Mackey-Glass noisy chaotic time series prediction by a swarm-optimized neural network

Journal of Physics: Conference Series **720** (2016) 012002 doi:10.1088/1742-6596/720/1/012002

Mackey–Glass noisy chaotic time series prediction by a swarm-optimized neural network

C H López-Caraballo, I Salfate, J A Lazzús, P Rojas, M Rivera and L Palma-Chilla

Departamento de Física y Astronomía, Universidad de La Serena, Avda. J. Cisternas 1200, Casilla 554, La Serena, Chile.

E-mail: clopez@dfuls.cl,jlazzus@dfuls.cl

Abstract. In this study, an artificial neural network (ANN) based on particle swarm optimization (PSO) was developed for the time series prediction. The hybrid ANN+PSO algorithm was applied on Mackey-Glass noiseless chaotic time series in the short-term and long-term prediction. The performance prediction is evaluated and compared with similar work in the literature, particularly for the long–term forecast. Also, we present properties of the dynamical system via the study of chaotic behaviour obtained from the time series prediction.

Then, this *standard* hybrid ANN+PSO algorithm was complemented with a Gaussian stochastic procedure (called *stochastic* hybrid ANN+PSO) in order to obtain a new estimator of the predictions that also allowed us compute uncertainties of predictions for noisy Mackey-Glass chaotic time series. We study the impact of noise for three cases with a white noise level (σ_N) contribution of 0.01, 0.05 and 0.1.

1. Introduction

The prediction of times series (TS) has played an important role in many science fields of practical application as engineering, biology, physics, meteorology, etc. In particular, and due to their dynamical properties, the analysis and prediction of chaotic time series have been of interest for the science community. In the literature, we found many methods focused on the prediction of chaotic time series, for example, those based on artificial neural network (ANN) models as the back-propagation algorithm [1], radial basic function [2], recurrent network [3], genetic algorithms [4], fuzzy system application [5], and wavelet approach [6], etc. These systems are usually modeled by delay-differential equations; standard examples are the Mackey–Glass equation [7], the Ikeda equation [8], and equation for an electronic oscillator with delayed feedback [9] (for more examples see [10]). The main goal is to forecast future values of a series known (or observed) up to a specific time ($x_{(t+T)}$, with T the ahead prediction) using past values. Note that the non-linear dependence level of a series on previous data points is of interest, particularly, because of the possibility of a chaotic behaviour of the input series.

Up to now, several short-term prediction methods have proven to be efficient in the forecast of chaotic time series and therefore in the characterization of dynamical system (for example, see [5, 11, 12, 6, 13] and references therein). In contrast, the long-term prediction still requires methods to improve the forecast effectiveness. Furthermore, in the literature the long-term forecast has not been widely studied.

On the other hand, the analysis of real-life times series require of consider the error propagation of input uncertainties. The observed data could be contaminated for different instrumental noise types as white noise or proportional to signal (the latter mainly arises from instrumental calibration). In modeling of chaotic times series, the prediction can be seen as erros-in-variable problem where the noisy is propagated into the prediction model. In the literature, the noisy impact on chaotic time series prediction have been scarce considered. We can found studies where the algorithms were tested from a theoretical point of view (for example, see [14, 15, 16, 17, 18]), and works where the implementation was applied of real-life time series (for example, see [19, 20, 15]). In the latter two references the authors proposed modification to the methods in order to improve the performance prediction in presence of noise.

In this work, we used a Mackey–Glass chaotic time series in order to study the short–term and long–term prediction with a hybrid ANN+PSO algorithm. Our performance prediction (of this *noiseless* cases) is compared with similar implementation available in the literature, in particular for the long–term forecast. Then, this *standard* hybrid ANN+PSO algorithm was complemented with a Gaussian stochastic procedure (called *stochastic* hybrid ANN+PSO) in order to obtain a new estimator of the predictions that also allowed us compute uncertainties of predictions for noisy Mackey-Glass chaotic time series. We study the impact of noise for three cases with a white noise level (σ_N) contribution of 0.01, 0.05 and 0.1.

