*International Journal of Innovative Computing, Information and Control*
Volume **9**, Number **10**, October **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 4151–4166

# GENETIC OPTIMIZATION OF ENSEMBLE NEURAL NETWORKS FOR COMPLEX TIME SERIES PREDICTION OF THE MEXICAN EXCHANGE

Martha Pulido, Oscar Castillo and Patricia Melin

Division of Research and Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California, México
marthapulido_84@hotmail.com; { ocastillo; pmelin }@tectijuana.mx

Received October 2012; revised February 2013

Abstract. *This paper describes an optimization method based on genetic algorithms for ensemble neural networks with fuzzy aggregation to forecast complex time series. The time series that was considered in this paper, to compare the hybrid approach with traditional methods, is the Mexican Stock Exchange, and the results shown are for the optimization of the structure of the ensemble neural network with type-1 and type-2 fuzzy logic integration. Simulation results show that the optimized ensemble approach produces good prediction of the Mexican Stock Exchange.*

**Keywords:** Ensemble neural network, Genetic algorithm, Optimization, Time series prediction

## 1. Introduction.

Time series are usually analyzed to understand the past and to predict the future, enabling managers or policy makers to make properly informed decisions. Time series analysis quantifies the main features in data, as it is the random variation. These facts, combined with improved computing power, have made time series methods widely applicable in government, industry, and commerce. In most branches of science, engineering, and commerce, there are variables measured sequentially in time. For example, reserve banks record interest rates and exchange rates each day. The government statistics department will compute the country's gross domestic product on a yearly basis. Newspapers publish yesterday's noon temperatures for capital cities from around the world. Meteorological offices record rainfall at many different sites with different resolutions. When a variable is measured sequentially in time over or at a fixed interval, known as the sampling interval, the resulting data form a time series [8].

Time series predictions are very important because based on them past events can be analyzed to know the possible behavior of future events and thus take preventive or corrective decisions to help avoid unwanted circumstances.

The choice and implementation of an appropriate method for prediction has always been a major issue for enterprises that seek to ensure the profitability and survival of business. The predictions give the company the ability to make decisions in the medium and long term, and due to the accuracy or inaccuracy of data this could mean predicted growth or profits and financial losses. It is very important for companies to know the behavior that will be the future development of their business, and thus be able to make decisions that improve the company's activities, and avoid unwanted situations, which in some cases can lead to the company's failure. In this paper a hybrid approach for time series prediction by using an ensemble neural network and its optimization with genetic algorithms is proposed. In the literature there has been recent work of time series [3-5,20,25,26,29,41,43], which indicate the importance of the topic.

Time series prediction has been a hot topic in recent years. This paper shows the results of an optimized ensemble neural network and its fuzzy response integration for predicting the time series of the Mexican Stock Exchange. The technique used to optimize the neural network ensemble is the genetic algorithm, as this can help in finding a good solution to a complex problem, which in this case is to find the best ensemble neural network architecture to obtain the minimum forecast error for the time series mentioned above and to integrate the results with type-1 and type-2 fuzzy systems.

The rest of the paper is organized as follows. Section 2 describes the basic concepts of the proposed method, Subsection 2.2 describes the concepts of neural networks, Subsection 2.3 describes the concepts of ensemble neural networks, Subsection 2.4 describes the concepts of optimization, Subsection 2.5 describes the concepts of genetic algorithms, and Subsection 2.6 describes the concepts of fuzzy systems as methods of integration. In Section 3, the problem and the proposed method for solving it are represented, Section 4 describes the simulation results of the proposed method, in Section 5 the comparison and the $t$ student test are presented, and Section 6 shows the conclusions.

## 2. **Preliminaries.** This section shows the basic concepts that are used in the proposed method:

### 2.1. **Time series prediction.** A time series is defined as a sequence of observations on a set of values that take a variable (quantitative) at different points in time. Time series are widely used today because organizations need to know the future behavior of certain relevant phenomena in order to plan, prevent, and so on, their actions. That is, to predict what will happen with a variable in the future from the behavior of that variable in the past [2]. The data can behave in different ways over time, and this may be a trend, which is the component that represents a long-term growth or decline over a period of time. A cycle is also possible, which refers to the wave motion that occurs around the trend, or may not have a defined or random manner; there are also seasonal variations (annual, biannual, etc.), which are behavior patterns that are repeated year after year at a particular time [3].

The word "prediction" comes from the Latin prognosticum, which means I know in advance. Prediction is to issue a statement about what is likely to happen in the future, based on analysis and considerations of past experiments. Making a forecast is to obtain knowledge about uncertain events that are important in decision-making [9]. Time series prediction tries to predict the future based on past data, which takes a series of real data $x_{t-n}, ..., x_{t-2}, x_{t-1}, x_t$ and then obtains the prediction of future data $x_{t+1}, x_{t+2}, ..., x_{t+n}$. The goal of time series prediction or a model is to observe the series of real data, so that future data may be accurately predicted [35].

### 2.2. **Neural network.** Neural networks (NNs) are composed of many elements (Artificial Neurons), grouped into layers that are highly interconnected (with the synapses), and this structure has several inputs and outputs, which are trained to react (or give values) in a desired way based on input stimuli. These systems emulate, in some way, the human brain. Neural networks are required to learn to behave (Learning) and someone should be responsible for the teaching or training, based on prior knowledge of the environment problem [19]. Artificial neural networks are inspired by the architecture of the biological nervous system, which consists of a large number of relatively simple neurons that work in parallel to facilitate rapid decision-making [32].

2.3. **Ensemble neural networks.** An Ensemble Neural Network is a learning paradigm where many neural networks are jointly used to solve a problem [37]. A Neural network ensemble is a learning paradigm where a collection of a finite number of neural networks is trained for the same task [39]. It originates from Hansen and Salamon's work [16], which shows that the generalization ability of a neural network system can be significantly improved through ensembling a number of neural networks, i.e., training many neural networks and then combining their predictions. Since these ensemble models behave remarkably well, recently it has become a very hot topic in both neural networks and machine learning communities [36], and has already been successfully applied to diverse areas such as face recognition [13,18], optical character recognition [11,14,28], scientific image analysis [8], medical diagnosis [9,45], and seismic signals classification [38].

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. Examples are as follows. Hampshire and Waibel [14] utilize different objective functions to train distinct component neural networks. Cherkauer [7] trains component networks with different number of hidden units. Maclin and Shavlik [25] initialize component networks at different points in the weight space. Krogh and Vedelsby [24] employ cross-validation to create component networks. Opitz and Shavlik [33] exploit a genetic algorithm to train diverse knowledge based component networks. Yao and Liu [42] regard all the individuals in an evolved population of neural networks as component networks.

2.4. **Optimization.** The process of optimization is the process of obtaining the '*best*', and assuming that it is possible to measure and change what is 'good' or 'bad'. In practice, one wishes the 'most' or 'maximum' (e.g., salary) or the 'least' or 'minimum' (e.g., expenses). Therefore, the word '*optimum*' takes the meaning of '*maximum*' or '*minimum*' depending on the circumstances; 'optimum' is a technical term which implies quantitative measurement and is a stronger word than 'best', which is more appropriate for everyday use. Likewise, the word '*optimize*', which means to achieve an optimum, is a stronger word than 'improve'. Optimization theory is the branch of mathematics encompassing the quantitative studies of optima and methods for finding them. Optimization practice, on the other hand, is the collection of techniques, methods, procedures, and algorithms that can be used to find the optima [1].

