SPRINGER BRIEFS IN APPLIED SCIENCES AND TECHNOLOGY · COMPUTATIONAL INTELLIGENCE

Jesus Soto
Patricia Melin
Oscar Castillo

Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction

Springer

SpringerBriefs in Applied Sciences and Technology

Computational Intelligence

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/10618

Jesus Soto · Patricia Melin
Oscar Castillo

Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction

Springer

Jesus Soto
Division of Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California
Mexico

Oscar Castillo
Division of Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California
Mexico

Patricia Melin
Division of Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California
Mexico

ISSN 2191-530X ISSN 2191-5318 (electronic)
SpringerBriefs in Applied Sciences and Technology
ISSN 2520-8551 ISSN 2520-856X (electronic)
SpringerBriefs in Computational Intelligence
ISBN 978-3-319-71263-5 ISBN 978-3-319-71264-2 (eBook)
https://doi.org/10.1007/978-3-319-71264-2

Library of Congress Control Number: 2017958718

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book focuses on the fields of hybrid systems, fuzzy systems, bio-inspired algorithms, and time series. This book describes the construction of ensembles of interval type-2 fuzzy neural network (IT2FNN) models and the optimization of their fuzzy integrators with bio-inspired algorithms for time series prediction. Interval type-2 and type-1 fuzzy systems are used to integrate the outputs of the ensemble of IT2FNN models which are used. Genetic Algorithms and Particle Swarm Optimization are the bio-inspired algorithms used for the optimization of fuzzy integrators. The Mackey–Glass, Mexican Stock Exchange, Dow Jones, and NASDAQ time series are used to test the performance of the proposed method. Prediction errors are evaluated by the following metrics: mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), mean percentage error (MPE), and mean absolute percentage error (MAPE).

In Chap. 1, a brief introduction to the book is presented, where the intelligence techniques that are used, the main contribution, motivations, application, and a general description of the proposed methods are mentioned.

In Chap. 2, we describe the State of the Art, basic theoretical and technical concepts about the areas of computational intelligence, forecasts as well as a brief introduction and operations are addressed, as all of them are of great importance for the development of this work.

In Chap. 3, we describe the Problem Statement and Development of the ensemble of IT2FNN models with optimization of the fuzzy integrators used GAs and PSO algorithms for time series prediction; we also describe the IT2FNN models (IT2FNN-1, IT2FNN-2, and IT2FNN-3). The development of the structure of the chromosome (in GA) and particles (in PSO) for the optimization of fuzzy integrators is also presented.

Chapter 4 presents the results of the proposed method for all study cases: ensemble of IT2FNN models with optimization of the fuzzy integrators used GA and PSO algorithms for time series prediction, ensemble of the IT2FNN models for the Mexican Stock Exchange, Dow Jones, and NASDAQ time series with which we work during the development of this work.

Chapter 5 presents the Conclusion of this research work, and future work is suggested.

We end this Preface of the book by giving thanks to all the people who have helped or encouraged us during the writing of this book. First of all, we would like to thank our colleagues and friends, namely Prof. Patricia Melin, Prof. Oscar Castillo, and Prof. Janusz Kacprzyk for always supporting our work and for motivating us to report this research work. We would also like to thank our families for their continuous support during the time that we spent in this project. Of course, we have to thank our institution, Tijuana Institute of Technology, for always supporting our projects. We must thank our supporting agencies, CONACYT and TNM, in our country for their help during this project. Finally, we thank our colleagues working in Soft Computing, who are too many to mention all by name.

Tijuana, Mexico

Jesus Soto
Patricia Melin
Oscar Castillo

Contents

Chapter 1
Introduction

Based on the evolution of a variable or a set of variables given in a time series, to predict future values of this variable we should seek the dynamic laws governing the real state of the system over time. This preliminary step is the prediction modeling process. In short, time series analysis aims at drawing conclusions about a complex system using past data.

The time series analysis consists of a description of the movements that compose it, then building models using these movements to explain the structure and predict the evolution of a variable over time [1, 2]. The main and fundamental procedure for the analysis of a time series is described below:

1. Collecting data of the time series, and trying to ensure that these data are reliable.
2. Representing the time series qualitatively by noting the presence of long-term trends, cyclical variations and seasonal variations.
3. Plot a graph or trend line and obtain the appropriate trend values using the least squares method.
4. When seasonal variations are present, obtained these and adjust the data to these seasonal variations (i.e. data seasonally).
5. Adjust the seasonally trend.
6. Represent the cyclical variations obtained in step 5.
7. Combining the results of steps 1–6 and any other useful information to make a prediction (if desired) and if possible discuss the sources of error and their magnitude.

In general the above ideas can help in solving the important problem of prediction in time series. Along with common sense, experience, skill and judgment of the researchers, such mathematical analysis can, however, be of value for predicting the short, medium and long term.

This book focuses on the construction of ensembles for the Interval Type-2 Fuzzy Neural Networks (IT2FNN) architectures and the optimization of the fuzzy integrators for time series prediction with Bio-inspired algorithms. Interval type-2

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2_1

and type-1 fuzzy systems are used to integrate the output (forecast) of each Ensemble of IT2FNN models are used. The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are used for the optimization of the parameters values of fuzzy integrators. The Mackey-Glass, Mexican Stock Exchange (BMV) , Dow Jones and NASDAQ time series are used to test of performance of the proposed method. Prediction errors are evaluated by the following metrics: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE).

As related work we can mention: Type-1 Fuzzy Neural Network (T1FNN) [3–7] and the IT2FNN [8–11], also the type-1 [12–15] and type-2 [16, 17] fuzzy evolutionary systems are typical hybrid systems in soft computing. These systems combine T1FLS generalized reasoning methods [18–22] and IT2FLS [23–25] with NN learning capabilities [26–28] and evolutionary algorithms [4, 29–33] respectively.

References

1. Brocklebank, J.C., Dickey, D.A.: SAS for Forecasting Series, pp. 6–140. SAS Institute Inc., Cary, NC, USA (2003)
2. Brockwell, P.D., Davis, R.A.: Introduction to Time Series and Forecasting, pp. 1–219. Springer, New York (2002)
3. Horikowa, S., Furuhashi T., Uchikawa, Y.: On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm. IEEE Trans. Neural Netw. 3 (1992)
4. Melin, P., Soto, J., Castillo, O., Soria, J.: A new approach for time series prediction using ensembles of ANFIS models. Experts Syst. Appl. **39**(3), 3494–3506 (2012)
5. Jang, J.S.R.: ANFIS: Adaptive-network-based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. **23**, 665–685 (1992)
6. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and Soft Computing. Prentice-Hall, New York (1997)
7. Lin, Y.C., Lee, C.H.: System identification and adaptive filter using a novel fuzzy neuro system. Int. J. Comput. Cogn. **5**(1), 2 (2007)
8. Hagras, H.: Comments on dynamical optimal training for interval type-2 fuzzy neural network (T2FNN). IEEE Trans. Syst. Man Cybern. Part B **36**(5), 1206–1209 (2006)
9. Wang, C.H., Cheng, C.S., Lee, T.T.: Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN). IEEE Trans. Syst. Man Cybern. Part B: Cybern. **34**(3), 1462–1477 (2004)
10. Lee, C.H., Lin, Y.C.: Type-2 fuzzy neuro system via input-to-state-stability approach. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) International Symposium on Neural Networks, vol. 4492, pp. 317–327. Springer, Heidelberg, LNCS (2007)
11. Lee, C.H., Hong, J.L., Lin, Y.C., Lai, W.Y.: Type-2 fuzzy neural network systems and learning. Int. J. Comput. Cogn. **1**(4), 79–90 (2003)
12. Ascia, G., Catania, V., Panno, D.: An integrated fuzzy-GA approach for buffer management. IEEE Trans. Fuzzy Syst. **14**(4), 528–541 (2006)
13. Pedrycz, W.: Fuzzy Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (1997)
14. Chiou, Y.C., Lan, L.W.: Genetic fuzzy logic controller: an iterative evolution algorithm with new encoding method. Fuzzy Sets Syst. **152**(3), 617–635 (2005)

15. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans. Fuzzy Syst. **3**, 260–270 (1995)
16. Gaxiola, F., Melin, P., Valdez, F., Castillo, O.: Optimization of type-2 fuzzy weight for neural network using genetic algorithm and particle swarm optimization. In: World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 22–28 (2013)
17. Wu, D., Wan-Tan, W.: Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers. Eng. Appl. Artif. Intell. **19**(8), 829–841 (2006)
18. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Mach. Stud. **7**, 1–13 (1975)
19. Zadeh, L.A.: Fuzzy logic, neural networks and soft computing. Commun. ACM **37**(3), 77–84 (1994)
20. Pedrycz, W.: Fuzzy Modelling: Paradigms and Practice. Kluwer Academic Press, Dordrecht (1996)
21. Takagi, T., Sugeno, M.: Derivation of fuzzy control rules from human operation control actions. In: Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55–60 (1983)
22. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)
23. Karnik, N.N., Mendel, J.M.: Applications of type-2 fuzzy logic systems to forecasting of time-series. Inf. Sci. **120**, 89–111 (1999)
24. Wu, D., Mendel, J.M.: A vector similarity measure for interval type-2 fuzzy sets and type-1 fuzzy sets. Inf. Sci. **178**, 381–402 (2008)
25. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, NJ (2001)
26. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing, Boston, MA (1996)
27. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, NJ (2003)
28. Haykin, S.: Adaptive Filter Theory. Prentice Hall, Englewood Cliffs. ISBN 0-13-048434-2 (2002)
29. Pulido, M., Melin, P., Castillo, O.: Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. Inf. Sci. **280**, 188–204 (2014)
30. Soto, J., Melin, P., Castillo, O.: Time series prediction using ensembles of ANFIS models with genetic optimization of interval type-2 and type-1 fuzzy integrators. Int. J. Hybrid Intel. Syst. **11**(3), 211–226 (2014)
31. Bonissone, P.P., Subbu, R., Eklund, N., Kiehl, T.R.: Evolutionary algorithms + domain knowledge = real-world evolutionary computation. IEEE Trans. Evol. Comput. **10**(3), 256–280 (2006)
32. Engelbrech, P.: Fundamentals of Computational of Swarm Intelligence: Basic Particle Swarm Optimization, pp. 93–129. Wiley, New York (2005)
33. Deb, K.: A population-based algorithm-generator for real-parameter optimization. Springer, Heidelberg (2005)

Chapter 2
State of the Art

In this chapter, we describe the state of the art of the computational intelligence techniques, which we use as a basis for this work.

2.1 Time Series

Time series is a set of measurements of some phenomenon or experiment sequentially recorded over time. These observations will be denoted by $\{x(t_1), x(t_2), \ldots, x(t_n)\} = \{x(t) : t \in T \subseteq R\}$ con $x(t_i)$ the value of the variable x in the time t_i. If $T = Z$ is said that the time series is discrete and if $T = R$ is said that the time series is continuous [1, 2].

A classic model for a time series, assumes that a $x_{(1)}, \ldots, x_{(n)}$ series can be expressed as the sum or product of its components: trend, cyclical, seasonal and irregular [3]. There are three time series models, which are generally accepted as good approximations to the true relationships between the components of the observed data. These are:

Additive.

$$X(t) = T(t) + C(t) + S(t) + I(t) \tag{2.1}$$

Multiplicative.

$$X(t) = T(t) \cdot C(t) \cdot S(t) \cdot I(t) \tag{2.2}$$

Mixed.

$$X(t) = T(t) \cdot S(t) \cdot I(t) \tag{2.3}$$

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2_2

where:

$X(t)$ observed series in time t
$T(t)$ trend component
$C(t)$ cyclic component
$S(t)$ seasonal component
$I(t)$ irregular or random component

A common assumption is that $I(t)$ is a random or white noise component with zero mean and constant variance. An additive model, is suitable, for example, when $S(t)$ does not depend on other components such as $T(t)$, if instead the seasonality varies with the trend, the most suitable model is a multiplicative model. It is clear that the multiplicative model can be transformed into additive, by taking logarithms. The problem that arises is to adequately model the components of the series.

Temporal phenomena are both complex and important in many real-world problems. Their importance stems from the fact that almost every kind of data contains time-dependent components, either explicitly coming in the form of time values or implicitly in the way that the data is collected from a process that varies with time [4]. A time series is an important class of complex data objects [5] and comes as a sequence of real numbers, each number representing a value reported at a certain time instant [6]. The popular statistical model of Box-Jenkins [7] is considered to be one of the most common choices for the prediction of time series. However, since the Box-Jenkins models are linear and most real world applications involve nonlinear problems, it is difficult for the Box-Jenkins models to capture the phenomenon of nonlinear time series and this brings a limitation to the accuracy of the generated predictions [8].

2.2 Interval Type-2 Fuzzy Neural Network

One way to build on IT2FNN is by fuzzifying a conventional neural network (NN). Each part of a NN (the activation function, the weights, and the inputs and outputs) can be fuzzified. A fuzzy neuron is basically similar to an artificial neuron, except that it has the ability to process fuzzy information.

The IT2FNN system is one kind of IT2-TSK-FIS inside a NN structure. An IT2FNN is proposed by Castro et al. in [9], with TSK reasoning and processing elements called IT2FN for defining antecedents, and the IT1FN for defining the consequents of rules R^k.

An IT2FN is composed by two adaptive nodes represented by squares, and two non-adaptive nodes represented by circles. Adaptive nodes have outputs that depend on their inputs, modifiable parameters and transference function while non-adaptive, on the contrary, depend solely on their inputs, and their outputs represent lower $\underline{\mu}_A(x)$ and upper $\overline{\mu}_A(x)$ membership functions. Parameters from adaptive nodes with uncertain standard deviation are denoted by $\left(w \in \left[w_{1,1}, w_{2,1}\right]\right)$

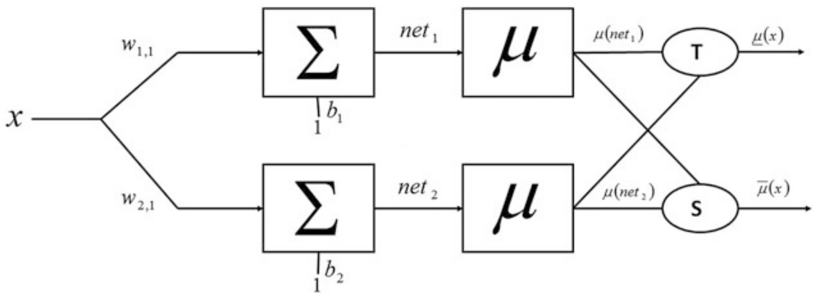


Fig. 2.1 Interval type-2 fuzzy neuron (IT2FN)

and with uncertain mean by $(b \in [b_1, b_2])$. The IT2FN (Fig. 2.1) with crisp input signals (x), crisp synaptic weights (w, b) and type-1 fuzzy outputs $\mu(net_1), \mu(net_2), \underline{\mu}(x), \overline{\mu}(x)$. This kind of neuron is build from two conventional neurons with transference functions $\mu(net_1)$, Gaussian, generalized bell and logistic for fuzzifier the inputs. Each neuron equation is defined as follows: the function μ is often referred to as an activation (or transfer) function. Its domain is the set of activation values, net, of the neuron model; we thus often use this function as $\mu(net_2)$. The variable net is defined as a scalar product of the weight and the vectors:

$$\begin{aligned} net_1 &= w_{1,1} + b_1; \mu_1 = \mu(net_1); \\ net_2 &= w_{2,1} + b_1; \mu_2 = \mu(net_2) \end{aligned} \tag{2.4}$$

The non-adaptive *t*-norm node (*T*) evaluates the lower membership function $\overline{\mu}(x)$ under *t*-norm algebraic product, while s-norm non-adaptive node (*S*), evaluate the upper membership function $\underline{\mu}(x)$ under the *s*-norm algebraic sum, as shown in Eq. (2.5):

$$\begin{aligned} \underline{\mu}(x) &= \mu(net_1) \cdot \mu(net_2), \\ \overline{\mu}(x) &= \mu(net_1) + \mu(net_2) - \underline{\mu}(x) \end{aligned} \tag{2.5}$$

Each IT2FN adapts an interval type-2 fuzzy set [10, 11], $\tilde{A}$, expressed in terms of the output $\underline{\mu}(x)$, of type-1 fuzzy neuron with *T*-norm and $\overline{\mu}(x)$ of type-1 fuzzy neuron with *S*-norm. An internal type-2 fuzzy set is denoted as:

$$\tilde{A} = \int_{x \in X} \left[\int_{\mu(x) \in [\overline{\mu}(x), \underline{\mu}(x)]} 1/\mu \right] / x \tag{2.6}$$

An IT1FN (Fig. 2.2) is built from two conventional adaptive linear neurons (ADALINE) [12] for adapting the consequents $y_k^j \in [{}_l y_{k \cdot r}^j y_k^j]$ from the rules R_k, for the output defined by

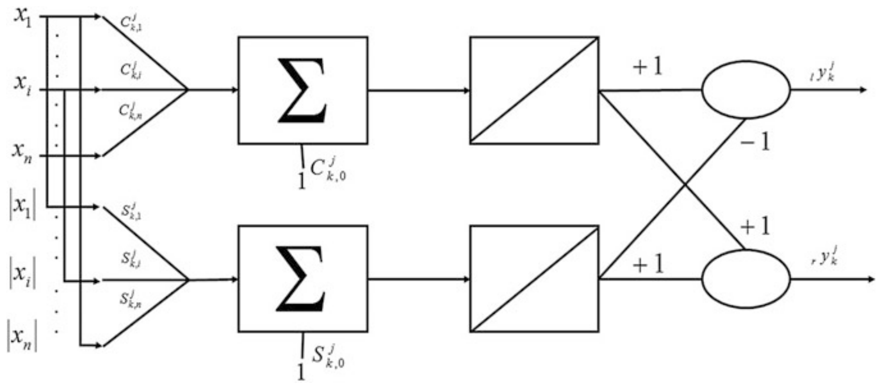


Fig. 2.2 Interval type-1 fuzzy neuron

$$
\begin{aligned}
{}_l y_k^j &= \sum_{1=1}^{n} C_{k,i}^j x_i + C_{k,0}^j - \sum_{1=1}^{n} S_{k,i}^j |x_i| - S_{k,0}^j, \\
{}_r y_k^j &= \sum_{1=1}^{n} C_{k,i}^j x_i + C_{k,0}^j + \sum_{1=1}^{n} S_{k,i}^j |x_i| + S_{k,0}^j.
\end{aligned}
\tag{2.7}
$$

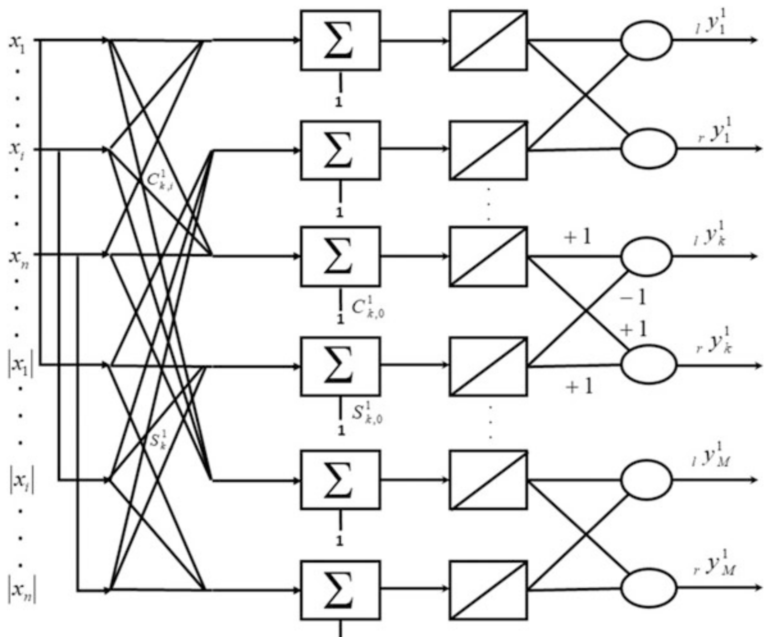


Fig. 2.3 Interval type-2 fuzzy neural network

Thus consequents can be adapted with linear networks. The network weights are the parameters of consequents $C_{k,i}^{j}, S_{k,i}^{j}$ for the kth rule. The outputs represents interval linear MFs of the rule's consequents (Fig. 2.3).

2.3 Ensemble Learning

The Ensemble consists of a learning paradigm where multiple component learners are trained for a same task, and the prediction of the component learners are combined for dealing with future instances [13]. Since an Ensemble is often more accurate than its component learners, such a paradigm has become a hot topic in recent years and has already been successfully applied to optical character recognition, face recognition, scientific image analysis, medical diagnosis and time series [14].

In general, a neural network ensemble is constructed in two steps, i.e. training a number of component neural networks and then combining the component predictions.

There are also many other approaches for training the component neural networks. Some examples are as follows. Hampshire and Waibel [15, 16] utilize different objective functions to train distinct component neural networks. Cherkauer [17] trains component networks with different number of hidden layers. Maclin and Shavlik [18] initialize component networks at different points in the weight space. Krogh and Vedelsby [19, 20] employ cross-validation to create component networks. Opitz and Shavlik [21, 22] exploit a genetic algorithm to train diverse knowledge based component networks. Yao and Liu [23] regard all the individuals in an evolved population of neural networks as component networks [24].

2.4 Interval Type-2 Fuzzy Systems

Type-2 fuzzy sets are used to model uncertainty and imprecision; originally they were proposed by Zadeh [25, 26] and they are essentially "fuzzy–fuzzy" sets in which the membership degrees are type-1 fuzzy sets (Fig. 2.4).

The structure of a type-2 fuzzy system implements a nonlinear mapping of on input to on output space. This mapping is achieved through a set of type-2 if-then fuzzy rules, each of which describes the local behavior of the mapping.

The uncertainty is represented by a region called footprint of uncertainty (FOU). When $\mu_{\tilde{A}}(x,u) = 1, \forall u \in l_x \subseteq [0,1]$ we have an interval type-2 membership function [27–30] (Fig. 2.5).

The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\overline{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$.

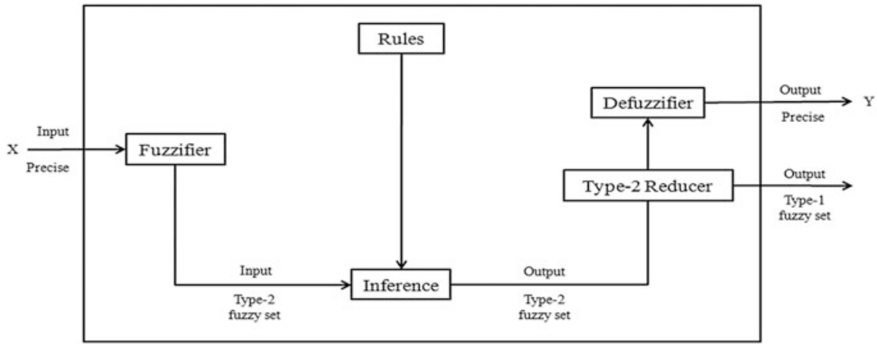
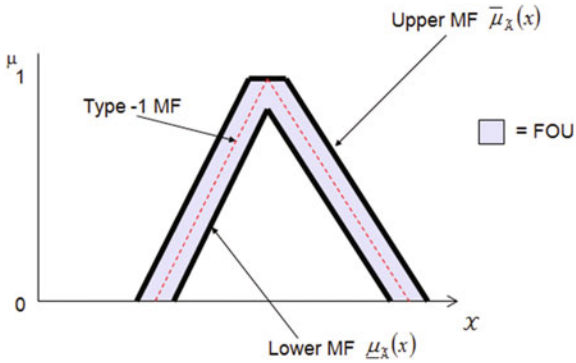


Fig. 2.4 Structure of the interval type-2 fuzzy logic system

Fig. 2.5 Interval type-2 membership function



A fuzzy logic system (FLS) described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain [9, 31]. On the other hand, type-2 FLSs are very useful in circumstances where it is difficult to determine an exact certainty value, and there are measurement uncertainties.

2.5 Genetic Algorithms

Genetic algorithms (GAs) are adaptive methods that can be used to solve search and optimization problems. They are based on the genetic process of living organisms. Over generations, the populations evolve in nature in accordance with the principles of natural selection and survival of the strongest, postulated by Darwin. By imitating this process, genetic algorithms are able to create solutions to real world problems. The evolution of these solutions towards optimal values of the problem depends largely on proper coding them. The basic principles of genetic algorithms were established by Holland [32, 33] and are well described in the works of Goldberg [34–36], Davis [37] and Michalewicz [38]. The large field of applications of GA is related to those problems for which there are no specialized techniques. Even if such technical exist and work well, improvements can be made with the same hybrid genetic algorithms.

A GA is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects (typically strings of fixed length which fit the model chains chromosomes), each of which is associated with a fitness in a new population (e.g. the next generation) operations using models according to the principle Darwinian reproduction and survival of the fittest and after having naturally presented a series of genetic operations [39].

To apply the genetic algorithm requires the following five basic components:

1. Representation of the potential solutions to the problem.
2. One way to create an initial population of possible solutions (usually a random process).
3. An evaluation function to play the role of the environment, classifying solutions in terms of their "fitness".
4. Genetic operators that alter the composition of the children that will occur for the next generation.
5. Values for the different parameters using the genetic algorithm (population size, crossover probability, mutation probability, maximum number of generations, etc.).

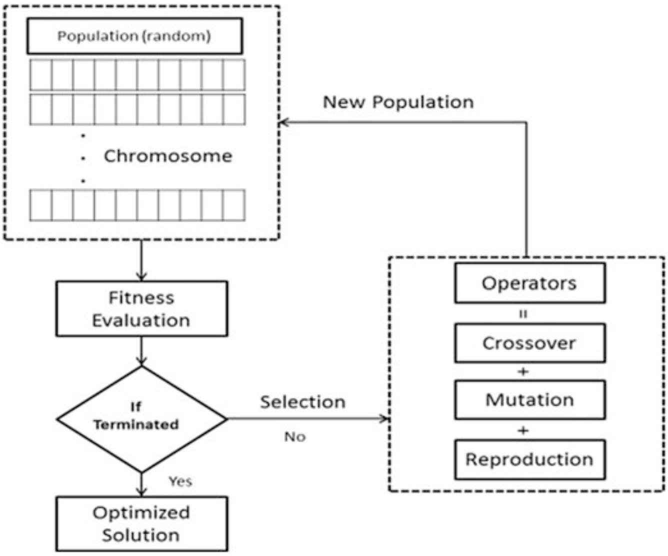


Fig. 2.6 Steps of the genetic algorithm

The basic operations of a genetic algorithm [40] are as follows illustrated in Fig. 2.6:

- Step 1: Represent the problem variable domain as a chromosome of a fixed length; choose the size of a chromosome population N, the crossover probability (pc) and the mutation probability (pm).
- Step 2: Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.
- Step 3: Randomly generate an initial population of chromosomes of size $N : x_1, x_2, \ldots, x_N$
- Step 4: Calculate the fitness of each individual chromosome: $f(x_1), f(x_2), \ldots, f(x_N)$.
- Step 5: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness.
- Step 6: Create a pair of offspring chromosomes by applying the genetic operators- crossover and mutation.
- Step 7: Place the created offspring chromosomes in the new population.
- Step 8: Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population, N.
- Step 9: Replace the initial (parent) chromosome population with the new (offspring) population.
- Step 10: Go to Step 4, and repeat the process until the termination criterion is satisfied.

