CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.94-103
https://doi.org/10.51537/chaos.1116084

On the Prediction of Chaotic Time Series using Neural Networks

Josué Alexis Martínez-García[*,1], **Astrid Maritza González-Zapata**[α,2], **Ericka Janet Rechy-Ramírez**[*,3] **and Esteban Tlelo-Cuautle**[α,4]

[*]University of Veracruz, Artificial Intelligence Research Institute, 91907, Veracruz, Mexico, [α]Instituto Nacional de Astrofisica, Optica y Electronica, Electronics Department, 72840, Puebla, Mexico.

ABSTRACT Prediction techniques have the challenge of guaranteeing large horizons for chaotic time series. For instance, this paper shows that the majority of techniques can predict one step ahead with relatively low root-mean-square error (RMSE) and Symmetric Mean Absolute Percentage Error (SMAPE). However, some techniques based on neural networks can predict more steps with similar RMSE and SMAPE values. In this manner, this work provides a summary of prediction techniques, including the type of chaotic time series, predicted steps ahead, and the prediction error. Among those techniques, the echo state network (ESN), long short-term memory, artificial neural network and convolutional neural network are compared with similar conditions to predict up to ten steps ahead of Lorenz-chaotic time series. The comparison among these prediction techniques include RMSE and SMAPE values, training and testing times, and required memory in each case. Finally, considering RMSE and SMAPE, with relatively few neurons in the reservoir, the performance comparison shows that an ESN is a good technique to predict five to fifteen steps ahead using thirty neurons and taking the lowest time for the tracking and testing cases.

KEYWORDS
Chaotic time series
Neural network
Echo state network
Long short-term memory
RMSE
Prediction technique

INTRODUCTION

Chaos has been a research area that includes several physical phenomena that can be modeled by deterministic mathematical equations, applied to real life problems and predicted applying artificial intelligence based techniques. For example: one can take a chaotic system as the well-known Lorenz oscillator to generate chaotic time series; afterwards, one can use the time series to try to predict several steps ahead applying prediction techniques. In this prediction problem one has the challenge of choosing the appropriate technique, which depends on the nature of the data to validate the prediction, e.g. some data can have slow variations and others fast changes in their dynamics. Some examples of chaotic time series in the real world are for example: sunspots, water run-off, electric changes, temperature, rainfalls, voice signals Lau and Wu (2008); Yang *et al.* (2005); Dhanya and Nagesh Kumar (2010); Jingjing *et al.* (2018), and so on. Clearly, these data is different and therefore the challenge is the development or application of known prediction techniques that guarantee a large prediction horizon with minimum error.

Some of the main characteristics that try to quantify chaotic behavior was introduced by Li and Yorke (1975). From this seminal work, one understand important concepts as fractal dimension, Lyapunov exponents, Fourier transform, Hilbert transform and the reconstruction of an attractor Liu (2010). Another seminal work was introduced by Wolf *et al.* (1985), for determining Lyapunov exponents from a time series, where the chaotic time series can be experimental or taken from simulation. In this manner, one can evaluate the Lyapunov exponent of a chaotic time series to validate if it is chaotic or not, and therefore, it is chaotic if the Lyapunov exponent is positive. On another point of view, it is said that chaotic time series present characteristics seemingly unpredictable due to their complexity Han *et al.* (2019c), and due to their high sensitivity to the initial conditions, as shown by Wolf *et al.* (1985).

Manuscript received: 13 May 2022,
Revised: 18 July 2022,
Accepted: 22 July 2022.

[1] josuealexis15@hotmail.com
[2] erechy@uv.mx
[3] amgonzalezz@uqvirtual.edu.co
[4] etlelo@inaoep.mx (**Corresponding Author**)

One can find a huge number of chaotic time series from physical phenomena or generated from mathematical models. For instance, the authors in Liu (2010) talk about tornadoes and human brain, in which the challenge is predicting the future behavior, thus requiring the development of prediction techniques. Fortunately, nowadays one can find contributions to chaotic time series prediction applying artificial intelligence, statistics, mathematics, electronics among other research areas. On this direction, some authors have shown the usefulness of applying Artificial Neural Networks (ANN) Ong and Zainuddin (2019); Chen and Han (2013); Pano-Azucena *et al.* (2021), fuzzy logic Miranian and Abdollahzade (2013); Heydari *et al.* (2016); Goudarzi *et al.* (2016), Bayes theoremLi *et al.* (2016); Wang *et al.* (2020b), Machine Learning (ML) Alemu (2018); Gromov and Borisenko (2015), multilayer Perceptrons Dalia Pano-Azucena *et al.* (2018); Zhao *et al.* (2014), recurrent neural networks (RNN) Li *et al.* (2012); Ardalani-Farsa and Zolfaghari (2010); Chandra and Zhang (2012); Xu *et al.* (2019), linear and nonlinear filters Wu and Song (2013); Ma *et al.* (2017); Yumei *et al.* (2019), optimization by evolutionary computation Samanta (2011); Chandra *et al.* (2017); Guo *et al.* (2016b), approximation by recursion Wang *et al.* (2017); Li-yun (2010); Han *et al.* (2019b), statistical methods Kurogi *et al.* (2018); Xu *et al.* (2019); Jokar *et al.* (2019), Wavelet transform Zhongda *et al.* (2017); Feng *et al.* (2019b), Lyapunov exponents Yong (2013), computer algorithms Guo *et al.* (2020); Hua *et al.* (2013); Jingjing *et al.* (2018); Zhou *et al.* (2017) and hybrid architectures Xiao *et al.* (2019); Fu *et al.* (2010); Han *et al.* (2017).

Among these techniques, the optimization by evolutionary computation and hybrid architectures have shown good results. In the case of optimization by evolutionary computation, one can find the application of Particle Swarm Optimization (PSO) Eberhart and Kennedy (1995), Differential Evolution (DE) Price *et al.* (2006), Cuckoo Search Yang and Deb (2010), Ant Colony Optimization (ACO) Dorigo *et al.* (2006), Fruit Fly Optimization Algorithm Xing and Gao (2014), Whale Optimization Algorithm Mirjalili and Lewis (2016), grey wolf optimizer Mirjalili *et al.* (2014) and co-evolution, where different optimization methods work together.

In the case of hybrid architectures for chaotic time series prediction, the most known are: Bayes theorem Swinburne (2004), Echo State Network (ESN) Jaeger (2007), ANN Drew and Monson (2000), Wavelet transform Zhang (2019a), long short-term memory (LSTM) and Least Square Support Vector Machine (LSSVM) Suykens and Vandewalle (1999).

In this manner, this paper provides a summary on chaotic time series prediction techniques and compares the performance of four techniques based on neural networks to predict chaotic time series from Lorenz chaotic system. The next section shows the most used models of Lorenz system and Mackey-Glass, and others, and shows a Table summarizing different prediction techniques, comparing the predicted steps, data used for the prediction and the associated root-mean-square error (RMSE) for each case. Afterwards, this paper compares four prediction techniques based on neural networks, namely: ESN, LSTM, ANN and 1-Dimension Convolutional Neural Network (1D-CNN). The prediction results are shown in the section before concluding this work.

TECHNIQUES FOR CHAOTIC TIME SERIES PREDICTION

In the current state of the art, one can find different techniques oriented to predict chaotic time series. The following papers were used for the classification of prediction techniques, predicted steps, number of points used for the prediction, and the associated RMSE: Alemu (2018); Shinozaki *et al.* (2020); Zhang and Jiang (2020); Su and Yang (2021); Zhang *et al.* (2020). The chaotic time series data was mainly taken from two chaotic systems: *Lorenz* and *Mackey-Glass*.

1. **Lorenz**:
   This is a deterministic system modeled by three ordinary differential equations (ODEs) introduced by Lorenz (1963), and given by (1), where chaotic behavior exists by setting $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$.

$$\begin{aligned} \frac{dx(t)}{dt} &= \sigma[y(t) - x(t)] \\ \frac{dy(t)}{dt} &= x(t)[\rho - z(t)] - y(t) \\ \frac{dz(t)}{dt} &= x(t)y(t) - \beta z(t) \end{aligned} \tag{1}$$

2. **Mackey-Glass**:
   This chaotic system was introduced by Mackey and Glass (1977), and denoted by (2), where τ is a delay parameter, and it can be set to $\tau \leq 4.43$ to produce a fixed point, $4.43 \leq \tau \leq 13.3$ to produce a stable limit cycle, $13.3 \leq \tau \leq 16.8$ to produce a double limit attraction, and $16.8 \leq \tau$ to generate chaotic behavior.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1 + x^c(t-\tau)} - bx(t) \tag{2}$$

When simulating a chaotic system, the amplitudes of the state variables can be as large as possible, however, for hardware implementation, it is desired to have amplitudes within the range $[-1, 1]$ or $[0, 1]$. In the validation of the steps predicted by each technique, the authors use different errors, such as: RMSE, Mean Square Error (MSE), Mean Absolute Error (MAE), Normalized RMSE (NRMSE), R^2, among others. However, in the majority of works, the most used measure is RMSE, which is defined by (3), where N is the total of attributes, $\tilde{y}_n$ is the predicted value and y_n^{target} the reference value.

$$RMSE = \sqrt{\frac{\Sigma_{n=1}^{N} \left(\tilde{y}_n - y_n^{target}\right)^2}{N}} \tag{3}$$

In addition, Symmetric Mean Absolute Percentage Error (SMAPE) is implemented like an accuracy measure based on percentage errors. This error is described in equation (4) and indicates the percent of accuracy of the real value versus the forecast value in descendent form, where N is the total of attributes, F_n is the predicted value and A_n is the actual value.

$$SMAPE = \frac{100\%}{n} \Sigma_{n=1}^{N} \frac{|F_n - A_n|}{\frac{(|F_n| + |A_n|)}{2}} \tag{4}$$

In Table 1, we list some prediction techniques including the type of chaotic data used by the associated technique, the predicted steps ahead, number of test points, and RMSE. It can be appreciated that hybrid and optimized techniques have low RMSE, and also, the low errors are associated to the techniques predicting 1 step ahead of chaotic time series. The minimum number of points for testing each technique is 500.

Table 1 Prediction techniques for chaotic time series, ordered from the lowest to the highest RMSE value.

Technique	Chaotic serie	Prediction	Test data	RMSE
Combining the phase space reconstruction and fuzzy logic Gholizade-Narm and Shafiee-Chafi (2015)	Mackey-Glass	1 step	600	2.26E-10
Hybrid Empirical Mode Decomposition - Neural Networks (HEMD-NN) Tang *et al.* (2020)	Mackey-Glass	1 step	2,000	5.31E-08
Efficient Extreme Learning Machine - Differential Evolution (EELM-DE) Guo *et al.* (2016b)	Lorenz	1 step	500	7.67E-08
Kernel Local Polynomial coefficient autoregressive Prediction (KLPP) Su and Li (2015b)	Henon	1 step	500	8.44E-07
Hybrid Elman-NARX neural networks Ardalani-Farsa and Zolfaghari (2010)	Mackey-Glass	1 step	1,000	3.72E-05
Radial Basis Function (RBF) neural network Zhang *et al.* (2013)	Drift sensor	2 step	4,000	4.87E-05
ESN optimized by Selective Opposition Grey Wolf Optimizer (SOGWO-ESN) Chen and Wei (2021)	Mackey-Glass	25 step	800	1.46E-04
Artificial Neural Networks (ANNs), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Least-Squares Support Vector Machines (LSSVM) Dalia Pano-Azucena *et al.* (2018)	Chaotic system	6 step	2,000	2.98E-04
Local Neuro-Fuzzy (LNF) - Least-Squares Support Vector Machines (LSSVMs) Miranian and Abdollahzade (2013)	Mackey-Glass	6 step	500	7.90E-04
Local Functional Coefficient Autoregressive (LFAR) Su and Li (2015a)	Mackey-Glass	500 step	500	1.30E-03
Structured Manifold - Broad Learning System (SM-BLS) Han *et al.* (2019a)	Lorenz	10 step	$\approx$ 4,400	2.45E-03
Hierarchical Delay-Memory Echo State Network (HDESN) Na *et al.* (2021)	Lorenz	12 step	2,000	2.65E-03
The Elman recurrent networkChandra and Zhang (2012)	Mackey-Glass	500 step	500	6.33E-03
Local Volterra model based on phase points clustering Han *et al.* (2018)	Lorenz	50 step	4,976	8.10E-03

PREDICTION TECHNIQUES BASED ON NEURAL NETWORKS

This section shows a comparison among prediction techniques based on most used neural networks.

Echo State Network

The prediction technique based on ESN was introduced by Jaeger (2007). It becomes to behave as a recurrent neural network and includes a reservoir that assigns random weights while a certain percentage or neurons are connected by accomplishing the property of echo, as shown by Lukoševičius (2012). The update equations are given in equations (5) and (6). In these equations $x(n)$ denotes the activation vector of the neurons in the reservoir, where n is the value of each neuron in the reservoir, α is the leaking rate denoted by $\in (0, 1]$ for the training. $\tilde{x}(n)$ is the update for each n, where $tanh(\cdot)$ holds the vertical concatenation of matrix W^{in} and W represents the inputs and recurrent weights of the matrices. The output layer is defined by equations (7) and (8).

$$\tilde{x}(n) = tanh(\mathrm{W}^{in}[1; u(n)] + \mathrm{W} \times (n-1)) \tag{5}$$

$$x(n) = (1-\alpha) \times (n-1) + \alpha\tilde{x}(n) \tag{6}$$

$$W^{out} = Y^{target} X^T (XX^T + \beta I)^{-1} \tag{7}$$

$$y(n) = W^{out}[1; u(n); x(n)] \tag{8}$$

In equation (8), $y(n)$ is the output layer, W^{out} is the weights output matrix determined by *Ridge regression* or also known as *Tikhonov* regularization, where β it's regularization coefficient. On other hand, $[\cdot;\cdot;\cdot]$ holds the verticality in the concatenation of the vector as mentioned above; X is the collect data of W (it mean the percent of connection in the reservoir).The simulation of this prediction technique includes a reservoir of 30 neurons and a spectral radius (SP) of 2.5.

Long Short-Term Memory

The technique known as Long Short-Term Memory (LSTM) is a kind of recurrent neural network that was introduced by Hochreiter and Schmidhuber (1997). Its main characteristic is the ability to retain one state of a sequence in a long term. An LSTM has three inputs and two outputs: x_t is the current input value as denoted by equations (10) and (11); while at the same time shares the input with h_{t-1}, that is the previous output value of the net, as described by equations (10) and (11). c_{t-1} denotes the input to the *cell state*; the outputs h_t are denoted by equations (13) and (14), and C_t is the unitary state of the current LSTM net given by equation (12). This LSTM in addition includes update gates of information to forget and update the cell state values. The description of the *forget gate* is given in f_t by equation (9) and the *update gate* is given in C_t by (12).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{11}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = o_t * tanh(C_t) \tag{14}$$

In these equations σ is a sigmoid function scaled within the values $[0, 1]$ for the updating of the C_t (*cell state*) for the consumption of the next time step LSTM. *tanh* denotes the activation function, $W_{[\cdot]}$ is the weight matrix for learning, $b_{[\cdot]}$ is the bias or every neuronal network in the LSTM, and x_t denotes the inputs to the LSTM net, h_{t-1} and c_{t-1} are the inputs from the previous time step and f_t means the forget gate. The simulation of this technique consists of the series connection of four LSTM.

Artificial Neural Network

The ANN was introduced by McCulloch and Pitts (1943), as a mathematical model described by a bio-inspiration of the neurons in the human brain. An ANN consists of an array of artificial neurons connected in a *feed forward* way. In this manner, it consists of at least three layers, namely: input layer, hidden layer and output layer. The input layer can be described by vector x_i; in the hidden layer take place the operations evaluated by the weights w_i and bias b, it includes the activation functions to each neuron denoted as f. The hidden layer operates on equation (15), and it can include more than one layer. In the output layer, the last evaluations take place to provide the learned data. The training of an ANN consists of epochs, and the most used training method is known as *Backpropagation* that is denoted by equations (16) and (17).

$$\sum_{i=1}^{i=n} (w_i * x_i) + b \tag{15}$$

$$E = \frac{\sum_{i=1}^{i=n} (t_i - a_i)^2}{2} \tag{16}$$

$$\Delta \mathrm{W} = -\alpha \frac{\partial EW}{\partial W} \tag{17}$$

In this case, equation (16) evaluates the mean square error of the target t_i and the output of the neuron a_i, which updates the weights by (17) when the net is back-propagated to learn in each epoch. The simulation of this technique was performed considering a hidden layer of 20, 15 and 10 neurons.

Convolutional Neural Network

The Convolutional Neural Network (CNN) is a kind of ANN introduced by Fukushima (1980). The difference with an ANN is the application devoted to bidirectional matrices, being quite effective for artificial vision tasks. However, its application is also suitable for time series prediction, plain images and signals from functional magnetic resonance images. The CNN consists of the main layers: Convolutional layer, which performs the convolution of the inputs with a kernel given in equation (18); the Max-pooling layer, which extracts the main characteristics form the convolution; and the third layer is a fully connected network (feed forward).

$$Y_j = g(b_j + \sum_i K_{ij} \otimes Y_i) \tag{18}$$

In equation (18), Y_j is the output of neuron j evaluated through a linear combination of the outputs Y_i of the neurons in the previous layer, each one operated with the convolutional core K_{ij} corresponding to that connection. This value is added to b_j and afterwards send to an activation function $g(\cdot)$ of non-linear type. For chaotic time series prediction, the CNN has a kernel that moves in one direction, i.e. guided by the time series. The simulation of this technique was performed using a Max-pooling layer (MP) and 50 neurons that are full-connected among them.

SIMULATION RESULTS

The simulation of the four prediction techniques described in the previous section, was performed using a personal computer with Intel i5-11400H processor of 64 bit at 2.70 GHz, with 8 Gb of RAM. The four techniques have similar characteristics to perform the prediction and was executed each one five times. In this manner, the training was executed using a random seed trying to get similar results. In all the cases, the Lorenz and Mackey-Glass systems was simulated to generate a data of 1500 points that were used for the training and 800 points for the test during the prediction, omitting the first 200 points that are the transitory state of the chaotic system. The four techniques were executed using a leaking rate of 0.001 with 180 epochs for the learning, except for the ESN. The prediction of the steps ahead was performed in an adjacent way with respect to the inputs and predicted steps.

The prediction capabilities of the four techniques is given considering four characteristics: (I) predicted steps, (II) errors (RMSE and SMAPE), (III) training and test time, and (IV) memory required during the training and test, as listed in Tables 2 and 3. In each prediction technique, five runs were executed for each step prediction, reporting the best result of this five executed. The RMSE is the total over the 800 test data of the predicted values. As one can see, the lowest RMSE at one step is provided by LSTM, while the lowest RMSE with the highest predicted steps ahead (15) was provided by CNN. However, the ESN provides the results in general with low RMSE for the prediction with different steps ahead, in addition the SMAPE presents de low variance that others models. Figures 1, 2, 3 and 4 show the better prediction for the chaotic time series results reaching 15 steps ahead applying ESN, LSTM, ANN and CNN techniques, respectively. In the experiments, considering Lorenz time series the techniques reported lower RMSE and SMAPE than when using Mackey-Glass time series, as shown in Tables 2 and 3.

The determination of the maximum predicted steps ahead given in Tables 2 and 3 was done according to the steps ahead (1, 3, 5, 10 and 15 steps). Details of the topologies of each prediction technique are also given in the Tables. For example, one can see the quantity of layers and neurons in each case. It is worth noting the stability of ESN, considering the RMSE and SMAPE, it has low execution time and memory requirement with respect to the results provided by the other prediction techniques. From the results shown in Figures 1, 2, 3, and 4, one can see that some models present high variance in the prediction, as reported in Tables 2 and 3, where SMAPE presents high values. One can also see that the prediction using Lorenz time series is much better providing low RMSE and SMAPE for ESN. However, when using the Mackey-Glass time series the prediction techniques present a similar RMSE result, as shown Table 3. ESN presented a low accuracy in Mackey-Glass with respect to the Lorenz time series. The reason for this are the values of the parameters, since it is a time series with a different behavior. Parameters such as number of neurons, spectral radius, among others, must be adjusted to obtain good results, compared to the other three models, since they adapt to the series with the passage of time. Finally, in Table 4 we show the best results of our experiments with each prediction technique and compared with results in the state of the art.

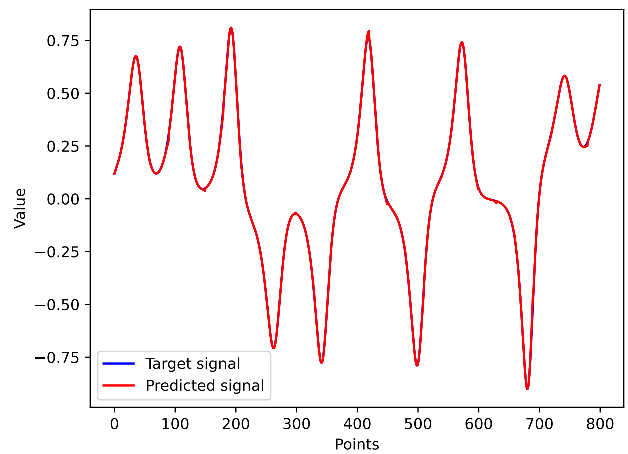

Figure 1 Lorenz time series prediction results by ESN reaching 15 steps ahead.

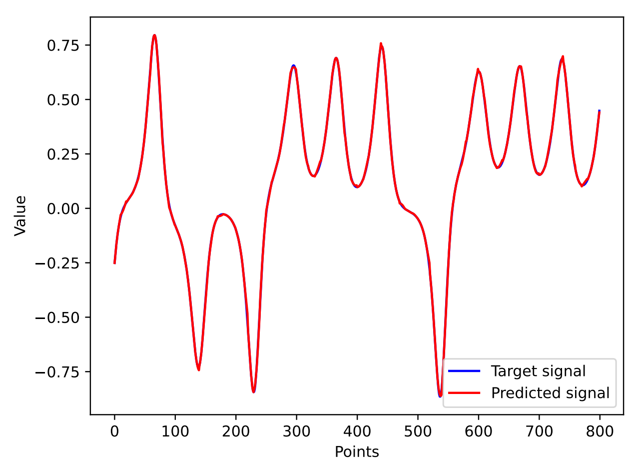

Figure 2 Lorenz time series prediction results by LSTM reaching 15 steps ahead.

Table 2 **Comparison of the best executions in the four prediction techniques with Lorenz time series, listing the predicted steps ahead, RMSE, SMAPE, training and testing times, and training and testing memory.**

ESN:	Step ahead	Training-time	Testing-time	Training-memory	Testing-memory	RMSE	SMAPE
Train=1,500	1	0.0156 Sec.	0.0625 Sec.	1.7728 Mb	0.2901 Mb	2.86E-02	12.52%
Test=800	3	0.0375 Sec.	0.0531 Sec.	1.8654 Mb	0.3746 Mb	2.11E-03	4.69%
Neurons=30	5	0.0317 Sec.	0.2157 Sec.	1.9689 Mb	0.5094 Mb	8.21E-04	0.75%
SP=2.5	10	0.0312 Sec.	0.0562 Sec.	2.1878 Mb	0.9013 Mb	1.30E-03	0.86%
Leaking rate=0.001	15	0.0316 Sec.	0.0467 Sec.	2.3818 Mb	1.2797 Mb	5.76E-03	1.47%
LSTM:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	364.0543 Sec.	0.9687 Sec.	5.2716 Mb	1.9529 Mb	2.22E-02	11.63%
Test=800	3	404.6772 Sec.	0.9218 Sec.	5.5292 Mb	1.9757 Mb	3.04E-02	4.71%
LSTMs=4	5	446.3018 Sec.	0.9278 Sec.	5.3843 Mb	1.9955 Mb	4.51E-03	3.17%
Epochs=180	10	556.7754 Sec.	0.9322 Sec.	5.2363 Mb	2.2375 Mb	6.89E-03	7.64%
Leaking rate=0.001	15	365.8241 Sec.	0.9443 Sec.	5.2806 Mb	1.9483 Mb	1.50E-02	7.86%
ANN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	265.9227 Sec.	0.1718 Sec.	1.7933 Mb	0.5118 Mb	2.59E-02	14.91%
Test=800	3	266.1119 Sec.	0.1812 Sec.	1.7667 Mb	0.6770 Mb	5.02E-03	11.63%
Layers=20,15,10	5	262.3324 Sec.	0.1624 Sec.	1.7926 Mb	0.7977 Mb	5.85E-03	3.65%
Epochs=180	10	263.9948 Sec.	0.1673 Sec.	1.8077 Mb	1.1445 Mb	6.28E-03	8.65%
Leaking rate=0.001	15	263.7734 Sec.	0.1685 Sec.	1.7948 Mb	1.4987 Mb	7.07E-03	6.81%
CNN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	268.1515 Sec.	0.1781 Sec.	1.8653 Mb	0.5576 Mb	2.27E-02	12.01%
Test=800	3	265.2860 Sec.	0.1875 Sec.	1.8602 Mb	0.6726 Mb	4.72E-03	6.76%
Layers=1MP,50	5	269.0973 Sec.	0.1866 Sec.	1.7919 Mb	0.8411Mb	6.64E-03	6.72%
Epochs=180	10	274.2548 Sec.	0.1875 Sec.	1.8689 Mb	1.1925 Mb	7.41E-03	8.92%
Leaking rate=0.001	15	274.9681 Sec.	0.1866 Sec.	1.8156 Mb	1.5433 Mb	5.45E-03	6.32%

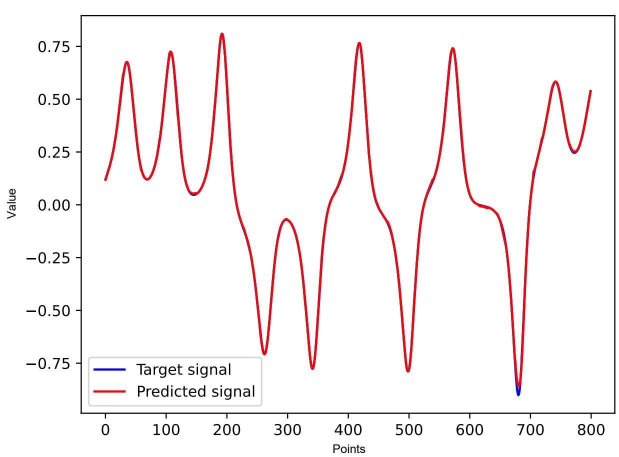

Figure 3 Lorenz time series prediction results by ANN reaching 15 steps ahead.

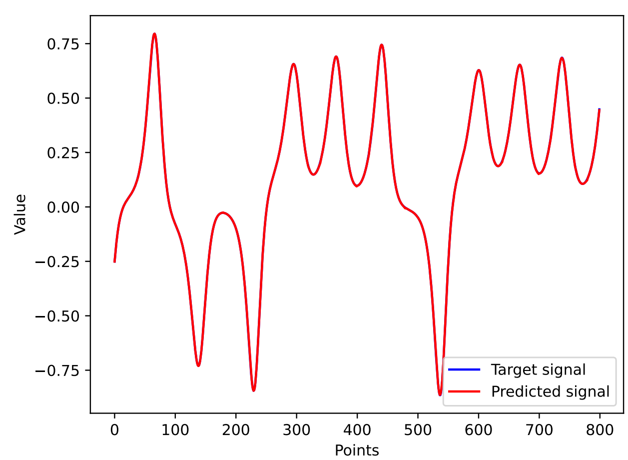

Figure 4 Lorenz time series prediction results by CNN reaching 15 steps ahead.

Table 3 Comparison of the best executions in the four prediction techniques with Mackey-Glass time series, listing the predicted steps ahead, RMSE, SMAPE, training and testing times, and training and testing memory.

ESN:	Step ahead	Training-time	Testing-time	Training-memory	Testing-memory	RMSE	SMAPE
Train=1,500	1	0.2656 Sec.	0.3125 Sec.	2.9410 Mb	0.3280 Mb	3.74E-02	8.23%
Test=800	3	0.0468 Sec.	0.0624 Sec.	3.0219 Mb	0.3865 Mb	1.49E-02	2.76%
Neurons=30	5	0.0312 Sec.	0.0468 Sec.	3.1151 Mb	0.5333 Mb	1.96E-02	3.14%
SP=2.5	10	0.0312 Sec.	0.0624 Sec.	3.3470 Mb	0.9001 Mb	8.47E-02	12.01%
Leaking rate=0.001	15	0.0312 Sec.	0.0624 Sec.	2.5823 Mb	1.2691 Mb	7.79E-02	11.41%
LSTM:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	363.7754 Sec.	0.9218 Sec.	5.2639 Mb	1.9470 Mb	3.49E-02	7.46%
Test=800	3	412.3002 Sec.	0.9446 Sec.	5.1749 Mb	1.9738 Mb	1.27E-02	2.26%
LSTMs=4	5	462.1199 Sec.	1.2499 Sec.	7.9011 Mb	2.4232 Mb	2.52E-02	3.96%
Epochs=180	10	555.2916 Sec.	0.9218 Sec.	5.1964 Mb	2.2387 Mb	2.38E-02	4.72%
Leaking rate=0.001	15	647.9304 Sec.	1.2812 Sec.	5.1711 Mb	2.6389 Mb	1.82E-02	3.60%
ANN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	266.2632 Sec.	0.1562 Sec.	1.7642 Mb	0.5119 Mb	3.59E-02	7.86%
Test=800	3	263.4166 Sec.	0.1562 Sec.	1.7559 Mb	0.6515 Mb	1.58E-02	2.37%
Layers=20,15,10	5	268.1604 Sec.	0.1874 Sec.	1.7469 Mb	0.7927 Mb	3.30E-02	4.65%
Epochs=180	10	266.1194 Sec.	0.1562 Sec.	1.7969 Mb	1.1753 Mb	1.52E-02	2.58%
Leaking rate=0.001	15	268.1438 Sec.	0.1528 Sec.	2.0190 Mb	0.1528 Mb	1.04E-02	2.00%
CNN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	267.5832 Sec.	0.1875 Sec.	1.8019 Mb	0.5631 Mb	3.51E-02	8.45%
Test=800	3	264.8904 Sec.	0.1875 Sec.	1.8881 Mb	0.7009 Mb	2.14E-02	4.37%
Layers=1MP,50	5	269.0176 Sec.	0.2031 Sec.	1.7811 Mb	0.8507 Mb	3.63E-02	6.83%
Epochs=180	10	272.9374 Sec.	0.1718 Sec.	1.7973 Mb	1.1921 Mb	1.61E-02	2.89%
Leaking rate=0.001	15	272.2324 Sec.	0.1875 Sec.	1.7894 Mb	1.5429 Mb	9.52E-03	2.14%

Table 4 Comparison of our better results with the state of the art

Technique	Chaotic serie	Prediction	RMSE
Our approach with LSTM	Lorenz	1 step	2.22E-02
Deep Hybrid Neural Network with Differential Neuroevolution Huang *et al.* (2020)	Lorenz	1 step	7.56E-02
Our approach with ESN	Lorenz	5 steps	8.21E-04
Adaptive Sparse Quantization Kernel Least Mean Square Algorithm Zhao *et al.* (2021)	Beijing PM 2.5	5 steps	3.15E-02
Improved Kernel Recursive Least Squares Algorithm Han *et al.* (2019b)	Lorenz	5 steps	4.41E-02
Co-evolutionary predictive algorithm Chandra *et al.* (2017)	Mackey-Glass	5 steps	5.90E-02
Our approach with ESN	Lorenz	10 steps	1.30E-03
Structured Manifold - Broad Learning System (SM-BLS) Han *et al.* (2019a)	Lorenz	10 steps	2.45E-03
Robust manifold broad learning system for large-scale noisy chaotic time series prediction Feng *et al.* (2019a)	Lorenz	10 steps	1.82E-01

CONCLUSION

This paper showed the state of the art in chaotic time series prediction using different prediction techniques. From Table 1, it was observed the usefulness of neural networks, so that four techniques were chosen to perform the prediction of time series taken data from Lorenz and Mackey-Glass systems. Tables 2 and 3 summarizes the prediction results provided by applying four techniques that are based on ESN, LSTM, ANN and CNN. As a result, one can see that the ESN is the technique providing better prediction results in its stability of results in the five executions realized. In addition, ESN obtained the low RMSE and SMAPE values. This means that the results provided by ESN have the lower variance in average compared to the other prediction technqiues, and it also requires lower computing resources.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Alemu, M. N., 2018 A fuzzy model for chaotic time series prediction. International Journal of Innovative Computing Information and Control **14**: 1767–1786.

Ardalani-Farsa, M. and S. Zolfaghari, 2010 Chaotic time series prediction with residual analysis method using hybrid elman-narx neural networks. Neurocomputing **73**: 2540–2553.

Chandra, R., Y.-S. Ong, and C.-K. Goh, 2017 Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction. Neurocomputing **243**: 21–34.

Chandra, R. and M. Zhang, 2012 Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. Neurocomputing **86**: 116–123.

Chen, D. and W. Han, 2013 Prediction of multivariate chaotic time series via radial basis function neural network. Complexity **18**: 55–66.

Chen, H.-C. and D.-Q. Wei, 2021 Chaotic time series prediction using echo state network based on selective opposition grey wolf optimizer. Nonlinear Dynamics **104**: 3925–3935.

Cheng, W., Y. Wang, Z. Peng, X. Ren, Y. Shuai, *et al.*, 2021 High-efficiency chaotic time series prediction based on time convolution neural network. Chaos Solitons & Fractals **152**.

Dalia Pano-Azucena, A., E. Tlelo-Cuautle, S. X. D. Tan, B. Ovilla-Martinez, and L. Gerardo de la Fraga, 2018 Fpga-based implementation of a multilayer perceptron suitable for chaotic time series prediction. Technologies **6**.

Dhanya, C. and D. Nagesh Kumar, 2010 Nonlinear ensemble prediction of chaotic daily rainfall. Advances in Water Resources **33**: 327–347.

Dorigo, M., M. Birattari, and T. Stutzle, 2006 Ant colony optimization. IEEE computational intelligence magazine **1**: 28–39.

Drew, P. J. and J. R. Monson, 2000 Artificial neural networks. Surgery **127**: 3–11.

Eberhart, R. and J. Kennedy, 1995 Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pp. 1942–1948, Citeseer.

Feng, S., W. Ren, M. Han, and Y. W. Chen, 2019a Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective. Neural Networks **117**: 179–190.

Feng, T., S. Yang, and F. Han, 2019b Chaotic time series prediction using wavelet transform and multi-model hybrid method. Journal of Vibroengineering **21**: 1983–1999.

Fu, Y.-Y., C.-J. Wu, J.-T. Jeng, and C.-N. Ko, 2010 Arfnns with svr for prediction of chaotic time series with outliers. Expert Systems with Applications **37**: 4441–4451.

Fukushima, K., 1980 Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics **36**: 193–202.

Ganjefar, S. and M. Tofighi, 2018 Optimization of quantum-inspired neural network using memetic algorithm for function approximation and chaotic time series prediction. Neurocomputing **291**: 175–186.

Gholizade-Narm, H. and M. R. Shafiee-Chafi, 2015 Using repetitive fuzzy method for chaotic time series prediction. Journal Of Intellıgent & Fuzzy Systems **28**: 1937–1946.

Goudarzi, S., M. B. Khodabakhshi, and M. H. Moradi, 2016 Interactively recurrent fuzzy functions with multi objective learning and its application to chaotic time series prediction. Journal Of Intellıgent & Fuzzy Systems **30**: 1157–1168.

Gromov, V. A. and E. A. Borisenko, 2015 Predictive clustering on non-successive observations for multi-step ahead chaotic time series prediction. Neural Computing & Applications **26**: 1827–1838.

Gromov, V. A. and A. N. Shulga, 2012 Chaotic time series prediction with employment of ant colony optimization. Expert Systems With Applıcations **39**: 8474–8478.

Guo, F., L. Lin, and C. Wang, 2016a Novel continuous function prediction model using an improved takagi-sugeno fuzzy rule and its application based on chaotic time series. Engıneering Applications Of Artificial Intelligence **55**: 155–164.

Guo, W., T. Xu, and Z. Lu, 2016b An integrated chaotic time series prediction model based on efficient extreme learning machine and differential evolution. Neural Computing & