ISSN 2007-9737

Bird Swarm Algorithm and Particle Swarm Optimization in Ensemble Recurrent Neural Networks Optimization for Time Series Prediction

Martha Pulido, Patricia Melin*

Tecnológico Nacional de México, Campus Tijuana,
Mexico

{martha.pulido, pmelin}@tectijuana.mx

Abstract. Recurrent Neural Networks have proven to provide good results in time series prediction. In this paper, an ensemble recurrent neural network, for prediction is used. Euro/Mexican pesos and Euro/Dollar series are utilized to develop a prediction model. The design of this consists of an ensemble recurrent neural network and the optimization of its structure is achieved with the bird swarm algorithm and Particle Swarm Optimization. The outputs of the networks are integrated with type-1 and type-2 fuzzy systems. These fuzzy systems are of the Mamdani type. The tests were realized with the designed method and a good result was obtained in the application to time series, as well as a comparison of the two optimization algorithms was made.

Keywords. Bird swarm algorithm, time series, recurrent neural networks, optimization, prediction.

1 Introduction

In nature, we find situations and processes for which there is no immediate explanation. A particular case of this is swarming, in which many simple individuals act in such a way that intelligent behavior emerges.

Swarm intelligence is the branch of artificial intelligence that seeks to study and simulate the complex behaviors found in these swarms, such as flocks of birds or ant colonies. In this case, an algorithm based on the behavior of birds was implemented, applying it to complex time series prediction.

A time series can represent anything from the prices of an item, unemployment rates, the maximum daily temperature, and wind speed, to the efforts and temperatures at various points of an instrumented civil work, etc., since it is a sequence of data arranged chronologically and are used to study the causal relationship between various variables that change over time.

In this work, we use recurrent neural networks (RNNs) since these are one of the most used models. For example, in voice recognition systems or video analysis, or in natural language processing, and these models are these models capable of processing different types of sequences (such as videos, conversations, text) and, unlike neural or convolutional networks, this type of neural network are capable of generating new sequences, likes time series.

RNNs use the concept of recurrence: to generate the output, which we will call activation from now on. The network uses not only the current input, but also the activation generated in the previous iteration. In a nutshell, RNNs use a certain type of “memory” to generate the desired output [17].

Some models based on RNNs can be found in finance [4], electronic commerce [5, 6], capital markets [7, 8], macroeconomy [9, 10], health [11, 12], signal processing, meteorology [13, 14], voice recognition [15] and traffic control [16].

When birds are fed with flocks of birds, we can have more information and a good foraging effectiveness and survival benefits.

doi: 10.13053/CyS-28-2-5027

Table 1. Parameters of the BSA

Parameter	Value
M1	100
pop	100
Dime	60
FQ1	9
Co1	0.5
Co2	0.5
a1	2
a2	2

Table 2. Parameters of the PSO

Parameters	Value
P	100
I	100
C1	2
C2	2
C	0-0.9
W	0-0.9

Table 3. Parameters

Parameters ERRN	Value	
	Min	Max
NN1	1	5
NL1	1	3
NN1	1	30

Table 4. Euro/Mexican pesos time series results of the optimization

No.	NM	NL	Time	PE
1	2	16, 16 18, 17	01:38:47	0.01806
2	2	10 15	01:40:41	0.01842
3	2	24, 22 16, 23	01:35:11	0.019807
4	2	25, 16 1, 6	01:40:41	0.01802
5	2	25, 17 15, 16	01:15:42	0.01827
6	2	6, 9 28, 4	01:45:35	0.019806
7	2	6, 9 28, 4	01:35:35	0.01827
8	1	17	01:42:17	0.01822
9	2	25 15	01:15:42	0.01835
10	2	6.9, 22 26, 2, 26	01:26:15	0.01807

If one bird finds food, others may feed from it [19] While foraging, birds often unite due to the Predation threat [20]. They constantly scan their surroundings and raise their heads.

These behaviors, interpreted as vigilance behavior [21], may be conducive to detecting Predators [22]. Studies showed that birds would randomly choose between foraging and keeping vigilance [23]. Birds often give alarm calls when they detect a Predator [24]. Thus, the whole group would fly off together. Birds in the border of a group have higher probability of being attacked by Predators than those in the center.

Studies suggest that animals foraging in the flock center may move to their neighbors to protect themselves from being attacked [25]. The birds move towards the center of the flock as they perceive them. This motion, however, may be affected by the interference induced by competing bird swarms [26].

For this reason, the birds move toward the center of the swarm. This work outlines the formation of the ensemble recurrent neural network (ERNN). This model is applied to time series prediction [27, 32].

The architecture is optimized with the bird swarm algorithm, (BSA) [29-32], and Particle Swarm Optimization (PSO). The responses of the ERNN modules are integrated with type-1 and type-2 fuzzy systems (IT2FS), [32-38]. The optimization of the RNN consists finding the best the number of hidden layer (NL), number of neurons (NN) and the number of modules (NM).

We integrate responses in the ERNN, with type-1 and Fuzzy System Type-2 (IT2FS) and in this way we achieve Prediction. The Mamdani fuzzy inference system (FIS) has five inputs (Pr1, Pr2, Pr3, Pr4, and Pr5) and one output that is called Prediction.

The number of inputs of the fuzzy system (FS) is according to the outputs of ERNN and Mamdani fuzzy FIS is created. This FIS five inputs, which are Pr1, Pr2, Pr3, Pr4, and Pr5, have a from the range from 0 to 1.4.