2. Hybrid algorithm

A feed-forward neural network was used to represent non-linear relationships among variables [1]. This ANN was implemented replacing standard back-propagation algorithm with particle swarm optimization (PSO). PSO is a population-based optimization tool, whose operation is based on these two equations [21, 22]:

$$v_j^{k+1} = \omega v_j^k + c_1 r_1 \left(\psi_j^k - s_j^k\right) + c_2 r_2 \left(\psi_g^k - s_j^k\right) \tag{1}$$

$$s_j^{k+1} = s_j^k + v_j^{k+1} \tag{2}$$

where s and v denote a particle position and its corresponding velocity in a search space, respectively. In general, the value of each component in v can be clamped to the range $[-v_{\max}, +v_{\max}]$ control excessive roaming of particles outside the search space [21, 22].

On the other hand, the procedure to calculating the output values, using the input values, are described in detail in [23]. The net inputs (N) are calculated for the hidden neurons coming from the inputs neurons. In the case of a neuron in the hidden layer is had:

$$N_i^h = \sum_i^n w_{i,j}^h p_i + b_{i,j}^h \tag{3}$$

where p_i is the vector of the inputs of the training, $w_{i,j}^h$ is the weight of the connection among the input neurons with the hidden layer h, and the term $b_{i,j}^h$ corresponds to the bias of the neuron of the hidden layer h, reached in its activation. The PSO algorithm is very different then any of the traditional methods of training [21]. Each neuron contains a position and velocity. The position corresponds to the weight of a neuron $\left(s_i^k \to w_{i,j}^h\right)$. The velocity is used to update the weight $\left(v_i^{k+1} \to w'_{i,j}\right)$. Starting from these inputs, the outputs (y_i) of the hidden neurons are calculated, using a transfer function f^h associated with the neurons of this layer. In the ANN, the hyperbolic tangent function ($tansig$) was used [24].

$$y_i = f^h \left(\sum_i^n w_{i,j}^h p_i + b_{i,j}^h \right) \tag{4}$$

Table 1. Adjustment parameters in the hybrid ANN+PSO algorithm.

ANN		PSO	
NN-type	feed-forward	Number of particles in swarm (N_{part})	50
Number of hidden layers	1	Number of iterations ($k_{\max}$)	1500
Transfer function	*tansig*	Cognitive component (c_1)	1.494
Number of iterations	1500	Social component (c_2)	1.494
Normalization range	[–1, 1]	Maximum velocity ($v_{\max}$)	12
Weight range	[–100, 100]	Minimum inertia weight ($\omega_{\min}$)	0.5
Bias range	[–10, 10]	Maximum inertia weight ($\omega_{\max}$)	0.7
Minimun error	1e–3	Objective function	RMSE

After finding the output values, the weights of all layers of the network are actualized $w_{i,j} \rightarrow w'_{i,j}$ by PSO, using equation 1 and 2 [22]. On each step, PSO compares each weight using the data set [21]. In this article, we used the mean square error (MSE) to determine network fitness for the entire training set. This process was repeated for the total number of patterns in the training set. For a successful process the objective of the algorithm is to modernize all the weights minimizing the total root mean squared error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(Y_i^{\text{true}} - Y_i^{\text{calc}}\right)^2}{n}}. \tag{5}$$

In PSO, the inertial weight ω, the constant c_1 and c_2, the number of particles N_{part} and the maximum speed of particle summary the parameters to syntonize for their application in a given problem. Table 1 shows the selected parameters for this hybrid algorithm.

3. Simulations of Mackey–Glass system

We computed the Mackey–Glass time series from the Mackey–Glass time-delay differential system [7, 25], which is described as follow:

$$\frac{dx}{dt} = \beta x(t) + \frac{\alpha x(t-\tau)}{1 + x(t-\tau)^{10}} \tag{6}$$

where x (unitless) is the series in the time t, and τ the time delay. Here, we assumed $\alpha = 0.2$, $\beta = 0.1$, $\tau = 17$ and $x(0) = 1.2$. Note that, if $\tau \geq 17$ the time series show a chaotic behaviour [4, 25]. Finally, the nominal Mackey–Glass chaotic time series was obtained from numerical integration via the method fourth order Runga Kutta. The series was computed with a time sampling of 1 second. Thus, $x(t)$ is derived for $0 \leq t \leq t_{\text{h}}$ with $x(t) = 0$ for $t < 0$, where t_{h} is the time horizon considered.