2.5. **Genetic algorithms.** Genetic algorithms were introduced for the first time by a Professor of the University of Michigan named John Holland [16]. A genetic algorithm is a mathematical highly parallel algorithm that transforms a set of mathematical individual objects with regard to the time using operations based on evolution. The Darwinian laws of reproduction and survival of the fittest can be used, and after having appeared of natural form a series of genetic operations is used [12,32]. Each of these mathematical objects is usually a chain of characters (letters or numbers) of fixed length that adjusts to the model of the chains of chromosomes, and one associates to them a certain mathematical function that reflects the fitness. A Genetic Algorithm (GA) [27] assumes that a possible solution to a problem can be seen as an individual and is represented by a set of parameters. These parameters are the genes of a chromosome, representing the structure of the individual. This chromosome is evaluated by a fitness function, providing a fitness value to each individual, as to the suitability it has to solve the given problem. Through genetic evolution using crossover operations (matching of individuals to produce children) and mutation that simulates the biological behavior, combined with a selection process, the population of individuals is eliminating those with less ability, and tends to improve

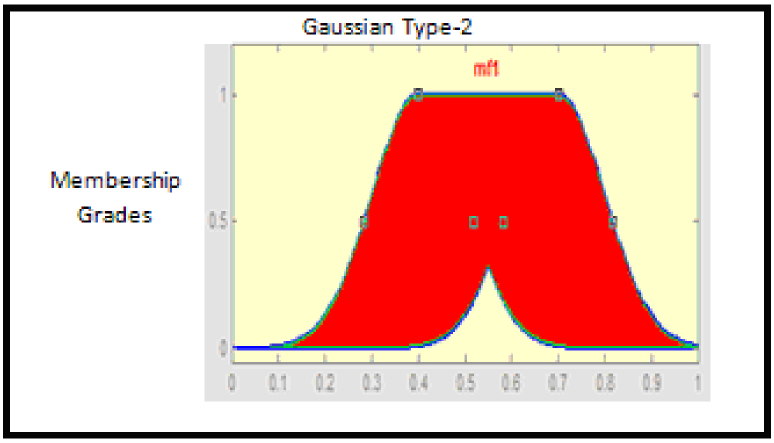


Figure 1. A type-2 fuzzy set representing a type-1 fuzzy set with uncertain standard deviation

overall fitness of the population, to produce a solution close to optimal performance for the given problem [17].

2.6. **Type-2 fuzzy systems.** The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-$n$ (Karnik and Mendel 1998) [21]. A higher-type number just indicates a higher "degree of fuzziness". The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2 [43].

As an example: Consider the case of a fuzzy set characterized by a Gaussian membership function with mean $m$ and a standard deviation that can take values in $[\sigma_1, \sigma_2]$, i.e.,

$$(x) = \exp\left\{-1/2\left[\frac{x-m}{\sigma}\right]^2\right\}; \quad \sigma \in [\sigma_1, \sigma_2] \tag{1}$$

Corresponding to each value of $\sigma$ we will get a different membership curve (as shown in Figure 1). So, the membership grade of any particular $x$ (*except* $x = m$) can take any of a number of possible values depending upon the value of $\sigma$, i.e., the membership grade is not a crisp number; it is a fuzzy set. Figure 1 shows the domain of the fuzzy set associated with $x = 0.7$; however, the membership function associated with this fuzzy set is not shown in Figure 1.

**Gaussian type-2 fuzzy set.** *A Gaussian type-2 fuzzy set is one in which the membership grade of every domain point is a Gaussian type-1 set contained in* $[0, 1]$.

**Interval type-2 fuzzy set.** *An interval type-2 fuzzy set is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in* $[0, 1]$.

**Footprint of uncertainty.** *Uncertainty in the primary memberships of a type-2 fuzzy set, $\tilde{A}$, consists of a bounded region that we call the "footprint of uncertainty" (FOU). Mathematically, it is the union of all primary membership functions [22].*

**Upper and lower membership functions.** *An "upper membership function" and a "lower membership functions" are two type-1 membership functions that are bounds for the FOU of a type-2 fuzzy set $\tilde{A}$. The upper membership function is associated with the upper bound of the FOU($\tilde{A}$). The lower membership function is associated with the lower bound of the FOU($\tilde{A}$).*

**Operations of Type-2 Fuzzy Sets**

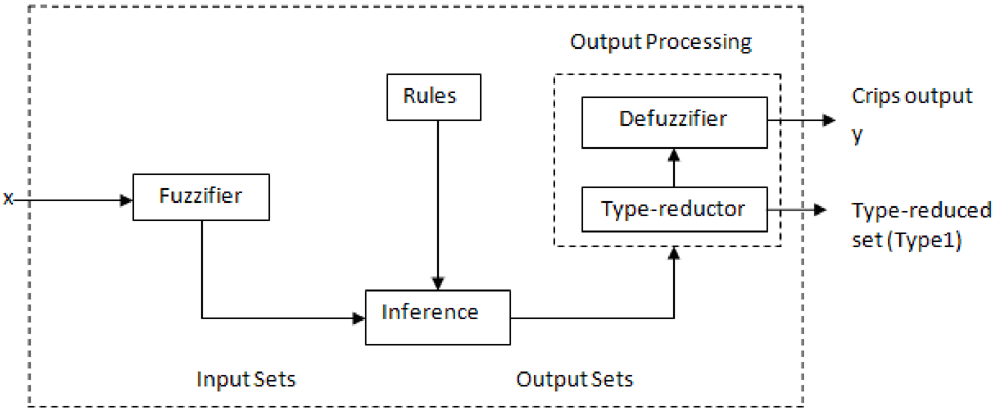


FIGURE 2. Structure of a type-2 fuzzy logic system

**Union of type-2 fuzzy sets.** *The union of $\tilde{A}_1$ and $\tilde{A}_2$ is another type-2 fuzzy set, just as the union of type-1 fuzzy sets $A_1$ and $A_2$ is another type-1 fuzzy set. More formally, we have the following expression:*

$$\tilde{A}_1 \cup \tilde{A}_2 = \int_{x \in X} \mu_{\tilde{A}_1 \cup \tilde{A}_2}(x)/x \tag{2}$$

**Intersection of type-2 fuzzy sets.** *The intersection of $\tilde{A}_1$ and $\tilde{A}_2$ is another type-2 fuzzy set, just as the intersection of type-1 fuzzy sets $A_1$ and $\tilde{A}_2$ is another type-1 fuzzy set. More formally, we have the following expression:*

$$\tilde{A}_1 \cap \tilde{A}_2 = \int_{x \in X} \mu_{\tilde{A}_1 \cup \tilde{A}_2}(x)/x \tag{3}$$

**Complement of a type-2 fuzzy set.** *The complement of a set is another type-2 fuzzy set, just as the complement of type-1 fuzzy set $A$ is another type-1 fuzzy set. More formally we have:*

$$\tilde{A}' = \int_{X} \mu_{\tilde{A}'1}(x)/x \tag{4}$$

*where the prime denotes complement in the above equation. In this equation $\mu_{\tilde{A}'1}$ is the secondary membership function.*

**Type-2 fuzzy rules.** *Consider a type-2 FLS having $r$ inputs $x_1 \in X_1, ..., x_r \in X_r$ and one output $y \in Y$. As in the type-1 case, we can assume that there are $M$ rules; but, in the type-2 case the $l$th rule has the form:*

$$R^1 : \ IF\ x_1\ is\ \tilde{A}_1^1\ and \ldots x_p\ is\ \tilde{A}_P^1,\ THEN\ y\ is\ \hat{Y}^1 \quad 1 = 1, \ldots, M \tag{5}$$

*The rules represent a type-2 relation between the input space $X_1 x \ldots x X_r$, and space output set space $Y$ of the type-2 fuzzy system.*

In the type-2 fuzzy system (Figure 2), as in the type-1 fuzzy system crisp inputs are first fuzzified into fuzzy input sets than then activate the inference block, which in the present case is associated with type-2 fuzzy sets [6].