The output is called Prediction, the range goes from 0 to 1.4 and is granulated into two membership functions (MFs) “Low”, “High” [50-53]. The main contribution of this work is the

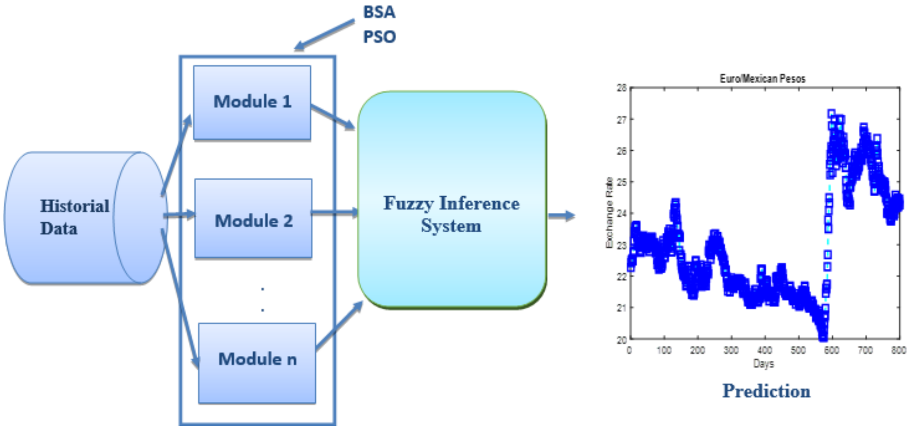

Fig. 1. General scheme

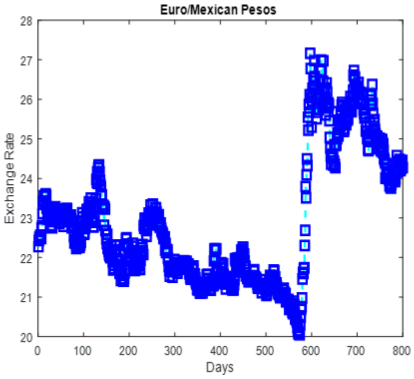

Fig. 2. Euro/Mexican pesos time series

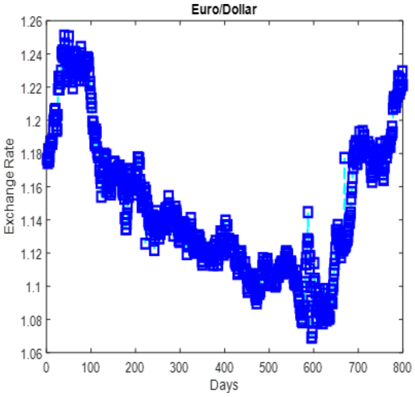

Fig. 3. Euro/Dollar time series

optimization of recurrent neural networks with the BSA algorithm, and PSO applied to time series, which represents a different way to optimize this kind of neural networks.

This paper is consisting of 4 parts: In section 2, the problem to be solved and the solution with the proposed method are mentioned, section 3 describes in the results obtained with the model and section 5 offers the conclusions.

2 Proposed Method

In this section, a recurrent neural network optimization model with the Bird Swarm algorithm (BSA) is offered. We have historical data that enters the ENN modules, and then responses from the ENN are integrated with type-1 and IT2FS for the Prediction of the time series.

In this case: Euro/Mexican pesos and Euro/Dollar time series. Figure 1 shows a general scheme of each of the steps of the method postulated in this paper which is explained in more detail below:

2.1 Parameters for BSA Applied to the RNN

Table 1 summarizes the parameters used for the optimization of ERNN with the BSA, where Dime is the number of dimensions used for optimization, M1 is the iteration number, pop is the bird population, FQ1 (FL) is the bird behavior frequency, a1 and a2 are values related to the vigilance behavior of the birds these effects are

If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Hh) If (P1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4L) and (Pr5 is P5eH) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Lo) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5L) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5H) then (Pr is Hh) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3H) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Lo) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is Pe5H) then (Pr is Lo) If (Pr1 is Pe1L) and (Pr2 is Pe2H) and (Pr3 is Pe3L) and (Pr4 is Pe4H) and (Pr5 is P5eH) then (Pr is Hh) If (Pr1 is Pe1H) and (Pr2 is Pe2L) and (Pr3 is Pe3H) and (Pr4 is Pe4L) and (Pr5 is Pe5L) then (Pr is Lo)

Fig. 4. IT2FS rules

Table 5. Results integration for the Euro/Mexican pesos

No.	Type-1 Fuzzy Integrator
1	0.3026
2	0.1408
3	0.1384
4	0.1122
5	0.3175
6	0.3022
7	0.1408
8	0.1383
9	0.1383
10	0.3021

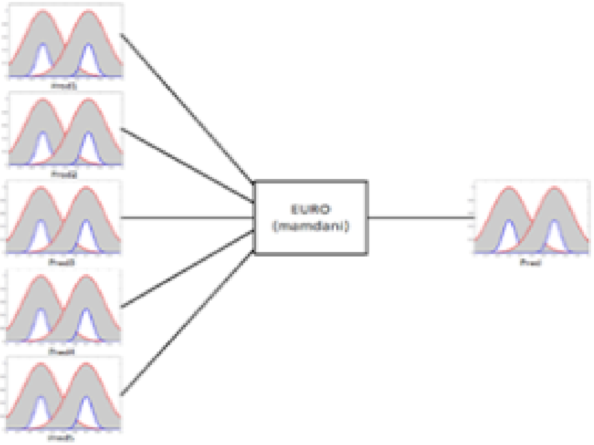

Fig. 5. IT2FS

indirect and direct. co1 cognitive is the coefficient and co2 is the accelerated social coefficient. Table 2 illustrated the parameters used for the optimization of ERNN with the PSO, where P is the number of particles, I is the iteration number, the cognitive component is C1, the Social Component is C2, Constriction coefficient of linear increase is Cand Inertia weight with linear decrease is (W).