The algorithm implementation. We create as input a vector from d points of the time series spaced Δ apart, i.e. $\mathbf{x}(t) = [x(t), x(t+\Delta), \cdots, x(t+(d-1)\Delta)]$, in order to predict the future $x(t+T)$ value. In the standard Mackey–Glass analysis, we consider four non consecutive points in the time series in order to generate each input vector to predict the $x(t+6)$ and $x(t+84)$ as:

$$x(t+6) = F\left[x(t), x(t-6), x(t-12), x(t-18)\right] \tag{7}$$

$$x(t+84) = F\left[x(t), x(t-6), x(t-12), x(t-18)\right] \tag{8}$$

where this standard test assume $d = 4$ and $\Delta = T = 6$ [2, 4]. As reference, in general we use the 80% and 20% of data for the training and prediction, respectively.

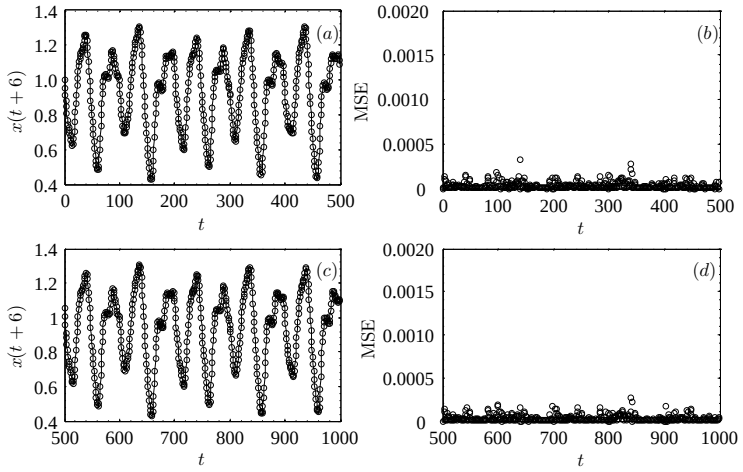

Figure 1. Training set and prediction set for the short-term $x(t+6)$.

Table 2. RRMSE for the computational methods found in the literature and our case studies.

Method	$\text{RMSE}_{x(t+6)}$	$\text{RMSE}_{x(t+84)}$
Linear model	0.5503	1.504
Cascade correlation NN	0.0624	0.170
Six order polynomial	0.0402	0.085
Back-propagation NN	0.0262	0.060
RBFNN	0.0114	0.097
This work *Case 1*	0.0053	0.038

4. Noiseless chaotic time series prediction

For the noiseless we generate one thousand data points ($t_{\text{h}} = 1000$), where the first five hundred were used for training while the others were used for testing.

Short-term prediction. The most basic architecture of a a neural network normally consisting of three or four layers [1, 26]. The input layer contains one neuron for each input parameter. The output layer has one node generating the scaled estimated value of the chaotic time series, $x(t+6)$. The optimum number of neurons was determined by adding neurons in systematic form and evaluating the MSE and RMSE of the sets during the learning process [21]. For this Case the optimum architecture was 4-6-1. In Figure 1 we present a comparison between recorded and predicted values of the standard configuration applied in the Mackey–Glass time series. Figure 1*a* shows the training set as a function of the time t. For this training set, the correlation coefficient R^2 was 0.99952. Figure 1*b* shows the MSE of training as a function of the time t, with a MSE_{max} of 0.00032. Figure 1*c* shows the prediction set. Note that for the prediction set, R^2 was 0.99953. Figure 1*d* shows the MSE obtained for the prediction set with MSE_{max} of 0.00027. Table 4 shows the RMSE of some computational methods [12, 6, 13, 5, 11] and the ANN+PSO method proposed in this work (for a more detailed table see [24]). The low errors found with the proposed method (MSE = 0.000028, and RMSE = 0.0053) indicate that it can estimate the Mackey–Glass time series $x(t+6)$ and $x(t+84)$ with better accuracy than other methods.