3. **Problem Statement and Proposed Method.** The goal of this work was to implement a Genetic Algorithm to optimize the ensemble neural network architectures for each of the modules, and thus to find a neural network architecture that yields optimum results in each of the Time Series that was considered. Figure 3 indicates the general architecture, where the historical data of each time series for prediction is indicated, then the data is provided to the modules that will be optimized with the genetic algorithm for the

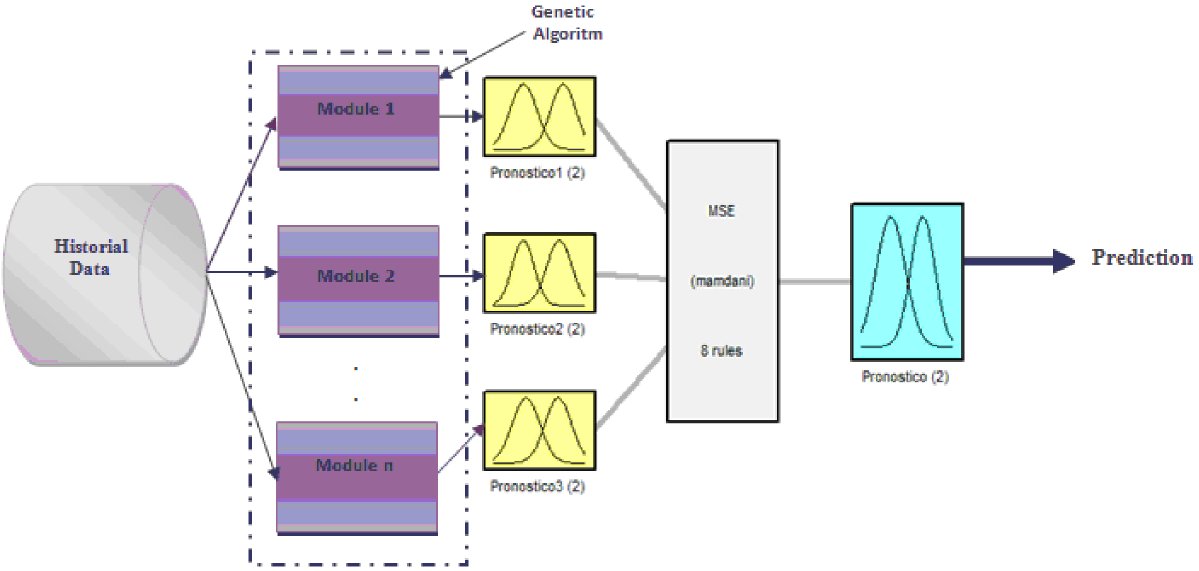


FIGURE 3. General architecture of the proposed ensemble neural network

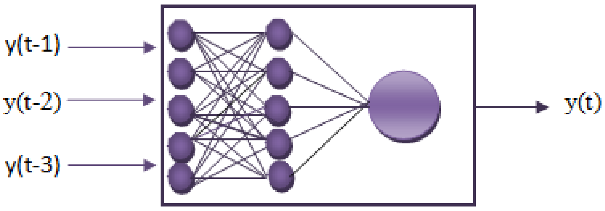


FIGURE 4. Architecture of the modules

ensemble network, and then results of these modules are integrated with an integration method based on fuzzy logic.

The architecture for each of the modules is based on three delays of the time series, as shown in Figure 4.

Historical data of the Mexican Stock Exchange time series was used for the ensemble neural network trainings, where each module was fed with the same information, unlike modular networks, where each module is fed with different data, which leads to architectures that are not uniform. The Mexican Stock Exchange (MSE) 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.

Derived from the monitoring of global trends and changes that have occurred in legislation, the MSE concluded with the conversion process, becoming a company whose shares are likely to be traded on the market stock exchange, with the initial public offering taking place on June 13 of 2008 with its shares representing its capital [31]. Data of the Mexican Stock Exchange time series: In this case 800 points are used, which correspond to a period from 11/09/05 to 01/15/09 (as shown in Figure 5). In this case 70% of the data were used for the ensemble neural network training and 30% to test the network [31].

After some tests were performed, another period of data from 04/08/08 to 05/09/11 for this series was also selected (as shown in Figure 6).

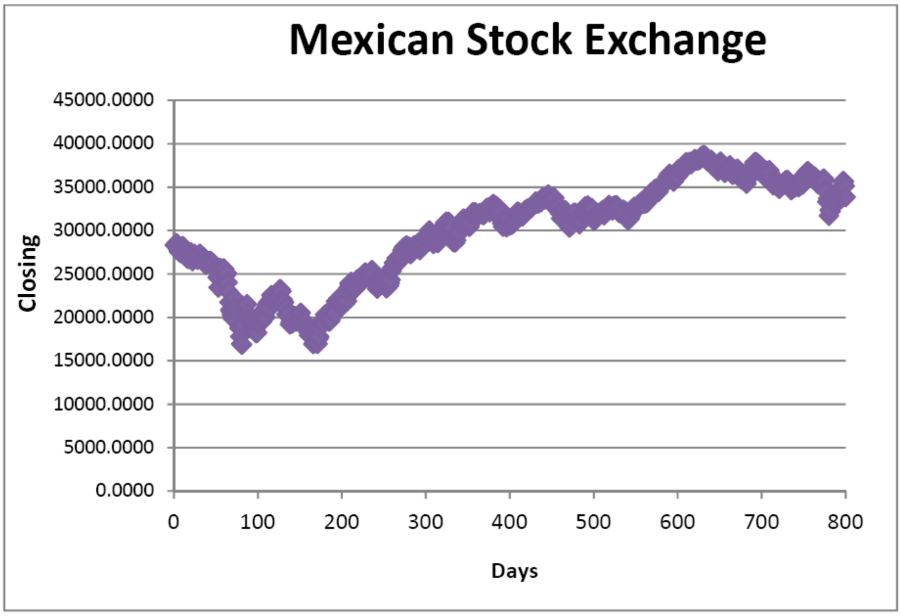


Figure 5. Mexican stock exchange of time series 1

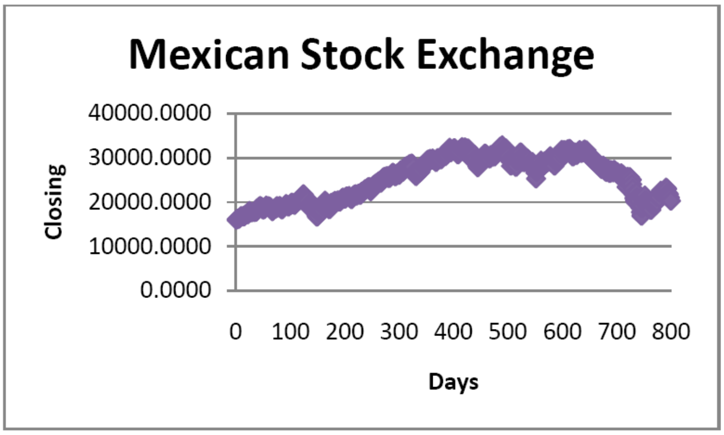


Figure 6. Mexican stock exchange of time series 2

For the optimization of the structure of the ensemble neural network, a genetic algorithm was used. The chromosome of the genetic algorithm represents the structure of the ensemble.

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

$$EM = \left( \sum_{i=1}^{D} |a_i - x_i| \right) \Big/ D \tag{6}$$

where $a$ corresponds to the predicted data depending on the output of the network modules, $X$ represents real data, $D$ the Number of Data points and $EM$ is the total average prediction error. The corresponding chromosome structure is shown in Figure 7.

The chromosome shows the proposed structure to optimize the architecture of the ensemble neural network with a genetic algorithm. First, the number of modules is represented in which a limit of 5 modules was considered, then the number of layers with a maximum of 3 layers and the number of neurons with a maximum of 30 neurons are shown. The selection method is the Roulette, the percentage of crossover and mutation

