2006). “The Application of Neural Networks to Forecast Fuzzy Time Series”, Physica A, 363:481-491.

[13] Kuo, R.J., Chen, C.H., and Hwang, Y.C. (2001). “An Intelligent Trading Decision Support System Through Integration of Genetic Algorithm Based Fuzzy Neural Network and Artifical Neural Network , Fuzzy Sets and Systems”, 118:21-45.

[14] Chen, S.M. and Chung, N.Y. (2006). “Forecasting Enrollments Using High-Order Fuzzy Time Series and Genetic Algorithms”, International Journal of Intelligent Systems, 21:485-501.

[15] Kim, D. and Kim, C. (1997). “Forecasting Time Series with Genetic Fuzzy Predictor Ensemble”, IEEE Transactions on Fuzzy Systems, 5(4):523-535.

[16] Egrioglu, E. (2012). “A New Time Invariant Fuzzy Time Series Forecasting Method Based on Genetic Algorithm”, Advances in Fuzzy System, 2012.

[17] Bas, E., Uslu, V.R., Yolcu, U. and Egrioglu, E. (2014). “A Modified Genetic Algoirthm for Fuzzy Time Series to Find the Optimal Interval Length”, Applied Intelligence, 42(2):453-463.

[18] Egrioglu, E., Aladag, C.H., Yolcu, U., Uslu, V.R. and Basaran, M.A. (2009). “A new Approach Based on Artificial Neural Networks for High Order Multivariate Fuzzy Time Series”, Expert Systems with Applications, 36:10589-10594.

[19] Chen, S.M. and Tanuwijaya, K. (2011). “Multivariate Fuzzy Forecasting Based on Fuzzy Time Series and Automatic Clustering Techniques”, Expert Systems with Applications, 38:10594-10605.

[20] Jilani, T.A., Burney, S.M.A and Ardil, C. (2008). “Multivariate High Order Fuzzy Time Series Forecasting for Car Road Accidents”, International Journal of Computational Intelligence, 4:15-20.

[21] Huarng, K.H (2007). “A Multivariate Heuristic Model for Fuzzy Time-Series Forecasting”, IEEE Transactions on System, Man and Cybernetics B, 37(4):836-846.

[22] Chau, K.W. (2006). “Particle Swarm Optimization Training Algorithm for ANNs in Stage Prediction of Shing Mun River”, Journal of Hydrology, 329:363-367.

[23] Park, F.I., Lee, D.J., Song, C.K and Chun, M.G (2010). “TAIEX and KOSPI Forecasting Based on Two-Factor High-Order Fuzzy Time Series and Particle Swarm Optimization”, Expert Systems with Applications, 37: 959-967.

[24] Kuo, I.H., Horng, S.J., Kao, T.W., Lin, T.L., Lee, C.L. and Pan,Y. (2009). “An Improved Method for Forecasting Enrollments Based on Fuzzy Time Series and Particle Swarm Optimization”, Expert Systems with Applications, 36:6108-6117.

[25] Aladag, C.H., Yolcu, U., Egrioglu, E. and Dalar, A.Z. (2012). “A New Time Invariant Fuzzy Time Series Method Based on Particle Swarm Optimization”, Applied Soft Computing, 12:3291-3299.

[26] Huang, C.M., Huang, C.J. and Wang, M.L. (2005). “A Particle Swarm Optimization to Identifying the ARMAX Model for Short-Term Load Forecasting”, IEEE Transactions on Power Systems, 20(2):1126-1133.

[27] Catalao, J. P. S., Pousinho, H. M. I., and Mendes, V.M.F. (2011). “Hybrid Wavelet-PSO-ANFIS approach for Short-Term Wind Power Forecasting in Portugal”, IEEE Transactions on Sustainable Energy, 2(1):50-59.

[28] Chabaa, S., Zeroual, A., and Antari, J. (2009). “ANFIS Method for Forecasting Internet Traffic Time Series”, IEEE Microwave Symposium, 7-12 June 2009, Boston.