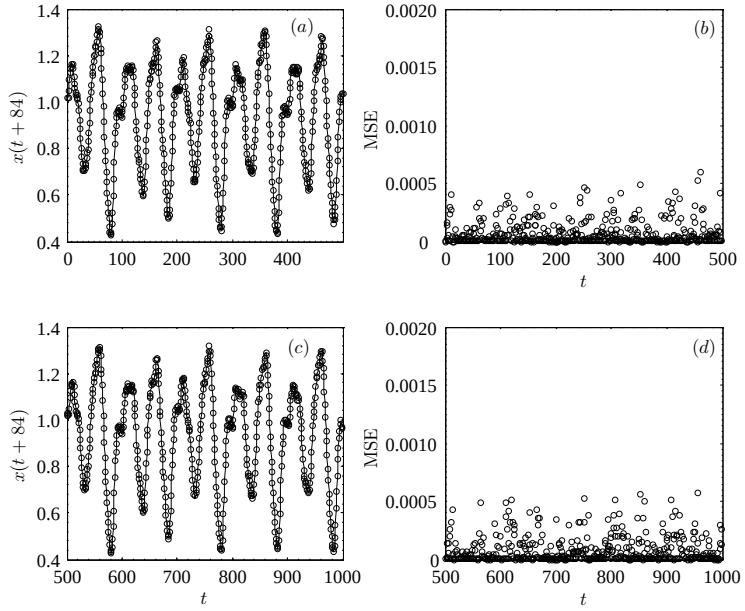

Figure 2. Training set and prediction set for the long-term $x(t + 84)$.

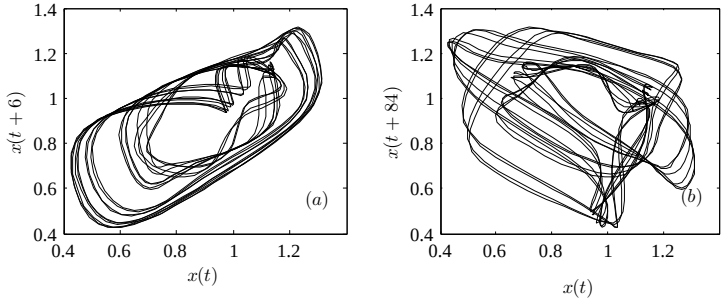

Figure 3. Chaotic attractors for the Mackey–Glass time series studies: (a) short-term representation; (b) long-term representation.

Long-term prediction. In the long-term prediction, the best architecture was 4-12-1. Figure 2 shows the recorded and predicted values of the long-term prediction. Figure 2*a* shows the training set as a function of the time t, with R^2 of 0.99877. Figure 2*b* shows the MSE for training set as a function of the time t, with MSE_{max} of 0.00059. Figure 2*c* shows the prediction set, with R^2 of 0.99853. Figure 2*d* shows the errors for the prediction set with MSE_{max} of 0.00057. Table 4 also shows a comparison between some computational methods found in the literature [27, 28, 29, 30] and our result for the long term prediction. This comparison was made using the normalized root mean squared error (NRMSE) [21]. The errors found with the proposed method (RMSE = 0.0080 and NRMSE = 0.0383) indicate that the method is efficient compared to others found in the literature [22].

Finally, in the Figure 3 we present a representation of the chaotic attractor for the two cases studies. This Figure shows that with the above parameters, the system operates in a high-dimensional regime. The Mackey–Glass system is infinite dimensional (because it is a time-delay equation) and, thus, has an infinite number of Lyapunov exponents (λ_i) [25]. The Lyapunov exponents of dynamical systems are one of a number of invariants that characterize the attractors of the system in a fundamental way [31].

5. Noisy chaotic time series prediction

We consider the noisy times series as the contribution of a noise level on the nominal case (without noise). Here, we consider as nominal case ($x^{\text{Noiseless}}$) the chaotic time series of Mackey-Glass with $\alpha = 0.2$, $\beta = 0.1$ and $\tau = 17$ (sampling of one second). We aim to the prediction of the short-term $x(t+6)$ with $x(t-0)$, $x(t-6)$, $x(t-12)$ and $x(t-18)$ terms. Also, we propose a complementary method that allow us compute the uncertainty of our forecast.