2.6 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a bio-inspired optimization method proposed by Eberhart and Kennedy [41–43] in 1995. PSO is as metaheuristic search technique based on a population of particles. The main idea of PSO comes from the social behavior of schools of fish and flocks of birds [44, 45]. In PSO each particle moves in a D-dimensional space based on its own past experience and those of other particles [46, 47]. Each particle has a position and a velocity represented by the vectors $x_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,D}\right)$ and $v_i = \left(v_{i,1}, v_{i,2}, \ldots, v_{i,D}\right)$ for the i-th particle. At each iteration, particles are compared with each other to find the best particle [48, 49]. Each particle records its best position as $v_i = \left(v_{i,1}, v_{i,2}, \ldots, v_{i,D}\right)$. The best position of all particles in the swarm is called the global best, and is represented as $G = (G_1, G_2, \ldots, G_D)$. The velocity of each particle is given by Eq. (2.13).

$$v_{id} = wv_{id} + C_1 \cdot rand() \cdot (pbest_{id} - x_{id}) + C_2 \cdot rand() \cdot (gbest - x_{id}) \tag{2.13}$$

In this equation $i = 1, 2, \ldots, M, d = 1, 2, \ldots D, C_1$ and C_2 are positive constants (known as acceleration constants), $rand_1()$ and $rand_2()$ are random numbers in

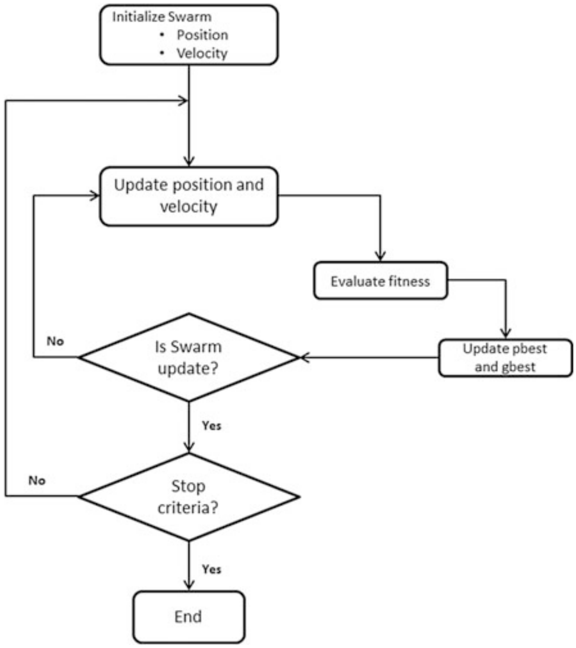


Fig. 2.7 Flowchart of the PSO algorithm

[0, 1], and w, introduced by Shi and Eberhart [50] is the inertia weight. The new position of the particle is determined by Eq. (2.14):

$$x_{id} = x_{id} + v_{id} \tag{2.14}$$

The basic functionally of the PSO is illustrated as follows (Fig. 2.7).

References

1. Cowpertwait, P., Metcalfe, A.: Time Series. Introductory Time Series with R, pp. 2–5. Springer, Dordrecht (2009)
2. Wei, W.W.S.: Time Series Analysis: Univariate and Multivariate Methods. Ed. Addison-Wesley **1**, 40–100 (1994)
3. Durbin, J., Koopman, S.J.: Time Series Analysis by State Space Methods. Oxford University Press (2001)
4. Erland, E., Ola, H.: Multivariate time series modeling, estimation and prediction of mortalities. Insur. Math. Econ. **65**, 156–171 (2015)
5. Weina, W., Witold, P., Xiaodong, L.: Time series long-term forecasting model based on information granules and fuzzy clustering. Eng. Appl. Artif. Intell. **41**, 17–24 (2015)
6. Shu-Xian, L., Xian-Shuang, Y., Hong-Yun, Q., Hai-Feng, H.: A novel model of leaky integrator echo state network for time-series prediction. Neurocomputing **159**, 58–66 (2015)
7. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control, 3rd edn. Prentice Hall, New Jersey (1994)

8. Akhter, M.R., Arun, A., Sastry, V.N.: Recurrent neural network and a hybrid model for prediction of stock returns. Expert Syst. Appl. **42**, 3234–3241 (2015)
9. Castro, J.R., Castillo, O., Melin, P., Rodriguez, A.: A hybrid learning algorithm for interval type-2 fuzzy neural networks: the case of time series prediction, vol. 15a, pp. 363–386. Springer, Berlin (2008)
10. Karnik, N.N., Mendel, J.M.: An Introduction to Type-2 Fuzzy Logic Systems. University of Southern California, Los Angeles, CA (1998)
11. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, NJ (2001)
12. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing, Boston, MA (1996)
13. Sharkey, A.: Combining artificial neural nets: ensemble and modular multi-net systems. Springer, London (1999)
14. Sollich, P., Krogh, A.: Learning with ensembles: how over-fitting can be useful. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems 8, pp. 190–196. MIT Press, Cambridge (1996)
15. Hampshire, J., Waibel, A.: A novel objective function for improved phoneme recognition using time-delay neural networks. IEEE Trans. Neural Netw. **1**(2), 216–228 (1990)
16. Wei, L.Y., Cheng, C.H.: A hybrid recurrent neural networks model based on synthesis features to forecast the Taiwan Stock Market. Int. J. Innovative Comput. Inf. Control **8**(8), 5559–5571 (2012)
17. Cherkauer, K.J.: Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks. In: Chan, P., Stolfo, S., Wolpert, D. (eds.) Proceedings of AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms, Portland, OR, AAAI Press, Menlo Park, CA, pp. 15–21 (1996)
18. Maclin, R., Shavlik, J.W.: Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks. In: Proceedings of IJCAI-95, Montreal, Canada, Morgan Kaufmann, San Mateo, CA, pp. 524–530 (1995)
19. Liu, F., Quek, C., See, G.: Neural network model for time series prediction by reinforcement learning. In: Proceedings of the International Joint Conference on the Neural Networks, Montreal, Canada (2005)
20. Soltani, S.: On the use of the wavelet decomposition for time series prediction. Neurocomputing **48**(1–4), 267–277 (2002)
21. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in neural information processing systems 7, MIT Press, Denver, CO, Cambridge, MA, pp. 231–238 (1995)
22. Opitz, D.W., Shavlik, J.W.: Generating accurate and diverse members of a neural network ensemble. in: D.S. Touretzky M.C., Mozer M.E., Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, Denver, CO, MIT Press, Cambridge, MA, pp. 535–541, (1996)
23. Yao, X., Liu, F.: Evolving neural network ensembles by minimization of mutual information. Int. J. Hybrid Intell. Syst. **1**, 12–21 (2004)
24. Xue, J., Xu, Z., Watada, J.: Building an integrated hybrid model for short-term and mid-term load forecasting with genetic optimization. Int. J. Innovative Comput. Inf. Control **8**(10), 7381–7391 (2012)
25. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Trans. Fuzzy Syst. **4**(2), 103–111 (1996)
26. Zadeh, L.A.: Fuzzy logic. Computer **1**(4), 83–93 (1988)
27. Castillo, O., Melin, P.: Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review. Inf. Sci. **205**, 1–19 (2012)
28. Castro, J.R., Castillo, O., Martínez, L.G.: Interval type-2 fuzzy logic toolbox. Eng. Lett. **15**(1), 89–98 (2007)

29. Karnik, N.N., Mendel, J.M., Qilian, L.: Type-2 fuzzy logic systems. Fuzzy Syst. IEEE Trans. **7**(6), 643–658 (1999)
30. Mendel, J.M.: Why we need type-2 fuzzy logic systems. Article is provided courtesy of Prentice Hall, By Jerry Mendel (2001)
31. Wu, D., Mendel, J.M.: A vector similarity measure for interval type-2 fuzzy sets and type-1 fuzzy sets. Inf. Sci. **178**, 381–402 (2008)
32. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Michigan (1975)
33. Holland, J.H.: Outline for a logical theory of adaptive systems. J. Assoc. Comput. Mach. **3**, 297–314 (1962)
34. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company (1989)
35. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms: motivation, analysis, and first results. Complex Syst. **3**, 493–530 (1989)
36. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In: Gregory, J.E.R. (ed) Foundations of Genetic Algorithms, pp. 69–93. Morgan Kaufmann Publishers, San Mateo, California (1991)
37. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold (1991)
38. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. AI Series. Springer, New York (1994)
39. Koza, J.R.: Genetic Programming. On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
40. Buckles, B.P., Petry, F.E.: Genetic Algorithms. IEEE Computer Society Press (1992)
41. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS), pp. 39–43 (1995)
42. Eberhart, R., Shi, Y., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Mateo, California (2001)
43. Engelbrech, P.: Fundamentals of Computational of Swarm Intelligence: Basic Particle Swarm Optimization, pp. 93–129. Wiley, New York (2005)
44. Escalante, H.J., Montes, M., Sucar, L.E.: Particle swarm model selection. J. Mach. Learn. Res. **10**, 405–440 (2009)
45. Kennedy, J., Eberhart, R.: Particle swam optimization. Proc. IEEE Int. Conf. Neural Network (ICNN) **4**, 1942–1948 (1995)
46. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multimodal complex space. IEEE Trans. Evol. Comput. **6**, 58–73 (2002)
47. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proc. IEEE Congr. Evol. Comput. **3**, 1951–1957 (1999)
48. Eberhart, R., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. Proc. IEEE Congr. Evol. Comput. **1**, 84–88 (2000)
49. Parsopoulos, K.E., Vrahatis, M.N.: Particle Swarm Optimization Intelligence: Advances and Applications. Information Science Reference, pp. 18–40. USA (2010)
50. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the IEEE Congress of Evolutionary Computation, pp. 69–73 (1998)

Chapter 3
Problem Statement and Development

The first goal of this book is the construction of the Ensembles of IT2FNN models and their optimization of the fuzzy integrators with GAs and PSO algorithms for time series prediction. The second goal is the design of interval type-2 and type-1 fuzzy systems to integrate the outputs (forecasts) of the IT2FNN models forming the Ensemble. The Genetic Algorithm (GAs) and Particle Swarm Optimization (PSO) were used for the optimization the parameters of the MFs of fuzzy integrators. The Mackey-Glass, Mexican Stock Exchange, Dow Jones, NASDAQ time series are used to test of performance of the proposed architectures (Fig. 3.1). When more than one forecasting technique seems reasonable for a particular application, then the forecast accuracy measures can also be used to discriminate between competing models. One can subtract the forecast value from the observed value of the data at that time point and obtain a measure of error. Therefore to evaluate the prediction error, we can apply the metrics to calculate the Mean Absolute Error (MAE) by Eq. (3.1), Mean Square Error (MSE) by Eq. (3.2), Root Mean Square Error (RMSE) by Eq. (3.3), Mean Percentage Error (MPE) by Eq. (3.4) and Mean Absolute Percentage Error (MAPE) by Eq. (3.5), respectively.

$$\mathrm{MAE} = \frac{1}{2}\sum_{t=1}^{n} |(A_t - P_t)| \tag{3.1}$$

$$\mathrm{MSE} = \frac{1}{2}\sum_{t=1}^{n} (A_t - P_t)^2 \tag{3.2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{2}\sum_{t=1}^{n} (A_t - P_t)^2} \tag{3.3}$$

$$\mathrm{MPE} = \frac{100\%}{n} = \sum_{t=1}^{n} \frac{(A_t - P_t)}{A_t} \tag{3.4}$$

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2_3

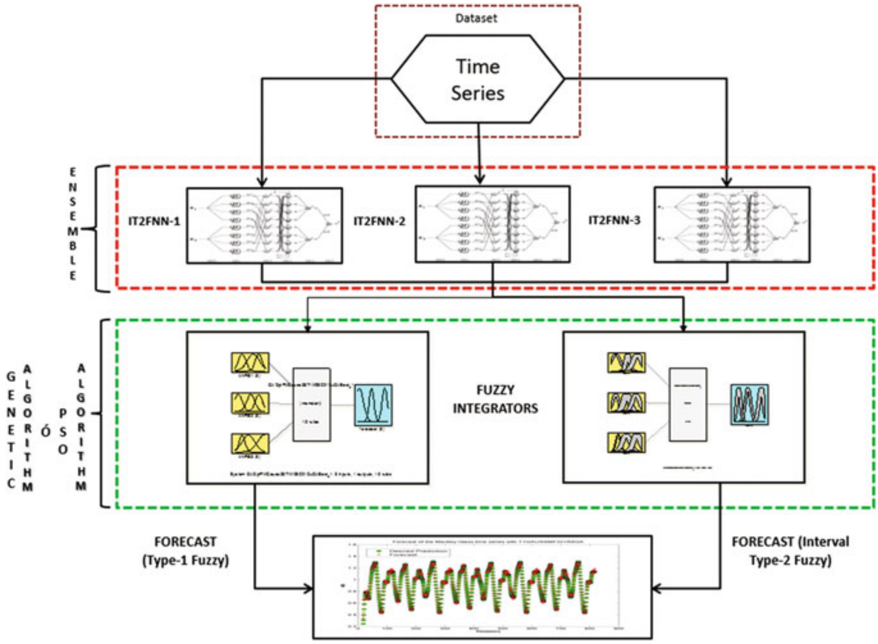


Fig. 3.1 The general proposed architecture

$$\text{MAPE} = \frac{100\%}{n} = \sum_{t=1}^{n} \left| \frac{(A_t - P_t)}{A_t} \right| \tag{3.5}$$

where *A,* corresponds to the real data of the time series, *P* corresponds to the forecast of the NNs or the aggregation models, *t* is the time variable, and *n* is the number of data points of the time series.

The general proposed architecture combines the ensemble of IT2FNN models and the use of fuzzy response integrators optimized with GA and PSO algorithms for time series prediction (Fig. 3.1).

3.1 Historical Data

The problem of predicting future values of a time series has been a point of reference for many researchers. The aim is to use the values of the time series known at a point x = t to predict the value of the series at some future point x = t + P. The standard method for this type of prediction is to create a mapping from D points of a Δ spaced time series, i.e. (x(t − (D − 1) Δ), …, x(t − Δ), x(t)), to a predicted future value x(t + P), for example the values D = 4 and Δ = P = 6 [1, 2] or 250 were used in this work. The data used in this book are the Mackey-Glass for τ = 13, 17, 30, 34, 68, 100, 136; the Mexican Stock Exchange, the Dow Jones and the NASDAQ time series.

3.1.1 Mackey-Glass Time Series

The chaotic time series data used is defined by the Mackey-Glass [3, 4] time series, whose differential equation is given by Eq. (3.6):

$$x(t) = \frac{0.2x(t-\tau)}{1 - x^{10}(t-\tau)} - 0.1x(t-\tau) \tag{3.6}$$

For obtaining the values of the time series at each point, we can apply the Runge-Kutta method [1, 2] for the solution of Eq. (3.6). The integration step was set at 0.1, with initial condition x(0) = 1.2, τ = 17, x(t) is then obtained for 0 ≤ t ≤ 1200, (We assume x(t) = 0 for t < 0 in the integration). From the Mackey-Glass time series we used 800 pairs of data points (Fig. 3.2), similar to [5–9]. The first 400 pairs of points are used for training (50%) and the other 400 pairs of points are used to validate the IT2FNN models.

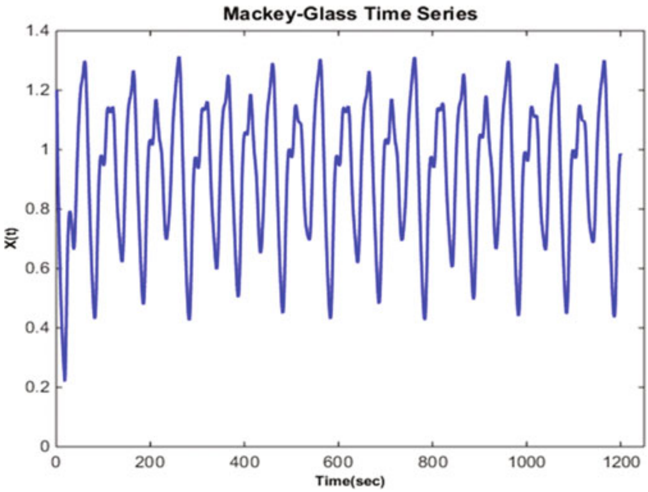


Fig. 3.2 Mackey-Glass time series

3.1.2 Mexican Stock Exchange

The Mexican Stock Exchange (BMV) is a financial institution that operates by a grant from the Department of Finance and Public Credit, with the goal of following closely the Securities Market of Values in Mexico [10, 11]with the initial public offering taking place on June 13 of 2008 with its shares representing its capital [12]. From the BMV time series we extracted 1250 pairs of data that correspond to a period from 01/03/2011 to 12/31/2015 (Fig. 3.3) and can be downloaded from daily live Yahoo database [13], where the first 625 pairs of points are used for training (50%) and the other 625 pairs of points are used to validate the IT2FNN models.

3.1.3 Dow Jones Time Series

The better represent the movements of the stock market at the time, the Dow Jones & Company designed a barometer of economic activity meter with twelve companies creating the Dow Jones stock index [14, 15]. Like the New York Times and Washington Post newspapers the company is open to the market but is controlled the by the private sector. So far, the company is controlled by the Bancroft family, which controls 64% of the shares entitled to vote [16]. From the Dow Jones Industries Average time series we are using 1250 pairs of data that correspond from 01/03/2011 to 12/31/2015 (Fig. 3.4) and can be downloaded from daily live Yahoo database [17], where the first 625 pairs of points are used for training (50%) and the other 625 pairs of points are used to validate the IT2FNN models.

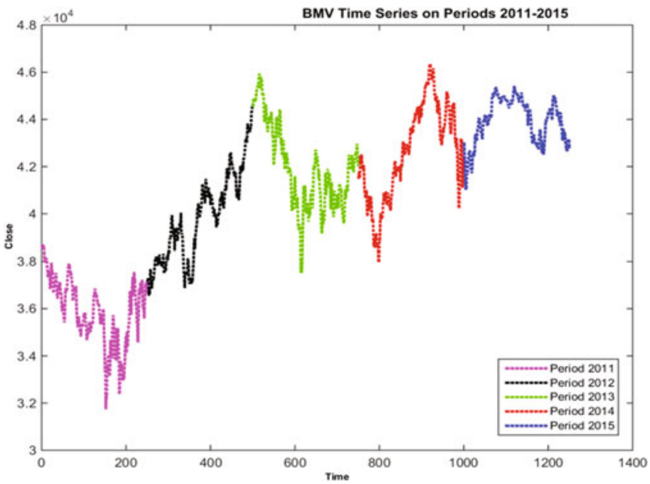


Fig. 3.3 Mexican Stock Exchange time series

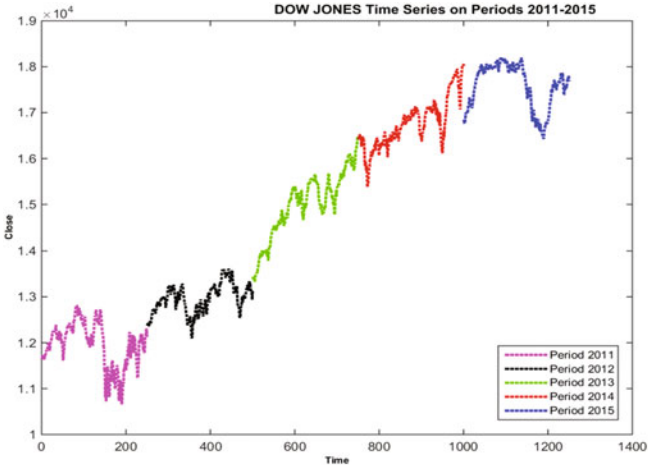


Fig. 3.4 Dow Jones time series

3.1.4 NASDAQ Time Series

NASDAQ is the largest U.S. electronic stock market. It has listed around 3300 companies; it may probably list most of the companies and, on average, trades more shares per day than other U.S. markets [18] price. The price is able to represent the tendency of variety of NASDAQ market in some sense. Therefore, the forecast of the price can benefit of analyzing the whole market [19, 20]. From the NASDAQ time series we are using 1250 pairs of data that correspond from 01/03/2011 to 12/31/2015 (Fig. 3.5) and can be downloaded from daily live Yahoo database [21], where the first 625 pairs of points are used for training (50%) and the other 625 pairs of points are used to validate the IT2FNN models.

3.2 Ensembles of IT2FNN Architectures

The ensembles of IT2FNN architectures imply a significant learning improvement comparatively to a single IT2NN and especially to the learning algorithms. Each IT2FNN works independently in its own domain. Each of the IT2FNN is build and trained for a specific task for each module. Three modules are used in each experiment of the ensemble of IT2FNN. In module 1 we have the IT2FNN-1, in module 2 we have the IT2FNN-2 and in module 3 we have the IT2FNN-3. Therefore each IT2FNN architecture has three/four input variables and one output variable that are described as follows:

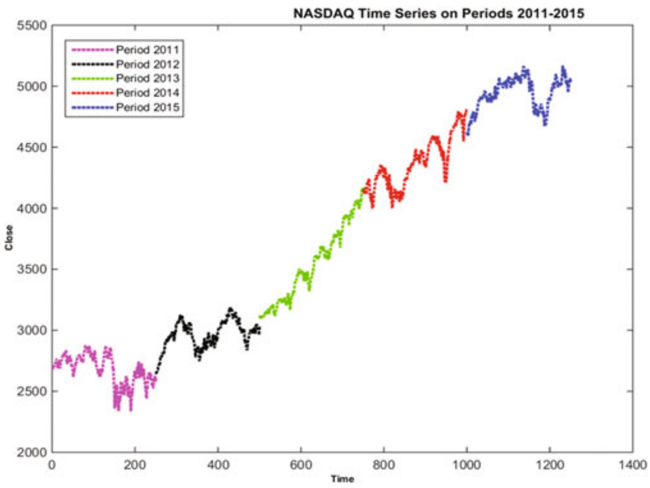


Fig. 3.5 NASDAQ time series

Mackey-Glass time series, in this case we predict $x(t)$ from three past (delays) values of the time series, that is, $x(t - 18)$, $x(t - 12)$, and $x(t - 6)$. Therefore the format of the training data is:

$$[x(t-18), x(t-12), x(t-6); x(t)] \tag{3.7}$$

where $t = 19$ to 818 and $x(t)$ is the desired prediction of the Mackey-Glass time series (Fig. 3.2).

The BMV, Dow Jones and NASDAQ time series, we predict $x(t)$ (corresponds to the 2015 period) from four past (delays) values of the time series, that is, $x(t - 1000)$ corresponds to the 2011 period, $x(t - 750)$ corresponds to the 2012 period, $x(t - 500)$ corresponds to the 2013 period, and $x(t - 250)$ corresponds to the 2014 period. Therefore the format of the training data is:

$$[x(t-1000), x(t-750), x(t-500), x(t-250); x(t)] \tag{3.8}$$

where $t = 1001$ to 1250 and $x(t)$ is the desired prediction of the time series (see Figs. 3.3, 3.4 and 3.5).

3.2.1 IT2FNN-1 Model

The IT2FNN-1 model has 5 layers (Fig. 3.6), consists of adaptive nodes with an equivalent function for the lower-upper membership in the fuzzification layer (layer 1). Non-adaptive nodes in the rules layer (layer 2) interconnect with the fuzzification layer (layer 1) in order to generate TSK IT2FIS rules antecedents. The

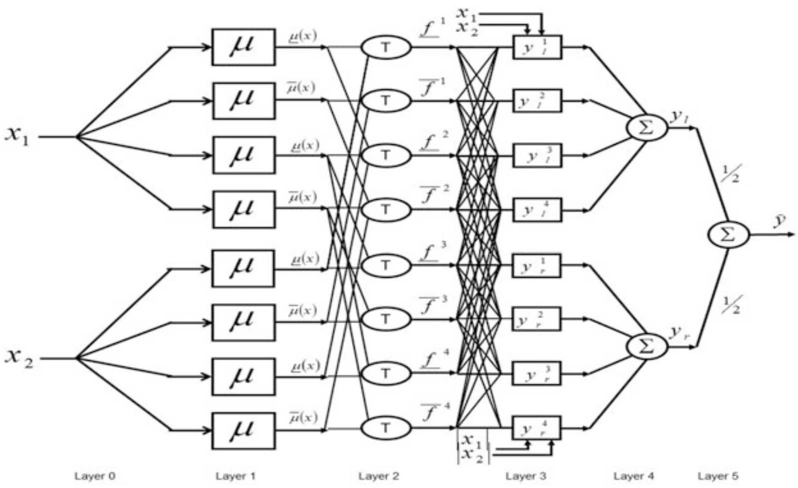


Fig. 3.6 IT2FNN-1 architecture

adaptive nodes in the consequent layer (layer 3) are connected to input layer (layer 0) to generate rules consequents. The non-adaptive nodes in type-reduction layer (layer 4) evaluate left-right values with KM algorithm [22]. The non-adaptive node in the defuzzification layer (layer 5) average left-right values.

For simplicity, we assume the IT2FNN-1 under consideration has ***n*** inputs and one output. The forward-propagation procedure is described as follows:

Layer 0: Inputs

$$x = (x_1, \ldots, x_i, \ldots, x_n,)^t$$

Layer 1: IT2 MFs $\tilde{\mu}_{k,i}(x_i) = \left\{ \underline{\mu}_{k,i}(x_i), \overline{\mu}_{k,i}(x_i) \right\}$

for example (Fig. 3.9a)

$$\begin{aligned}
&\mu_{k,i}(x_i) = \left\{ \underline{\mu}_{k,i}(x_i), \overline{\mu}_{k,i}(x_i) \right\} = i\,gaussmtype2\left(x_i\left[\sigma_{k,i}, {}^1m_{k,i}, {}^2m_{k,i}\right]\right); k = 1,2,\ldots,M; i = 1,2,\ldots,n,\\
&{}^1\mu_{k,i}\left(x_i\left[\sigma_{k,i}, {}^1m_{k,i}\right]\right), e^{-\frac{1}{2}\left(\frac{x_i - {}^1m_{k,i}}{\sigma_{k,i}}\right)^2}; {}^2\mu_{k,i}\left(x_i\left[\sigma_{k,i}, {}^1m_{k,i}\right]\right), e^{-\frac{1}{2}\left(\frac{x_i - {}^2m_{k,i}}{\sigma_{k,i}}\right)^2},\\
&\overline{\mu}_{k,i}(x_i) = \begin{cases} {}^1\mu\left(x_i\left[\sigma_{k,i}, {}^1m_{k,i}\right]\right), & x_i < {}^1m_{k,i},\\ 1, & {}^1m_{k,i} \le x_i \ge {}^2m_{k,i}\\ {}^2\mu\left(x_i\left[\sigma_{k,i}, {}^2m_{k,i}\right]\right), & x_i > {}^2m_{k,i}, \end{cases}\\
&\underline{\mu}_{k,i}(x_i) = \begin{cases} {}^2\mu\left(x_i\left[\sigma_{k,i}, {}^2m_{k,i}\right]\right), & x_i \le \frac{{}^1m_{k,i} - {}^2m_{k,i}}{2},\\ {}^1\mu\left(x_i\left[\sigma_{k,i}, {}^1m_{k,i}\right]\right), & x_i > \frac{{}^1m_{k,i} - {}^2m_{k,i}}{2}. \end{cases}
\end{aligned} \tag{3.9}$$

Layer 2: Rules

$$\underline{f}^k = \overset{n}{\underset{i=1}{\widetilde{*}}} \left(\underline{\mu}_{k,i}\right); \overline{f}^k = \overset{n}{\underset{i=1}{\widetilde{*}}} \left(\overline{\mu}_{k,i}\right) \tag{3.10}$$

Layer 3: Consequents left-right firing points

$$\begin{aligned} y_l^k &= \sum_{i=1}^{n} C_{k,i}x_i + C_{k,0} - \sum_{i=1}^{n} S_{k,i}|x_i| - S_{k,0}, \\ y_r^k &= \sum_{i=1}^{n} C_{k,i}x_i + C_{k,0} + \sum_{i=1}^{n} S_{k,i}|x_i| + S_{k,0}. \end{aligned} \tag{3.11}$$

Layer 4: Left-right points (type-reduction using KM algorithm)

$$\begin{aligned} \hat{y}_l &= \hat{y}_l\left(\overline{f}^1, \ldots, \overline{f}^L, \underline{f}^{L+1}, \ldots, \underline{f}^M, \quad y_l^1, \ldots, y_l^M\right) = \frac{\sum_k^M f_l^k \cdot y_l^k}{\sum_{k=1}^M f_l^k} = \frac{\sum_{k=1}^L \overline{f}^k \cdot y_l^k + \sum_{k=L+1}^L \underline{f}^k \cdot y_l^k}{\sum_{k=1}^L \overline{f}^k + \sum_{k=L+1}^L \underline{f}^k} \\ \hat{y}_r &= \hat{y}_r\left(\overline{f}^1, \ldots, \overline{f}^R, \underline{f}^{R+1}, \ldots, \underline{f}^M, \quad y_r^1, \ldots, y_r^M\right) = \frac{\sum_k^M f_r^k \cdot y_r^k}{\sum_{k=1}^M f_r^k} = \frac{\sum_{k=1}^R \overline{f}^k \cdot y_r^k + \sum_{k=R+1}^M \underline{f}^k \cdot y_l^k}{\sum_{k=1}^R \overline{f}^k + \sum_{k=R+1}^M \underline{f}^k} \end{aligned} \tag{3.12}$$

Layer 5: Defuzzification

$$\hat{y} = \frac{\hat{y}_l + \hat{y}_r}{2}. \tag{3.13}$$

IT2FNN uses backpropagation method with heuristic techniques (variable learning rate backpropagation, and resilient backpropagation), which were developed from a performance analysis of the standard steepest descent algorithm and numerical optimization techniques for IT2FNN training: conjugate gradient, quasi-Newton, and Levenberg-Marquardt for learning how to determine premise parameters (to find the parameters related to interval type-2 MFs) and consequent parameters. The learning procedure has two parts: in the first part the input patterns are propagated, the consequent parameters and the premise parameters are assumed to be fixed for the current cycle through the training set. In the second part the pattern are propagated again, and at this moment, backpropagation is used to modify the premise parameters, and consequent parameters. These two parts are considered an epoch.

Give an input-output training pair $\left\{\left(x_p : t_p\right)\right\} \forall p = 1, \ldots, q$, in order to get the design of the IT2FNN, the error function (E) must minimized.

$$e_p = t_p - \hat{y}_p \tag{3.14}$$

$$E_p = \frac{1}{2}e_p^2 = \frac{1}{2}\left(t_p - \hat{y}_p\right)^2 \tag{3.15}$$

$$E = \sum_{p=1}^{q} E_p \tag{3.16}$$

3.2.2 IT2FNN-2 Model

The IT2FNN-2 model has 6 layers (Fig. 3.7), if uses NN for fuzzifying the inputs (layers 1 to 2). The non-adaptive nodes in the rules layer (layer 3) interconnect with the lower-upper linguistic values layer (layer 2) to generate TSK IT2FIS rules antecedents. The non-adaptive nodes in the consequents layer (layer 4) are connected with the input layer (layer 0) to generate rule consequents. The non-adaptive nodes in type-reduction layer (layer 5) evaluate left-right values with KM algorithm. The non-adaptive node in defuzzification layer (layer 6) averages left-right values.

The forward-propagation procedure is described as follows:

Layer 0: Inputs

$$x = (x_1, \ldots, x_i, \ldots, x_n)^t$$

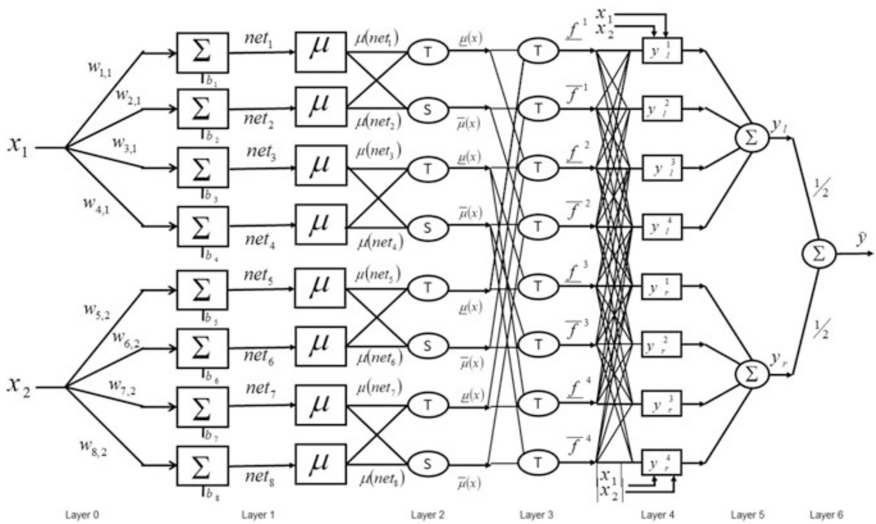


Fig. 3.7 IT2FNN-2 architecture

Layer 1: Every node ℓ in this layer is a square (Fig. 3.7) with node function

for $k = 1$ to M

for $i = 1$ to n

$$\begin{aligned} {}^{1}net_{k,i} &= {}^{1}w_{k,i}x_i + {}^{1}b_{k,i};\, {}^{1}\mu_{k,i}(x_i) = \mu\left({}^{1}net_{k,i}\right), \\ {}^{2}net_{k,i} &= {}^{2}w_{k,i}x_i + {}^{2}b_{k,i};\, {}^{2}\mu_{k,i}(x_i) = \mu\left({}^{2}net_{k,i}\right) \end{aligned} \tag{3.17}$$

end

end

where μ is the transfer function which can be Gaussian, GBell or logistic (e.g. Gaussian with uncertain mean "igaussmtype2" and transfer function GBell with uncertain mean "igbellmtype2") (see Fig. 3.9b).

Layer 2: Every node ℓ in this layer is a circle labeled with *T*-norm and *S*-norm alternated.

$$\begin{aligned} &\underline{\mu}_{k,i}(x_i) = {}^{1}\mu_{k,i}(x_i) \cdot {}^{2}\mu_{k,i}(x_i) \\ &\overline{\mu}_{k,i}(x_i) = {}^{1}\mu_{k,i}(x_i) + {}^{2}\mu_{k,i}(x_i) - \underline{\mu}_{k,i}(x_i) \\ &k = 1, 2, \ldots, M;\, i = 1, 2, \ldots, n \end{aligned} \tag{3.18}$$

Layer 3 to 6: This layer are equivalent to layers 2 to 5 on the IT2FNN-1 architecture.

3.2.3 IT2FNN-3 Model

The IT2FNN-3 model has 7 layers (Fig. 3.8). Layer 1 has adaptive nodes for fuzzifying the inputs; layer 2 has non-adaptive nodes with the interval fuzzy values (Fig. 3.9c). Layer 3 (rules) has non-adaptive nodes for generating firing strength of TSK IT2FIS rules. Layer 4, lower and upper values the rules firing strength are normalized. The adaptive nodes in layer 5 (consequent) are connected to layer 0 for generating the rules consequents. The non-adaptive nodes in layer 6 evaluate values from the left-right interval. The non-adaptive node in layer 7 (defuzzification) evaluates average of interval left-right values.

The forward-propagation procedure is described as follows: the first 3 layers (0 to 3) are identical to the corresponding layers on the IT2FNN-2 architecture.

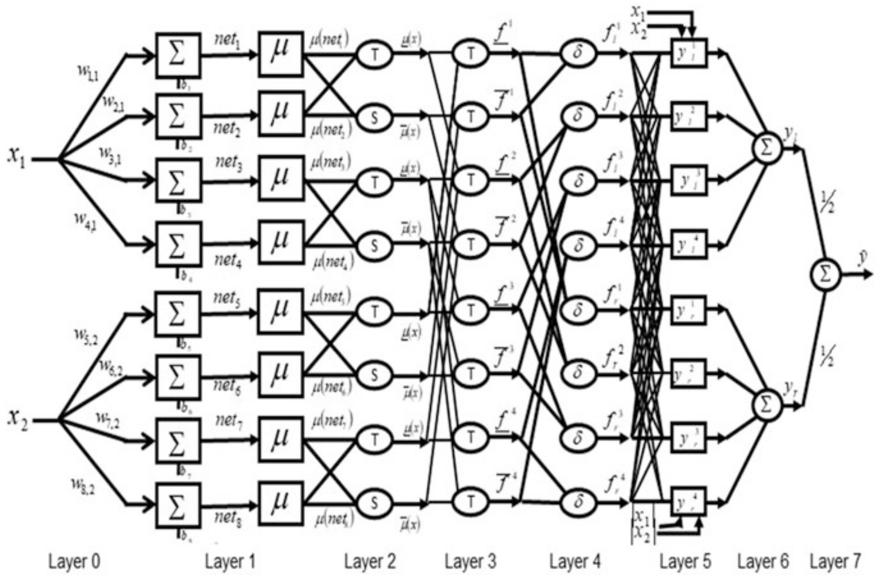


Fig. 3.8 IT2FNN-3 architecture

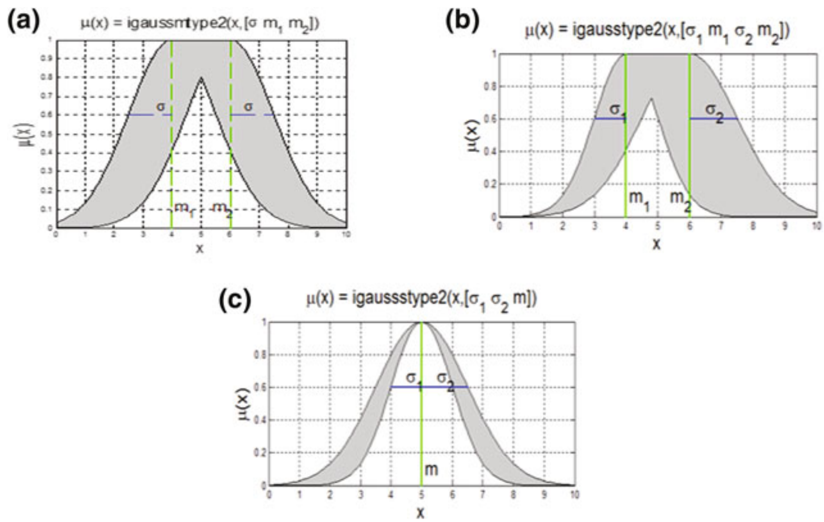


Fig. 3.9 The MFs used for training the IT2FNN architectures

Layer 4: Every node ℓ in this layer is a circle labeled $\wp$ which evaluates the left-most and right-most firing points denoted by:

$$f_l^k = \frac{\overline{w}_l^k \overline{f}^k + \underline{w}_l^k \underline{f}^k}{\overline{w}_l^k + \underline{w}_l^k}; f_r^k = \frac{\underline{w}_r^k \underline{f}^k + \overline{w}_r^k \overline{f}^k}{\underline{w}_r^k + \overline{w}_r^k} \tag{3.19}$$

where w values are adjustable weights.

Layer 5 is equivalent to layer 3 on the IT2FNN-1 architecture

Layer 6: The two nodes in this layer are circles labeled with "Σ" that evaluates the two end-points, y_l and y_r:

$$\hat{y}_l = \frac{\sum_{k=1}^{M} f_l^k \cdot y_l^k}{\sum_{k=1}^{M} f_l^k}; \hat{y}_r = \frac{\sum_{k=1}^{M} f_r^k \cdot y_r^k}{\sum_{k=1}^{M} f_r^k} \tag{3.20}$$

Layer 7: The single node in this layer is a circle labeled "Σ" that computes the output.

$$\widehat{y} = \frac{\hat{y}_l + \hat{y}_r}{2} \tag{3.21}$$

3.3 Fuzzy Integrators

The design of the type-1 (Fig. 3.10) and interval type-2 (Fig. 3.11) fuzzy inference systems integrators are of Mamdani type and have 3 inputs (IT2FNN1, IT2FNN2 and IT2FNN3) and 1 output (Forecast), so each Input-Output variable is assigned two MFs with linguistic labels "Small and Large" and have 8 if-then rules. The design of the if-then rules for the fuzzy inference system depends on the number of membership functions used in each input variable using the system [e.g. the fuzzy inference system uses 3 input variables which each entry contains two membership functions, therefore the total number of possible combinations for the fuzzy-rules is 8 (e.g. 2*2*2 = 8)], therefore we used 8 fuzzy-rules for the experiments (Fig. 3.12) because the performance is better and minimized the prediction error of the Mackey-Glass, BMV, Dow Jones and NASDAQ time series.

In the type-1 fuzzy integrators we used different MFs [Gaussian, Generalized Bell, and Triangular (Fig. 3.13a)] and for the interval type-2 fuzzy integrators we used different MFs [igaussmtype2, igbelltype2 and itritype2 (Fig. 3.13b)] [23] to observe the behavior of each of them and determine which one provides better forecast of the time series.

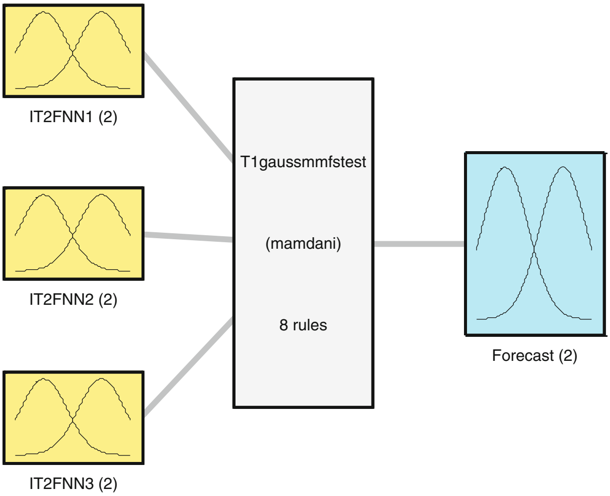


Fig. 3.10 Structure of the type-1 fuzzy integrator

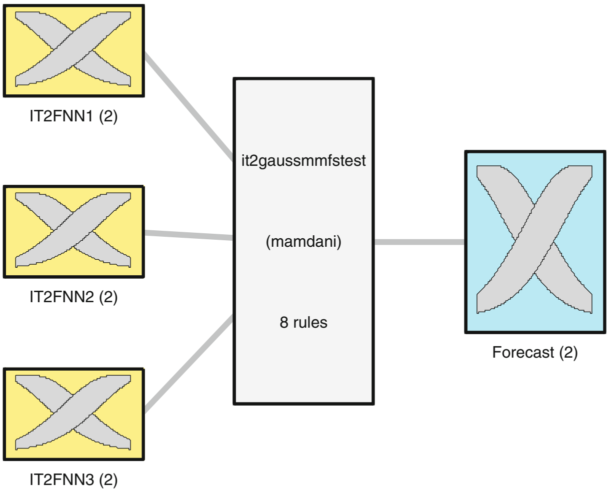


Fig. 3.11 Structure of the interval type-2 fuzzy integrator

1. If (IT2FNN1 is Small) and (IT2FNN2 is Small) and (IT2FNN3 is Small) then (Forecast is Small) (1)
2. If (IT2FNN1 is Small) and (IT2FNN2 is Small) and (IT2FNN3 is Large) then (Forecast is Small) (1)
3. If (IT2FNN1 is Small) and (IT2FNN2 is Large) and (IT2FNN3 is Small) then (Forecast is Small) (1)
4. If (IT2FNN1 is Small) and (IT2FNN2 is Large) and (IT2FNN3 is Large) then (Forecast is Large) (1)
5. If (IT2FNN1 is Large) and (IT2FNN2 is Small) and (IT2FNN3 is Small) then (Forecast is Small) (1)
6. If (IT2FNN1 is Large) and (IT2FNN2 is Small) and (IT2FNN3 is Large) then (Forecast is Large) (1)
7. If (IT2FNN1 is Large) and (IT2FNN2 is Large) and (IT2FNN3 is Small) then (Forecast is Large) (1)
8. If (IT2FNN1 is Large) and (IT2FNN2 is Large) and (IT2FNN3 is Large) then (Forecast is Large) (1)

Fig. 3.12 If-then rules for the fuzzy integrators

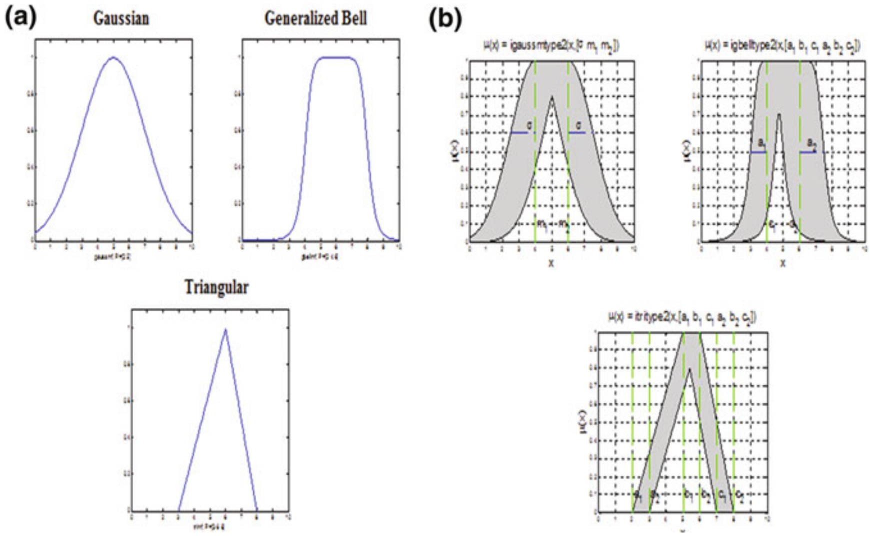


Fig. 3.13 Type-1 MFs (**a**) and interval type-2 MFs (**b**) for the fuzzy integrators

3.4 Optimization of the Fuzzy Integration with the Genetic Algorithm

We optimize the parameters values of the MFs in each type-1 and interval type-2 fuzzy integrators with GAs. The representation of GAs is of Real-Values and the chromosome size will depend of the number of MFs that are used in each design of the type-1 and interval type-2 fuzzy inference system integrators.

The objective function is defined to minimize the prediction error as follows:

$$f(t) = \sqrt{\frac{\sum_{t=1}^{n} (a_t - p_t)^2}{n}} \tag{3.22}$$

where a, corresponds to the real data of the time series, p corresponds to the output of the fuzzy integrators, t is de sequence time series, and n is the number of data points of time series.

The general structure of the chromosome (individuals) represents the parameters of the MFs of fuzzy integrators. The number of parameters varies according to the type of the MFs for the type-1 fuzzy system (e.g. two parameter are needed to represent a Gaussian MF are "σ" and "μ") for this case the GA optimized 16 parameters of the type-1 fuzzy integrator (Fig. 3.14). The interval type-2 fuzzy system (e.g. three parameter are needed to represent "igaussmtype2" MF's are "σ", "μ_1" and "μ_2") for this case the GA optimized 24 parameters of the interval type-2 fuzzy integrator (Fig. 3.15). Therefore the number of parameters that each fuzzy inference system integrator has depends of the MFs type (Fig. 3.13) assigned to each input and output variables.

The optimization was performed for the parameter values of the MFs (inputs and outputs) of fuzzy integrators. The parameters for the Genetic algorithm used to optimize the fuzzy integrators are described in Table 3.1.

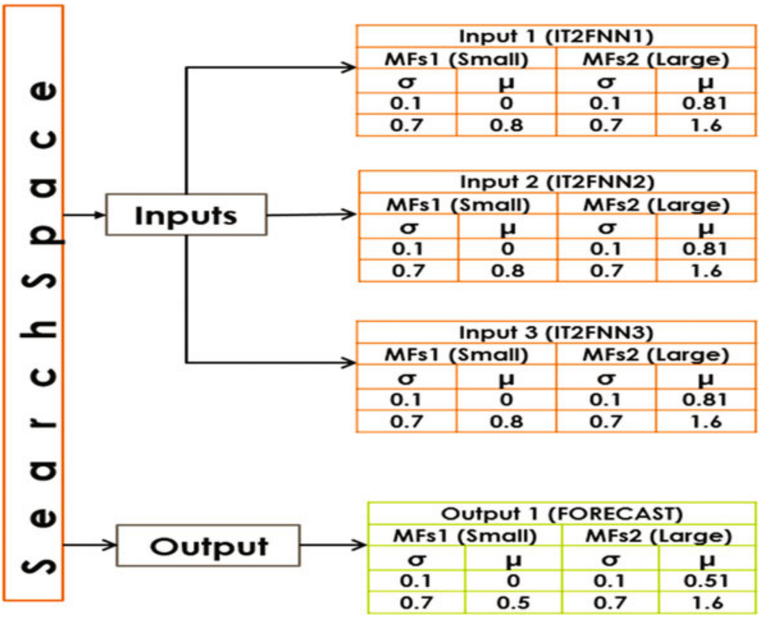


Fig. 3.14 Representation of chromosome of the GAs for the optimization of the Gaussian MFs

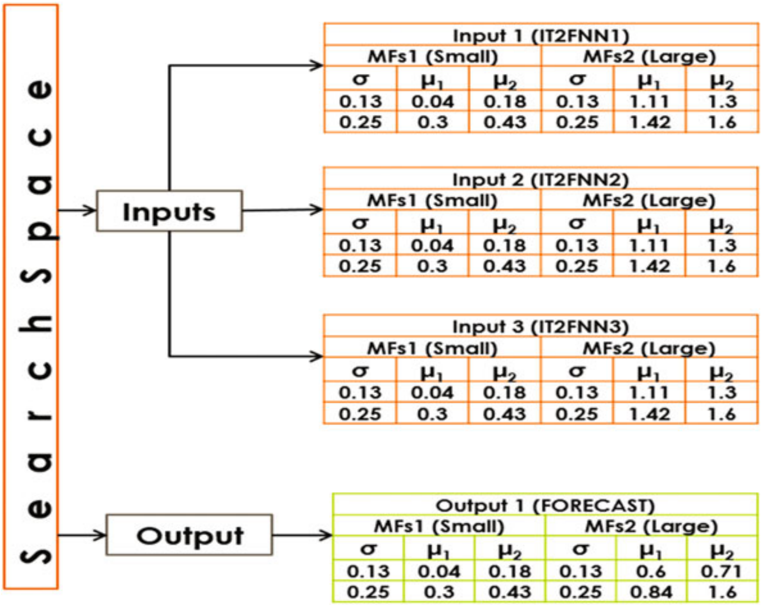


Fig. 3.15 Representation of chromosome of the GA for the optimization of the igaussmtype2 MFs

Table 3.1 The parameters of the GA used for optimization the fuzzy integrators

Representation of phenotypic	Real-values
Selection	Stochastic Universal Sample
Crossing or recombination	Discrete Recombination [0.8]
Mutation	0.1
Individuals	100
Generations	100
Iterations or run	31

3.5 Optimization of the Fuzzy Integrators with the Particle Swarm Optimization

We optimize the parameter values of the MFs in each type-1 and interval type-2 fuzzy integrators with PSO. The representation in PSO is of Real-Values and the particle size will depend on the number of the MFs that are used in each design of the fuzzy integrators.

The objective function to evaluate the performance of the PSO is similar to the Eq. (3.22). The general structure of the particles represents the parameters of the

Table 3.2 The parameters of the PSO used for optimization the fuzzy integrators

Parameters	Value
Particles	100
Iterations	65
Inertia weight "ω"	Linear decrement [0.88 − 0]
Constriction "C"	Linear increment [0.01 − 0.9]
r_1, r_2	Random
c_1	Linear decrement [2 − 0.5]
c_2	Linear increment [0.5 − 2]

MFs of the fuzzy integrators similar to the chromosome of the GAs (Figs. 3.14 and 3.15). Therefore the PSO are used to optimization the MFs (Fig. 3.13) of fuzzy integrators. The parameters for the PSO used to optimize the fuzzy integrators are described in Table 3.2.

References

1. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. 23, 665–685 (1992)
2. Jang, J.S.R.: Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 762–767 (1991)
3. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science **197**, 287–289 (1997)
4. Mackey, M.C.: Mackey-Glass. McGill University, Canada, http://www.sholarpedia.org/-article/Mackey-Glass_equation. 5 Sept 2009
5. Gaxiola, F., Melin, P., Valdez, F., Castillo, O.: Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction. Inf. Sci. **260**, 1–14 (2014)
6. Melin, P., Soto, J., Castillo, O., Soria, J.: A new approach for time series prediction using ensembles of ANFIS models. Experts Syst. Appl. **39**(3), 3494–3506 (2012)
7. Pulido, M., Melin, P., Castillo, O.: Genetic optimization of ensemble neural networks for complex time series prediction. IJCNN, pp. 202–206 (2011)
8. Pulido, M., Melin, P., Castillo, O.: Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. Inf. Sci. **280**, 188–204 (2014)
9. Soto, J., Melin, P., Castillo, O.: Time series prediction using ensembles of ANFIS models with genetic optimization of interval type-2 and type-1 fuzzy integrators. Int. J. Hybrid Intell. Syst. **11**(3), 211–226 (2014)
10. Castellanos, S.G., Martínez, L.: Development of the Mexican Bond Market. In: Borensztein, E., Cowan, K., Eichengreen, B., Panizza, U. (eds.) Bond Markets in Latin America: On the Verge of a Big Bang?, pp. 51–58. MIT Press, Cambridge (2008)
11. Sidaoui, J.: The Mexican financial system: reforms and evolution 1995–2005. BIS Papers **28**, 277–293 (2006)
12. López, F., Santillán, R.J., Cruz, S.: Volatility dependence structure between the Mexican Stock Exchange and the World Capital Market. Investigación Económica **74**(293), 69–97 (2015)

13. https://es-us.finanzas.yahoo.com/q/hp?s=%5EMXX+Precios+historicos (7 May 2015)
14. Dow Jones Company. http://www.dowjones.com (10 Jan 2014)
15. Historic Dow Jones Data, Yahoo Finance, http://finance.yahoo.com (10 Jan 2014)
16. Dow Jones Indexes. http://www.djindexes.com (5 Sept 2014)
17. https://es-us.finanzas.yahoo.com/q/hp?s=%5EEDJI+Precios+historicos (8 May 2015)
18. http://business.nasdaq.com/discover/nasdaq-story/index.html (27 April 2015)
19. Blau, B.M., Van-Ness, B.F., Van-Ness, R.A.: Information in short selling: comparing NASDAQ and the NYSE. Rev. Fin. Econ. **20**(1), 1–10 (2011)
20. Pagano, M.S., Peng, L., Schwartz, R.A.: A call auction's impact on price formation and order routing: evidence from the NASDAQ stock market. J. Fin. Markets **16**(2), 331–361 (2013)
21. https://es-us.finanzas.yahoo.com/q/hp?s=%5EIXIC+Precios+historicos (9 May 2015)
22. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, NJ (2001)
23. Castro, J.R., Castillo, O., Martínez, L.G.: Interval type-2 fuzzy logic toolbox. Eng. Lett. **15**(1), 89–98 (2007)

Chapter 4
Simulation Studies

In this section we present results obtained of the ensemble of IT2FNN models and the use of fuzzy integrators as response optimized with GA and PSO algorithms for time series prediction.

4.1 Mackey-Glass Time Series

This section presents the simulation and test results obtained by applying the proposed prediction method to the Mackey-Glass time series for $\tau = 13, 17, 30, 34, 68, 100, 136$, using different approach of the ensemble of IT2FNN architectures and the two types of optimization of the fuzzy integrators with the GAs and PSO algorithms, used in this work.

4.1.1 Ensemble of the IT2FNN Architectures for Mackey-Glass

The ensemble of IT2FNN architectures has three models as follows: the IT2FNN-1 model optimizes the parameters of the "igaussmtype2" MFs (Fig. 3.9a), the learning rate is 0.03 and the desired error is 0.00001; the IT2FNN-2 model optimizes the parameters of the "igausstype2" MFs (Fig. 3.9b), the learning rate is

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2_4

0.011 and the desired error is 0.000001; and the IT2FNN-3 model optimizes the parameters of the "igaussstype2" MFs (Fig. 3.9c), the learning rate is 0.02 and the desired error is 0.0000001. The number the epochs for training the IT2FNN models is 800.