[29] Chang, B. (2008). “Resolving the Forecasting Problems of Overshoot and Volatility clustering using ANFIS Coupling Nonlinear Heteroscedasticity with Quantum Tuning”, Fuzzy Set and Systems, 159(23):3183-3200.

[30] Chen, B. and Ma, Z. (2009). "Short-Term Traffic Flow Prediction Based on ANFIS", ICCSN `09, International Conference on Communication Software and Networks, 27-28 February 2009, Hvar.

[31] Chen, D. and Zhang, J. (2005). "Time Series Prediction Based on Ensemble ANFIS", Proceddings of the Fourth International Conference on Machine Learning and Cybermetics, 3552-3556.

[32] Egrioglu, E. and Yolcu, U. (2014). "A new Adaptive Network Based Fuzzy Inference System for Time Series Forecasting", Aloy Journal of Soft Computing and Applications, 2:25-32.

[33] Beyhan, S. and Alci, M. (2010). "Fuzzy Functions Based ARX Model and New Fuzzy Basis Function Models for Nonlinear System Identification", Applied Soft Computing, 10:439-444.

[34] Aladag, C.H., Turksen, I.B., Dalar, A.Z., Egrioglu, E. and Yolcu, U. (2014). "Application of Type-1 Fuzzy Functions Approach for Time Series Forecasting", Turkish Journal of Fuzzy Systems, 5(1):1-9.

[35] Tseng, F.M., Yu, H.C. and Tseng, G.H. (2002). "Combining Neural Network Model with Seasonal Time Series ARIMA Model", Technological Forecasting and Social Change, 69:71-87.

[36] Bas, E., Egrioglu, E., Yolcu, U. and Aladag, C.H. (2015). "Fuzzy Time Series Network used to Forecast Linear and Nonlinear Time Series", Applied Intelligence, 43(2):343-355.

[37] Lee, Y.S. and Tong, L.I. (2011). "Forecasting Time Series Using a Methodology Based on Autoregressive Integrated Moving Average and Genetic Algorithm", Knowledge-Based Systems, 24(1):66-72.

[38] Chen, K.Y. and Wang, C.H. (2007). "A Hybrid SARIMA and Support Vector Machines in Forecasting the Production Values of the Machinery Industry in Taiwan", Expert System Applications, 32(1):254-264.

[39] Pai, P.F. and Lin, C.S., (2005). "A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting", Omega, 33(7):497-505.

[40] Zhang, G. (2003). "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model", Neurocomputing, 50:159-175.

[41] BuHamra, S., Smaoui, N. and Gabr, M. (2003). "The Box-Jenkins Analysis and Neural Networks: Prediction and Time Series Modeling", Applied Mathematical Modelling, 27(10):805-815.

[42] Jain, A. and Kumar, A.M., (2007). "Hybrid Neural Network Models for Hydrological Time Series Forecasting", Applied Soft Computing, 7:585-592.

[43] Yolcu, U., Aladag, C.H. and Egrioglu, E. (2013). "A New Linear&Nonlinear Artifical Neural Network Model for Time Series Forecasting", Decision Support Systems, 54(3):1340-1347.

[44] Zimmermann, H. J., (2010). "Fuzzy Set Theory", John Wiley&Sons, Inc. WIREs Computational Statistics, 2:317-332.

[45] Hathaway, R.J. and Bezdek, J.C. (1993). "Switching Regression Models and Fuzzy Clustering", IEEE Transactions on Fuzzy Systems, 1(3):195-204.

[46] Bezdek, J.C, Ehrlich, R. and Full, W. (1983). “FCM: The Fuzzy c-Means Clustering Algorithm”, Computers & Geosciences, 2(3):191-203.

[47] Jang, J.S.R., Sun, C.T. and Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, First Edition, Prentice Hall, NJ.

[48] Chennakesava, R.A. (2008). Fuzzy Logic and Neural Networks: Basic Concepts and Applications, New Age International Pvt Ltd Publishers, New Dellhi.