5.1. Simulations: Noisy chaotic time series

The Mackey–Glass noisy chaotic times series, $x_i \equiv x(t)$, is generated as:

$$x_{\text{i}} = x_{\text{i}}^{\text{Noiseless}} + \eta_{\text{i}} \tag{9}$$

where η_{i} is the particular contribution of noise of the i–element. It is estimated as $\eta_{\text{i}} = GR(\sigma_{N,i})$, with $GR(\sigma_{N,i})$ a random number generator following a Gaussian distribution with mean zero and standard deviation equal to $\sigma^2_{N,i}$. Note that $\sigma^2_{N,i}$ corresponds to the noise level considered. Here, we assume that the original data are effected by a white noise, i. e., the noise level is the same in each i–element ($\sigma_{N,i} = \sigma_N$). We consider three white noise levels: $\sigma_N = 0.01$, $\sigma_N = 0.05$ and $\sigma_N = 0.1$. These values are nearly related with the 1 %, 5 % y 10 % of the pick–to–pick amplitude ($I_{\max} - I_{\min} \sim max\{x^{Noiseless}\} - min\{x^{Noiseless}\}$) of nominal case.

Figure 4 shows the noisy chaotic time series of Mackey-Glass used in our work. As expected, the noisy time series with $\sigma_N = 0.01$ (blue line) is the closest to the nominal case. However, the case with $\sigma_N = 0.05$ and $\sigma_N = 0.1$ show a slightly more modified shape from the noiseless case, in particular with $\sigma_N = 0.1$ (red line).

5.2. The performance prediction

The *standard* hybrid ANN+PSO procedure applied in section 4 does not provides information about uncertainties of the predictions. Thus, the *standard* hybrid ANN+PSO algorithm was complemented with a Gaussian stochastic procedure (called *stochastic* hybrid ANN+PSO) in order to obtain a new estimator of predictions that also allow us compute uncertainties of predictions for noisy chaotic time series with a priori knowledge of the noise level.

• The *standard* ANN+PSO is the same procedure described in the section 2; however, here the root mean square error (RMSE, equation 5 in the learning process) is computed as:

$$RMSE = \sqrt{\frac{1}{\text{n}} \sum_{\text{i=1}}^{\text{n}} \frac{\left(y_{\text{i}}^{cal} - y_{\text{i}}^{true}\right)^2}{\sigma^2_{\text{N,i}}}} \tag{10}$$

where $\sigma_{\text{N,i}}$ is the noise level of each i–element ($\sigma_{\text{N,i}} = \sigma_{\text{N}}$ for white noise). The prediction values obtained with this method are identified as y_i. Note that we continue use the RMSE (as defined in equation 5) for evaluation of the prediction performance and the comparison with literature. So that the noise level σ_N only is used in the learning phase (equation 10).

• The *stochastic* ANN+PSO is a procedure executed subsequent to the *standard* ANN+PSO. Once established the architecture (neuron number in hidden layer n, the transfer

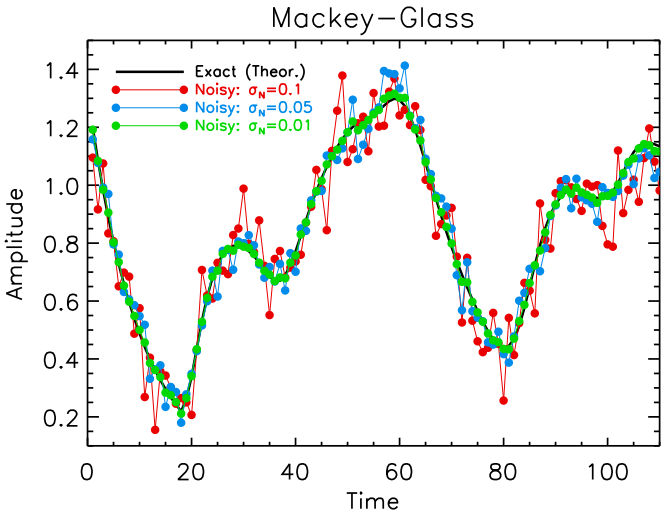

Figure 4. Noisy Mackey-Glass chaotic time series with $\alpha = 0.2$, $\beta = 0.1$ and $\tau = 17$. The black solid line is the noiseless case. The green, blue and red lines correspond to the series with a white noise level (σ_N) contribution of 0.01, 0.05 and 0.1, respectively.