2.2 Parameters for PSO Applied to and RNN

The following equation is used to minimize the prediction error, the objective function we

Table 6. Results for the IT2FS for the Euro/Mexican pesos

No.	PE 0.3	PE 0.4	PE 0.5
1	0.2219	0.2210	0.2286
2	0.2263	0.2268	0.2274
3	0.2215	0.2236	0.2339
4	0.2212	0.2222	0.2228
5	0.2229	0.2339	0.2348
6	0.2222	0.2225	0.2231
7	0.2221	0.2229	0.2235
8	0.2261	0.2268	0.2371
9	0.2256	0.2268	0.2387
10	0.2268	0.2274	0.2389

Table 7. Euro/Dollar pesos time series results of the optimization

No.	NM	NL	Time	PE
1	**3**	18 15 17	01:05:22	0.00042061
2	**3**	5, 18 15, 13 17, 18	01:19:33	0.00055028
3	**4**	17, 5, 7 12, 25, 13 20, 27, 22	01:28:04	0.00042102
4	**2**	10, 11 11, 12 15, 17 18, 16	01:45:20	0.0005802
5	**3**	18, 19, 17 13, 13.18	01:10:15	0.00050568
6	**3**	22, 19, 14 27, 17, 12 16, 17, 19	01:11:03	0.00051168
7	**3**	8, 22, 21 11, 23, 21 16, 6, 23	01:33:18	0.00052428
8	**3**	18, 19, 25 15, 22, 19 16, 11, 23	01:22:20	0.00528091
9	**4**	7, 7, 12 21, 2, 19 21, 13, 12	01:51:21	0.00505413
10	**4**	5, 27, 28 20, 20, 6 13, 12, 12 12, 26, 6	01:51:22	0.00042107

employed with the bird swarm algorithm and PSO:

$$\mathrm{ERMod} = \frac{1}{d}\sum_{i=1}^{d} |pd_i - rd_i|, \tag{1}$$

$$\text{PE} = \frac{1}{T}(\text{ERMod}_1 + \text{ERMod}_2 + \ldots + \text{ERMod}_N), \tag{2}$$

where PE is Prediction Error, d the data number, pd corresponds to the predicted data for each of the 5 modules, rd represents the real data, ERMod is the Module Prediction error, T represent module number determined by the BSA and the Predicion Error is the average prediction error for the modules. Table 2 represents the search space parameters.

2.3 Data Base

Fig. 2 presents the plot of the data Euro/Mexican pesos [18], where we use 800 data that belong to the time of 12/08/2017 to 31/12/2020. We apply 30% to test the RNN and 70% for the training of the RNN. Fig. 3 exhibits the plot of the data Euro/Dollar [18], where we use 800 data that belong to the time 12/08/2017 to 31/12/2020. We apply 30% to test the RNN and 70% for the training of the RNN.

2.4 Description of the Architecture of Type-1 and IT2FS

The following equation shows how the Fuzzy System is calculated:

$$y = \frac{\sum_{i=1 d_i}^{n} m\left(d_i\right)}{\sum_{i=1}^{n} m\left(d_i\right)_i}, \tag{3}$$

where m corresponds MFs and represents the input data.

Table 8. Results of the integration for Euro/Dollar

No.	Type-1 fuzzy integrator
1	0.1225
2	0.1228
3	0.1432
4	0.2223
5	0.225
6	0.2822
7	0.1736
8	0.3220
9	0.2217
10	0.2325

Table 9. Euro/Mexican pesos time series results of the PSO

No	Modules	Layers	Time	P
1	3	16,18 20,22 15,17	01:07:17	0.01922
2	2	22,23 24.25	01:02:23	0.02202
3	3	24,24 23,21 19.20	01:05:11	0.01988
4	2	27,21 24,23	01:07:22	0.02107
5	4	27,15 20,21 25,24 23,22	01:07:25	0.01977
6	2	26,29 28,27	01:07:32	0.02106
7	3	6,9 28,4	01:35:35	0.01827
8	2	20,23 21,22	01:07:17	0.01738
9	2	25,23 21,22	01:07:12	0.01732
10	3	6,9,22 26,2,26 22,21,10	01:06:13	