The obtained results of the ensemble of IT2FNN architectures are shown on Table 4.1. The best error is of 0.002517717 and the average error is of 0.00591527 with the IT2FNN-3 for the Mackey-Glass (τ = 13); the best error is of 0.000254857 and the average error is of 0.000513248 with the IT2FNN-1 for the Mackey-Glass (τ = 17); the best error is of 0.00089312 and the average error is of 0.004463189 with the IT2FNN-1 for the Mackey-Glass (τ = 30); the best error is of 0.000307511 and the average error is of 0.010427016 with the IT2FNN-3 for the Mackey-Glass (τ = 34); the best error is of 0.00085505 and the average error is of 0.003818732 with the IT2FNN-2 for the Mackey-Glass (τ = 68); the best error is of 0.000612878 and the average error is of 0.00327431 with the IT2FNN-1 for the Mackey-Glass (τ = 100); and the best error is of 0.000331059 and the average error is of 0.002382512 with the IT2FNN-1 for the Mackey-Glass (τ = 136).

Table 4.1 Results for the ensemble of IT2FNN for the Mackey-Glass time series

IT2FNN	RMSE	
	Best	Average
IT2FNN-1-Tau = 13	0.008596153	0.010146361
IT2FNN-2-Tau = 13	0.007919986	0.010166644
IT2FNN-3-Tau = 13	0.002517717	0.00591527
IT2FNN-1-Tau = 17	0.000258457	0.00513248
IT2FNN-2-Tau = 17	0.000281517	0.00554012
IT2FNN-3-Tau = 17	0.005466984	0.021365826
IT2FNN-1-Tau = 30	0.00089312	0.004463189
IT2FNN-2-Tau = 30	0.00124898	0.004503043
IT2FNN-3-Tau = 30	0.001404767	0.012277316
IT2FNN-1-Tau = 34	0.00077769	0.004371837
IT2FNN-2-Tau = 34	0.001349607	0.004943326
IT2FNN-3-Tau = 34	0.000307511	0.010427016
IT2FNN-1-Tau = 68	0.000988737	0.004635047
IT2FNN-2-Tau = 68	0.000855055	0.003818732
IT2FNN-3-Tau = 68	0.001238312	0.008696349
IT2FNN-1-Tau = 100	0.000612878	0.00327431
IT2FNN-2-Tau = 100	0.000782409	0.003720222
IT2FNN-3-Tau = 100	0.001152992	0.005881063
IT2FNN-1-Tau = 136	0.000331059	0.002382512
IT2FNN-2-Tau = 136	0.001351276	0.004299122
IT2FNN-3-Tau = 136	0.001133525	0.005586892

4.1.1.1 IT2FNN-1 Model

The forecast obtained for the IT2FNN-1 for the Mackey-Glass (τ = 17) time series shown in Fig. 4.1, the evolution error is shown in Fig. 4.2, and the optimization structure of the IT2FNN-1 with backpropagation learning algorithm show in Fig. 4.3, the forecast obtained for the Mackey-Glass (τ = 13 and τ = 30) time series shown in Figs. 4.4 and 4.5.

4.1.1.2 IT2FNN-2 Model

The forecast obtained for the IT2FNN-2 for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.6, the evolution error is shown in Fig. 4.7, and the optimization structure of IT2FNN-2 with backpropagation learning algorithm shown in Fig. 4.8, the forecast obtained for the Mackey-Glass (τ = 34 and τ = 68) time series shown in Figs. 4.9 and 4.10.

4.1.1.3 IT2FNN-3 Model

The forecast obtained for the IT2FNN-3 for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.11, the evolution error is shown in Fig. 4.12, and the

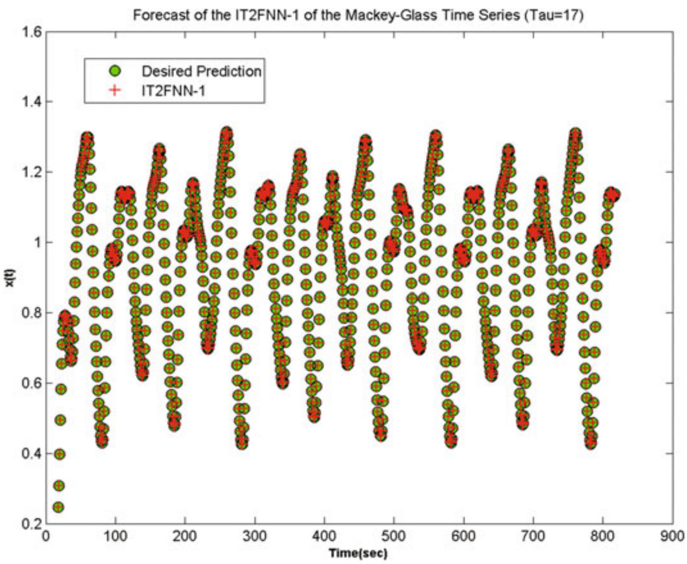


Fig. 4.1 Forecast of IT2FNN-1 for the Mackey-Glass (τ = 17) time series

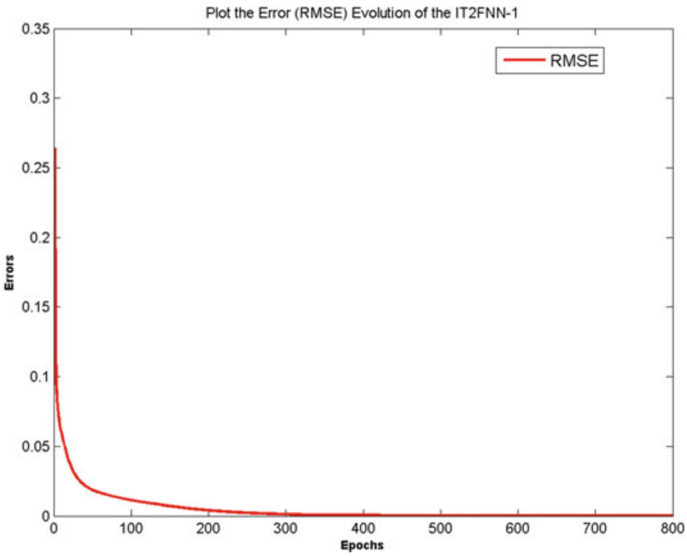


Fig. 4.2 Evolution error (RMSE) of IT2FNN-1 for the Mackey-Glass time series

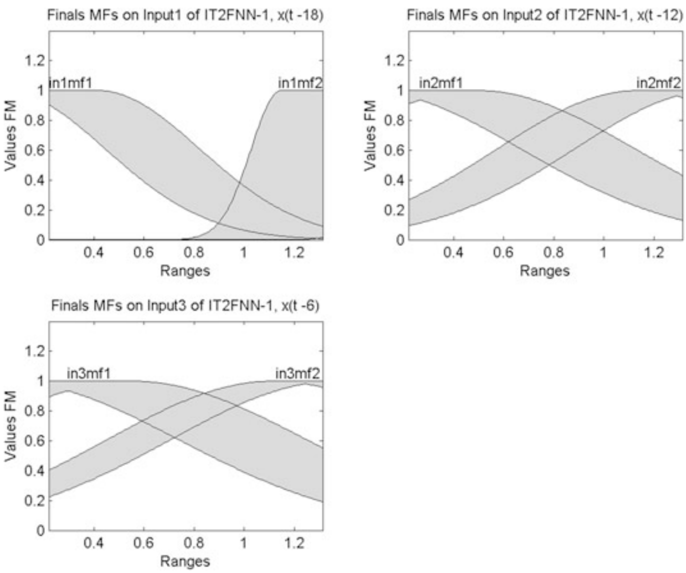


Fig. 4.3 Final MFs after training the IT2FNN-1 model

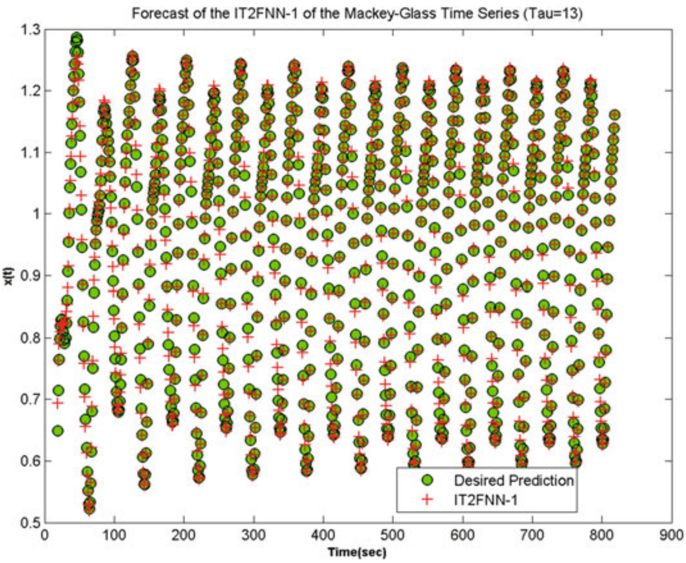


Fig. 4.4 Forecast of IT2FNN-1 for the Mackey-Glass (τ = 13) time series

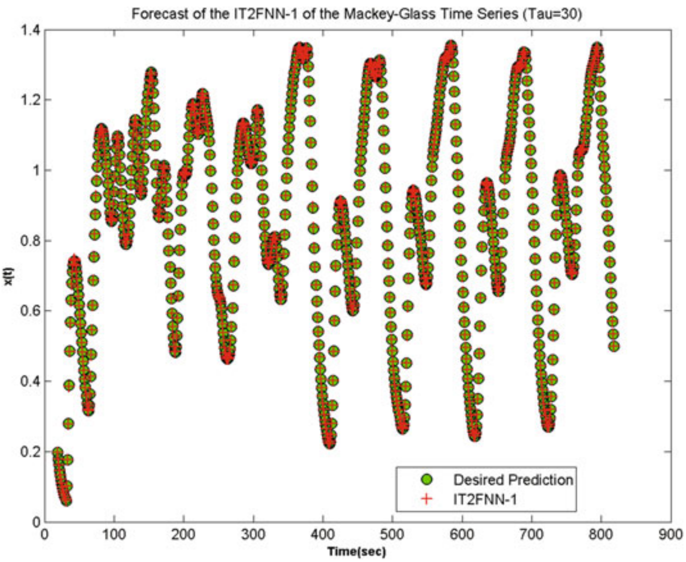


Fig. 4.5 Forecast of IT2FNN-1 for the Mackey-Glass (τ = 30) time series

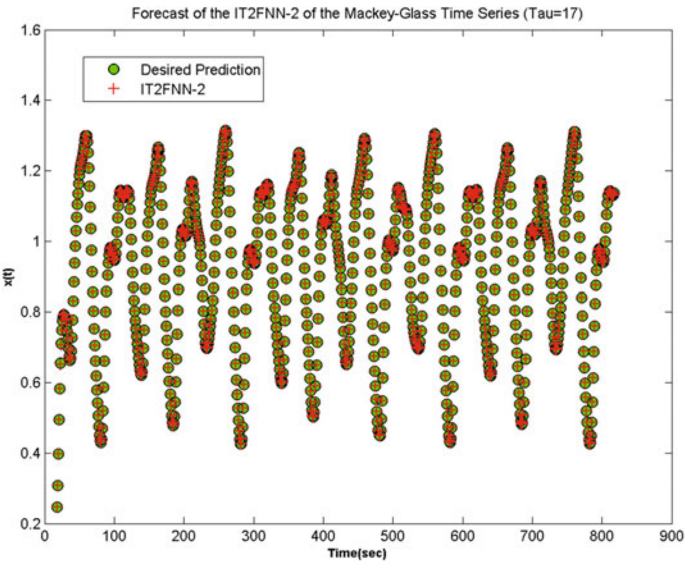


Fig. 4.6 Forecast of IT2FNN-2 for the Mackey-Glass (τ = 17) time series

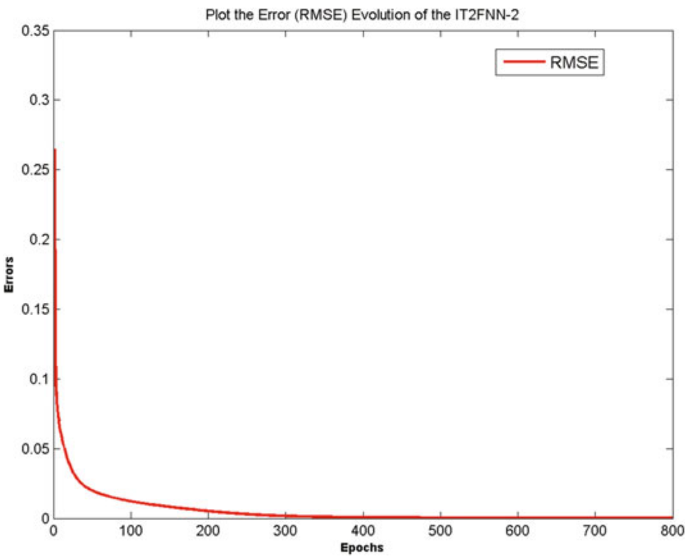


Fig. 4.7 Evolution error (RMSE) of IT2FNN-2 for the Mackey-Glass time series

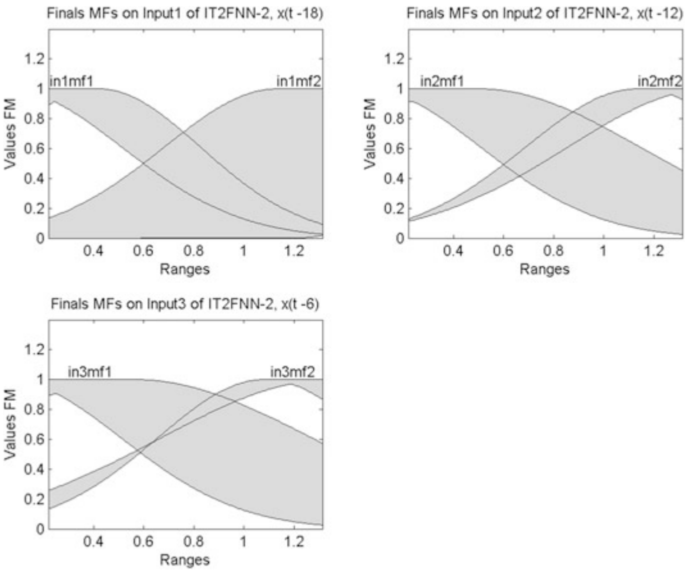


Fig. 4.8 Final MFs after training the IT2FNN-2 model

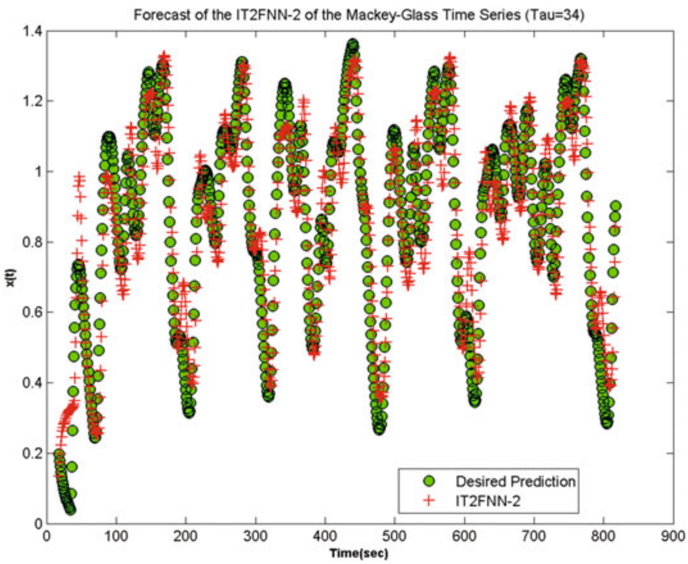


Fig. 4.9 Forecast of IT2FNN-2 for the Mackey-Glass (τ = 34) time series

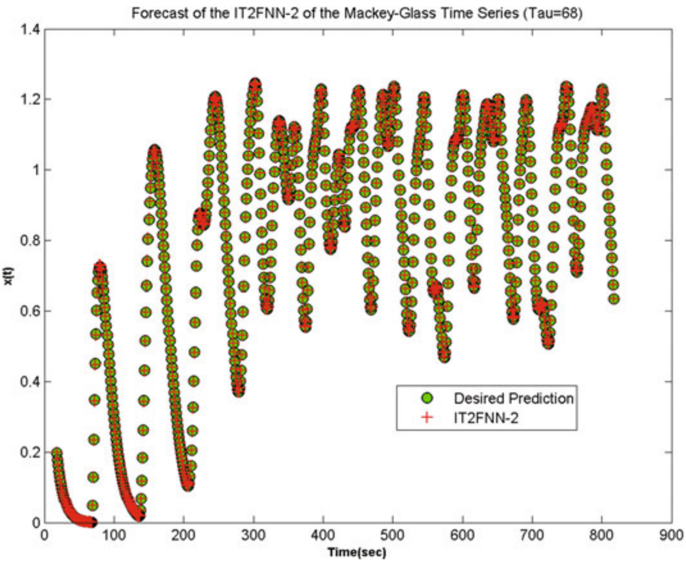


Fig. 4.10 Forecast of IT2FNN-2 for the Mackey-Glass ($\tau = 68$) time series

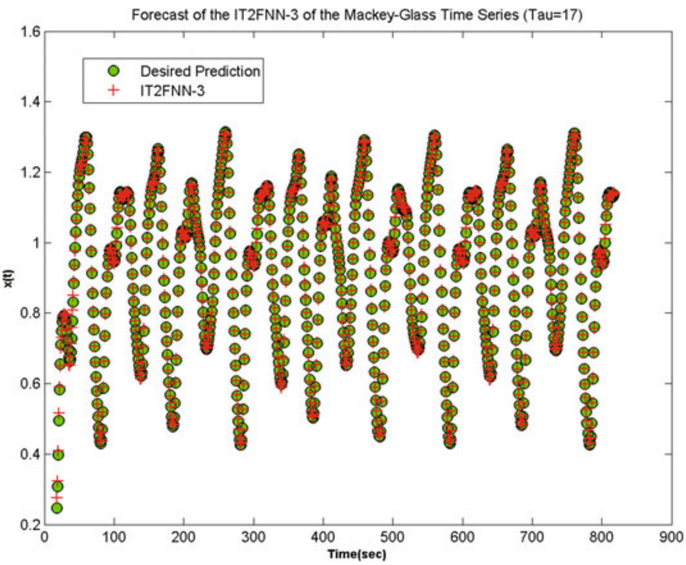


Fig. 4.11 Forecast of IT2FNN-3 for the Mackey-Glass ($\tau = 17$) time series

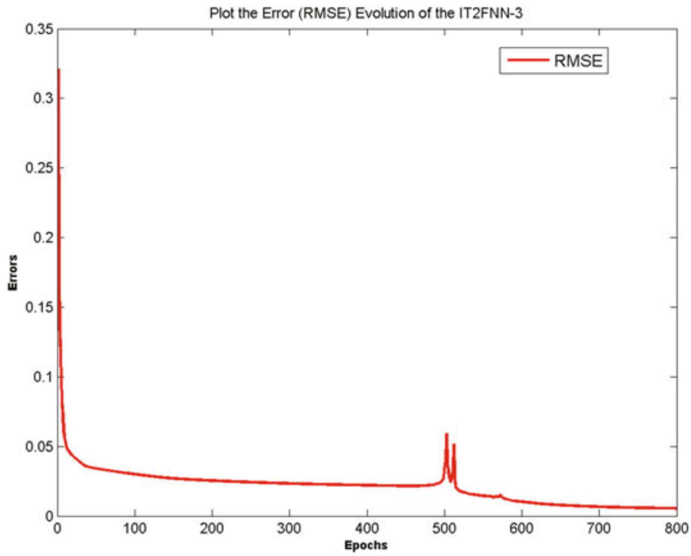


Fig. 4.12 Evolution error (RMSE) of IT2FNN-3 for the Mackey-Glass time series

optimization structure of IT2FNN-3 with backpropagation learning algorithm shown in Fig. 4.13, the forecast obtained for the Mackey-Glass ($\tau = 100$ and $\tau = 136$) time series is shown in Figs. 4.14 and 4.15.

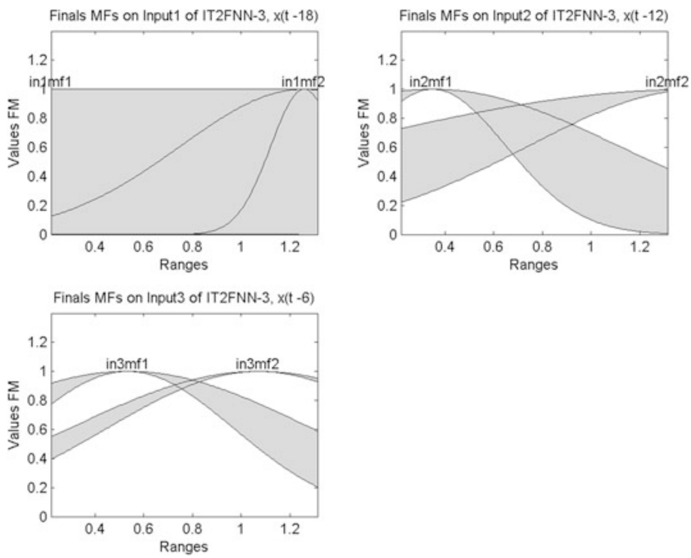


Fig. 4.13 Final MFs after training the IT2FNN-3 model

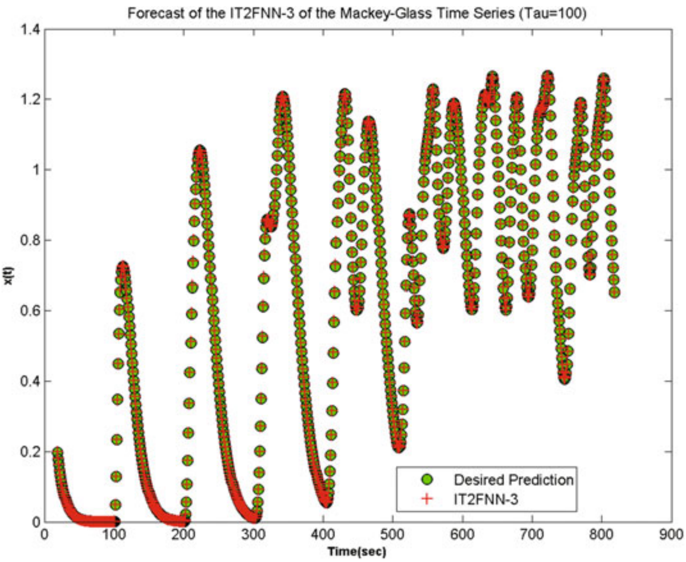


Fig. 4.14 Forecast of IT2FNN-3 for the Mackey-Glass (τ = 100) time series

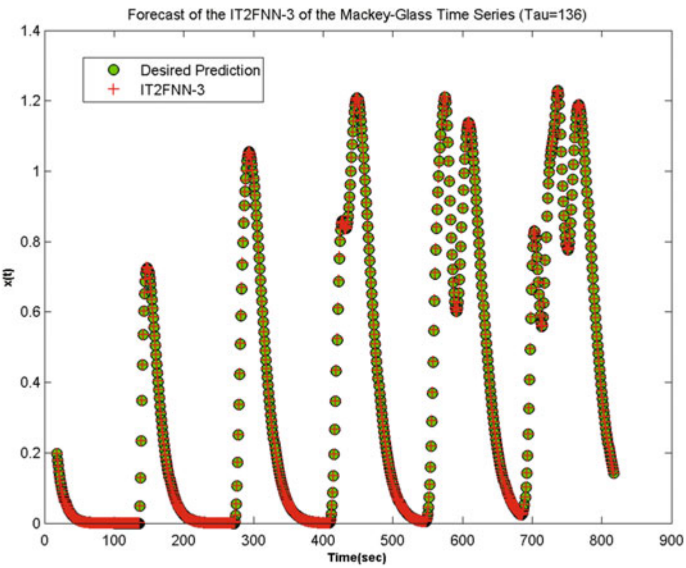


Fig. 4.15 Forecast of IT2FNN-3 for the Mackey-Glass (τ = 136) time series

4.1.2 *Optimization of the Fuzzy Integrators with the Genetic Algorithm*

The obtained results with optimized the fuzzy integrators with the GAs are shown on Table 4.2. The best error is of 0.02142164 and the average error is of 0.02255155 for the type-1 fuzzy integrator (T1FIS) using “Gbell” MFs, and the best error is of 0.02023097 and the average error is of 0.02033528 for the interval type-2 fuzzy integrator (IT2FIS) using “itritype2” MFs for the Mackey-Glass (τ = 17) time series.

We are presenting 10 experiments in Table 4.2, but the average error was calculated considering 30 experiments with the same parameters and conditions for the GAs. Therefore to evaluate the performance of the 30 experiments for this work, we applied different metrics to calculated average errors as shown in Table 4.3.

The forecast obtained of the optimized T1FIS using “Gauss” MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.16, the performance of the evolution error is shown in Fig. 4.17, and the optimization structure of T1FIS using “Gauss” MFs with the GAs shown in Fig. 4.18.

The forecast obtained of the optimized T1FIS using “Gbell” MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.19, the performance of the evolution error is shown in Fig. 4.20, and the optimization structure of T1FIS using “Gbell” MFs with the GAs shown in Fig. 4.21.

The forecast obtained of optimized the T1FIS using “Triangular” MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.22, the performance of the evolution error is shown in Fig. 4.23, and the optimization the structure of T1FIS using “Triangular” MFs with the GAs shown in Fig. 4.24.

The forecast obtained of the optimized interval type-2 fuzzy integrators using “igaussmtype2” MFs for the Mackey-Glass (τ = 17) time series is shown in

Table 4.2 Result of the optimization of fuzzy integrator with the GAs

No. exp.	Type-1 fuzzy integrators			Interval type-2 fuzzy integrators		
	Gaussian	Gbell	Triangular	igaussmtype2	igbelltype2	itritype2
1	0.02359056	0.02629806	0.08221695	0.021158482	0.02118375	0.02071019
2	0.0228442	0.02433066	0.08161724	0.021033109	0.02081994	0.02043439
3	0.0223283	0.02338475	0.08161615	0.021001443	0.02061247	0.02035049
4	0.02209832	0.02249878	0.08161613	0.020976213	0.02056261	0.02032144
5	0.02189447	0.02163905	0.08161613	0.020962286	0.02052157	0.02030222
6	0.02173872	0.02154845	0.08161613	0.020947947	0.02049882	0.02027343
7	0.02165485	0.02148594	0.08161613	0.020936484	0.02048822	0.02025226
8	0.02159362	0.02146486	0.08161613	0.020921625	0.02047444	0.02024165
9	0.02156282	0.02144333	0.08161613	0.020913384	0.02045962	0.02023573
10	0.02152446	0.02142164	0.08161613	0.020907752	0.02044867	0.02023097
Average	0.02208303	0.02255155	0.08167632	0.020975873	0.02060701	0.02033528

Table 4.3 Performance of the optimization of fuzzy integrator with the GAs

Metrics	Type-1 fuzzy integrators			Interval type-2 integrators		
	Gaussian	Gbell	Triangular	igaussmtype2	igbelltype2	itritype2
RMSE (Best)	0.02152446	0.021421638	0.081616127	0.020907752	0.020448674	0.02023097
RMSE (Average)	0.022083031	0.022551551	0.081676324	0.020975873	0.02060701	0.020335278
MSE	0.000563653	0.000593601	0.007011427	0.000455647	0.000447241	0.000435666
MAE	0.016962307	0.017643378	0.066130265	0.015458052	0.015277128	0.015126578
MPE	−8.337207925	−5.889802358	−1.418200646	−4.415973906	−3.886701193	−2.082901948
MAPE	1.917797038	1.977747607	7.491090409	1.746785771	1.731109592	1.696076726

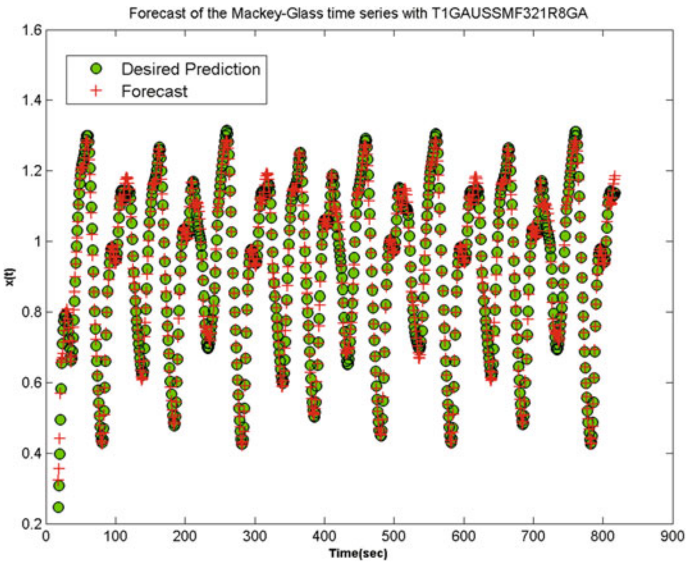


Fig. 4.16 Forecast of T1FIS using "Gauss" MFs for the Mackey-Glass time series

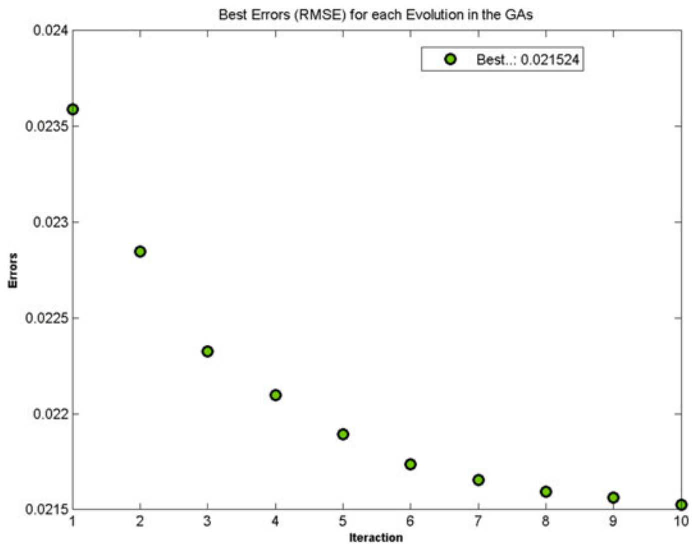


Fig. 4.17 Evolution error (RMSE) of the GAs for the T1FIS using "Gauss" MFs

Fig. 4.25, the performance of the evolution error is shown in Fig. 4.26, and the optimization structure of the interval type-2 fuzzy integrators using "igaussmtype2" MFs with the GAs is shown in Fig. 4.27.

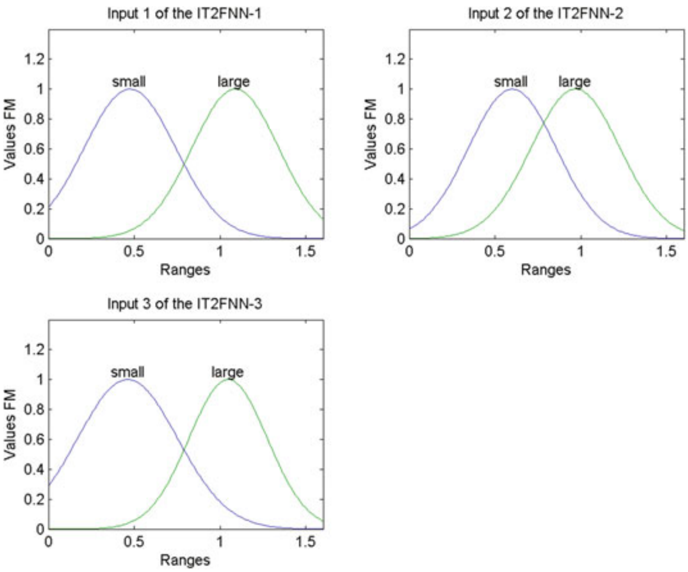


Fig. 4.18 Final MFs after optimized the T1FIS using "Gauss" MFs

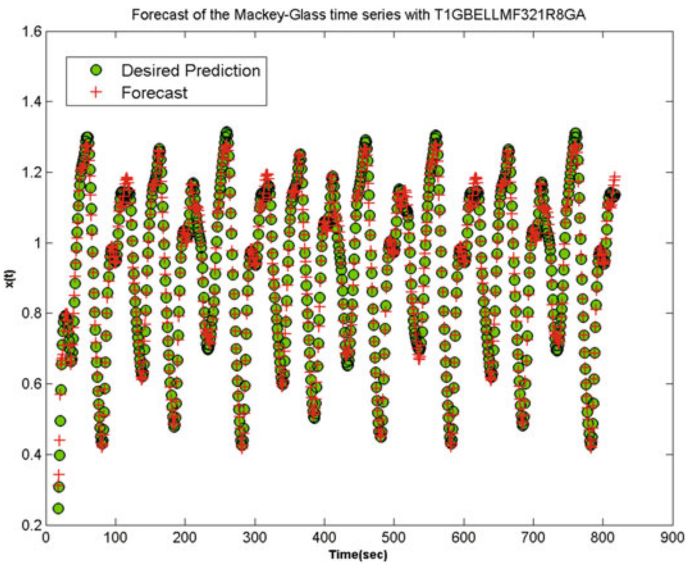


Fig. 4.19 Forecast of T1FIS using "Gbell" MFs for the Mackey-Glass time series

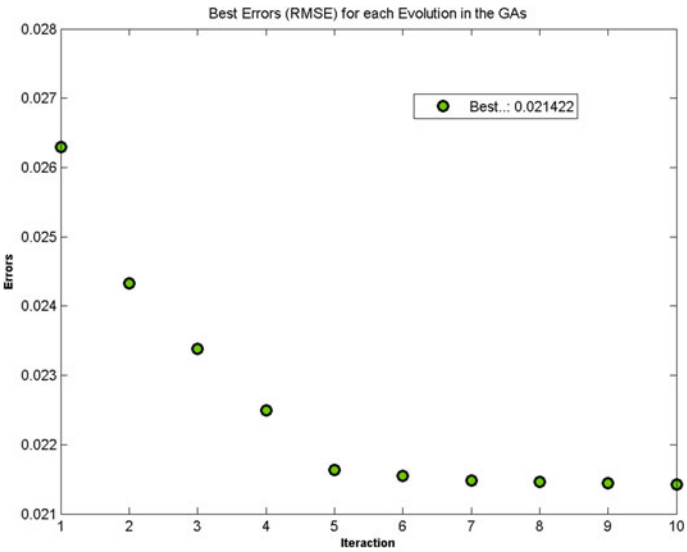


Fig. 4.20 Evolution error (RMSE) of the GAs for the T1FIS using "Gbell" MFs

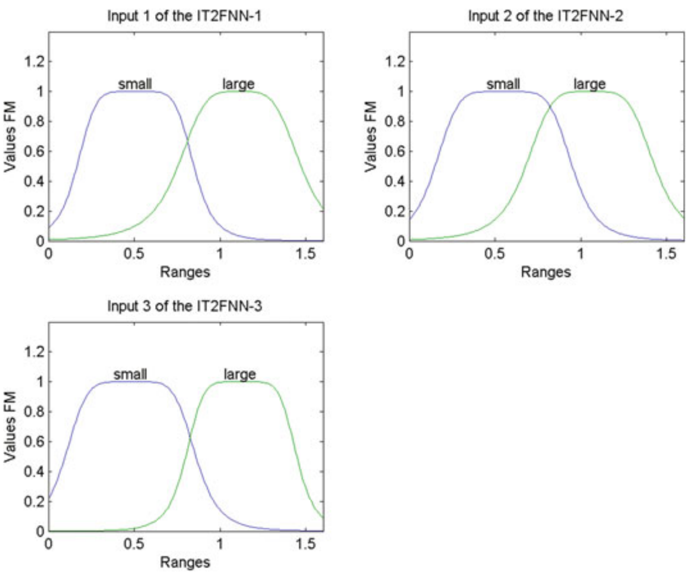


Fig. 4.21 Final MFs after optimized the T1FIS using "Gbell" MFs

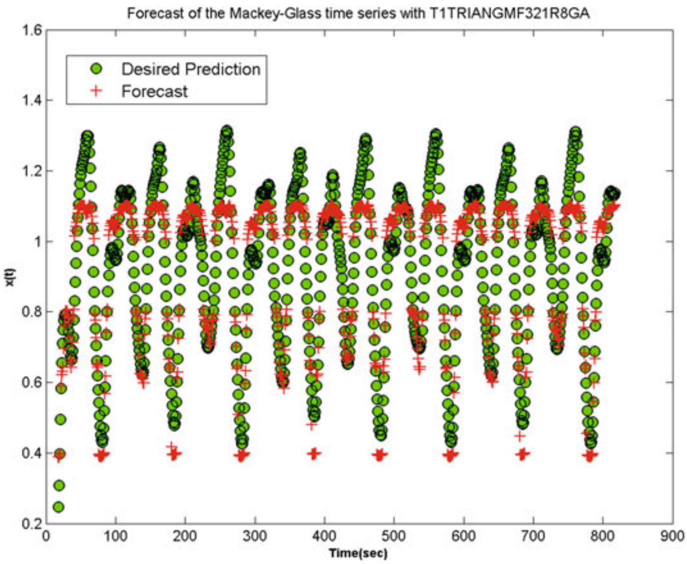


Fig. 4.22 Forecast of T1FIS using "Triangular" MFs for the Mackey-Glass time series

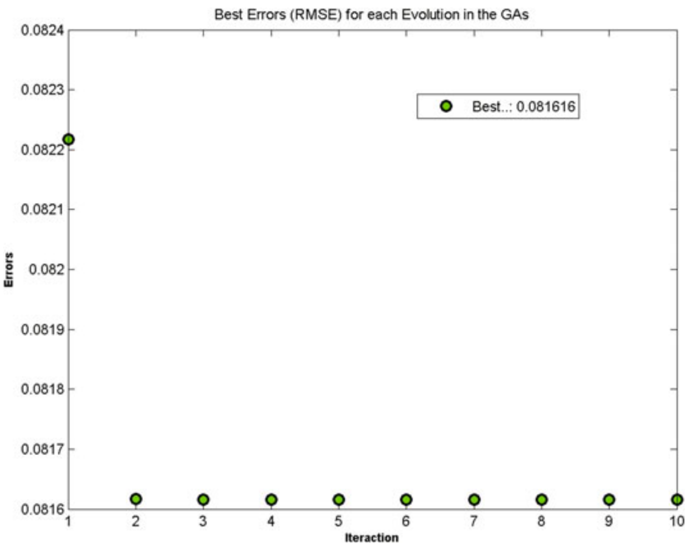


Fig. 4.23 Evolution error (RMSE) of the GAs for the T1FIS using "Triangular" MFs

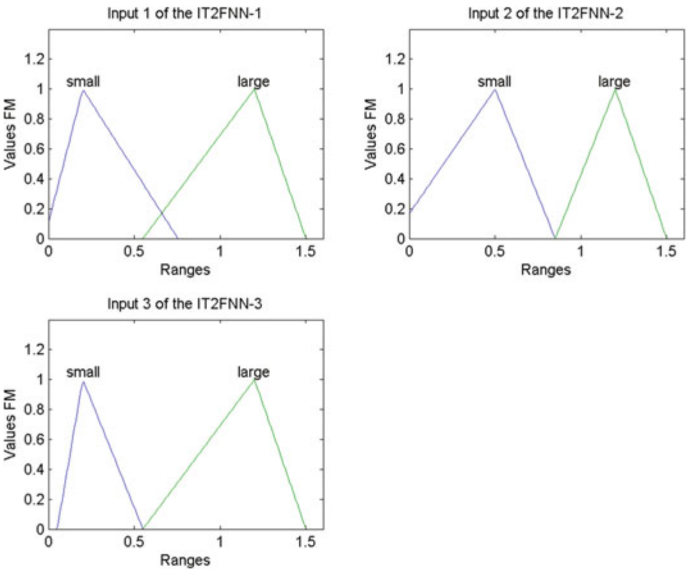


Fig. 4.24 Final MFs after optimized the T1FIS using "Triangular" MFs

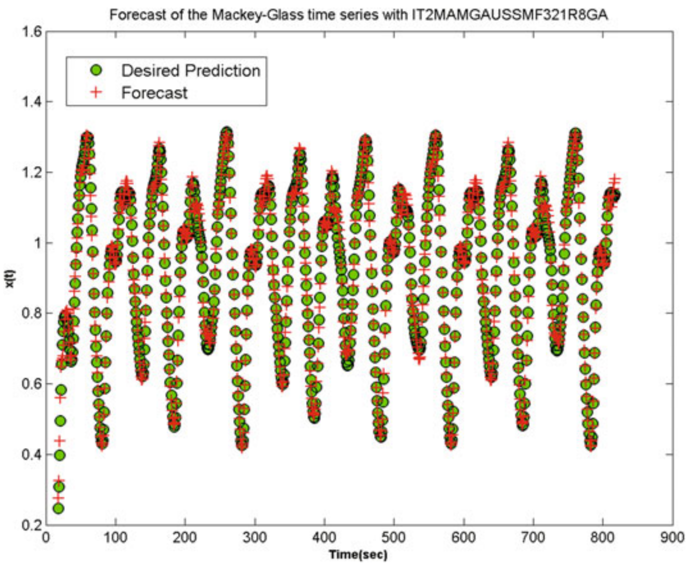


Fig. 4.25 Forecast of IT2FIS using "igaussmtype2" MFs for the Mackey-Glass time series

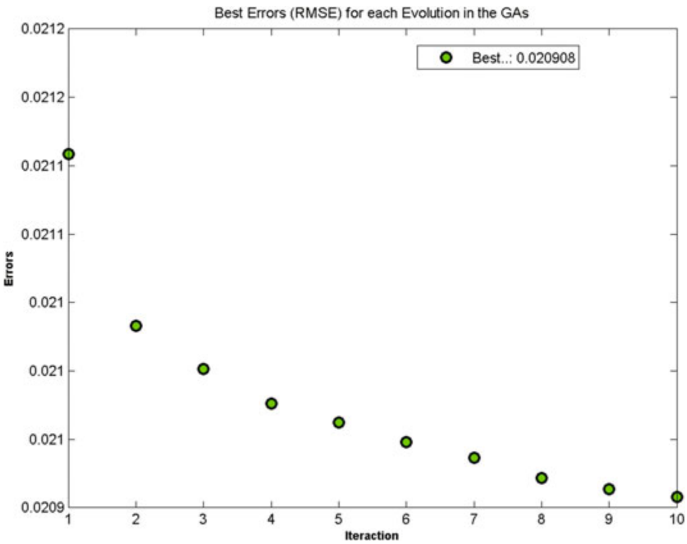


Fig. 4.26 Evolution error (RMSE) of the GAs for the IT2FIS using "igaussmtype2" MFs

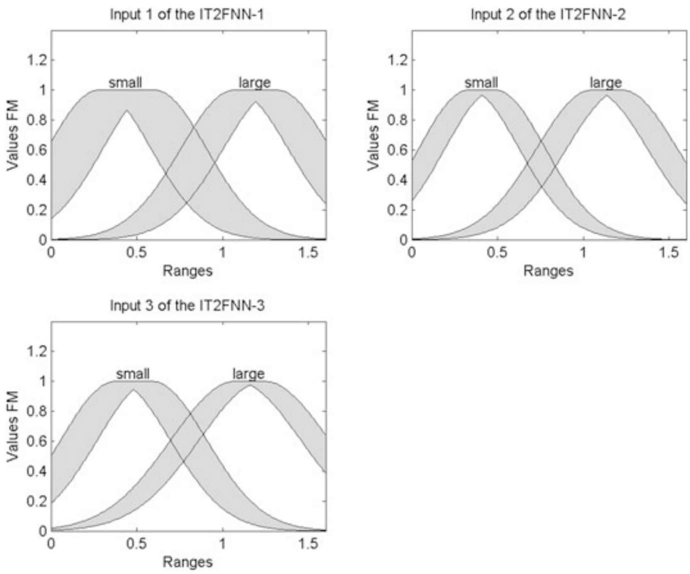


Fig. 4.27 Final MFs after optimized the IT2FIS using "igaussmtype2" MFs

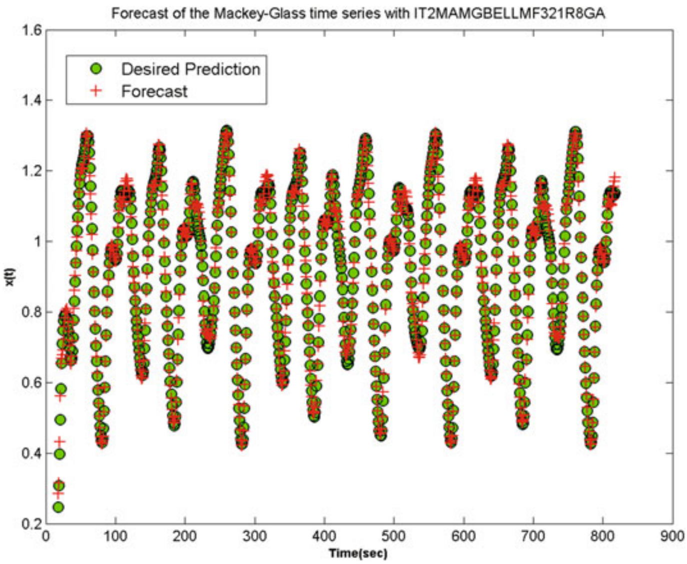


Fig. 4.28 Forecast of IT2FIS using "igbelltype2" MFs for the Mackey-Glass time series

The forecast obtained of the optimized interval type-2 fuzzy integrators using "igbelltype2" MFs for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.28, the performance of the evolution error is shown in Fig. 4.29, and the optimization

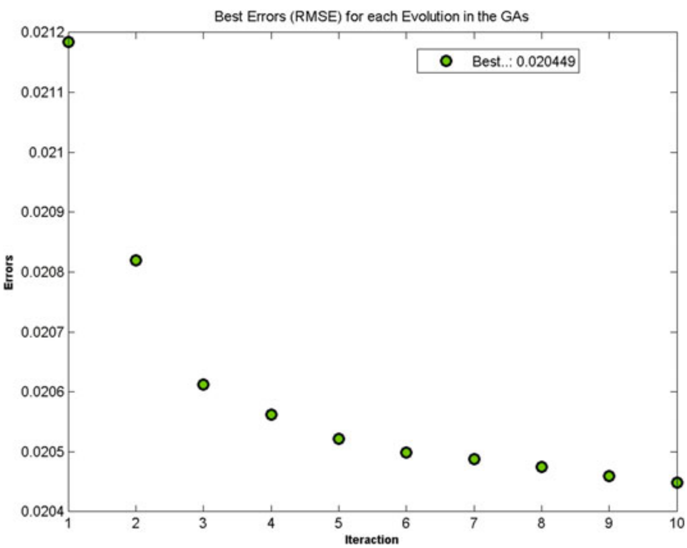


Fig. 4.29 Evolution error (RMSE) of the GAs for the IT2FIS using "igbelltype2" MFs

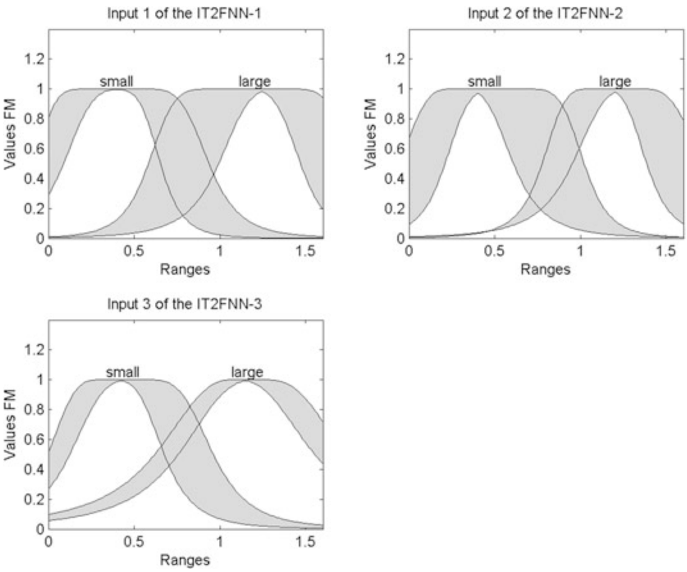


Fig. 4.30 Final MFs after optimized the IT2FIS using "igbelltype2" MFs

structure of the interval type-2 fuzzy integrators using "igbelltype2" MFs with the GAs is shown in Fig. 4.30.

The forecast obtained of optimized the interval type-2 fuzzy integrators using "itritype2" MFs for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.31, the performance of the evolution is error shown in Fig. 4.32, and the optimization structure of the interval type-2 fuzzy integrators using "itritype2" MFs with GAs shown in Fig. 4.33.

4.1.3 Optimization of the Fuzzy Integrators with the Particle Swarm Optimization

The obtained results with optimized the fuzzy integrators with the PSO are shown on Table 4.3. The best error is of 0.035228102 and the average error is of 0.036484603 for the type-1 fuzzy integrator using "Gbell" MFs, and the best error is of 0.023691987 and the average error is of 0.023691987 for the interval type-2 fuzzy integrator using "igbellype2" MFs for the Mackey-Glass (τ = 17) time series.

We are presenting 10 experiments in Table 4.4, but the average error was calculated considering 30 experiments with the same parameters and conditions for the PSO. Therefore to evaluate the performance of the 30 experiments for this work, we applied different metrics to calculate average errors as shown in Table 4.5.

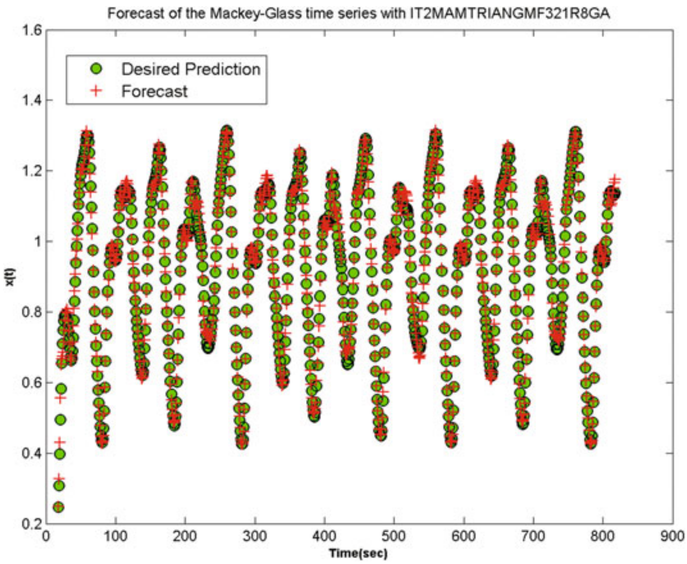


Fig. 4.31 Forecast of IT2FIS using “itritype2” MFs for the Mackey-Glass time series

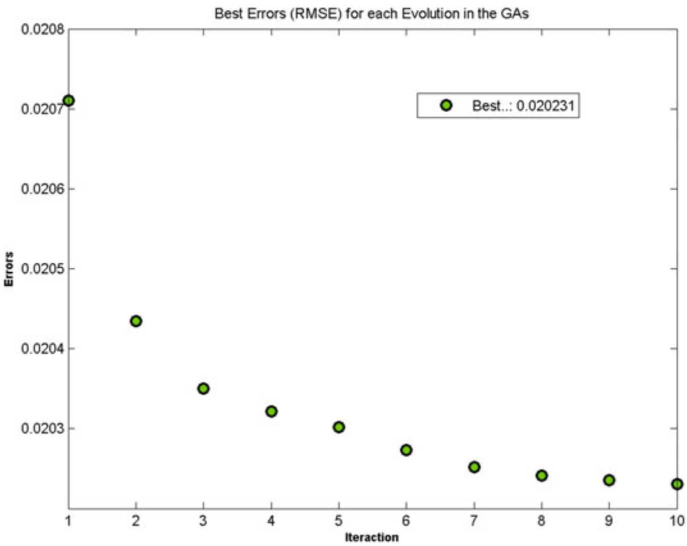


Fig. 4.32 Evolution error (RMSE) of the GAs for the IT2FIS using “itritype2” MFs

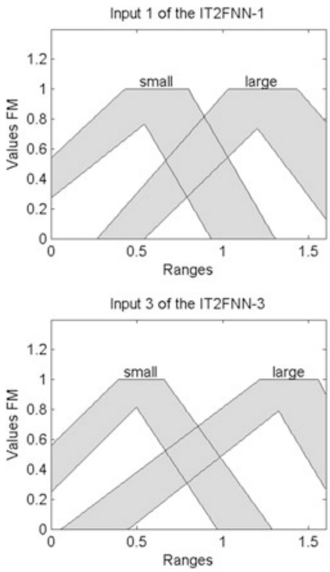


Fig. 4.33 Final MFs after optimized the IT2FIS using "itritype2" MFs

The forecast obtained of the optimized T1FIS using "Gaussian" MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.34, the performance of the evolution error is shown in Fig. 4.35, and the optimization structure of T1FIS using "Gaussian" MFs with the PSO is shown in Fig. 4.36.

The forecast obtained of the optimized T1FIS using "Gbell" MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.37, the performance of the evolution error is shown in Fig. 4.38, and the optimization structure of T1FIS using "Gbell" MFs with the PSO is shown in Fig. 4.39.

The forecast obtained of optimized the T1FIS using "Triangular" MFs for the Mackey-Glass (τ = 17) time series shown in Fig. 4.40, the performance of the evolution error is shown in Fig. 4.41, and the optimization structure of the T1FIS using "Triangular" MFs with the PSO is shown in Fig. 4.42.

The forecast obtained of the optimized interval type-2 fuzzy integrators using "igaussmtype2" MFs for the Mackey-Glass (τ = 17) time series is shown in

Table 4.4 Result of the optimization of fuzzy integrator with the PSO

No. Exp.	Type-1 fuzzy integrators			Interval type-2 fuzzy integrators		
	Gaussian	Gbell	Triangular	igaussmtype2	igbelltype2	itritype2
1	0.038239735	0.038527952	0.081862239	0.024438775	0.023724407	0.025115433
2	0.038181885	0.037436519	0.081612326	0.024433153	0.023720482	0.025115387
3	0.038066524	0.037057909	0.081426474	0.024418709	0.023719104	0.025115354
4	0.037946889	0.036709338	0.081283351	0.024347737	0.023713049	0.025115331
5	0.03760558	0.036438493	0.081142876	0.024287919	0.023702786	0.025115316
6	0.037294437	0.036133836	0.081004562	0.024261472	0.023687797	0.025115285
7	0.036723101	0.035971393	0.080849129	0.024243773	0.023678693	0.025115256
8	0.036510778	0.035753022	0.080176079	0.024228428	0.023670658	0.025115218
9	0.036203161	0.03558947	0.079900848	0.024208057	0.023654475	0.025115174
10	0.035946912	0.035228102	0.079753554	0.024183221	0.023648414	0.025115114
Average	0.0372719	0.036484603	0.080901144	0.024305124	0.023691987	0.025115287

Table 4.5 Performance of the optimization of fuzzy integrators with the PSO

Metrics	Type-1 fuzzy integrator			Type-2 fuzzy integrator		
	Gaussian	Gbell	Triangular	igaussmtype2	igbelltype2	Itritype2
RMSE (Best)	0.031543721	0.034842843	0.106253992	0.01891173	0.020071084	0.0205731
RMSE (Average)	0.047506067	0.050711845	0.117611133	0.023752013	0.022696277	0.02576771
MSE	0.007214748	0.005309127	0.019661961	0.002960679	0.001445358	0.00205462
MAE	0.058668759	0.054341832	0.113583028	0.034935372	0.025597427	0.03070206
MPE	0.495293832	0.238903859	0.548675423	0.045305119	−1.41435995	1.6958679
MAPE	6.918855495	6.407519242	14.02215078	4.13271772	2.957239916	3.41521921

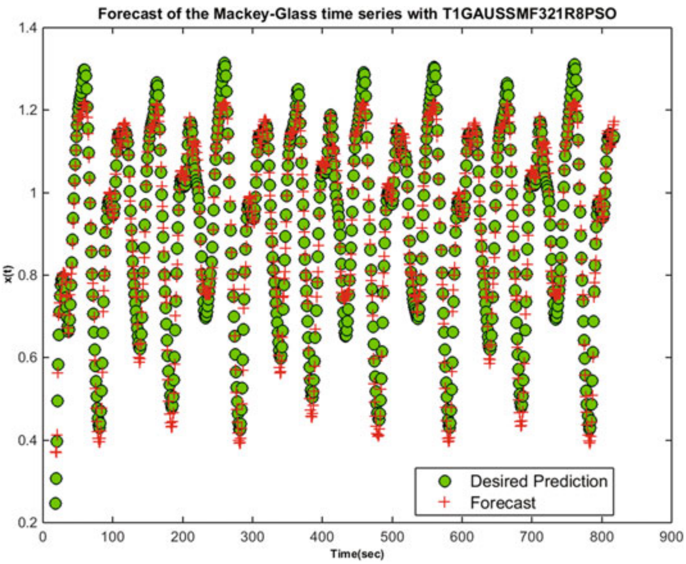


Fig. 4.34 Forecast of T1FIS using "Gaussian" MFs for the Mackey-Glass time series

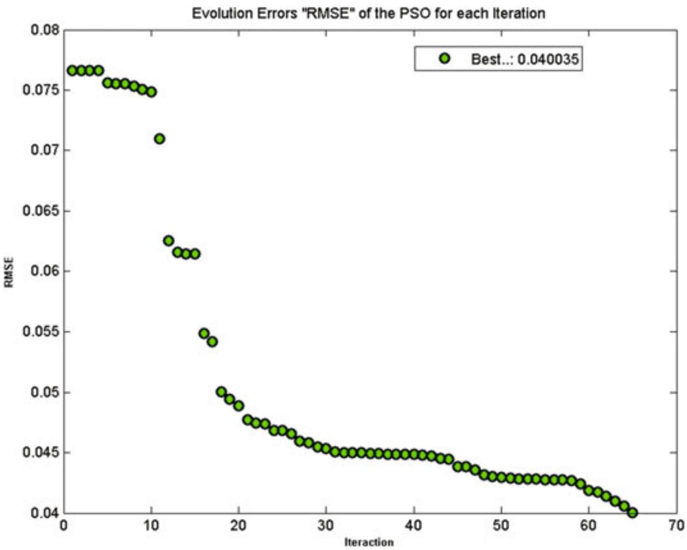


Fig. 4.35 Evolution error (RMSE) of the PSO for the T1FIS using "Gaussian" MFs

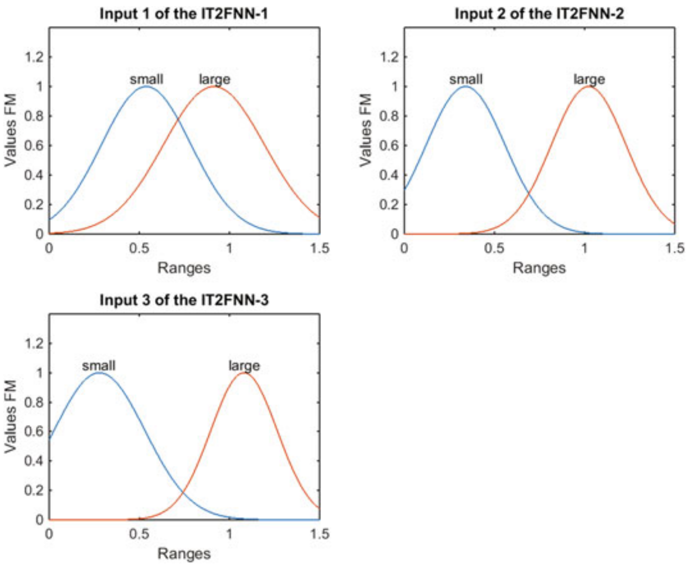


Fig. 4.36 Final MFs after optimized the T1FIS using "Gaussian" MFs

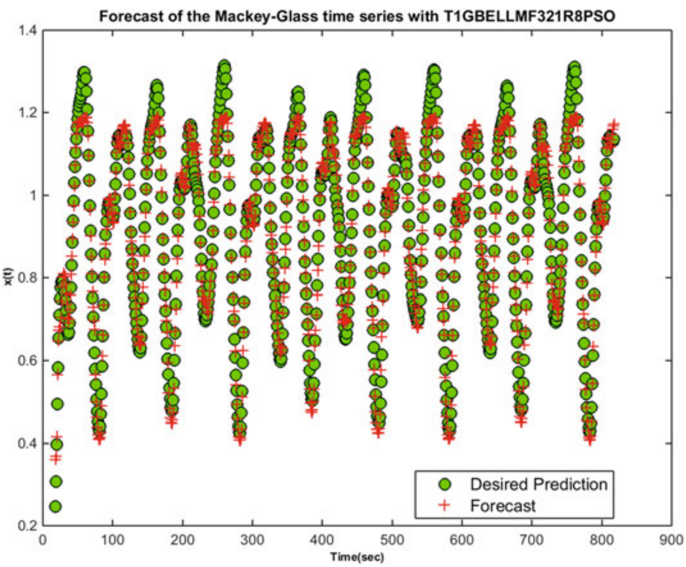


Fig. 4.37 Forecast of T1FIS using "Gbell" MFs for the Mackey-Glass time series

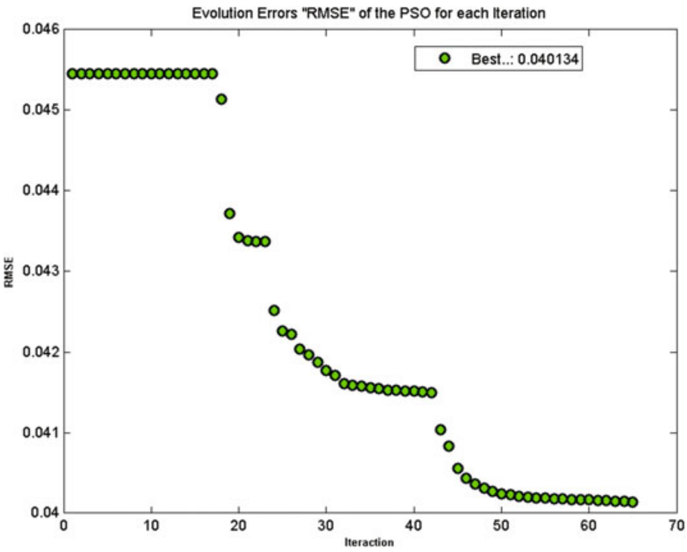


Fig. 4.38 Evolution error (RMSE) of the PSO for the T1FIS using "Gbell" MFs

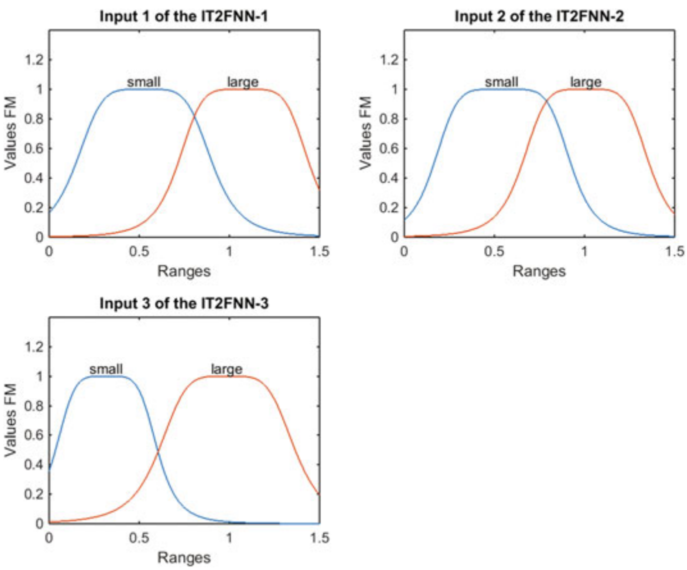


Fig. 4.39 Final MFs after optimized the T1FIS using "Gbell" MFs with PSO

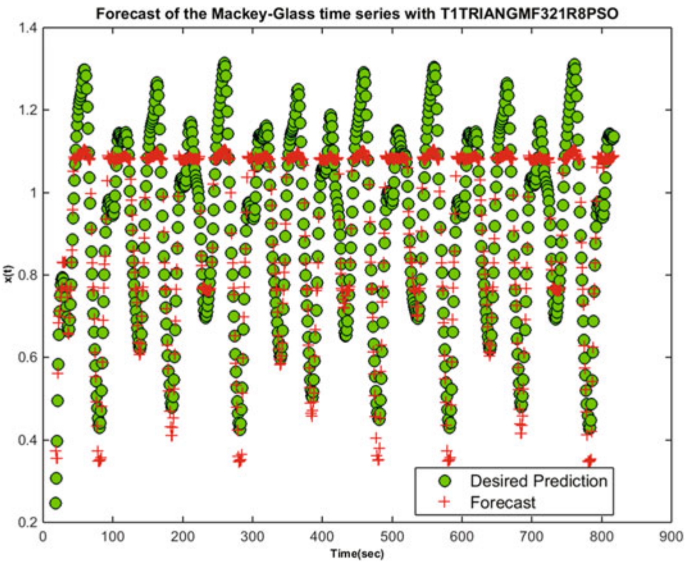


Fig. 4.40 Forecast of T1FIS using "Triangular" MFs for the Mackey-Glass time series

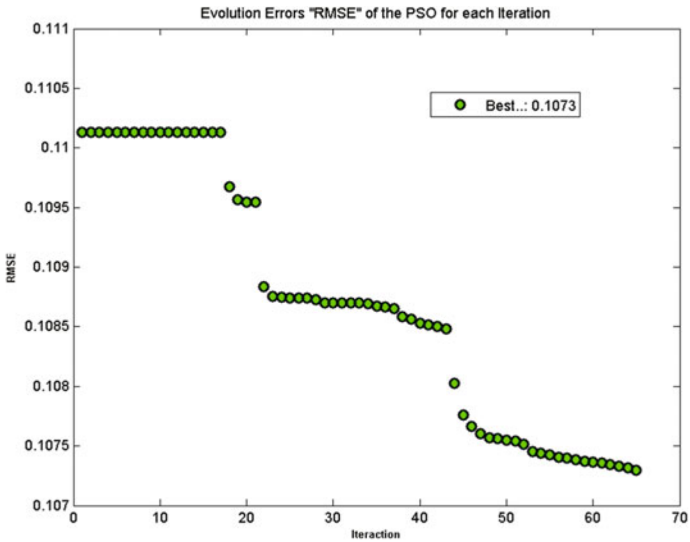


Fig. 4.41 Evolution error (RMSE) of the PSO for the T1FIS using "Triangular" MFs

Fig. 4.43, the performance of the evolution error is shown in Fig. 4.44, and the optimization structure of the interval type-2 fuzzy integrators using "igaussmtype2" MFs with the PSO is shown in Fig. 4.45.

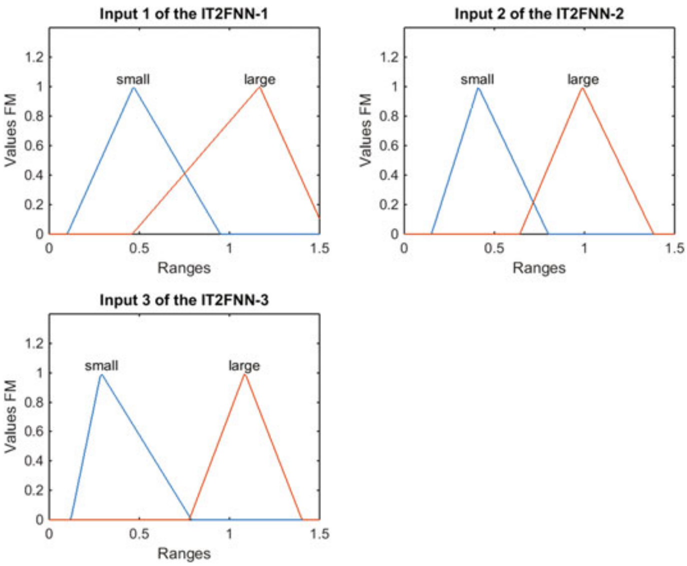


Fig. 4.42 Final MFs after optimized the T1FIS using "Triangular" MFs with PSO

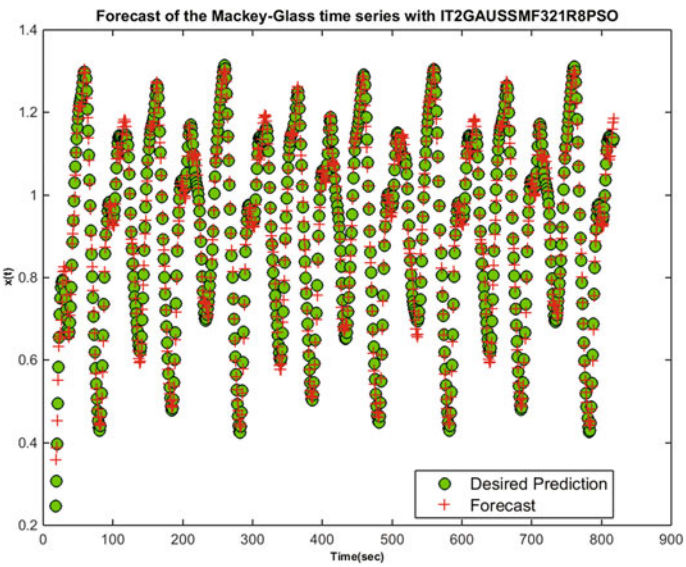


Fig. 4.43 Forecast of IT2FIS using "igaussmtype2" MFs for the Mackey-Glass time series

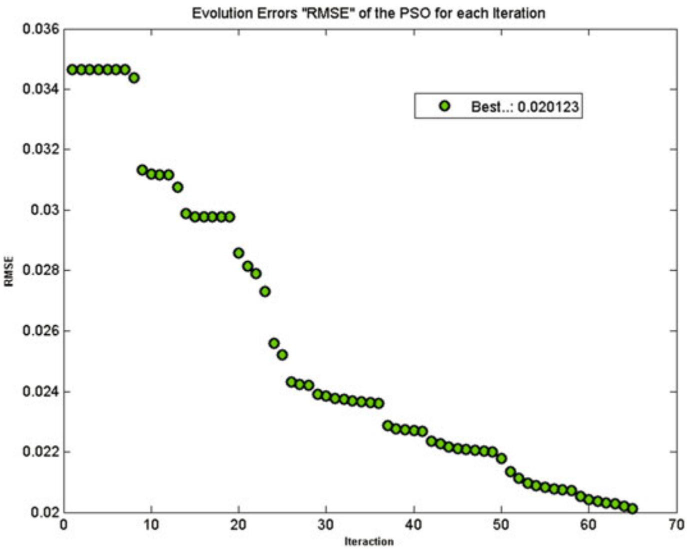


Fig. 4.44 Evolution error (RMSE) of the PSO for the IT2FIS using "igaussmtype2" MFs

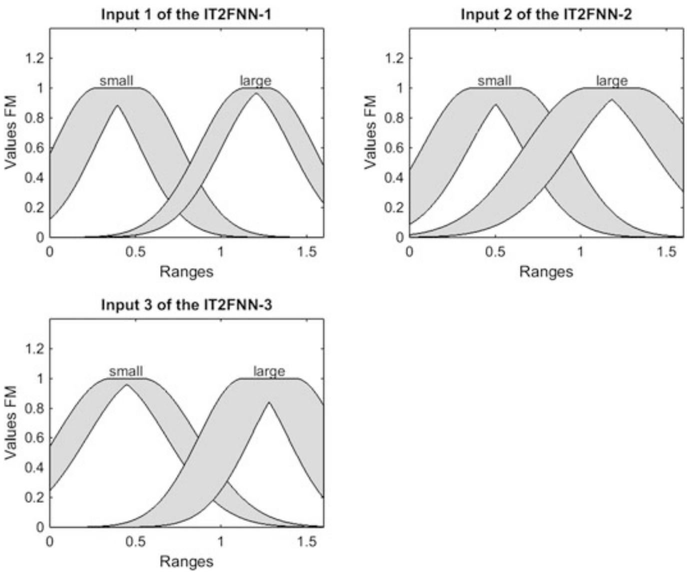


Fig. 4.45 Final MFs after optimized the IT2FIS using "igaussmtype2" MFs

The forecast obtained of the optimized interval type-2 fuzzy integrators using "igbelltype2" MFs for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.46, the performance of the evolution error is shown in Fig. 4.47, and the optimization

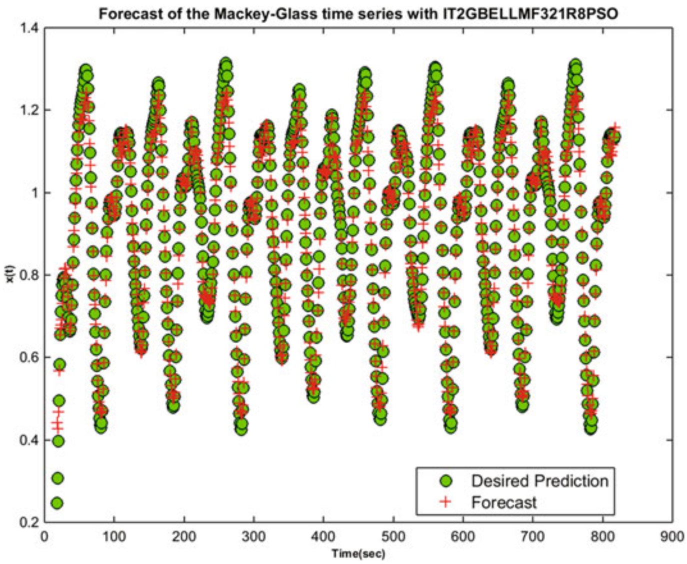


Fig. 4.46 Forecast of IT2FIS using "igbelltype2" MFs for the Mackey-Glass time series

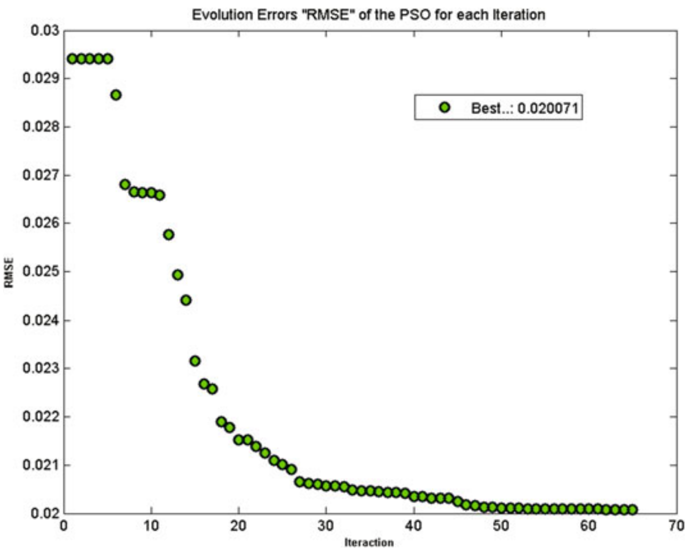


Fig. 4.47 Evolution error (RMSE) of the PSO for the IT2FIS using "igbelltype2" MFs

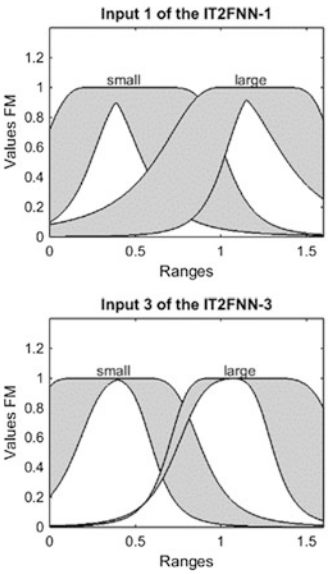


Fig. 4.48 Final MFs after optimized the IT2FIS using "igbelltype2" MFs

structure of the interval type-2 fuzzy integrators using "igbelltype2" MFs with the PSO is shown in Fig. 4.48.

The forecast obtained of the optimized interval type-2 fuzzy integrators using "itritype2" MFs for the Mackey-Glass (τ = 17) time series is shown in Fig. 4.49,

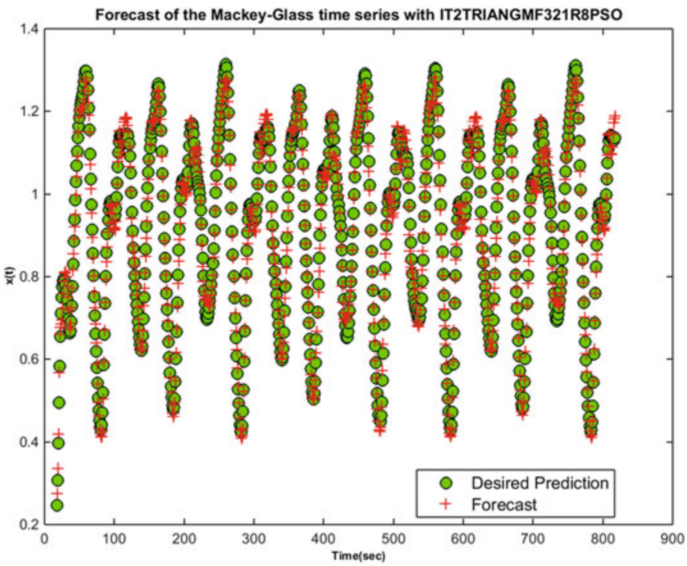


Fig. 4.49 Forecast of IT2FIS using "itritype2" MFs for the Mackey-Glass time series

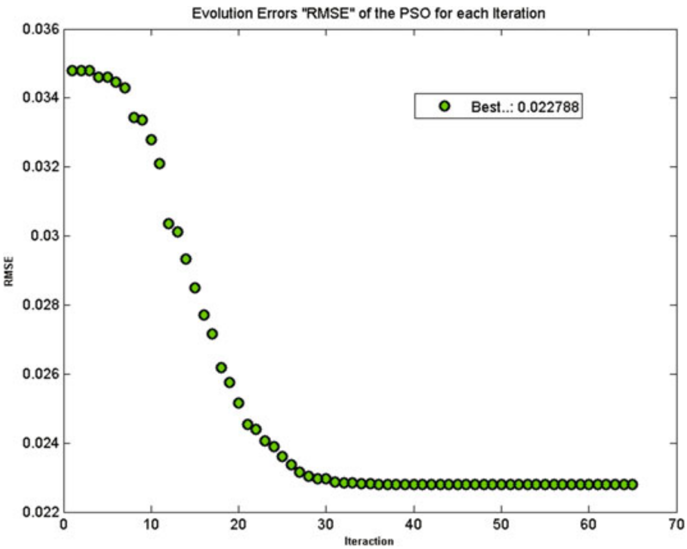


Fig. 4.50 Evolution error (RMSE) of the PSO for the IT2FIS using “itritype2” MFs

the performance of the evolution error is shown in Fig. 4.50, and the optimization structure of the interval type-2 fuzzy integrators using “itritype2” MFs with the PSO is shown in Fig. 4.51.

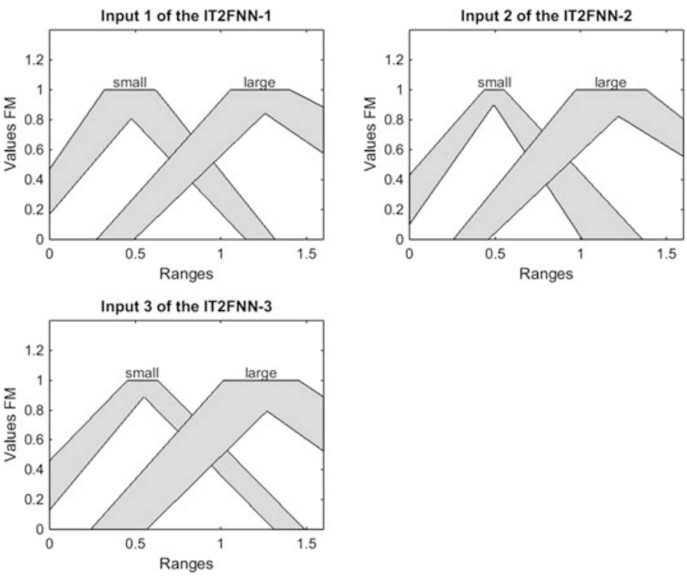


Fig. 4.51 Final MFs after optimized the IT2FIS using “itritype2” MFs

4.2 Mexican Stock Exchange Time Series

This section presents the simulation and test results obtained by applying the proposed prediction method to the Mexican Stock Exchange (BMV) time series for periods (01/03/2011–12/31/2015) (Fig. 3.3) using different approach of the ensemble of IT2FNN architectures, used in this work.

4.2.1 Ensemble of IT2FNN Architectures for BMV Time Series

The ensemble of IT2FNN architectures has three models as follows: the IT2FNN-1 model optimize the parameters of the "igaussmtype2" MFs (Fig. 3.9a), the learning rate is 0.03 and the desired error is 0.00001; the IT2FNN-2 model optimize the parameters of the "igausstype2" MFs (Fig. 3.9b), the learning rate is 0.011 and the desired error is 0.000001; and the IT2FNN-3 model optimize the parameters of the "igaussstype2" MFs (Fig. 3.9c), the learning rate is 0.02 and the desired error is 0.0000001. The number the epochs for training the IT2FNN models is 100.

The obtained results of the ensemble of IT2FNN architectures are shown on Table 4.6. The RMSE (best) is of 0.010127619, the RMSE (average) is of 0.016586239, the MSE is 0.001738454, the MAE is 0.012085755, the MPE is 1.284208192 and the MAPE 0.275038065 with the IT2FNN-1 model. Therefore the IT2FNN-1 model is better than the IT2FNN-2 and IT2FNN-3 models.

4.2.1.1 IT2FNN-1 Model

The forecast obtained for the IT2FNN-1 for the BMV time series is shown in Fig. 4.52, the evolution error is shown in Fig. 4.53, and the optimization structure of the IT2FNN-1 with backpropagation (BP) learning algorithm is shown in Fig. 4.54.

Table 4.6 Performance of the ensemble of IT2FNN for the BMV time series

Metrics	IT2FNN-1	IT2FNN-2	ITFNN-3
RMSE (Best)	0.010127619	0.022896849	0.010126143
RMSE (Average)	0.016586239	0.02748781	0.018984807
MSE	0.001738454	0.002373369	0.002859776
MAE	0.012085755	0.023283565	0.015326454
MPE	1.284208192	2.469791397	1.626561735
MAPE	0.275038065	0.385345148	0.511999726

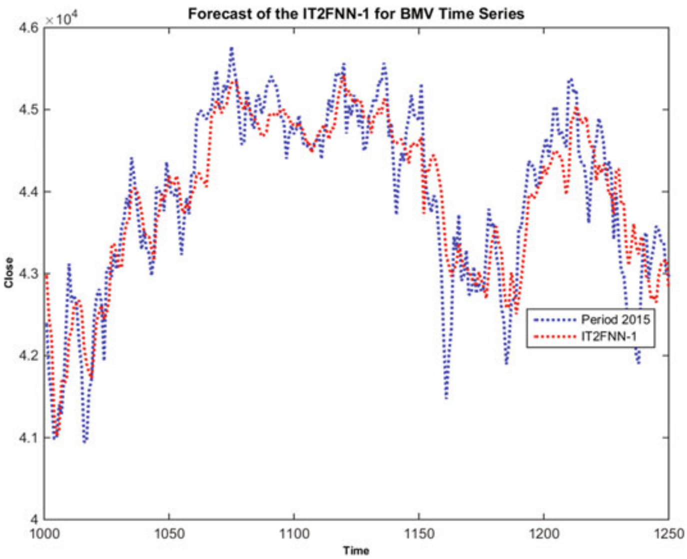


Fig. 4.52 Forecast of IT2FNN-1 for the BMV time series

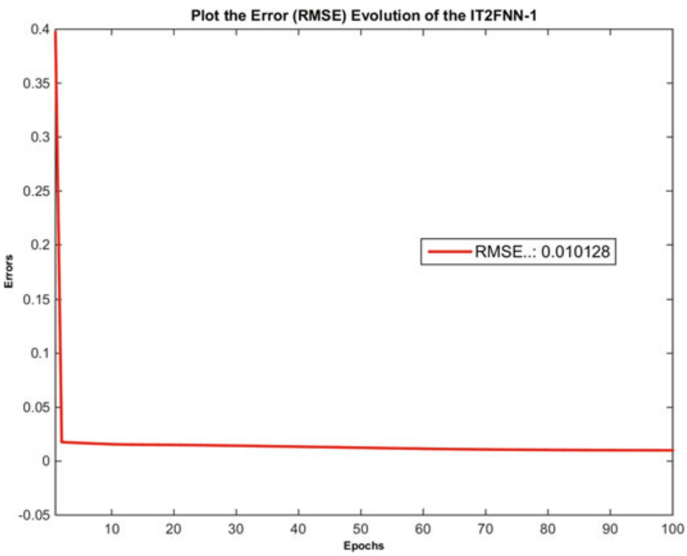


Fig. 4.53 Evolution error (RMSE) of IT2FNN-1 for the BMV time series

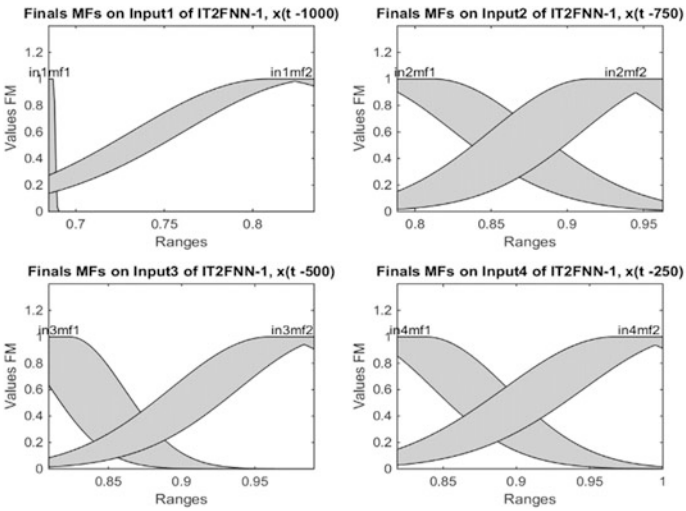


Fig. 4.54 Final MFs after training the IT2FNN-1 model with the BP algorithm

4.2.1.2 IT2FNN-2 Model

The forecast obtained for the IT2FNN-2 for the BMV time series shown in Fig. 4.55, the evolution error is shown in Fig. 4.56, and the structure optimization of the IT2FNN-2 with BP learning algorithm is shown in Fig. 4.57.

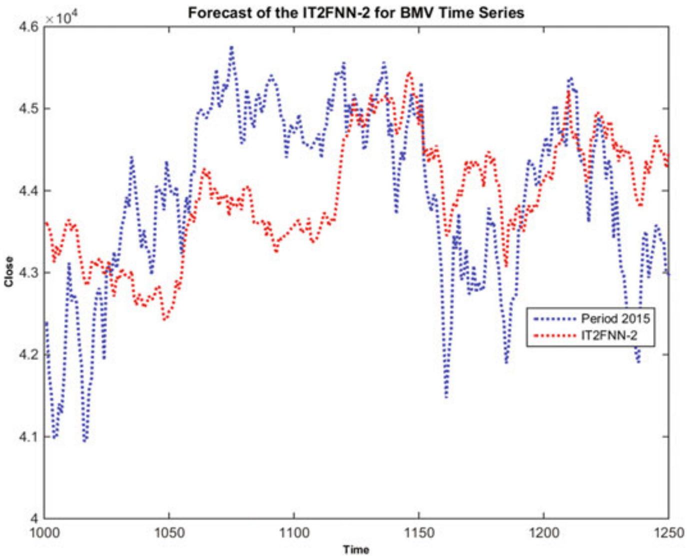


Fig. 4.55 Forecast of the IT2FNN-2 for the BMV time series

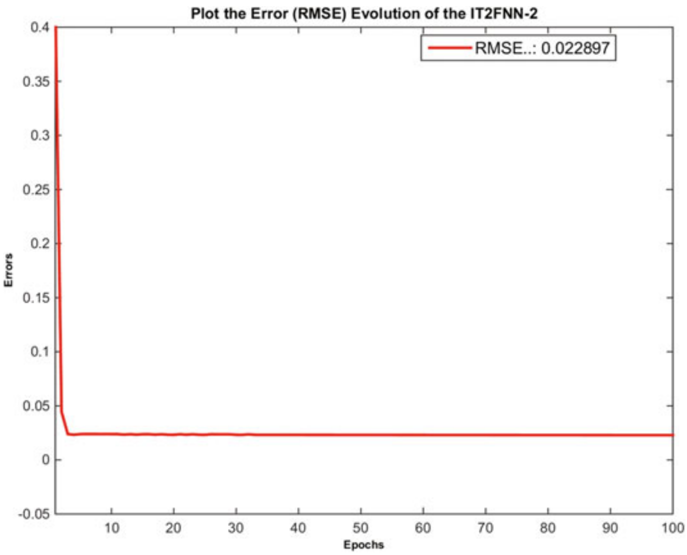


Fig. 4.56 Evolution error (RMSE) of IT2FNN-2 for the BMV time series

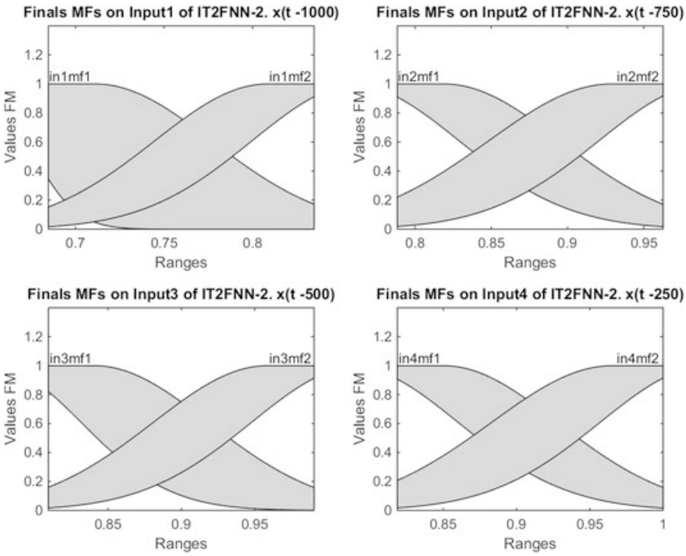


Fig. 4.57 Final MFs after training the IT2FNN-2 model with BP algorithm

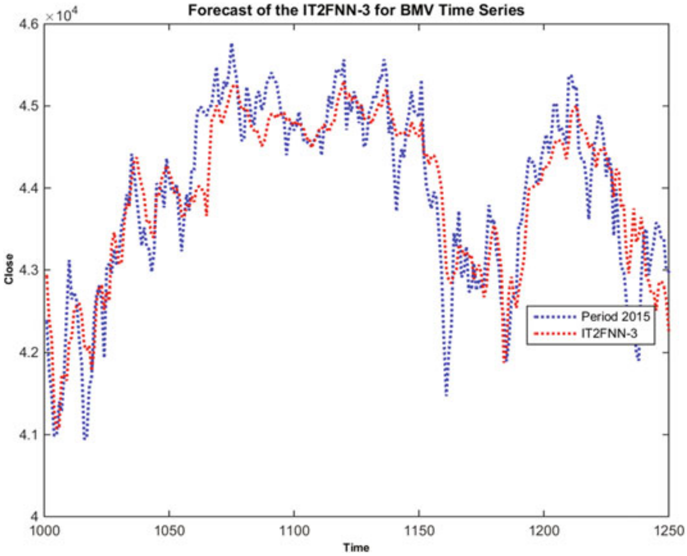


Fig. 4.58 Forecast of IT2FNN-3 for the BMV time series

4.2.1.3 IT2FNN-3 Model

The forecast obtained for the IT2FNN-3 for the BMV time series is shown in Fig. 4.58, the evolution error is shown in Fig. 4.59, and the structure optimization of IT2FNN-3 with BP learning algorithm is shown in Fig. 4.60.

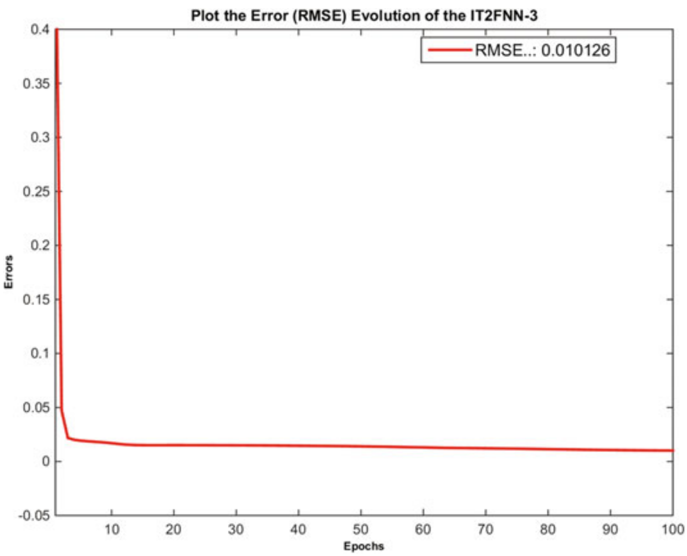


Fig. 4.59 Evolution error (RMSE) of IT2FNN-3 for the BMV time series

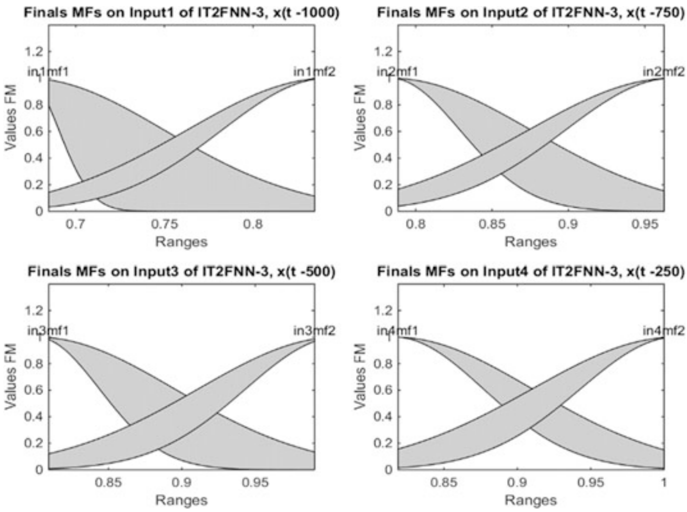


Fig. 4.60 Final MFs after training the IT2FNN-3 model with BP algorithm

4.3 Dow Jones Time Series

This section presents the simulation and test results obtained by applying the proposed prediction method to the Dow Jones time series for periods (01/03/2011–12/31/2015) (Fig. 3.4) using a different approach of the ensemble of IT2FNN architectures, used in this work.

4.3.1 Ensemble of IT2FNN Architectures for Dow Jones Time Series

The ensemble of IT2FNN architectures has three models as follows: the IT2FNN-1 model optimize the parameters of the “igaussmtype2” MFs (Fig. 3.9a), the learning rate is 0.03 and the desired error is 0.00001; the IT2FNN-2 model optimize the parameters of the “igausstype2” MFs (Fig. 3.9b), the learning rate is 0.011 and the desired error is 0.000001; and the IT2FNN-3 model optimize the parameters of the “igaussstype2” MFs (Fig. 3.9c), the learning rate is 0.02 and the desired error is 0.0000001. The number the epochs for training the IT2FNN models is 100.

Table 4.7 Performance of the ensemble of IT2FNN for the Dow Jones time series

Metrics	IT2FNN-1	IT2FNN-2	ITFNN-3
RMSE (Best)	0.015844833	0.01329307	0.01307153
RMSE (Average)	0.020874526	0.01909446	0.018482224
MSE	0.001743898	0.002022886	0.002138805
MAE	0.015181591	0.014462062	0.013647136
MPE	1.598124859	1.521469583	1.436647482
MAPE	0.236962281	0.346789986	0.320293965

The obtained results of the ensemble of IT2FNN architectures are shown on Table 4.7. The RMSE (best) is of 0.01307153, the RMSE (average) is of 0.018482224, the MSE is 0.002138805, the MAE is 0.013647136, the MPE is 1.436647482 and the MAPE 0.320293965 with the IT2FNN-3 model. Therefore the IT2FNN-3 model is better than the IT2FNN-1 and IT2FNN-2 models.

4.3.1.1 IT2FNN-1 Model

The forecast obtained for the IT2FNN-1 for the Dow Jones time series is shown in Fig. 4.61, the evolution error is shown in Fig. 4.62, and the structure optimization of the IT2FNN-1 with BP learning algorithm is shown in Fig. 4.63.

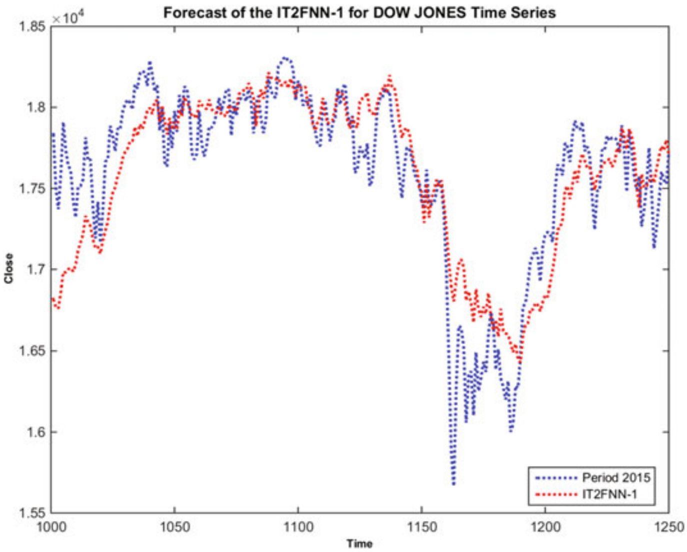


Fig. 4.61 Forecast of IT2FNN-1 for the Dow Jones time series

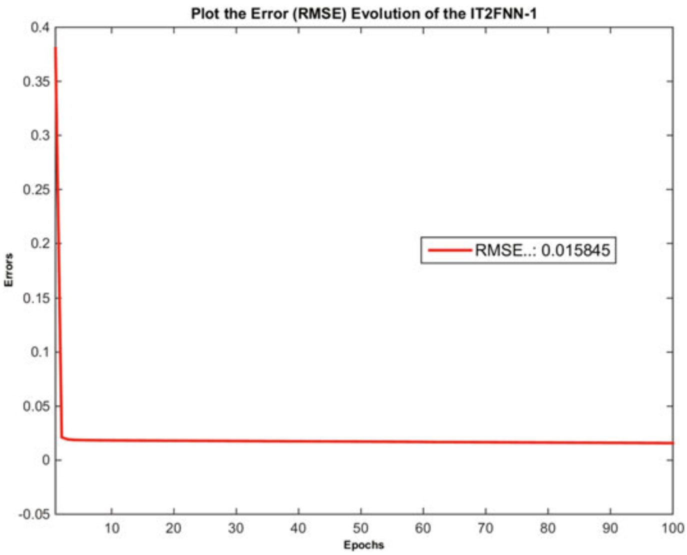


Fig. 4.62 Evolution error (RMSE) of IT2FNN-1 for the Dow Jones time series

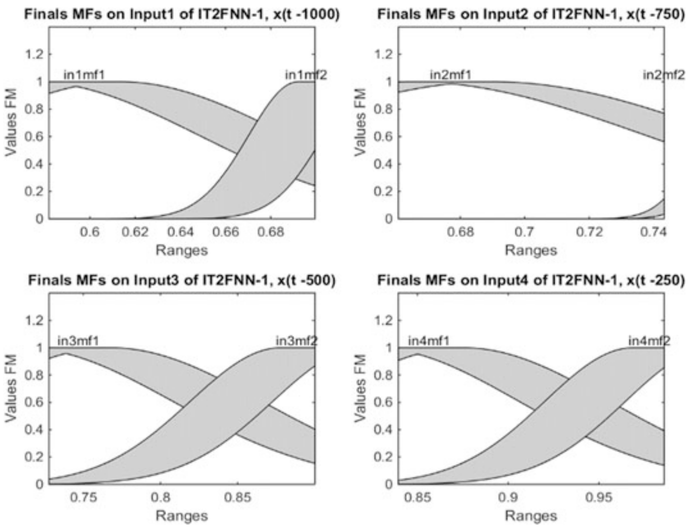


Fig. 4.63 Final MFs after training the IT2FNN-1 model with BP algorithm

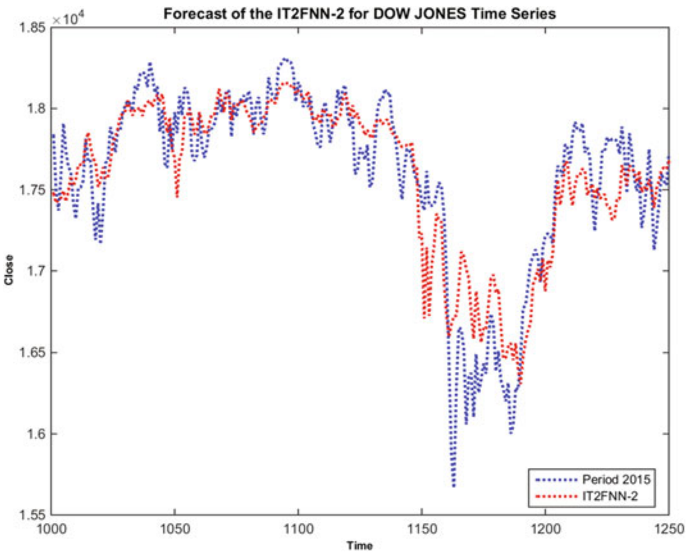


Fig. 4.64 Forecast of IT2FNN-2 for the Dow Jones time series

4.3.1.2 IT2FNN-2 Model

The forecast obtained for the IT2FNN-2 for the Dow Jones time series is shown in Fig. 4.64, the evolution error is shown in Fig. 4.65, and the structure optimization of the IT2FNN-2 with BP learning algorithm is shown in Fig. 4.66.

4.3.1.3 IT2FNN-3 Model

The forecast obtained for the IT2FNN-3 for the Dow Jones time series shown in Fig. 4.67, the evolution error is shown in Fig. 4.68, and the optimization structure of the IT2FNN-3 with BP learning algorithm is shown in Fig. 4.69.

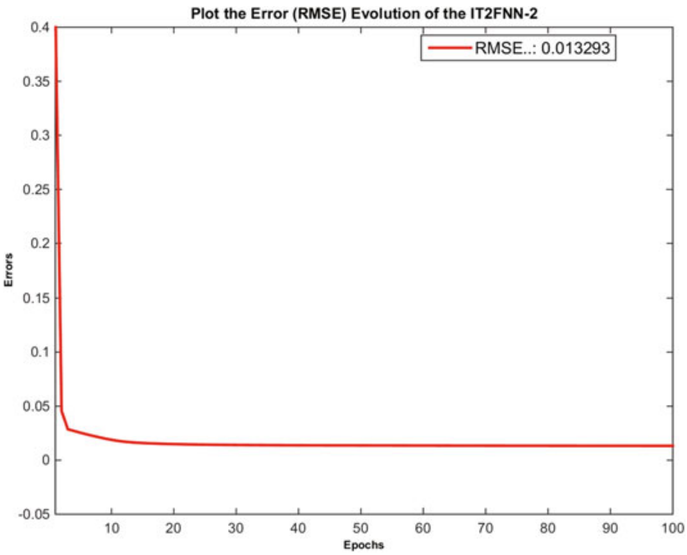


Fig. 4.65 Evolution error (RMSE) of IT2FNN-2 for the Dow Jones time series

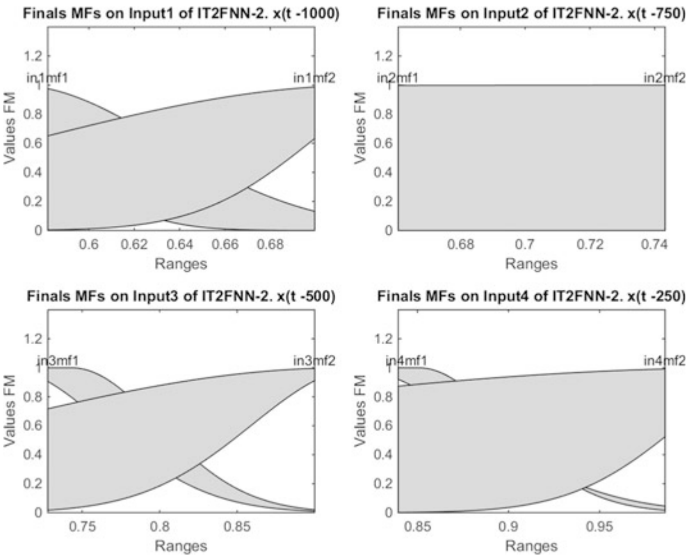


Fig. 4.66 Final MFs after training the IT2FNN-2 model with BP algorithm

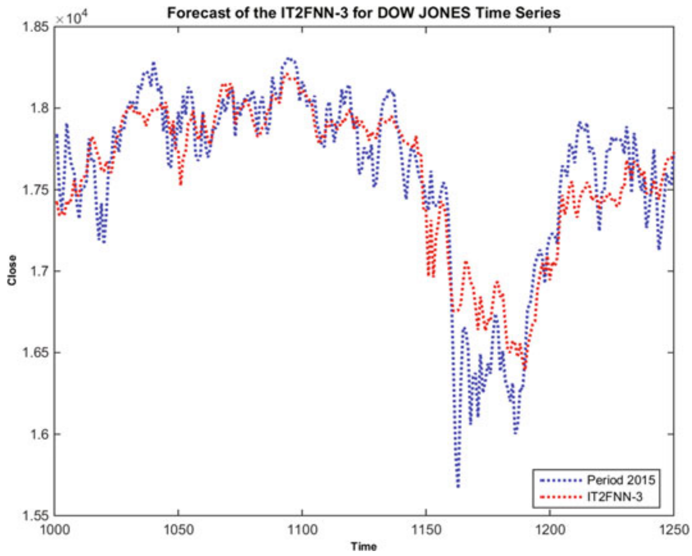


Fig. 4.67 Forecast of IT2FNN-3 for the Dow Jones time series

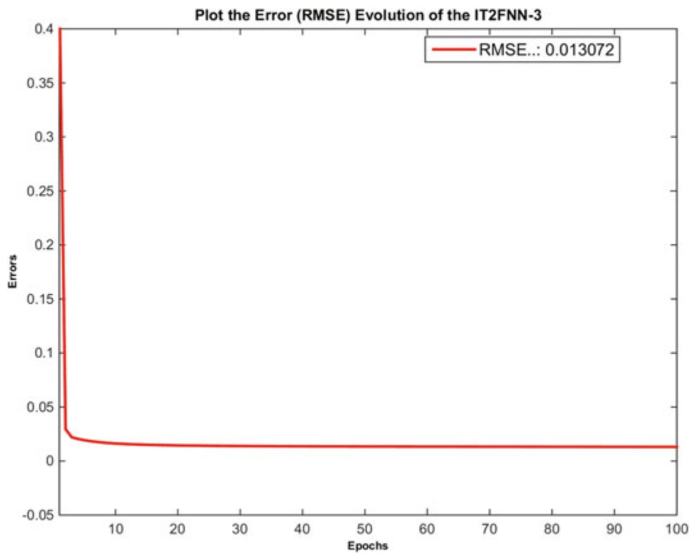


Fig. 4.68 Evolution error (RMSE) of IT2FNN-3 for the Dow Jones time series

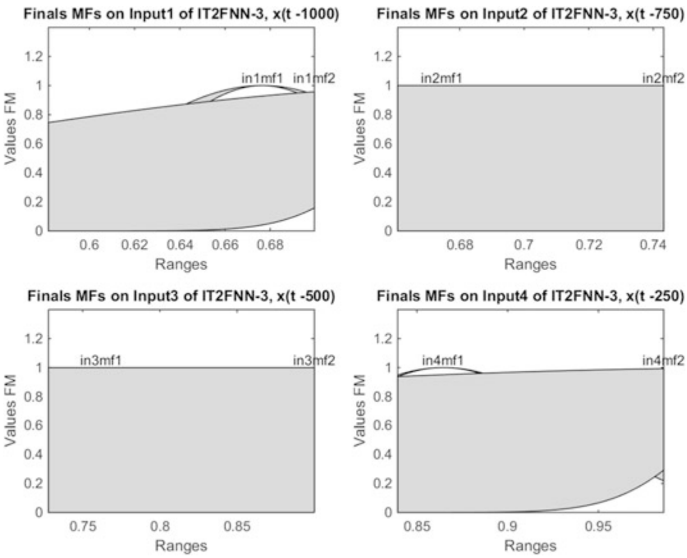


Fig. 4.69 Final MFs after training the IT2FNN-3 model with BP algorithm

4.4 NASDAQ Time Series

This section presents the simulation and test results obtained by applying the proposed prediction method to the NASDAQ time series for periods (01/03/2011–12/31/2015) (Fig. 3.5) using a different approach of the ensemble of IT2FNN architectures, used in this work.

4.4.1 Ensemble of IT2FNN Architectures for NASDAQ Time Series

The ensemble of IT2FNN architectures has three models as follows: the IT2FNN-1 model optimizes the parameters of the "igaussmtype2" MFs (Fig. 3.9a), the learning rate is 0.03 and the desired error is 0.00001; the IT2FNN-2 model optimizes the parameters of the "igausstype2" MFs (Fig. 3.9b), the learning rate is 0.011 and the desired error is 0.000001; and the IT2FNN-3 model optimizes the parameters of the "igaussstype2" MFs (Fig. 3.9c), the learning rate is 0.02 and the desired error is 0.0000001. The number the epochs for training the IT2FNN models is 100.

Table 4.8 Performance of the ensemble of IT2FNN for the NASDAQ time series

Metrics	IT2FNN-1	IT2FNN-2	ITFNN-3
RMSE (Best)	0.011711953	0.01318047	0.013617022
RMSE (Average)	0.016485694	0.017226806	0.020196196
MSE	0.001635756	0.001412437	0.003081807
MAE	0.012063554	0.012381383	0.01588996
MPE	1.288842865	1.324005862	1.691563953
MAPE	0.240159673	0.191975465	0.513682447

The obtained results of the ensemble of IT2FNN architectures are shown on Table 4.8. The RMSE (best) is of 0.011711953, the RMSE (average) is of 0.016485694, the MSE is 0.001635756, the MAE is 0.012063554, the MPE is 1.288842865 and the MAPE 0.240159673 with the IT2FNN-1 model. Therefore the IT2FNN-1 model is better than the IT2FNN-2 and IT2FNN-3 models.

4.4.1.1 IT2FNN-1 Model

The forecast obtained for the IT2FNN-1 for the NASDAQ time series is shown in Fig. 4.70, the evolution error is shown in Fig. 4.71, and the optimization structure of IT2FNN-1 with BP learning algorithm is shown in Fig. 4.72.

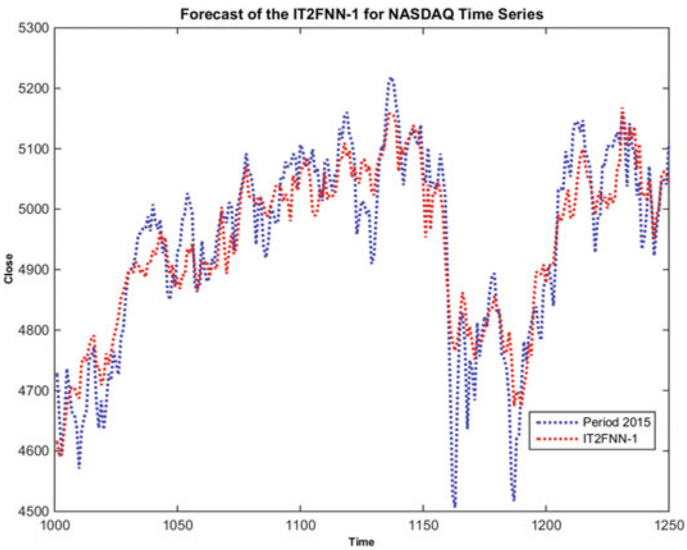


Fig. 4.70 Forecast of IT2FNN-1 for the NASDAQ time series

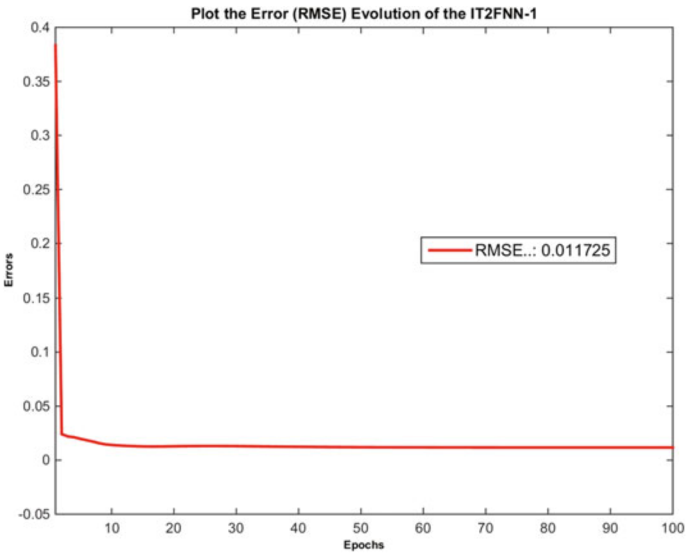


Fig. 4.71 Evolution error (RMSE) of IT2FNN-1 for the NASDAQ time series

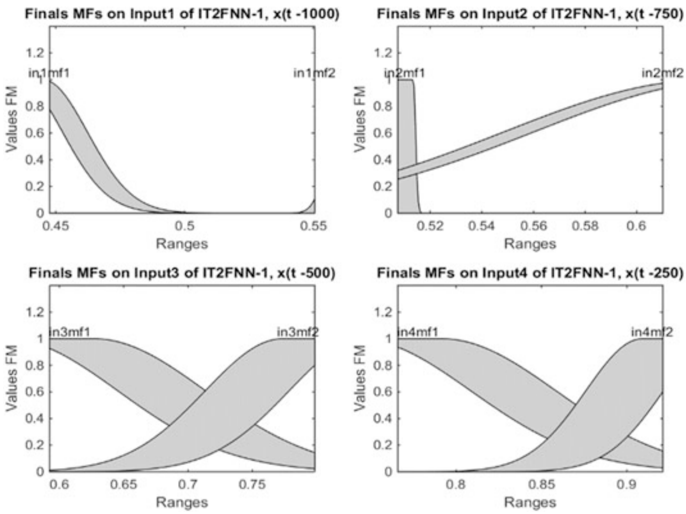


Fig. 4.72 Final MFs after training the IT2FNN-1 model with BP algorithm

4.4.1.2 IT2FNN-2 Model

The forecast obtained for the IT2FNN-2 for the NASDAQ time series is shown in Fig. 4.73, the evolution error is shown in Fig. 4.74, and the optimization structure of the IT2FNN-2 with BP learning algorithm is shown in Fig. 4.75.

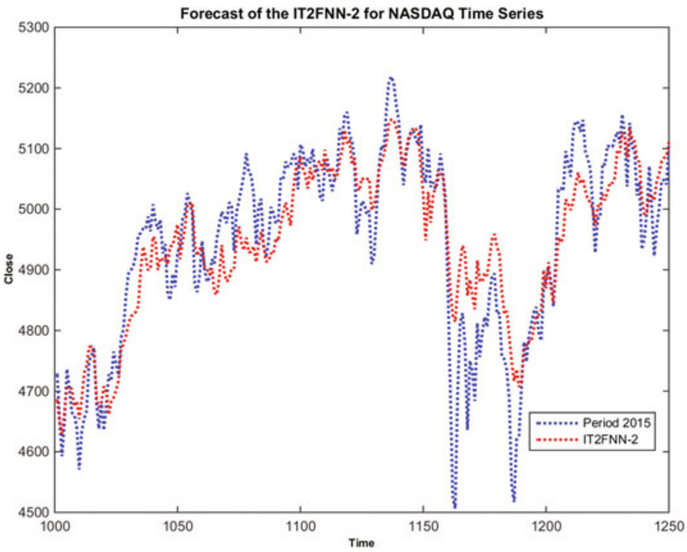


Fig. 4.73 Forecast of IT2FNN-2 for the NASDAQ time series

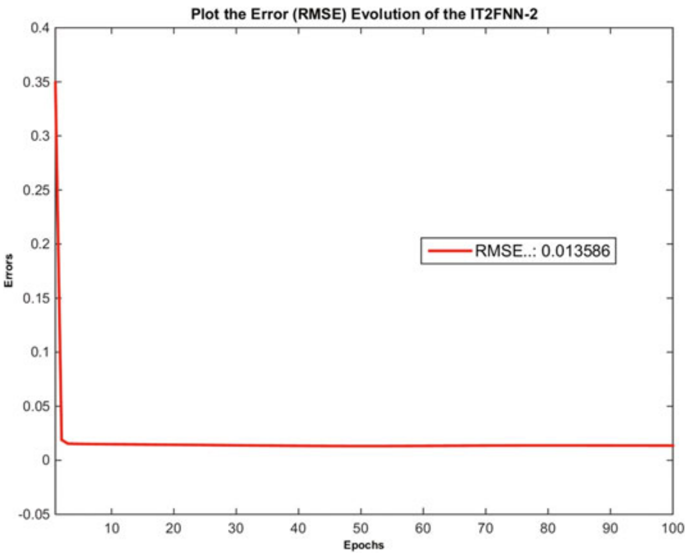


Fig. 4.74 Evolution error (RMSE) of IT2FNN-2 for the NASDAQ time series

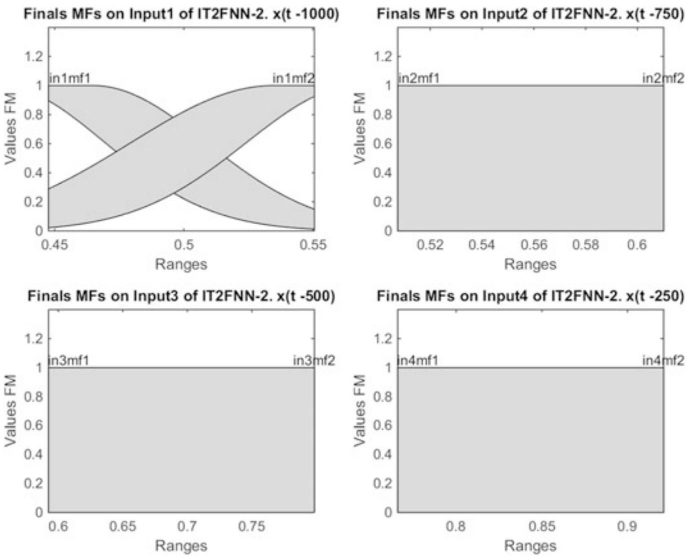


Fig. 4.75 Final MFs after training the IT2FNN-2 model with BP algorithm

4.4.1.3 IT2FNN-3 Model

The forecast obtained for the IT2FNN-3 for the NASDAQ time series is shown in Fig. 4.76, the evolution error is shown in Fig. 4.77, and the optimization structure of the IT2FNN-3 with BP learning algorithm is shown in Fig. 4.78.

4.5 Statistical Comparison Results of the Optimization of the Fuzzy Integrators

We also perform a statistical comparison of all the results obtained of the proposed model (Fig. 3.1) for the Mackey-Glass time series. The statistical test used for comparison is the *Z-scores*, whose parameters are defined in Table 4.9. In applying the statistic *Z-scores*, with significance level of 0.05, and the alternative hypothesis stating that the μ_1 is lower than the μ_2; $H_a(\mu_1 < \mu_2)$ (Fig. 4.79), and of course the null hypothesis tells us that the μ_1 is greater than or equal to the μ_2; $H_0(\mu_1 \geq \mu_2)$, with a rejection region for all values that fall below -1.732. We are presenting 30

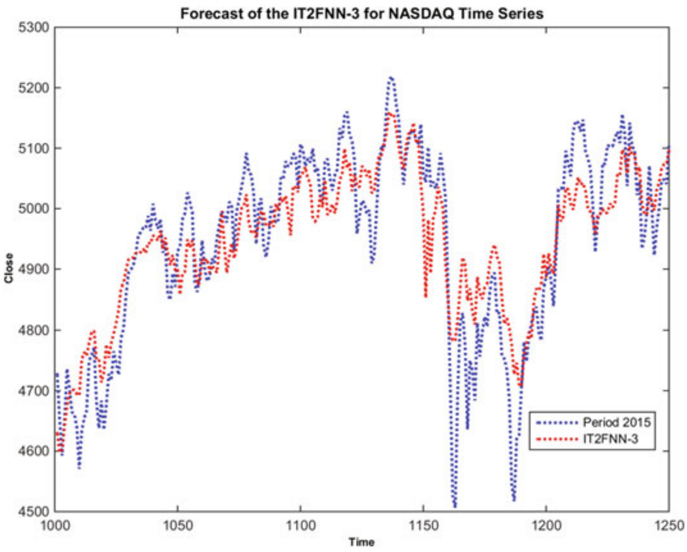


Fig. 4.76 Forecast of IT2FNN-3 for the NASDAQ time series

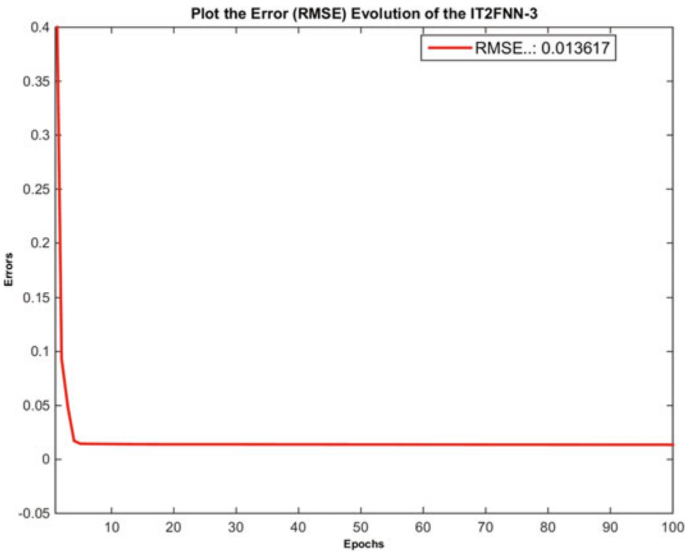


Fig. 4.77 Evolution error (RMSE) of IT2FNN-3 for the NASDAQ time series

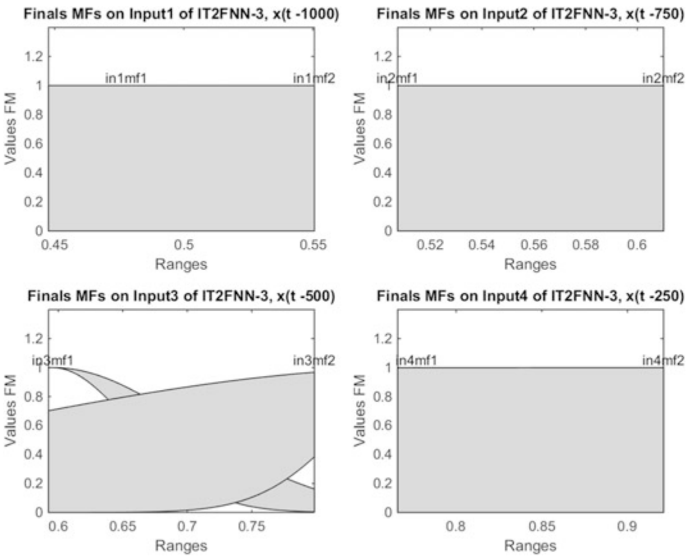


Fig. 4.78 Final MFs after training the IT2FNN-3 model with BP algorithm

Table 4.9 Statistical Z-scores parameters

Parameter	Value
Confidence interval	95%
Significance level (α)	5%
Null hypothesis (H_0)	$\mu_1^* \geq \mu_2^*$
Alternative hypothesis (H_a)	$\mu_1 < \mu_2$
Critical value	−1.645

μ_1—Average error of the optimization of fuzzy integrators with the GAs

μ_2—Average error of the optimization of fuzzy integrators with the PSO

experiments with the same parameters and conditions for the GAs and PSO algorithms for this work, so the n_1 and n_2 are equal 30.

The main objective of applying the statistical *Z-scores* is to analyze the performance and thus find if there is significant evidence of the proposed model results being better for the Mackey-Glass time series. The optimization of the fuzzy integrators results are generated from GAs and PSO algorithms. The results of the statistical *Z-scores* are shown in Table 4.10, so there is significant evidence to reject the null hypothesis because the value of $p < 0.05$ and the value of $z < -1.645$ and we accepted the alternative hypothesis. Therefore the results obtained of the optimization of fuzzy integrators with GAs are better than the PSO.

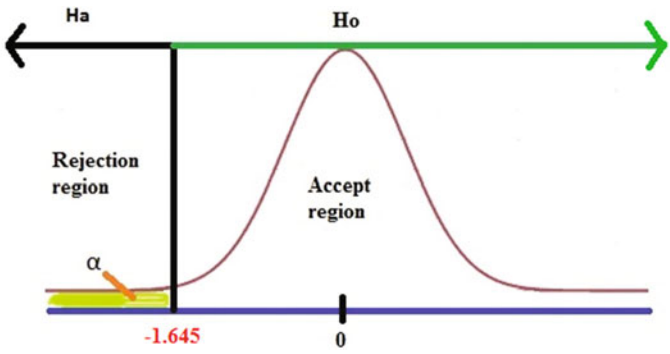


Fig. 4.79 Lower-Tailed Test $(\mu_1 < \mu_2)$

Table 4.10 Results of the Z-scores parameters

Optimization of the Type-1 fuzzy integrator using Gaussian MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	Z	$p < 0.05$	
0.02208303	0.000638122	0.0372719	0.000821093	−75.895	0	Significant
Optimization of the Type-1 Fuzzy Integrator using GBell MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	Z	$p < 0.05$	
0.02255155	0.001567122	0.0364846	0.000936185	−39.660	0	Significant
Optimization of the Type-1 Fuzzy Integrator using Triangular MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	z	$p < 0.05$	
0.08167633	0.000180209	0.08090114	0.00069041	5.645	0.0566351	Not Significant
Optimization of the Interval Type-2 Fuzzy Integrator using igaussmtype2 MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	z	$p < 0.05$	
0.02097587	0.00007548	0.02430513	0.00009724	−148.123	0	Significant
Optimization of the Interval Type-2 Fuzzy Integrator using igbelltype2 MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	z	$p < 0.05$	
0.02060701	0.000218508	0.02369199	0.00002811	−72.821	0	Significant
Optimization of the Interval Type-2 Fuzzy Integrator using itritype2 MFs						
GAs		PSO		Parameters		Evidence
μ_1	σ_1	μ_2	σ_2	z	$p < 0.05$	
0.02033528	0.000138626	0.02511529	0.000000010	−179.170	0	Significant

Based on the statistical *Z-scores* results, we can make the conclusion that the results obtained of the optimization of fuzzy integrators with the GAs are better than the PSO for the Mackey-Glass time series.

Chapter 5
Conclusion

Ensembles of IT2FNN models and the optimization of their fuzzy integrators using the GA and PSO algorithms for time series prediction, was proposed in this book.

We design the Ensemble of IT2FNN architectures for time series prediction. Three modules were used in each experiment of the Ensembles architectures. In module 1 the training was with the IT2FNN-1 model, in module 2 the training was with the IT2FNN-2 model and in module 3 the training was with the IT2FNN-3 model.

Genetic Algorithm and Particle swarm optimization were used to optimization the parameters of the membership functions of fuzzy integrators. We used type-1 (Gaussian, Generalized Bell and Triangular) and interval Type-2 (igaussmtype-2, igbelltype2 and itritrype2) MFs.

Gaussian membership functions in type-1 and type-2 fuzzy systems produced better results in predicting time series that were tested in the fuzzy integrators on this proposed model. I think that behave better with the kind of values that are represent in the time series.

Based on the statistical *Z-scores* results, we can make the conclusion that the results obtained of the optimization of fuzzy integrators with GAs are better than the PSO for the Mackey-Glass (τ = 13, 17, 30, 34, 68, 100, 136) time series show in Table 4.10.

The results of ensemble of the IT2FNN architectures for Mackey-Glass (τ = 13, 17, 30, 34, 68, 100, 136), Mexican Stock Exchange, Dow Jones and NASDAQ time series showed efficient results in the prediction error and the performance obtained of proposed method is good for this research.

Prediction errors obtained in this book are evaluated by the following metrics: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE). Therefore according to the results obtained by these metrics it can be concluded that the proposed model provides good performance in solving problems of time series.

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2_5

The objectives and goals in this book were achieved satisfactorily, because the results reported from experiments are efficient prediction errors because they are very small.

Future works

We envision applying others fuzzy integrator methods for the ensemble of IT2FNN models, like Sugeno fuzzy integration or the Choquet integral.

Apply other methods to optimize of fuzzy integrators, like the Gravitational Search Algorithm (GSA), Differential Evolution (DE), Cuckoo Search (CS), Firefly, Bat algorithms (BAT) and other ones metaheuristic algorithms for optimization.

Apply the proposed method to problems of control systems.

Apply the proposed method to problems pattern recognition.

Apply the method for obtain data prediction of another time series, like the, Dollar exchange series and Political elections.

Apply the proposed method to problems clustering, classification and control systems.

Appendix

IT2FNN-1 (Source Code)

```
function [Y,Yl,Yr,sigma,M1,M2,C,S,c1,c2,RMSE,MSE,MAE,ME,MAPE,
MPE] = TRAINIT2TSKNNGMF1(X,D,sigma,M1,M2,C,S,alpha,TOL,ITERMAX)
%
[L,n] = size(X) ;
[N,~] = size(M1);
OK   = false;
iter = 0;
while (~OK),
   for i=1:L % data
     xe  = [X(i,:) D(i)];
     Z  = [xe' ; abs(xe')];
     UU=[];
     LL=[];
     for j=1:N % rules
       Uu=1;
       Ll=1;
       for m=1:n % variables
         Param=[sigma(j,m),M1(j,m),M2(j,m)];
         mf=evalimftype2(X(i,m),Param,'igaussmtype2');
         ll = mf.LU{1}(:,1);
         uu = mf.LU{1}(:,2);
         Uu=Uu*uu;
         Ll=Ll*ll;
       end % variables
       UU=[UU,Uu];
       LL=[LL,Ll];
     end % reglas
     %
     c2  = [C +S]*Z;
     c1  = [C -S]*Z;
     %
```


J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2

```
    [r_out,I2l,I2u,wr]= rightpoint(c2',LL,UU);
    [l_out,I1u,I1l,wl]= leftpoint(c1',LL,UU);
    %
    Yl(i)=l_out;
    Yr(i)=r_out;
    Y(i) =(l_out+r_out)/2;
    e(i) =D(i)-Y(i);

    % Seleciona las MFs que contribuyen al punto a la derecha (yr)
    ME10=M1;
    ME20=M2;
    sigma0=sigma;
    LE=length(I2u); % numero de fu(I2u)
    for t=1:LE % fu(I2u)
      for k=1:n % variables
        if X(i,k)<M1(I2u(t),k)
          l=I2u(t);
          M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
        elseif  X(i,k)>M2(I2u(t),k)
          l=I2u(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
        end
      end % variables
    end % fu(I2u)
    %
    LE=length(I2l); % numero de fl(I2l)
    %
    for t=1:LE % numero de fl(I2l)
      for k=1:n % variables
        if X(i,k)<(M1(I2l(t),k)+M2(I2l(t),k))/2
          l=I2l(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
```

```
^2)/(sigma0(l,k)^3))…
          *(c2(l)-r_out)*wr(l)/sum(wr);
        else
          l=I2l(t);
          M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))…
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))…
          *(c2(l)-r_out)*wr(l)/sum(wr);
        end
      end % variables
    end % fl(I2u)
    % Seleciona las MFs que contribuyen al punto a la izquierda (yl)
    LE=length(I1l); % numero de fl(I1l)
    for t=1:LE % fl(I1l)
      for k=1:n % variables
        if X(i,k)<(M1(I1l(t),k)+M2(I1l(t),k))/2
          l=I1l(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))…
          *(c1(l)-l_out)*wl(l)/sum(wl);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))…
          *(c1(l)-l_out)*wl(l)/sum(wl);
        else
          l=I1l(t);
          M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))…
          *(c1(l)-l_out)*wl(l)/sum(wl);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))…
          *(c1(l)-l_out)*wl(l)/sum(wl);
        end
      end % variables
    end % fl(I1l)
    %
    LE=length(I1u); % numero de fu(I1u)
    %
    for t=1:LE % fu(I1u)
      for k=1:n % variables
        if  X(i,k)< M1(I1u(t),k)
          l=I1u(t);
          M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))…
```

```
          *(c1(l)-l_out)*wl(l)/sum(wl);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
          *(c1(l)-l_out)*wl(l)/sum(wl);
        elseif X(i,k)> M2(I1u(t),k)
          l=I1u(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
          *(c1(l)-l_out)*wl(l)/sum(wl);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))...
          *(c1(l)-l_out)*wl(l)/sum(wl);
        end
      end % variables
    end % fu(I1u)
    %
    for l=1:N % reglas
      for k=1:n % variables
        if sigma(l,k) < 0
          sigma(l,k)=abs(sigma(l,k));
        end
      end
    end
    %
    % fa1=wl/sum(wl); % wl = {fu(1),...,fu(L),fl(L+1),...,fl(N)} = {fu(I1u),fl
(I1l)}
    % fa2=wr/sum(wr); % wr = {fl(1),...,fl(R),fu(R+1),...,fu(N)} = {fl(I2l),fu
(I2u)}
    %
    fa1=wr'/sum(wr);
    fa2=wl'/sum(wl);
    % Termino de constantes de la funcion lineal
    C(:,n+1)=C(:,n+1)+alpha*e(i)*(fa1+fa2)/2;
    S(:,n+1)=S(:,n+1)+alpha*e(i)*(fa1-fa2)/2;
    for j=1:n, % variables
      C(:,j)=C(:,j)+alpha*e(i)*X(i,j)    *(fa1+fa2)/2;
      S(:,j)=S(:,j)+alpha*e(i)*abs(X(i,j))*(fa1-fa2)/2;
    end % variables
    % Reconstruye las funciones de membresia tipo-2 por intervalos
    S=abs(S);
    ME1=[];
    ME2=[];
    for t=1:N % reglas
```

```
      P=[M2(t,:)',M1(t,:)']';
      m2=max(P);
      m1=min(P);
      ME1=[ME1,m1'];
      ME2=[ME2,m2'];
    end % reglas
    M1=ME1';
    M2=ME2';
  end % end for i (datos)

  iter = iter + 1;

  % Calcular errores RMSE, MSE, ME, MAE, MPE, MAPE
  RMSE(iter) = sqrt(mse(e));
%   RMSE(iter)=errperf(D',Y,'rmse');
  MSE(iter)=errperf(D',Y,'mse');
  ME(iter)=sum(D'-Y)/size(D,1);
  MAE(iter)=errperf(D',Y,'mae');
  MPE(iter)=(sum((D'-Y)./D')/size(D,1))*100;
  for p=1:length(D),
   if (isnan(D(p,1))) || (D(p,1)==0),
     D(p,1)=1;%D(p-1,1);
   end
  end
  MAPE(iter)=errperf(D',Y,'mape');

  fprintf('*** Epochs %d *** RMSE %f *** MSE %f *** ME %f *** MAE %
f ***** MPE %f ***** MAPE %f *****\n', iter, RMSE(iter),MSE(iter),ME
(iter),MAE(iter),MPE(iter),MAPE(iter));

  if (iter==ITERMAX)
    %% Plot evolution error RMSE
    h = figure;
    plot(RMSE,'-
r','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8);
    axis([-inf inf -0.05 .4]);
    xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
    ylabel('Errors','fontsize',8,'fontweight','b','color','k');
    legend(['\fontsize{12}RMSE..: ',num2str(RMSE(iter))],'loca-
tion','best');
  %   legend(['Best..: ', num2str(BestResultMinIter
(Iter,1))],'location','best');
```

```
    title('Plot the Error (RMSE) Evolution of the IT2FNN-1');
    % Best Errors (MSE) for each Evolution in the GAs
    print(h, '-dpng','-r150',[folders,'EvolutionRMSE']);
  %   drawnow;
    close(h);

    %% Plotear evolucion de los errores por Iteracion Metrics (RMSE, MSE,
MAE y ME)
    h=figure;
    plot(RMSE,'-
r','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8); hold on
    plot(MSE,'-
b','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g',
'MarkerSize',8);
    plot(MAE,'-.
g','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','b',
'MarkerSize',8);
  %   plot(ME,'-
cs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','b',
'MarkerSize',8);
    axis([-inf inf -0.05 .4]);
    hold off
    xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
    ylabel('Errors','fontsize',8,'fontweight','b','color','k');
    legend(['\fontsize{12}RMSE..: ',num2str(RMSE(iter))],['\fontsize
{12}MSE..: ',num2str(MSE(iter))],['\fontsize{12}MAE..: ',num2str(MAE
(iter))],'location','best');
  %   legend(['- STD..: ',num2str(BestResultMinIterSTDMen(Iter))],
['Best..: ',num2str(BestResultMinIter(Iter,1))],['+ STD..: ',num2str
(BestResultMinIterSTDMas(Iter))],'location','best');
    title
('Comparison the Error Metrics for the Performance of the IT2FNN-1');
    print(h, '-dpng','-r150',[folders,'BestMetricsPlotEvolution']);
  %   drawnow;
    close(h);

    %% Plotear evolucion de los errores por Iteracion Metrics (MAPE y MPE)
    h=figure;
  %   plot(MPE,'-
ro','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8); hold on
    plot(MAPE,'-
b','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g',
'MarkerSize',8);
```

```
  %   hold off
    xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
    ylabel('Errors','fontsize',8,'fontweight','b','color','k');
  %   legend('\fontsize{12}MAPE','location','best');
    legend(['\fontsize{12}MAPE..: ',num2str(MAPE(iter))],'loca-
tion','best');
    title('Plot the Error (MAPE) Evolution of the IT2FNN-1');
    print(h, '-dpng','-r150',[folders,'BestMAPEEvolution']);
  %   drawnow;
    close(h);
  end

  % fprintf(1,'%6i  %12.8f\n',iter,rmse(iter));
  if iter >= ITERMAX
    OK = false;
    disp('procedimiento excede el numero maximo de iteraciones');
    break;
  end
  if RMSE <= TOL
    OK = true;
    disp('procedimiento terminado con exito');
    break;
  end
end
```

Index

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2