functions f^h) and weights of each neuron ($w^h_{i,j}$ and $b^h_{i,j}$, equations 3 and 4) the neural network act as a function (called *function ANN*) whose output only depend on the input vector. The idea is generate simulations from the input data ($d_i \equiv d(t)$) via Gaussian random number generator in order to propagate the intrinsic data noise through of the *function ANN*. For each i–element of the input time series we generate k–simulations as:

$$d_{i,k} = d_i + GR_k(\sigma_{N,i}) \tag{11}$$

where the noise level $\sigma_{N,i}$ is known ($\sigma_{N,i} = \sigma_N$ for white noise). Note that we recover the data properties as $d_i =< d_{i,k} >$ and $\sigma_{N,i} =< d^2_{i,k} >^{1/2}$, where $< >$ is over the index k.

Now, and for the i-th element, each k input data $d_{i,k}$ provides an output $y_{i,k}$. These $y_{i,k}$ are used in the estimation of a new estimator of prediction ($\hat{y}_i$) and an error on the prediction ($\sigma_{\hat{y}}$) as follow:

$$\hat{y}_i =< y_{i,k} > \quad \text{and} \quad \sigma_{\hat{y}} =< y^2_{i,k} >^{1/2} . \tag{12}$$

In summary, and for each noisy time series, firstly the *standard* ANN+PSO is run in order to obtain the prediction y_i and characterize the *function ANN*. Then, we execute the *stochastic* ANN+PSO to compute $\hat{y}_i$ and $\sigma_{\hat{y}}$.

We applied the *standard* and *stochastic* ANN+PSO to our three noisy Mackay–Glass cases. In the *stochastic* ANN+PSO we run 1000 simulations ($k = 1000$). In left panel of Figure 5, we compare the y_i and $\hat{y}$ predictions with the original noisy MG-CTS for our more noisy case ($\sigma_N = 0.1$). As expected, even on this high noise level case, the y_i and $\hat{y}_i$ are in total agreement. One of a main goals of this section is estimate the uncertainty on the prediction, so that the measurement and the error bars obtained from the *stochastic* ANN+PSO for the noisy time series with $\sigma_N = 0.1$ are presented in the right panel of Figure 5. We confirm that our forecast and the input data, for the strong noise contribution, are in agreement at one-sigma (at 68.5% of confidential level) when the error bars are considered. Therefore, both picture of Figure 5 show that the *standard* and their complement (*stochastic* ANN+PSO) are a robust tool in the predictability of time series with a noise contribution.