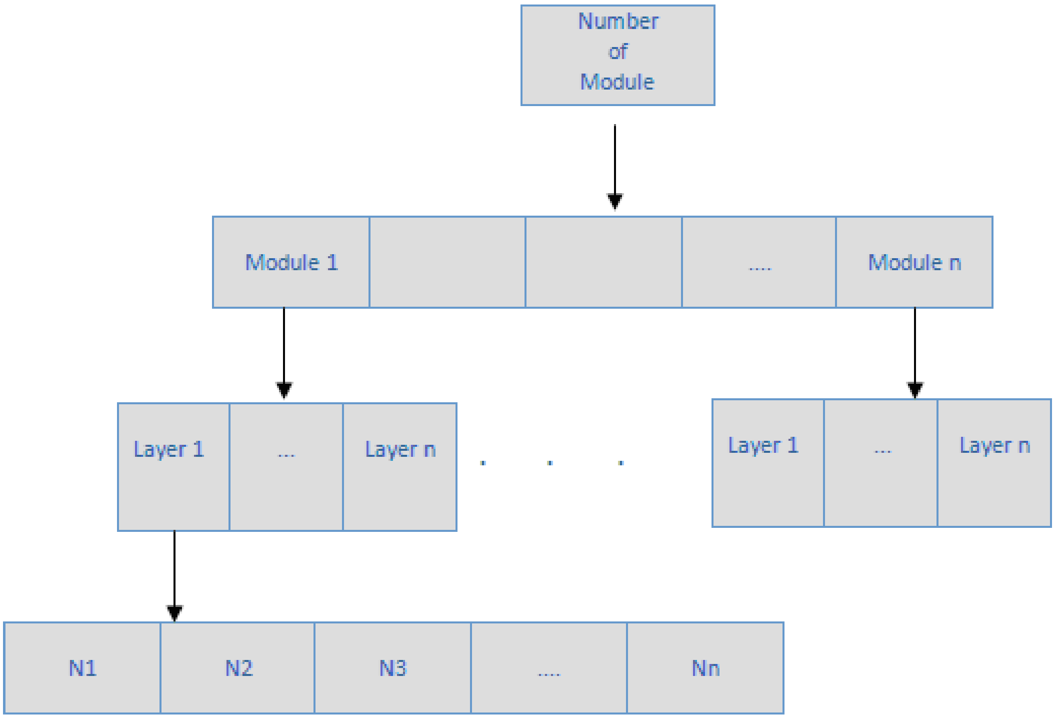


FIGURE 7. Chromosome structure to optimize the ensemble neural network

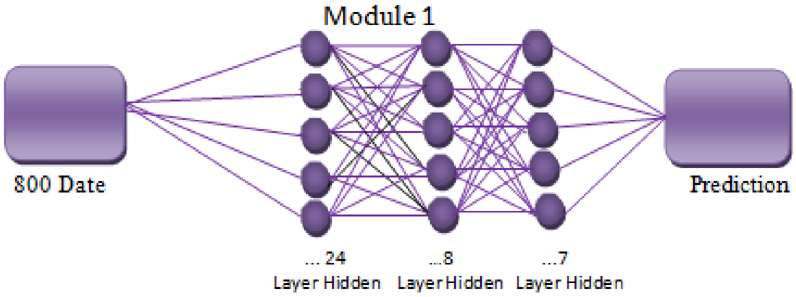


FIGURE 8. Monolithic neural network architecture

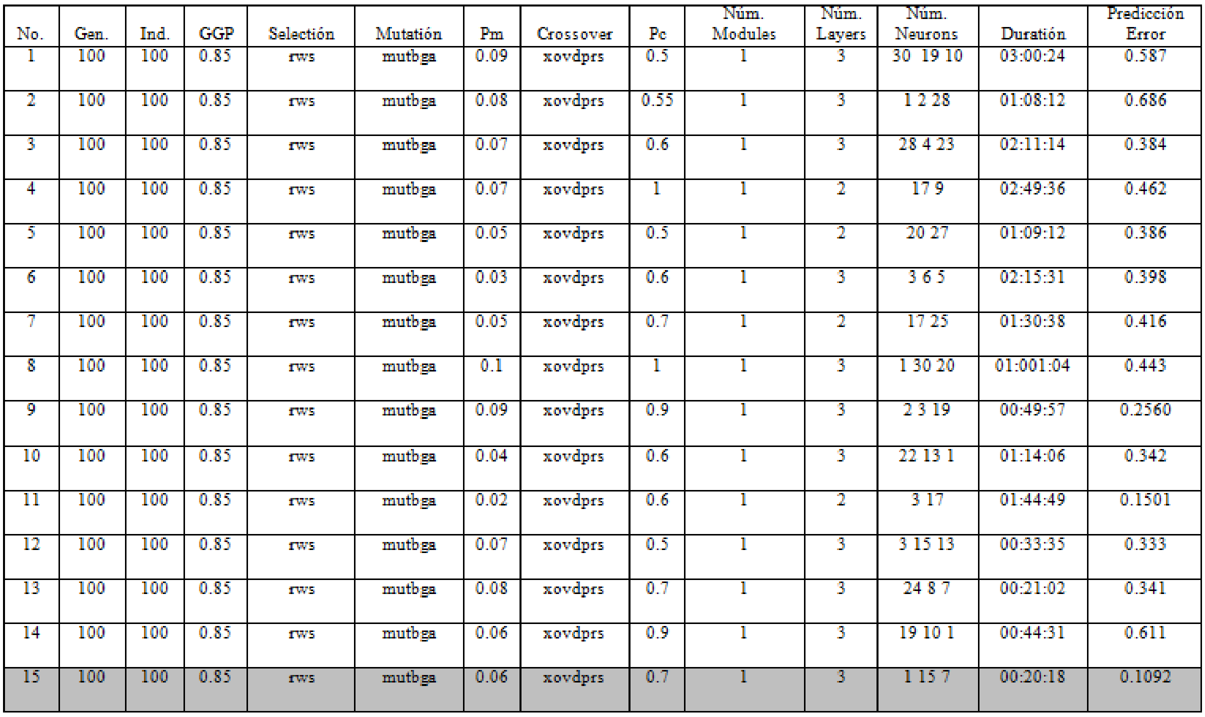
are changed at random, as suggested in current works, like in [24]. The crossover rate is changed in the range of [0.5, 1] and the mutation rate is changed in the range of [0, 0.1].

4. **Simulation Results.** In this section the simulation results obtained with the genetic algorithm optimization of the ensemble neural network for the prediction of the Mexican Stock Exchange are presented. A genetic algorithm was first used to optimize the structure of a monolithic neural network and the best obtained architecture is represented in Figure 8.

In this architecture there are two layers in each module. In module 1, in the first layer there are 24 neurons, 8 neurons in second layer and 7 neurons in third layer. The Levenberg-Marquardt (LM) training method was used and 3 delays for the network were considered.

Table 1 shows the genetic algorithm results, where the best achieved prediction error is of 0.1092, which is indicated in row number 15.

Table 1. Genetic algorithm results for the monolithic network



| No. | Gen. | Ind. | GGP | Selectión | Mutatión | Pm | Crossover | Pc | Núm. Modules | Núm. Layers | Núm. Neurons | Duratión | Predicción Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 100 | 0.85 | rws | mutbga | 0.09 | xovdprs | 0.5 | 1 | 3 | 30 19 10 | 03:00:24 | 0.587 |
| 2 | 100 | 100 | 0.85 | rws | mutbga | 0.08 | xovdprs | 0.55 | 1 | 3 | 1 2 28 | 01:08:12 | 0.686 |
| 3 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.6 | 1 | 3 | 28 4 23 | 02:11:14 | 0.384 |
| 4 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 1 | 1 | 2 | 17 9 | 02:49:36 | 0.462 |
| 5 | 100 | 100 | 0.85 | rws | mutbga | 0.05 | xovdprs | 0.5 | 1 | 2 | 20 27 | 01:09:12 | 0.386 |
| 6 | 100 | 100 | 0.85 | rws | mutbga | 0.03 | xovdprs | 0.6 | 1 | 3 | 3 6 5 | 02:15:31 | 0.398 |
| 7 | 100 | 100 | 0.85 | rws | mutbga | 0.05 | xovdprs | 0.7 | 1 | 2 | 17 25 | 01:30:38 | 0.416 |
| 8 | 100 | 100 | 0.85 | rws | mutbga | 0.1 | xovdprs | 1 | 1 | 3 | 1 30 20 | 01:001:04 | 0.443 |
| 9 | 100 | 100 | 0.85 | rws | mutbga | 0.09 | xovdprs | 0.9 | 1 | 3 | 2 3 19 | 00:49:57 | 0.2560 |
| 10 | 100 | 100 | 0.85 | rws | mutbga | 0.04 | xovdprs | 0.6 | 1 | 3 | 22 13 1 | 01:14:06 | 0.342 |
| 11 | 100 | 100 | 0.85 | rws | mutbga | 0.02 | xovdprs | 0.6 | 1 | 2 | 3 17 | 01:44:49 | 0.1501 |
| 12 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.5 | 1 | 3 | 3 15 13 | 00:33:35 | 0.333 |
| 13 | 100 | 100 | 0.85 | rws | mutbga | 0.08 | xovdprs | 0.7 | 1 | 3 | 24 8 7 | 00:21:02 | 0.341 |
| 14 | 100 | 100 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.9 | 1 | 3 | 19 10 1 | 00:44:31 | 0.611 |
| 15 | 100 | 100 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.7 | 1 | 3 | 1 15 7 | 00:20:18 | 0.1092 |

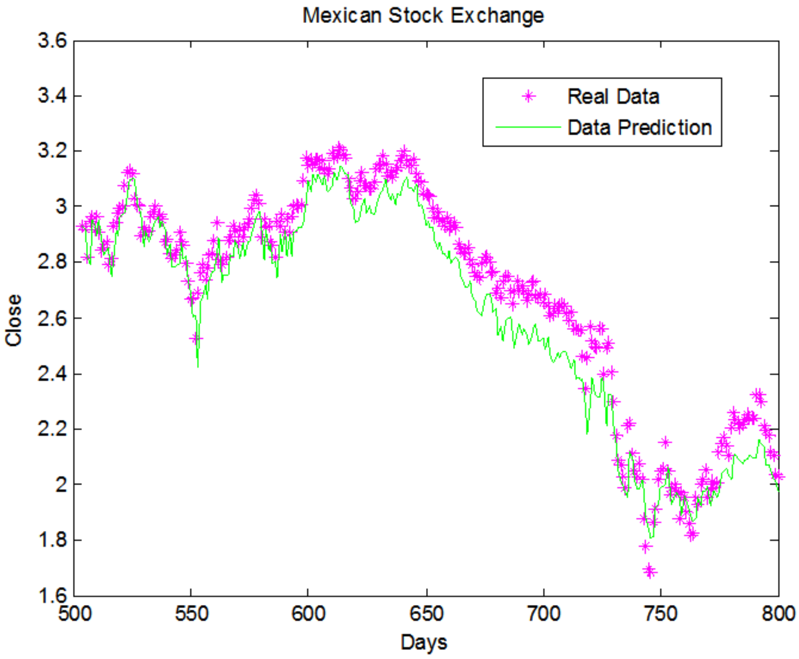


Figure 9. Prediction with the optimized monolithic neural network of the MSE

Figure 9 shows the plot of real data against the predicted data generated by the monolithic neural network optimized with the genetic algorithm. After using a genetic algorithm to optimize the structure of the ensemble neural network and having considered 5 modules at the most, the best achieved architecture is shown in Figure 10.

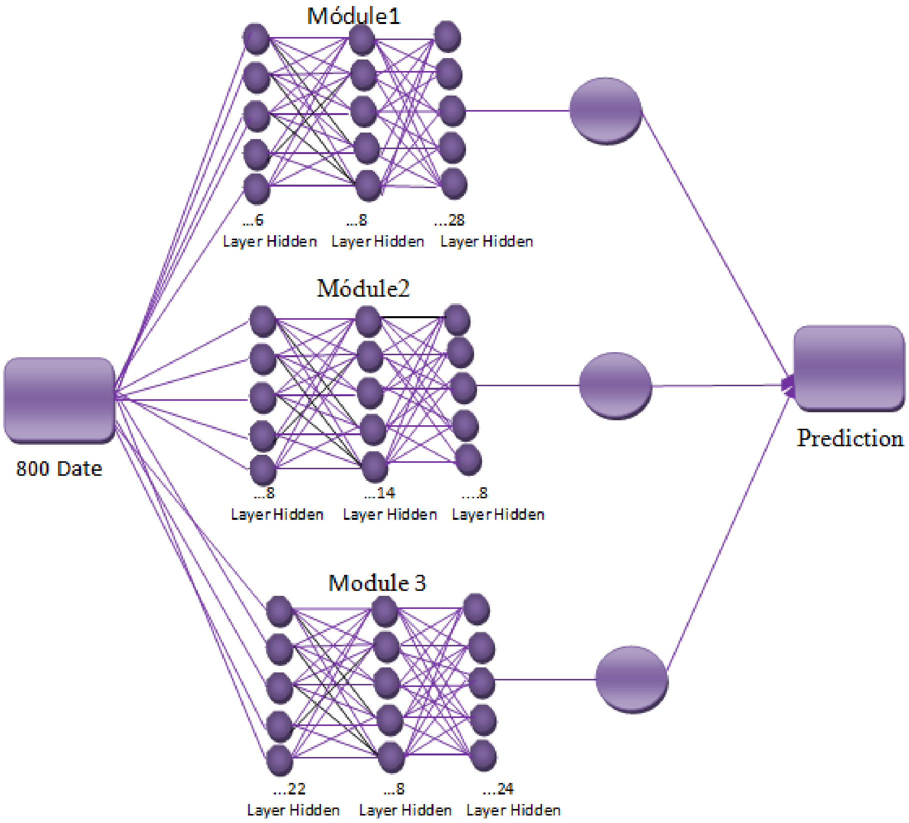


FIGURE 10. Ensemble neural network architecture

In this architecture there are three layers in each module. In module 1, there are 6 neurons in the first layer, 8 neurons in the second layer and 28 neurons in the third layer. In module 2, there are 8 neurons in the first layer, 14 neurons in the second and 8 in the third layer, and in module 3 there are 22 neurons in the first layer, 8 neurons in the second and 24 neurons in the third layer. The Levenberg-Marquardt (LM) training method was used; 3 delays for the network were considered. Table 2 shows the genetic algorithm results (as shown in Figure 9), where the best prediction error is of 0.0112, which is indicated in row number three.

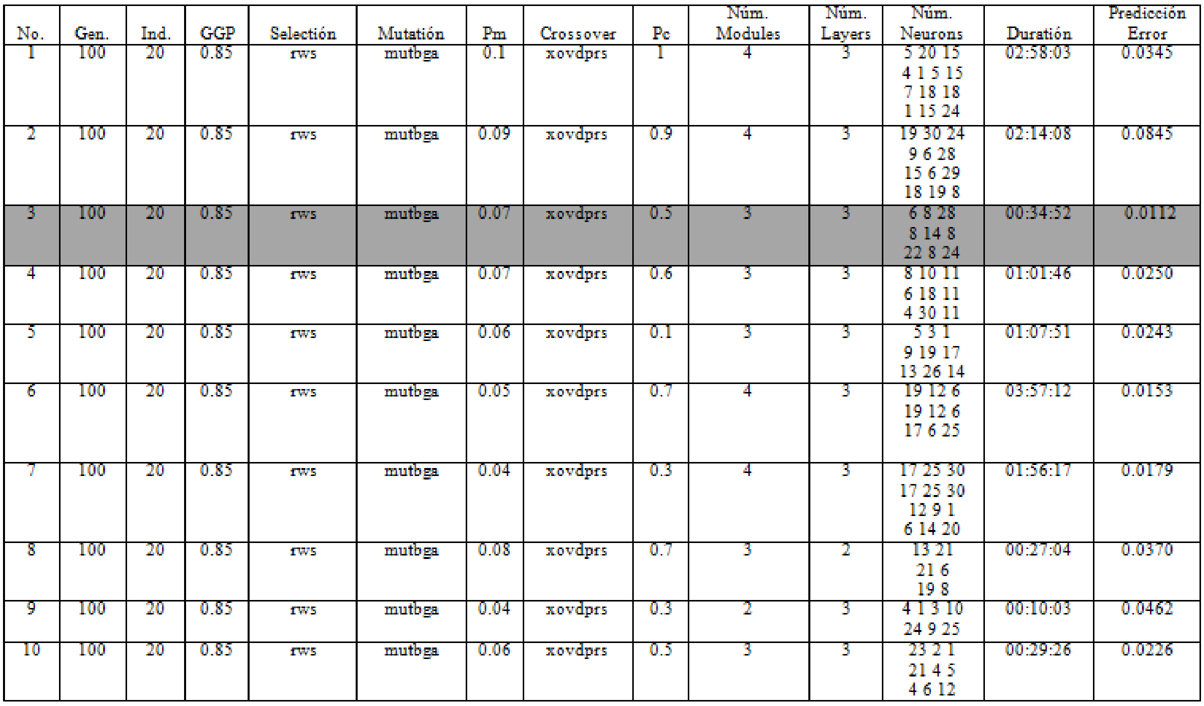
Figure 11 shows the plot of real data against the predicted data generated by the ensemble neural network optimized with the genetic algorithm. Fuzzy integration is performed by implementing a type-1 fuzzy system in which the best result was in the evolution of row number 6 of Table 3 with an error of 0.1467.

Fuzzy integration is also performed by implementing a type-2 fuzzy system in which the results were as follows: for the best evolution with a degree of uncertainty of 0.3 a forecast error of 0.0193 was obtained, and with a degree of uncertainty of 0.4 a forecast error of 0.0278 was obtained and with a degree of uncertainty of 0.5 a forecast error of 0.0293 was obtained, as shown in Table 4.

Fuzzy integration is performed by implementing a type-1 fuzzy system, where the best result was obtained in evolution 4 with an error of 0.1023, which is shown in Table 6.

Fuzzy integration is also performed by implementing a type-2 fuzzy system in which the results were for the best evolution as follows: with a degree of uncertainty of 0.3 yielded a forecast error of 0.0232, with a degree of uncertainty of 0.4 an error of 0.0120 and with a degree of uncertainty of 0.5 an error of 0.0192, as shown in Table 7.

Table 2. Genetic algorithm results for the ensemble neural network



| No. | Gen. | Ind. | GGP | Selectión | Mutatión | Pm | Crossover | Pc | Núm. Modules | Núm. Layers | Núm. Neurons | Duratión | Predicción Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 20 | 0.85 | rws | mutbga | 0.1 | xovdprs | 1 | 4 | 3 | 5 20 15<br>4 1 5 15<br>7 18 18<br>1 15 24 | 02:58:03 | 0.0345 |
| 2 | 100 | 20 | 0.85 | rws | mutbga | 0.09 | xovdprs | 0.9 | 4 | 3 | 19 30 24<br>9 6 28<br>15 6 29<br>18 19 8 | 02:14:08 | 0.0845 |
| 3 | 100 | 20 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.5 | 3 | 3 | 6 8 28<br>8 14 8<br>22 8 24 | 00:34:52 | 0.0112 |
| 4 | 100 | 20 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.6 | 3 | 3 | 8 10 11<br>6 18 11<br>4 30 11 | 01:01:46 | 0.0250 |
| 5 | 100 | 20 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.1 | 3 | 3 | 5 3 1<br>9 19 17<br>13 26 14 | 01:07:51 | 0.0243 |
| 6 | 100 | 20 | 0.85 | rws | mutbga | 0.05 | xovdprs | 0.7 | 4 | 3 | 19 12 6<br>19 12 6<br>17 6 25 | 03:57:12 | 0.0153 |
| 7 | 100 | 20 | 0.85 | rws | mutbga | 0.04 | xovdprs | 0.3 | 4 | 3 | 17 25 30<br>17 25 30<br>12 9 1<br>6 14 20 | 01:56:17 | 0.0179 |
| 8 | 100 | 20 | 0.85 | rws | mutbga | 0.08 | xovdprs | 0.7 | 3 | 2 | 13 21<br>21 6<br>19 8 | 00:27:04 | 0.0370 |
| 9 | 100 | 20 | 0.85 | rws | mutbga | 0.04 | xovdprs | 0.3 | 2 | 3 | 4 1 3 10<br>24 9 25 | 00:10:03 | 0.0462 |
| 10 | 100 | 20 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.5 | 3 | 3 | 23 2 1<br>21 4 5<br>4 6 12 | 00:29:26 | 0.0226 |

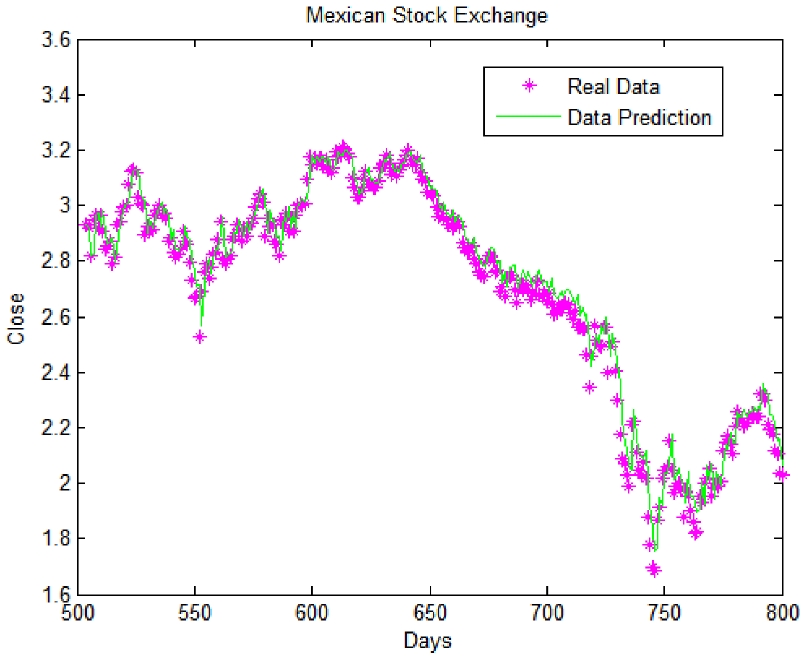


Figure 11. Prediction with the optimized ensemble neural network of the MSE

Table 8 summarizes the result of the monolithic and ensemble neural networks, and it can be noted that for the prediction of this time series the ensemble neural network is better than the monolithic neural network. The errors obtained with the monolithic neural network (MNN) in series number 1 is of 0.1092 and 0.084 for series number 2, and for the ensemble neural network (ENN) the error obtained in series number 1 is of 0.0112 and 0.0132 for series number 2.

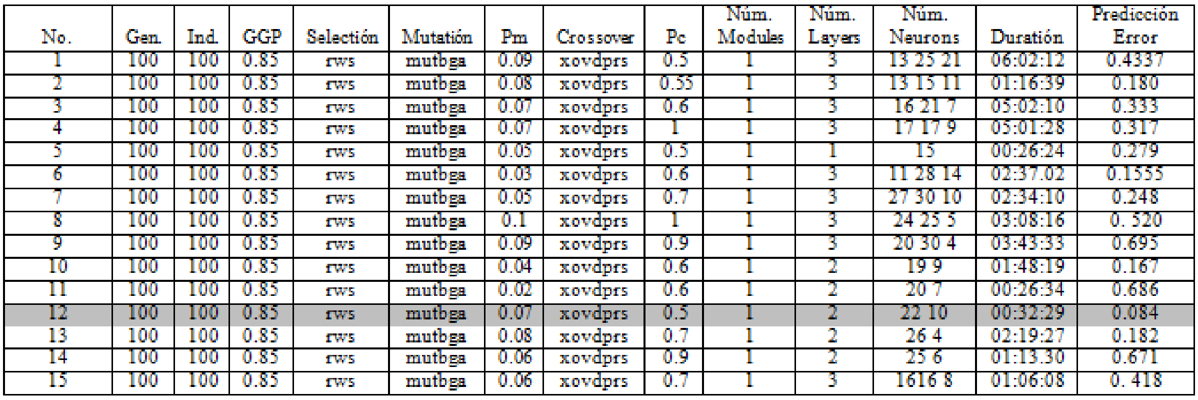
TABLE 3. Results of type-1 fuzzy integration of MSE

| **Evolution** | **Type-1 Fuzzy Integration** |
|---|---|
| Evolution 1 | 0.1630 |
| Evolution 2 | 0.3169 |
| Evolution 3 | 0.1801 |
| Evolution 4 | 0.2151 |
| Evolution 5 | 0.1811 |
| **Evolution 6** | **0.1467** |
| Evolution 7 | 0.2554 |
| Evolution 8 | 0.1759 |
| Evolution 9 | 0.3274 |
| Evolution 10 | 0.2773 |

TABLE 4. Results of type-2 fuzzy integration of MSE

| **Evolution** | **0.3 Uncertainty** | **0.4 Uncertainty** | **0.5 Uncertainty** |
|---|---|---|---|
| Evolution 1 | 0.0178 | 0.0453 | 0.028 |
| Evolution 2 | 0.0479 | 0.0494 | 0.01968 |
| Evolution 3 | 0.0189 | 0.0182 | 0.0217 |
| **Evolution 4** | **0.0193** | **0.0278** | **0.0293** |
| Evolution 5 | 0.0448 | 0.0276 | 0.0277 |
| Evolution 6 | 0.0131 | 0.0120 | 0.01953 |
| Evolution 7 | 0.0134 | 0.0164 | 0.01958 |
| Evolution 8 | 0.0453 | 0.0267 | 0.0257 |
| Evolution 9 | 0.0462 | 0.0599 | 0.0575 |
| Evolution 10 | 0.0147 | 0.0263 | 0.0282 |

TABLE 5. Genetic algorithm results for the monolithic network for the time series 2



| No. | Gen. | Ind. | GGP | Selectión | Mutatión | Pm | Crossover | Pc | Núm. Modules | Núm. Layers | Núm. Neurons | Duratión | Predicción Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 100 | 0.85 | rws | mutbga | 0.09 | xovdprs | 0.5 | 1 | 3 | 13 25 21 | 06:02:12 | 0.4337 |
| 2 | 100 | 100 | 0.85 | rws | mutbga | 0.08 | xovdprs | 0.55 | 1 | 3 | 13 15 11 | 01:16:39 | 0.180 |
| 3 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.6 | 1 | 3 | 16 21 7 | 05:02:10 | 0.333 |
| 4 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 1 | 1 | 3 | 17 17 9 | 05:01:28 | 0.317 |
| 5 | 100 | 100 | 0.85 | rws | mutbga | 0.05 | xovdprs | 0.5 | 1 | 1 | 15 | 00:26:24 | 0.279 |
| 6 | 100 | 100 | 0.85 | rws | mutbga | 0.03 | xovdprs | 0.6 | 1 | 3 | 11 28 14 | 02:37.02 | 0.1555 |
| 7 | 100 | 100 | 0.85 | rws | mutbga | 0.05 | xovdprs | 0.7 | 1 | 3 | 27 30 10 | 02:34:10 | 0.248 |
| 8 | 100 | 100 | 0.85 | rws | mutbga | 0.1 | xovdprs | 1 | 1 | 3 | 24 25 5 | 03:08:16 | 0. 520 |
| 9 | 100 | 100 | 0.85 | rws | mutbga | 0.09 | xovdprs | 0.9 | 1 | 3 | 20 30 4 | 03:43:33 | 0.695 |
| 10 | 100 | 100 | 0.85 | rws | mutbga | 0.04 | xovdprs | 0.6 | 1 | 2 | 19 9 | 01:48:19 | 0.167 |
| 11 | 100 | 100 | 0.85 | rws | mutbga | 0.02 | xovdprs | 0.6 | 1 | 2 | 20 7 | 00:26:34 | 0.686 |
| 12 | 100 | 100 | 0.85 | rws | mutbga | 0.07 | xovdprs | 0.5 | 1 | 2 | 22 10 | 00:32:29 | 0.084 |
| 13 | 100 | 100 | 0.85 | rws | mutbga | 0.08 | xovdprs | 0.7 | 1 | 2 | 26 4 | 02:19:27 | 0.182 |
| 14 | 100 | 100 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.9 | 1 | 2 | 25 6 | 01:13.30 | 0.671 |
| 15 | 100 | 100 | 0.85 | rws | mutbga | 0.06 | xovdprs | 0.7 | 1 | 3 | 1616 8 | 01:06:08 | 0. 418 |

5. **Comparison of Results.** This section shows a comparison of results with two studies that were carried out. It is important to note that the outcome of this is that the time series prediction is very useful, as investors seek to place the money to get an attractive return and companies that need capital to invest in the development of their businesses,

Table 6. Results type-1 fuzzy integration of MSE for time series 2

| Evolution | Type-1 Fuzzy Integration |
|---|---|
| Evolution 1 | 0.1630 |
| Evolution 2 | 0.1212 |
| Evolution 3 | 0.1997 |
| **Evolution 4** | **0.1023** |
| Evolution 5 | 0.1730 |
| Evolution 6 | 0.2188 |
| Evolution 7 | 0.4368 |
| Evolution 8 | 0.1042 |
| Evolution 9 | 0.3020 |
| Evolution 10 | 0.2405 |

Table 7. Results type-2 fuzzy integration of MSE for time series number 2

| **Evolution** | **0.3 Uncertainty** | **0.4 Uncertainty** | **0.5 Uncertainty** |
|---|---|---|---|
| Evolution 1 | 0.0171 | 0.0183 | 0.0167 |
| Evolution 2 | 0.0190 | 0.0188 | 0.0163 |
| Evolution 3 | 0.0225 | 0.0206 | 0.0196 |
| Evolution 4 | 0.0229 | 0.0205 | 0.0198 |
| Evolution 5 | 0.0233 | 0.0249 | 0.0199 |
| **Evolution 6** | **0.0232** | **0.0120** | **0.0192** |
| Evolution 7 | 0.0183 | 0.0184 | 0.0166 |
| Evolution 8 | 0.0215 | 0.0191 | 0.0191 |
| Evolution 9 | 0.020 | 0.0163 | 0.0164 |
| Evolution 10 | 0.0228 | 0.0176 | 0.0222 |

Table 8. Results of the monolithic and ensemble neural network for the time series

| Time Series | MNN | ENN |
|---|---|---|
| Mexican Stock Exchange Series 1 | 0.1092 | 0.0112 |
| Mexican Stock Exchange Series 2 | 0.084 | 0.0132 |

and can be analyzed to predict events before an event and thus can take preventive or remedial help avoid unwanted circumstances, or impair any process.

A comparison of results for this time series of the Mexican Stock Exchange for the period of 11/09/05 to 01/15/09 with the paper entitled: "An Ensemble Neural Network Architecture with Fuzzy Response Integration for Complex Time Series Prediction" [31], shows that the best result was using an ensemble neural network architecture with 1 layer using 3 delays. The error obtained by the average integration was 0.0479, by the average weighted integration was 0.0519 and by fuzzy integration was 0.1094, respectively. It was also decided to implement an evolutionary approach to optimize the membership functions and rules of this system and the obtained error was of 0.045848. The genetic algorithm was applied to optimize the monolithic neural network for the Mexican Stock Exchange time series: 0.1092 (as shown in Figure 6 and Table 1). In this paper, the best result when applying the genetic algorithm to optimize the ensemble neural network was:

Table 9. Results of $t$ student test for the time series

| Time Series | N (Type-1) | N (Type-2) | T Value | P Value |
|---|---|---|---|---|
| Mexican Stock Exchange Series 1 | 10 | 30 | 15.59 | 0.00 |
| Mexican Stock Exchange Series 2 | 10 | 30 | 10.29 | 0.00 |

0.00112 (as shown in Figure 8 and Table 2). This shows that our hybrid ensemble neural approach produces better results for this time series.

After a comparison of results for this time series of the Mexican Stock Exchange for the period of 11/09/05 to 01/15/09 with the paper entitled: "A new Approach for Time Series Prediction Using Ensembles of ANFIS Models" [26], where the best result was when integration by weighted average was used (an error of 0.0123 was obtained). Also, in this paper with integration by average we obtained an error of 0.0127, which shows a very similar result. In this case, no significant difference can be found. In the literature there are only two references as to forecast the time series of the Mexican stock exchange, this is why the comparison was made with only these two studies.

**Statistical *t* students test.** In this section, results with the $t$ student test are shown, and it can be noticed that there is statistical evidence of significant difference in the results between the type-1 and type-2 fuzzy systems. In other words, it can be said that for this time series prediction is better with type-2 fuzzy integration than with type-1 fuzzy integration. The number of samples used for the type-1 were 10 and 30 samples for type-2, the value obtained for type-1 in series number 1 is 15.59 and 10.29 for series number 2, so that the results show that in our tests more than 99% confidence was obtained of significant improvement with type-2 fuzzy logic according to the achieved value of $P$, as shown in Table 9.

## 6. Conclusions.

In conclusion, it can be said that the obtained results with the proposed method for the optimization of the neural network ensemble show a good prediction for the time series of the Mexican stock exchange, since it has managed to minimize the prediction error of the series against historical results obtained in other works. After achieving these results, the efficiency of the algorithms applied to optimize the neural network ensemble architecture was verified. In this case, the method was efficient but it also has certain disadvantages, sometimes the results may not be good, but genetic algorithms can be considered as good techniques for solving search and optimization problems. In conclusion, the use of ensemble neural networks with type-2 fuzzy integration could be a good choice in predicting complex time series.

**Acknowledgment.** We would like to express our gratitude to the CONACYT, Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

## REFERENCES

[1] A. Andreas and W. Sheng, Introduction optimization, *Practical Optimization Algorithms and Engineering Applications*, pp.1-4, 2007.

[2] P. D. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, Springer-Verlag, New York, 2002.

[3] O. Castillo and P. Melin, Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory, *IEEE Trans. on Neural Networks*, vol.13, no.6, pp.1395-1408, 2002.

[4] O. Castillo and P. Melin, Simulation and forecasting complex economic time series using neural networks and fuzzy logic, *Proc. of the International Neural Networks Conference*, vol.3, pp.1805-1810, 2001.

[5] O. Castillo and P. Melin, Simulation and forecasting complex financial time series using neural networks and fuzzy logic, *Proc. of the IEEE the International Conference on Systems, Man and Cybernetics*, vol.4, pp.2664-2669, 2001.

[6] O. Castillo and P. Melin, *Type-2 Fuzzy Logic: Theory and Applications*, Springer-Verlag, New York, 2008.

[7] K. J. Cherkauer, Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks, *Proc. of AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Portland, OR, USA, pp.15-21, 1996.

[8] P. Cowpertwait and A. Metcalfe, *Time Series, Introductory Time Series with R*, Springer Dordrecht Heidelberg, London, New York, 2009.

[9] P. Cunningham, J. Carney and S. Jacob, Stability problems with artificial neural networks and the ensemble solution, *Artificial Intelligence in Medicine*, vol.20, no.3, pp.217-225, 2000.

[10] N. Davey, S. Hunt and R. Frank, *Time Series Prediction and Neural Networks*, University of Hertfordshire, Hatfield, UK, 1999.

[11] H. Drucker, R. Schapire and P. Simard, Improving performance in neural networks using a boosting algorithm, *Advances in Neural Information Processing Systems*, vol.5, pp.42-49, 1993.

[12] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[13] S. Gutta and H. Wechsler, Face recognition using hybrid classifier systems, *Proc. of ICNN-96*, Washington, DC, USA, pp.1017-1022, 1996.

[14] J. Hampshire and A. Waibel, A novel objective function for improved phoneme recognition using time-delay neural networks, *IEEE Trans. on Neural Networks*, vol.1, no.2, pp.216-228, 1990.

[15] L. K. Hansen and P. Salomon, Ensemble methods for handwritten digit recognition, *Proc. of IEEE Workshop on Neural Networks for Signal Processing*, Helsingoer, Denmark, pp.333-342, 1992.

[16] L. K. Hansen and P. Salomon, Neural network ensembles, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.12, no.10, pp.993-1001, 1990.

[17] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[18] F. J. Huang, Z. Huang, H.-J. Zhang and T. H. Chen, Pose invariant face recognition, *Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000.

[19] J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1996.

[20] N. Karnik and M. Mendel, Applications of type-2 fuzzy logic systems to forecasting of time-series, *Information Sciences*, vol.120, no.1-4, pp.89-111, 1999.

[21] N. Karnik and M. Mendel, Introduction to type-2 fuzzy logic systems, *IEEE Trans. on Signal Processing*, vol.2, pp.915-920, 1998.

[22] N. Karnik and M. Mendel, Operations on type-2 fuzzy sets, *Fuzzy Sets and Systems*, vol.122, pp.327-348, 2001.

[23] A. Kehagias and V. Petridis, Predictive modular neural networks for time series classification, *Neural Networks*, vol.10, no.1, pp.31-49, 1997.

[24] A. Krogh and J. Vedelsby, Neural network ensembles, cross validation, and active learning, in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky and T. Leen (eds.), Cambridge, MA, MIT Press, 1995.

[25] R. Maclin and J. W. Shavlik, Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks, *Proc. of IJCAI-95*, Montreal, Canada, pp.524-530, 1995.

[26] L. P. Maguire, B. Roche, T. M. McGinnity and L. J. McDaid, Predicting a chaotic time series using a fuzzy neural network, *Information Sciences*, vol.112, no.1-4, pp.125-136, 1998.

[27] K. Man, K. Tang and S. Kwong, Genetic algorithms and designs, *Introduction, Background and Biological Background*, Springer-Verlag London Limited, 1998.

[28] J. Mao, A case study on bagging, boosting and basic ensembles of neural networks for OCR, *Proc. of IJCNN-98*, Anchorage, AK, vol.3, pp.1828-1833, 1998.

[29] P. Melin, O. Castillo, S. Gonzalez, J. Cota, W. Trujillo and P. Osuna, *Design of Modular Neural Networks with Fuzzy Integration Applied to Time Series Prediction*, Springer Berlin/Heidelberg, 2007.

[30] P. Melin, J. Soto, O. Castillo and J. Soria, A new approach for time series prediction using ensembles of ANFIS models, *Expert System with Applications*, 2011.

[31] *Mexico Bank Database*, http://www.banxico.org.mx, 2011.

[32] I. M. Multaba and M. A. Hussain, Application of neural networks and other learning, *Technologies in Process Engineering*, 2001.

[33] D. W. Opitz and J. W. Shavlik, Generating accurate and diverse members of a neural network ensemble, in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (eds.), MIT Press, 1996.

[34] E. A. Plummer, *Time Series Forecasting with Feed-Forward Neural Networks: Guidelines and Limitations*, University of Wyoming, 2000.

[35] M. Pulido, A. Mancilla and P. Melin, An ensemble neural network architecture with fuzzy response integration for complex time series prediction, *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*, vol.257, pp.85-110, 2009.

[36] A. Sharkey, *One Combining Artificial of Neural Nets*, Department of Computer Science, University of Sheffield, U.K., 1996.

[37] A. Sharkey, *Combining Artificial Neural Nets: Ensemble and Modular Multi-net Systems*, Springer-Verlag, London, 1999.

[38] Y. Shimshoni, Intrator classification of seimic signal by integrating ensemble of neural networks, *IEEE Trans. on Signal Processing*, vol.461, no.5, pp.1194-1201, 1998.

[39] P. Sollich and A. Krogh, Learning with ensembles: How over-fitting can be useful, in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (eds.), Denver, CO, MIT Press, 1996.

[40] R. M. Tong and H. T. Nguyen, *Fuzzy Sets and Applications: Selected Papers*, John Wiley, New York, 1987.

[41] R. N. Yadav, P. K. Kalra and J. John, Time series prediction with single multiplicative neuron model, *Soft Computing for Time Series Prediction, Applied Soft Computing*, vol.7, no.4, pp.1157-1163, 2007.

[42] X. Yao and Y. Liu, Making use of population information in evolutionary artificial neural networks, *IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics*, vol.28, no.3, pp.417-425, 1998.

[43] L. A. Zadeh, *Fuzzy Sets and Applications: Selected Papers*, John Wiley, New York, 1987.

[44] L. Zhao and Y. Yang, PSO-based single multiplicative neuron model for time series prediction, *Expert Systems with Applications*, vol.36, no.2, pp.2805-2812, 2009.

[45] Z.-H. Zhou, Y. Jiang, Y.-B. Yang and S.-F. Chen, Lung cancer cell identification based on artificial neural network ensembles, *Artificial Intelligence in Medicine*, vol.24, no.1, pp.25-36, 2002.