[49] Nyugen, H. T. and Wu, B. (2006). Fundamentals of Statistics with Fuzzy Data, Springer, The Netherlands.

[50] Fanelli, A. M., Pedrycz, W. and Petrosino, A. (2011). Fuzzy Logic and Applications, 9th International Workshop, WILF 2011, Trani, Italy, Springer, Berlin.

[51] Celikyilmaz, A. and Turksen, B. (2009). Modeling Uncertainty with Fuzzy Logic: with Recent Theory and Applications, Springer, Berlin.

[52] Knapp, R. B., Fuzzy Sets and Pattern Recognition, http://hci.sapp.org/lectures/knapp/fuzzy/fuzzy.pdf, 26 October 2014.

[53] Sugeno Fuzzy Models, http://www.bindichen.co.uk/post/AI/takagi-sugeno-fuzzy-model.html, 26 October 2014.

[54] Brockwell, P. J. and Davis, R.A. (2002). Introduction to Time Series and Forecasting, Second Edition, Springer, Newyork.

[55] Wei, W. W. S. (2006). Time Series Analysis: Univariate and Multivariate Methods, Second Edition, Pearson Education, Inc., USA.

[56] Kennedy, J. and Eberhart, R. (1995). “Particle Swarm Optimization”, Proceedings of the IEEE International Conference on Neural Networks, IEEE press, Piscataway, NJ, 1942-1948.

[57] R Core Team. R: A language and environment for statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org/, 7 May 2016.

[58] Egrioglu, E. and Aladag, C.H., Yolcu, U., Bas, E. (2014). “A New Adaptive Network Based Fuzzy Inference System for Time Series Forecasting”, Aloy Journal of Soft Computing and Applications, 2(1): 25-32.

APPENDIX-A

THE R CODES OF THE PROPOSED METHOD

The R codes of the proposed method are listed below. In the first algorithm, the codes of the evaluation function are given. In the second algorithm, the codes of the proposed method are given.

Algorithm 1

The codes of evaluation function for particle swarm optimization are given below.

Inputs of the evaluation function

x: Input matrix

y: Original time series values

B: coefficients of the regression equation

c: number of cluster

```
evalFunc<- function(x,y,B,c)
{
  fitness<-numeric(c)
  for (i in 1:c)
  {
    fitness[i]<- sum(  t(y -  x[,,i] %*% B[,i]) %*% (y -  x[,,i] %*% B[,i] ) ) #SSE
  }
  fit<-sum(fitness)
  return(list(fit))
}
```

Algorithm 2

R codes of recurrent type-1 fuzzy functions approach are given below.

Inputs of the proposed method

numberParticle: the number of particles

c: the number of clusters

da: data set

iterasyon: the number of iterations

lengthTest: the length of the test data set

p: the number of lags for AR part

```
T1FFPSOARMA <- function (numberParticle,c,da,iterasyon,lengthTest,p)
{
  da<-as.matrix(da)
  numberParam<-p+4
  da1<-embed(da,p)
  lengthTrain<-length(da1[,1]) - lengthTest
  training<-embed( da,p)[1:lengthTrain, ]
  constant<-rep(1,lengthTrain)
  et1<-numeric(lengthTrain)
  x1<-as.matrix(x1<-cbind( constant,training[,2:p],et1))
  y<-da[1:lengthTrain]
  cc<-rep(1,lengthTest)
  et2<-numeric(lengthTest)
  test<-as.matrix(cbind(cc,da1[(lengthTrain+1):length(da1[,1]),2:p],et2))
  cl<-cmeans(x1,c,100,verbose=TRUE,m=2) # degrees of memberships and cluster centers are calculated
  v<-cl$centers[,2]
```

```
mu<-cl$membership
fore<-numeric(length(test[,1])*c)
fore<-matrix(fore,length(test[,1]),c)
dista<-numeric(length(test[,1])*c)
dista<-matrix(dista,length(test[,1]),c)
distaa<-numeric(length(test[,1])*c)
distaa<-matrix(dista,length(test[,1]),c)
sumdist<-numeric(c)
##degrees of memberships are being calculated for test data set
for (i in 1:length(test[,1]))
  for(j in 1:c)
 {
    dista[i,j]<-dist(rbind( v[j] , test[i,]))
  }
for (i in 1:length(test[,1]))
{
  for(j in 1:c)
  {
    for (k in 1:c)
    {
      sumdist[k]<-(dista[i,j]/dista[i,k])^2
    }
    distaa[i,j]<- 1 / sum(sumdist) # degrees of memberships for forecasts are being
calculated.
  }
}
```

```
 train<-array(0,dim=c(lengthTest,numberParam,c))
 for(i in 1:c)
 train[,,i]<-as.matrix(cbind(distaa[,i],exp(distaa[,i]),distaa[,i]^2,test))
 ## training data set is being obtained.
 x<-array(0,dim=c(length(training[,1]),numberParam,c,numberParticle))
 xx<-array(0,dim=c(length(training[,1]),numberParam,c))
 for(j in 1:numberParticle)
  for(i in 1:c)
  {
   x[,,i,j]<-cbind(mu[,i],exp(mu[,i]),(mu[,i] ^2),x1)
  }
## initial positions and initial velocities are being generated
 position<-
array(rnorm(numberParam*numberParticle*c),dim=c(numberParam,c,numberParticle))
 position1<-array(0,dim=c(numberParam,c,numberParticle))
 v<-
array(rnorm(numberParam*numberParticle*c),dim=c(numberParam,c,numberParticle))
 v1<-array(0,dim=c(numberParam,c,numberParticle))
## initial personal bests are being obtained
 pbest<-position
 gbest<-pbest[,,1]
 Ytah<-matrix(numeric(length(training[,1])*c), length(training[,1]),c)
 Ytah1<-numeric(length(training[,1]))
 ## initial disturbance terms (et) are being obtained.
 for (k in 1:numberParticle)
 {
  et<-0
```

```
 for(n in 1:length(training[,1]))
 {
  x[n,numberParam,,k]<-et
  for(j in 1:c)
  {
   Ytah[n,j]<-x[n,,j,k] %*% position[,j,k]
  }
  Ytah1[n]<-mu[n,]%*%Ytah[n,]
  et<-y[n] - Ytah1[n]
 }
}
## initial global best value is being obtained
for (k in 1:numberParticle)
{
 if ( evalFunc(x[,,,k],y,pbest[,,k],c)[[1]] < evalFunc(x[,,,k],y,gbest,c)[[1]] )
  gbest<-pbest[,,k]
}
##disturbance terms are being obtained for test data set
Yfore<-array(0,c(length(train[,1,1]),c))
Yfore1<-numeric(length(train[,1,1]))
ett<-numeric(c)
for(n in 1:length(train[,1,1]))
{
 train[n,numberParam,]<-ett
 for(j in 1:c)
 {
```

```
    Yfore[n,j]<-train[n,,j] %*% gbest[,j]
  }
  Yfore1[n]<-mu[n,]%*%Yfore[n,]
  ett<-da1[lengthTrain:length(da1[,1]),1][n] - Yfore1[n]
  }
i<-0
repeat
{
  i<-i+1
  ## new positions and velocities are being calculated
  for (k in 1:numberParticle)
  {
    r1<-runif(n=1,min=0,max=1)
    r2<-runif(n=1,min=0,max=1)
    for( j in 1:c)
    {
   v1[,j,k]<- v[,j,k] + 2*r1*(pbest[,j,k]-position[,j,k]) + 2*r2 * (gbest[,j] - position[,j,k])
      for(l in 1:numberParam)
        v1[l,j,k]<- max(-1,min(v1[l,j,k],1))
      position1[,j,k]<-position[,j,k]+v1[,j,k]
    }
  }
  ## personal bests are being updated
  for (k in 1:numberParticle)
  {
    if ( evalFunc(x[,,,k],y,position1[,,k],c)[[1]] < evalFunc(x[,,,k],y,pbest[,,k],c)[[1]] )
```

```
    pbest[,,k]<- position1[,,k]
}
##global best is being updated
for (k in 1:numberParticle)
{
  if ( evalFunc(x[,,,k],y,pbest[,,k],c)[[1]] < evalFunc(x[,,,k],y,gbest,c)[[1]] )
  {
    gbest<-pbest[,,k]
    xx<-x[,,,k]
  }
}
v<-v1
position<-position1
## disturbance terms are being updated
for (k in 1:numberParticle)
{
  et<-0
  for(n in 1:length(training[,1]))
  {
    x[n,numberParam,,k]<-et
    for(j in 1:c)  # forecast are being calculated for each cluster for training data set
    {
      Ytah[n,j]<-x[n,,j,k] %*% position[,j,k]
    }
    Ytah1[n]<-mu[n,]%*%Ytah[n,]
    et<-y[n] - Ytah1[n]
```

```
    }
   }
   ###########################
   ett<-0
   for(n in 1:length(train[,1,1]))
   {
     train[n,numberParam,]<-ett
     for(j in 1:c)  # forecasts are being calculated for each cluster for test data set
     {
       Yfore[n,j]<-train[n,,j] %*% gbest[,j]
     }
     Yfore1[n]<-mu[n,]%*%Yfore[n,]
     ett<-da1[lengthTrain:length(da1[,1]),1][n] - Yfore1[n]
   }
   if (i==iterasyon)
     break;
  }
##root mean squared error is being calculated
 zz<-sqrt(sum((Yfore1-da1[(lengthTrain+1):length(da1[,1]),1])^2) / lengthTest)
##mean absolute percentage error is being calculated
 yy<-(1/lengthTest * sum(abs((da1[(lengthTrain+1):length(da1[,1]),1] - Yfore1) / da1[(lengthTrain+1) : length(da1[,1]),1]) ))
 #### forecasts are being calculated
 yytah<-array(0,dim=c(lengthTrain,c))
 for (i in 1:c)
 {
```

```
    yytah[,i]<-xx[,,i]%*%gbest[,i]
  }
  muu<-array(0,lengthTrain)
  for (i in 1:lengthTrain)
    round(muu[i]<-mu[i,]%*%yytah[i,])
  return(list(zz,yy,Yfore1))
}
```

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Nihat TAK

Date of birth and place : 30.07.1984 - Şişli

Foreign Languages :English

E-mail :nihattak@gmail.com

EDUCATION

Degree	Department	University	Date of Graduation
Phd	Statistics	Yıldız Technical University	2016
Graduate	Statistics	University of Arkansas	2012
Undergraduate	Computer and Statistcs Sciences	Karadeniz Technical University	2008
High School	Computer Hardware	Gultepe Vocational High School	2001

WORK EXPERIENCE

Year	Corporation/Institute	Enrollment
2012-2016	Kırklareli Unversity	Research Assistant
2012-2012	University of Arkansas	Research Assistant
2008-2009	Otokod Otomasyon A.Ş.	Computer Technicihan

PUBLISHMENTS

Conference Papers

1. Tak, N. , Evren, A. A., Tez, M. and Eğrioğlu (2016). E., Geri Beslemeli Bulanik Fonksiyon Yaklaşımı ve T.C. Altını Verisinin Analizi, Xth International Statistics Days Conference, 7-9 October 2016, Giresun University, Turkey.

2. Tak, N. , Evren, A. A., Tez, M. and Eğrioğlu, E. (2016). Recurrent Fuzzy Regression Function for Time Series Forecasting, International Conference on Trends and Perspectives in Linear Statistical Inference, 22-25 August 2016, Marmara University, Turkey.

3. Tak, N. (2015). Zaman Serileri Analizinde 1. Tip Bulanık Fonksiyon Yaklaşımı, 16. Uluslararası Ekonometri, Yöneylem Araştırması ve İstatistik Sempozyumu, 07-12 Mayıs 2015, Trakya Universitesi, Türkiye.