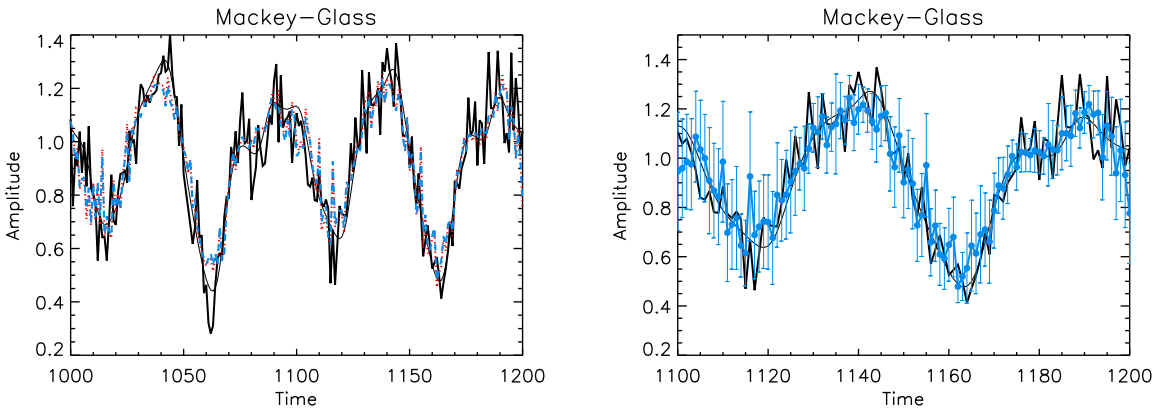

Figure 5. Predictions and uncertainties of noisy Mackey-Glass chaotic time series with a white noise of $\sigma_N = 0.1$. The black solid line correspond to the original noisy MG-CTS. *Left panel:* the dotted (red) and dashed (blue) lines are y_i (from *standard* ANN+PSO) and $\hat{y}_i$ (from *stochastic* ANN+PSO) predictions, respectively. *Right panel:* blue points correspond to the $\hat{y}_i$ prediction and the error bars are their uncertainties $\sigma_{\hat{y}}$.

In each case, and in order to evaluate the forecast performance, we used the y^{input} and the y^{output} to estimate the RMSE (following equation 9). The values computed are showed in Table 5.2. Note that both methods (*standard* and *stochastic*) provide very similar values of RMSE. As expected, the RMSE increases as growing the noise level from a 0.005 to 0.15 for noiseless and $\sigma_N = 0.1$ cases, respectively. Therefore, we confirm that a higher noise level of data leads to a poor estimation of the prediction estimator. We can use the ratio $r_{\mathrm{RMSE}} = \mathrm{RMSE}_{noisy}/\mathrm{RMSE}_{noiseless}$ to study the impact of noise on the performance efficiency of our implementation (see values in Table 5.2). In the worst case, the performance efficiency (r_{RMSE}) is strongly affected by one order of magnitude with respect to noiseless case. Even so, the *standard* and *stochastic* ANN+PSO confirm be a powerful tools for making prediction of chaotic times series.

In the literature, we dont find a similar implementation (due to the ahead prediction, white noise level, etc.) that allow us a straightforward comparison of results. For example, we can contrast our results with those presented by Sheng et al. 2012 [20]. They applied the Echo State Network base on dual estimation (ESN) on a noisy Mackey-Glass time series ($\alpha = 0.2$, $\beta = 0.1$ and $\tau = 17$ with a sampling of 2 second) with a white noise level of $\sigma = 0.1$ (variance of 0.01). However, the ahead prediction was one, which is considerable lower than ours. Depend on the prediction performance they obtained RMSE of 0.05 for Generic ESN (hereafter GESN) and 0.04 for CKF/KF based ESN (henceforth CESN). These RMSE are almost two times better than our RMSE= 0.129 in a similar noise contribution ($\sigma_N = 0.1$). However, the impact of the noise noisy on the performance efficiency is lower in our implementation. In fact, we have a r_{RMSE} of 9.4 meanwhile we computed r_{RMSE} of 1161 and 33.5 for GESN and CESN, respectively.

6. Conclusions

In this work, a hybrid algorithm based on artificial neural network and particle swarm optimization (ANN+PSO) was used in the short-term and long-term prediction of Mackey–Glass chaotic time series. We also study the impact of noise on our hybrid method when a white

Table 3. Parameters used in the evaluation of the prediction performance of the *Standard* and *Stochastic* ANN+PSO approach.

	Noisy Mackey–Glass chaotic time series			
	Noiseless	$\sigma_N = 0.01$	$\sigma_N = 0.05$	$\sigma_N = 0.1$
RMSE	0.0138	0.016	0.067	0.129
r_{RMSE}	1	1.159	4.855	9.348

noise contribution affects the input data. The ANN+PSO was complemented with a procedure (*stochastic* ANN+PSO) in order to propagate the input uncertainties through of our method, and to obtain an estimation of error of the prediction.

Based on the results and discussion presented in this study, the following main conclusions are obtained: i) ANN+PSO method proposed in this work can be properly trained and used in the short-term prediction and long-term prediction of time series with acceptable accuracy; ii) the current value $x(t)$ and the past values used have influential effects on the good training and predicting capabilities of the chosen network; iii) simulation without noise contribution shows that this hybrid ANN+PSO algorithm is a very powerful tool for making prediction of chaotic time series, and the low deviations found with the proposed method shows a better accuracy than other methods available in the literature; iv) when the ANN+PSO is applied on simulated noisy times series our method carry out the prediction with acceptable accuracy (when the input noise level is considered); v) moreover, the performance efficiency is nearly related with the input noise level, for example, in the $\sigma_N = 0.1$ case the performance is worsened around to 10 % ($r_{RMSE} = 9.4$); and vi) the hybrid ANN+PSO algorithm is a robust tool for making prediction of chaotic time series, including cases where the input data are contaminated by a noise contribution.

Acknowledgments

The authors thank the Direction of Research of the University of La Serena (DIULS) for the special support that made possible the preparation of this paper.

References

[1] Karunasinghe D S K and Liong S Y 2006 *J. of Hydrol.* **323** 92-105
[2] Chng E S, Chen S and Mulgrew B 1996 *IEEE Trans. on Neural Networks* **7** 190-4
[3] Zhang J S and Xiao X C 2000 *Chinese Phys. Lett.* **17** 88
[4] Mirzae H 2009 *Chaos Sol. and Frac.* **41** 2681-89
[5] Jang J S R, Sun C T and Mizutani E 1997 *Neuro-fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence.* (Upper Saddle River NJ: Prentice-Hall)
[6] Chen Y, Yang B and Dong J 2006 *Neurocomputing* **69** 449-65
[7] Mackey M C and Glass L 1977 *Science* **197** 287-9
[8] Ikeda K 1979 *Optics Communications* **30** 257-61
[9] Chua L O, Kocarev L, Eckert K and Itoh M 1992 *Int. J. of Bif. and Chaos* **2** 705-8
[10] Bezruchko B P, Karavaev A, Ponomarenko V and Prokhorov M 2001 *Phys. Rev. E* **64** 056216(1)-(6)
[11] Kim D and Kim D 1997 *IEEE Trans. on Fuzzy Systems* **5** 523-35
[12] Chen Y, Yang B and Dong J 2004 *Int. J. of Neural Systems* **14** 125-37
[13] Fu Y Y, Wu C J, Jeng J T and Ko C N 2010 *Exp. Sys. with Appl.* **37** 4441-51
[14] Girard A, Rasmussen C, Quinonero-Candela J and Murray-Smith R 2003 *Advances in Neural Information Processing Systems* (MIT Press) pp 529536.
[15] Sapankevych N and Sankar R 2009 *Comp. Intel. Mag. IEEE* **4** 2438
[16] Haykin S and Principe J 1998 *Sig. Proc. Mag. IEEE* **15** 66
[17] Li D, Han M and Wang J 2012 *IEEE Trans. on Neural Networks and Lear. Sys.* **23** 78799
[18] Han M and Wang X 2013 *Proc. The 2013 Int. Joint Conf. on Neural Networks (Dallas)* vol (no., 1-5

[19] Leung H, Lo T and Wang S 2001 *IEEE Trans. on Neural Networks* **13** 116372
[20] Sheng C, Zhao J, Liu Y and Wang W 2012 *Neurocomputing* **82** 186-95
[21] Lazzús J A 2010 *Fluid Phase Equilibria* **289** 176-84
[22] Lazzús J A 2010 *Comp. and Math. with Appl.* **60** 2260-69
[23] Lazzús J A 2009 *Therm. Acta* **489** 53
[24] Lazzús J A, Salfate I and Montecinos S 2014 *Neural Network World* **24** 601-17
[25] Farmer J D 1982 *Phys. D* **4** 366-93
[26] Han M and Wang Y 2009 *Exp. Sys. with Appl.* **36** 1280-90
[27] Awad M, Pomares H, Rojas I, Salameh O and Hamdon M 2009 *Int. Arab J. of Inf. Tech.* **6** 138-43
[28] Bersini H, Duchateau A and Bradshaw N 1997 *Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems (Barcelona Spain)* vol (Barcelona: 21 IEEE Press) pp 1417-22
[29] Martinetz T M, Berkovich S G and Schulten K J 1993 *IEEE Trans. on Neural Networks* **4** 558-69
[30] Whitehead B A and Choate T D 1996 *IEEE Trans. on Neural Networks* **7** 869-80
[31] Brown R, Bryant P and Abarbanel H D I 1997 *Phys. Rev. A* **43** 2787-2806