2.5 Type-2 Memberships Functions

Below are equations of type-2 fuzzy system membership functions:

$$\widetilde{u}(x) = [\mu(x), \widetilde{u}(x)] = \mathrm{gausstype2}(x, [\sigma_x m_1, m_2]), \quad (4)$$

where “igaussstype2” stands for the Gaussian generalized type-2 membership function with uncertain mean:

$$m_x = \frac{m_1 + m_2}{2}, \quad (5)$$

$$p_x = \mathrm{gaussmf}(x,\, [\sigma_x m_1,\, m_2]), \quad (6)$$

$$\Longrightarrow\ p_x = \exp\left[\frac{1}{2}\left(\frac{x - m_x}{\sigma_x}\right)^2\right], \quad (7)$$

$$p_x = \mathrm{gaussmf}(x,\, [\sigma_x,\, m_x]), \quad (8)$$

$$\widetilde{u}(x,\, \mu) = \mathrm{gaussmf}(\mu,\, [\sigma_x,\, p_x]), \quad (9)$$

$$\Longrightarrow\ \widetilde{u}(x, \mu) = \exp\left[\frac{1}{2}\left(\frac{x - p_x}{\sigma_u}\right)^2\right], \quad (10)$$

where $p_x =$ is of the center of the function and σ_u is the implication of the function. MF too high:

$$\mathrm{MFToohigh} = e^{-\frac{1}{2}}\left(\frac{x - 234}{25.2}\right)^2, \quad (11)$$

where $p_x = 234$ and $\sigma_u = 25.2$, where $p_x = 234$ and $\sigma_u = 25.2$. The fuzzy system that was used has five inputs that are Pr1, Pr2, Pr3, Pr4, and Pr5, and an output called Pr, where the parameters have a range from 0 to 1.4 and the variables are denominated “Low” and “High”, in Figure 5 as present. The Figure 4 lists the rules used in the type-1 and IT2FL.

The rules were obtained based on previous work used in the fuzzy system [36]. The Figure 4 represents the rules employed in the fuzzy system. The rules were formulated based on previous work used in the fuzzy system where 32 rules were used and they are all possible rules since we have 5 inputs and one output, and each input has two MFs and the consequents are determined by trial and error [36].

Table 10. Results for the IT2FS for the Euro/Dollar

No.	PE 0.3	PE 0.4	PE 0.5
1	0.3827	0.3849	0.3864
2	0.4162	0.4165	0.4168
3	0.4168	0.4236	0.4238
4	0.4229	0.4239	0.4249
5	0.4422	0.4436	0.4443
6	0.4987	0.4222	0.4229
7	0.4668	0.4544	0.4277
8	0.4492	0.4898	0.4899
9	0.4449	0.4453	0.4456
10	0.4966	0.4988	0.4994

Table 11. Results integration for the Euro/Mexican pesos for PSO

No.	Type-1 fuzzy integrator
1	0.3228
2	0.3417
3	0.3263
4	0.3123
5	0.3442
6	0.3227
7	0.3409
8	0.3386
9	0.3323
10	0.3122

Table 12. Results for the IT2FS for the Euro/Mexicans pesos for PSO

No.	PE 0.3	PE 0.4	PE 0.5
1	0.2330	0.2332	0.2377
2	0.2383	0.2385	0.2366
3	0.2318	0.2338	0.2380
4	0.2333	0.2346	0.2348
5	0.2388	0.2389	0.2391
6	0.2382	0.2385	0.2381
7	0.2371	0.2373	0.2376
8	0.2381	0.2384	0.2387
9	0.2390	0.2392	0.2396
10	0.2377	0.2379	0.2381

3 Results

In this section, the results of the experiments of the presented model are outlined. In this case, the bird algorithm and particle swarm optimization were used for the optimization of the ERNN, and the type-1 and IT2FS integration and 10 experiments were carried out and the time series of the Euro/Mexican pesos and Euro/Dollar pesos were considered. Table 4 represents the Euro/Mexican pesos time series results of the Optimization of ERNN for 10 experiments, where 100 generations and 100 population is used.

Table 5 shows the Euro/Mexican pesos time series results of 10 type-1 fuzzy system experiments. Table 6 represents the Euro/Mexican pesos time series results of the IT2FLS for 10 experiments with different levels of uncertainty.

Table 7 represents Euro/Dollar time series results of the Optimization of ERNN for 10 experiments, where 100 generations and 100 population is used. Table 8 shows the Euro/Dollar time series results of 10 type-1 fuzzy system experiments. Table 9 represents the Euro/Mexican pesos time series results of the PSO of ERNN 10 experiments, where 100 particles and 100 iterations is used. Table 10 represents the Euro/Dollar time series results of 10 IT2FS experiments with different uncertainty levels.

Table 11 shows the Euro/Mexican pesos time series results for 10 type-1 fuzzy system experiments for PSO. Table 12 represents the Euro/Mexican pesos time series results of the IT2FLS 10 experiment for PSO with different levels of uncertainty.

Table 13 shows the Euro/Dollar time series results of 10 type-1 fuzzy system experiments for PSO. Table 14 represents the Euro/Dollar time series results of 10 IT2FS experiments for PSO with different levels of uncertainty.

To compare the two optimization algorithms, Bird Sarm Algorithm and Particle Swarm, for Euro/Mexican pesos time series, the t statistical test was used and we can conclude that there is significant evidence between the optimization of the NN with the BSA and PSO. In other words, it can be said that the BSA algorithm is better.

Table 13. Euro/Dollar pesos time series results of the PSO

No.	Mod.	Layers	Time	P
1	2	15,17 15,16	01:08:35	0.00052058
2	2	17,15 15,15	01:08:33	0.00051022
3	3	16,15,15 13,15,16 15,17,12	01:08:31	0.00052323
4	3	21,20,19 22,18,20 21,20,18 20,22,22	01:08:27	0.00062023
5	2	15,18,19 16,15,17	01:08:26	0.00056032
6	4	21,20,19 22,18,20 21,20,18 20,22,22	01:09:03	0.00051168
7	3	21,22, 15,16 16,17	01:08:55	0.00053328
8	4	20,19 21,22	01:08:28	0.00628082
9	2	21,20 22,18 21,20 20,22	01:09:01	0.00605332
10	3	2,27 20,21 17,20	01:09:09	0.00661133

Table 14. Results of the integration for Euro/Dollar for PSO

No.	PE 0.3	PE 0.4	PE 0.5
1	0.4327	0.4449	0.4451
2	0.4522	0.4523	0.4528
3	0.4238	0.4242	0.4244
4	0.4266	0.4268	0.4270
5	0.4292	0.4298	0.4301
6	0.4287	0.4289	0.4290
7	0.4266	0.4268	0.4270
8	0.4292	0.4294	0.4296
9	0.4275	0.4277	0.4279
10	0.4223	0.4226	0.4229

Table 15. t-Statistical test for Euro/Mexican pesos

Algorithm	N	Mean	Standard Deviation	Mean Error
BSA	29	0.018518	0.00635	0.00012
PSO	29	0.01955	0.00154	0.00029

Table 16. t-Statistical test for Euro/Dollar

Algorithm	N	Mean	Standard deviation	Mean Error
BSA	29	0.00355	0.00927	0.0017
PSO	29	0.00261	0.00261	0.00049

The number of samples was 29, and a 95% confidence interval was used. and produce a P value of 0.002 and T value of -3.34 and Critical Value of 0.0668. To compare the two optimization algorithms of BSA and PSO for Euro/Dollar Peso time series, t-statistic test was used and we can conclude that there is no significant evidence between the optimization of with the BSA and PSO algorithm. In other words, it can be said that the BSA algorithm and the PSO are equal since the results are almost the same, the samples used were 29 and a 95% confidence interval was used.

4 Conclusions

Recurrent neural networks are an excellent model for time series data. In this paper we, so we conclude that this type of network produces good results, as well as the optimization of the structure of this network, and then the fuzzy integration of the network responses to test this method was used Euro/Dollar and Euro/Mexican pesos time series. When performing statistical tests, we had the goal finding out it both algorithms are good at solving time series problems.

In the Euro/Mexican time series the BSA algorithm was better and for the Euro/Dollar time series, both were goodComparisons were made with the two optimization algorithms Bird Swarm Algorithm and Particle Swarm, for Euro/Mexican pesos time series, the statistical t test was used and we can conclude that there is significant evidence between the optimization with the BSA and PSO.

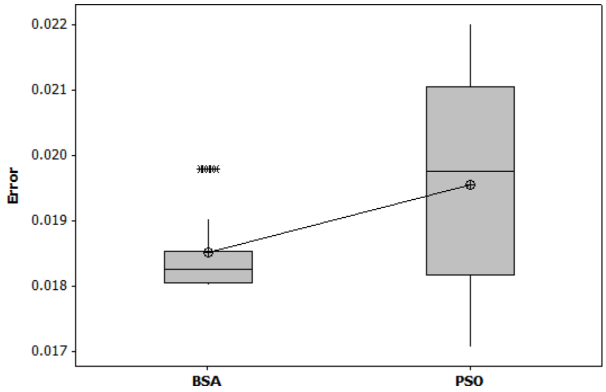

Fig. 6. Box plot for Euro/Mexican pesos time series

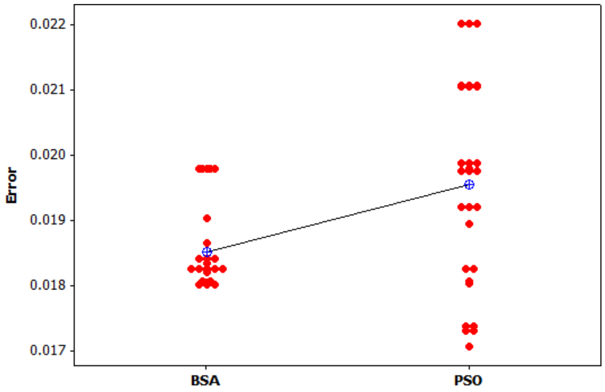

Fig. 7. Graph of values for Euro/Mexican pesos time series

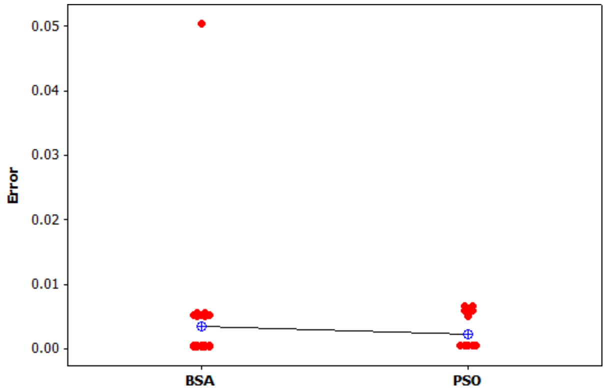

Fig. 8. Graph of values for Euro/Dollar time series

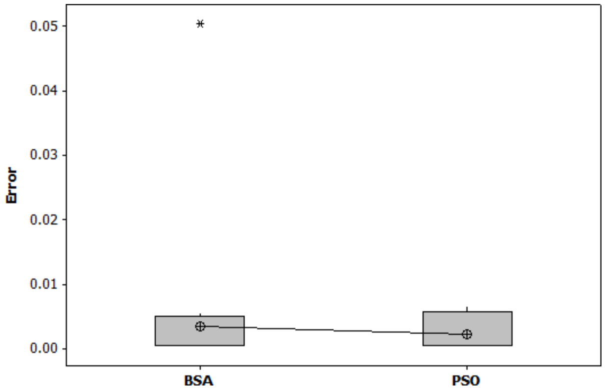

Fig. 9. Graph of values for Euro/Mexican pesos time series

In other words, it can be said that the BSA algorithm is better, and for Euro/Dollar Peso time series, the t-statistic test was also used and we can conclude that there is no significant evidence between the optimization with the BSA algorithm and PSO. In other words, it can be said that the BSA algorithm and the PSO are the same. As future work, we plan to consider other similar metaheuristics (instead of BSA and PSO), like in [53-57], and also undertake other application areas [58-61].

Acknowledgments

We would like to express our gratitude to CONAHCYT and Tijuana Institute of Technology for the resources granted for this research.

References

1. **Anderson, T. R. (2002).** Biology of the ubiquitous house parrow: From genes to Populations. Oxford University Press.

2. **Ahmad, M., Javaid, N., Niaz, I. A., Shafiq, S., Rehman, O. U., Hussain, H. M. (2018).** Application of bird swarm algorithm for solution of optimal power flow problems. Complex, Intelligent, and Software Intensive Systems, pp. 280–291. DOI: 10.1007/978-3-319-936 59-8_25.

3. **Gad, A. G. (2022).** Particle swarm optimization algorithm and its applications: A systematic review. Archives of Computational Methods in Engineering, Vol. 29, No. 5, pp. 2531–2561. DOI: 10.1007/s11831-021-09694-4.

4. **Beauchamp, G. (2003).** Group-size effects on vigilance: A search for mechanisms. Behavioural Processes, Vol. 63, No. 3, pp. 111–121. DOI: 10.1016/s0376-6357(03) 00002-0.

5. **Bednekoff, P. A., Lima, S. L. (1998).** Randomness, chaos and confusion in the study of antipredator vigilance. Trends in Ecology and Evolution, Vol. 13, No. 7, pp. 284–287. DOI: 10.1016/s0169-5347(98) 01327-5.

6. **Brockwell, P. J., Davis R. A. (2002).** Introduction to time series and forecasting, Springer Texts in Statistics. DOI: 10.1007/0-3 87-21657-X_8.

7. **Cai, L., Zhang, Y., Ji, W. (2018).** Variable strength combinatorial test data generation using enhanced bird swarm algorithm. Proceedings of the 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 391–398. DOI: 10.1109/snpd.2018.8441104.

8. **Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y. (2018).** Recurrent neural networks for multivariate time series with missing values. Scientific Reports, Vol. 8, No. 1, pp. 6085. DOI: 10.1038/s41598-018-24271-9.

9. **Correa, P. P., Cipriano, A., Nunez, F., Salas, J. C., Lobel, H. (2021).** Forecasting copper electrorefining cathode rejection by means of recurrent neural networks with attention mechanism. IEEE Access, Vol. 9, pp. 79080–79088. DOI: 10.1109/access.202 1.3074780.

10. **Castillo, O., Melin, P. (2007).** Comparison of hybrid intelligent systems, neural networks and interval type-2 fuzzy logic for time series prediction. Proceedings of the International Joint Conference on Neural Networks, pp. 3086–3091. DOI: 10.1109/ijcnn.2007.437 1453.

11. **Castillo, O., Melin, P. (2002).** Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1395–1408. DOI: 10.1109/ tnn.2002.804316.

12. **Castro, J. R., Castillo, O., Melin, P., Mendoza, O., Rodríguez-Díaz, A. (2010).** An interval type-2 fuzzy neural network for chaotic time series prediction with cross-validation and akaike test. Soft Computing for Intelligent Control and Mobile Robotics, pp. 269–285. DOI: 10.1007/978-3-642-15534-5_17.

13. **Castillo, O. (2012).** Type-2 fuzzy logic in intelligent control applications, pp.1–188. DOI: 10.1007/978-3-642-24663-0.

14. **Castillo O., Melin P. (2001).** Simulation and forecasting complex economic time series using neural networks and fuzzy logic. International Joint Conference on Neural Networks, Vol. 3, pp. 1805–1810. DOI: 10.1 109/IJCNN.2001.938436.

15. **Castillo O., Melin P. (2008).** Type-2 fuzzy systems. Type-2 Fuzzy logic Theory and Application, pp. 30–43. DOI: 10.1007/978-3 -540-76284-3.

16. **Cowpertwait, P. S., Metcalfe, A. V. (2009).** Introductory time series with R. Springer Science and Business Media, pp. 2–5. DOI: 10 .1007/978-0-387-88698-5.

17. **Davey, N., Hunt, S., Frank, R. (1999).** Time series prediction and neural networks. University of Hertfordshire, Hatfield.

18. **Euro Database (2021).** Futuros petróleo crudo WTI - Jul 2024 (CLN4). mx.investing .com/commodities/crude-oil.

19. **Escudero, P., Alcocer, W., Paredes, J. (2021).** Recurrent neural networks and arima models for euro/dollar exchange rate forecasting. Applied Sciences, Vol. 11, No. 12, pp. 5658. DOI: 10.3390/app11125658.

20. **Fathi, M., Kashani, M. H., Jameii, S. M., Mahdipour, E. (2021).** Big data analytics in

weather forecasting: a systematic review. Archives of Computational Methods in Engineering, Vol. 29, No. 2, pp. 1247–1275. DOI: 10.1007/s11831-021-09616-4.

21. **Nakip, M., Guzelis, C., Yildiz, O. (2021).** Recurrent trend predictive neural network for multi-sensor fire detection. IEEE Access, Vol. 9, pp. 84204–84216. DOI: 10.1109/access.2 021.3087736.

22. **Simões-Hoffmann, L. F., Parquet-Bizarria, F. C., Parquet-Bizarria, J. W. (2020).** Detection of liner surface defects in solid rocket motors using multilayer perceptron neural networks. Polymer Testing, Vol. 88, pp. 106559. DOI: 10.1016/j.polymertesting .2020.106559.

23. **Ismail, F. H., Houssein, E. H., Hassanien, A. E. (2018).** Chaotic bird swarm optimization algorithm. Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2018, pp. 294–303. DOI: 10.1 007/978-3-319-99010-1_27.

24. **Kennedy, J., Eberhard, R. (1975).** Particle swarm optimization. Proceedings of ICCN95 International Conference on Neuronal Networks, Vol. 4, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

25. **Connor, J., Martin, R., Atlas, L. (1994).** Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 240–254. DOI: 10 .1109/72.279188.

26. **Karnik, N. N., Mendel, J. M. (1998).** Introduction to type-2 fuzzy logic systems. IEEE International Conference on Fuzzy Systems Proceedings, Vol. 2. pp. 915–920. DOI: 10.1109/FUZZY.1998.686240.

27. **Kennedy, J., Eberhart, R. C., Shi, Y. H. (2001).** Swarm intelligence Algorithm Burlington, MK: Morgan Kaufmann Publishers.

28. **Koops, M. A., Giraldeau, L. A. (1996).** Producer–scrounger foraging games in starlings: A test of rate-maximizing and risk-sensitive models. Animal Behaviour, Vol. 51, No. 4, pp. 773–783. DOI: 10.1006/anbe.1996. 0082.

29. **Krausz, R. R. (2013).** Living in groups. Transactional Analysis Journal, Vol. 43, No. 4, pp. 366–374. DOI: 10.1177/ 0362153713519414.

30. **Lima, S. L., Dill, L. M. (1990).** Behavioral decisions made under the risk of predation: a review and prospectus. Canadian Journal of Zoology, Vol. 68, No. 4, pp. 619–640. DOI: 10 .1139/z90-092.

31. **Tan, W. W., Chua, T. W. (2007).** Uncertain rule-based fuzzy logic systems: Introduction and new directions. IEEE Computational Intelligence Magazine, Vol. 2, No. 1, pp. 72–73. DOI: 10.1109/mci.2007.357196.

32. **Meng, X. B., Gao, X. Z., Lu, L., Liu, Y., Zhang, H. (2016).** A new bio-inspired optimisation algorithm: Bird swarm algorithm. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 28, No. 4, pp. 673–687. DOI: 10.1080/0952813x.2015.1042530.

33. **Han, M., Xi, J., Xu, S., Yin, F. L. (2004).** Prediction of chaotic time series based on the recurrent predictor neural network. IEEE Transactions on Signal Processing, Vol. 52, No. 12, pp. 3409–3416. DOI: 10.1109/tsp.20 04.837418.

34. **Tang, P., Zhang, Q., Zhang, X. (2023).** A recurrent neural network based generative adversarial network for long multivariate time series forecasting. Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, pp. 181–189. DOI: 10.1145/359110 6.3592306.

35. **Pulido, M., Melin, P., Mendoza, O. (2018).** Optimization of ensemble neural networks with type-1 and interval type-2 fuzzy integration for forecasting the Taiwan stock exchange. In: Gaweda, A., Kacprzyk, J., Rutkowski, L., Yen, G. (eds) Advances in Data Analysis with Computational Intelligence Methods. Studies

in Computational Intelligence, Springer, Cham, Vol 738. DOI: 10.1007/978-3-319-67946-4_6.

36. **Pulliam, H. R., Pyke, G. H., Caraco, T. (1982).** The scanning behavior of juncos: A game-theoretical approach. Journal of Theoretical Biology, Vol. 95, No. 1, pp. 89–103. DOI: 10.1016/0022-5193(82)90289-2.

37. **Pulliam, H. R (1973).** On the advantages of flocking. Journal of Theoretical Biology, Vol. 38, No. 2, pp. 419–422. DOI: 10.1016/0022-5193(73)90184-7.

38. **Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S. (2011).** Extensions of recurrent neural network language model. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5528–5531. DOI: 10.1109/icassp.2011.5947611.

39. **Petneházi, G. (2019).** Recurrent neural networks for time series forecasting. DOI: 10.48550/ARXIV.1901.00069.

40. **Chandra, R., Zhang, M. (2012).** Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction. Neurocomputing, Vol. 86, pp. 116–123. DOI: 10.1016/j.neucom.2012.01.014.

41. **Piccialli, F., Giampaolo, F., Prezioso, E., Camacho, D., Acampora, G. (2021).** Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion. Information Fusion, Vol. 74, pp. 1–16. DOI: 10.1016/j.inffus.2021.03.004.

42. **Pulido, M., Melin, P. (2021).** Ensemble recurrent neural networks for complex time series prediction with integration methods. In: Castillo, O., Melin, P. (eds) Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications. Studies in Computational Intelligence, Springer, Cham. Vol. 940, pp. 71–83. DOI: 10.1007/978-3-030-68776-2_4.

43. **Pulido, M., Melin, P. (2022).** Ensemble recurrent neural network using genetic algorithm applied in times series prediction. Computación y Sistemas, Vol. 26, No. 2, pp. 683–700. DOI: 10.13053/cys-26-2-4251.

44. **D'Urso, P., Giovanni, L., Massari, R. (2019).** Trimmed fuzzy clustering of financial time series based on dynamic time warping. Annals of Operations Research, Vol. 299, No. 1, pp. 1379–1395. DOI: 10.1007/s10479-019-03284-1.

45. **Sherstinsky, A. (2020).** Fundamentals of recurrent neural network (RNN) and long short term memory (LSTM) network. Physica D: Nonlinear Phenomena, Vol. 404, pp. 132306. DOI: 10.1016/j.physd.2019.132306.

46. **Saha, S., Raghava, G. P. S. (2006).** Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. Proteins: Structure, Function, and Bioinformatics, Vol. 65, No. 1, pp. 40–48. DOI: 10.1002/prot.21078.

47. **Walid, A., Alamsyah, I. U. (2017).** Recurrent neural network for forecasting time series with long memory pattern. Journal of Physics: Conference Series, Vol. 824, No. 1, pp. 012038. DOI: 10.1088/1742-6596/824/1/012038.

48. **Wang, X., Wang, Y., Peng, J., Zhang, Z., Tang, X. (2022).** A hybrid framework for multivariate long-sequence time series forecasting. Applied Intelligence, Vol. 53, No. 11, pp. 13549–13568. DOI: 10.1007/s10489-022-04110-1.

49. **Wang, Y., Fang, S., Zhang, C., Xiang, S., Pan, C. (2022).** TVGCN: Time-variant graph convolutional network for traffic forecasting. Neurocomputing, Vol. 471, pp. 118–129. DOI: 10.1016/j.neucom.2021.11.006.

50. **Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G. (2017).** A dual-stage attention-based recurrent neural network for time series prediction. DOI: 10.48550/arXiv.1704.02971.

51. **Zhang, J., Man, K. F. (1998).** Time series prediction using recurrent neural network in multidimension embedding phase space. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp. 1868–1873. DOI: 10.1109/ICSMC.1998.728168.

52. **Zhang, J. S., Xiao, X. C. (2000).** Predicting chaotic time series using recurrent neural network. Chinese Physics Letters, Vol. 17, No. 2, pp. 88–90. DOI: 10.1088/0256-307x/17/2/004.

53. **Amador-Angulo, L., Ochoa, P., Peraza, C., Castillo, O. (2023).** Fuzzy dynamic adaptation of an artificial fish swarm algorithm for the optimization of benchmark functions. In: Castillo, O., Melin, P. (eds) Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics. Studies in Computational Intelligence, Springer, Cham, Vol. 1096, pp. 99–114. DOI: 10.1007/978-3-031-28999-6_6.

54. **Castillo, O., Lizárraga, E., Soria, J., Melin, P., Valdez, F. (2015).** New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system. Information Sciences, Vol. 294, pp. 203–215. DOI: 10.1016/j.ins.2014.09.040.

55. **Amador-Angulo, L., Mendoza, O., Castro, J., Rodríguez-Díaz, A., Melin, P., Castillo, O. (2016).** Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. Sensors, Vol. 16, No. 9, pp. 1458. DOI: 10.3390/s16091458.

56. **Sánchez, D., Melin, P., Castillo, O. (2017).** A grey wolf optimizer for modular granular neural networks for human recognition. Computational Intelligence and Neuroscience, Vol. 2017, pp. 1–26. DOI: 10.1155/2017/4180510.

57. **González, B., Valdez, F., Melin, P., Prado-Arechiga, G. (2015).** Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition. Expert Systems with Applications, Vol. 42, No. 14, pp. 5839–5847. DOI: 10.1016/j.eswa.2015.03.034.

58. **Tai, K., El-Sayed, A. R., Biglarbegian, M., Gonzalez, C., Castillo, O., Mahmud, S. (2016).** Review of recent type-2 fuzzy controller applications. Algorithms, Vol. 9, No. 2, pp. 39. DOI: 10.3390/a9020039.

59. **Ontiveros, E., Melin, P., Castillo, O. (2020).** Comparative study of interval type-2 and general type-2 fuzzy systems in medical diagnosis. Information Sciences, Vol. 525, pp. 37–53. DOI: 10.1016/j.ins.2020.03.059.

60. **Melin, P., Castillo, O. (2004).** A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks. International Journal of General Systems, Vol. 33, No. 2–3, pp. 289–304. DOI: 10.1080/03081070310001633608.

61. **Moreno, J. E., Sanchez, M. A., Mendoza, O., Rodríguez-Díaz, A., Castillo, O., Melin, P., Castro, J. R. (2020).** Design of an interval type-2 fuzzy model with justifiable uncertainty. Information Sciences, Vol. 513, pp. 206–221. DOI: 10.1016/j.ins.2019.10.042.

Article received on 26/12/2023; accepted on 07/05/2024.
**Corresponding author is Patricia Melin.*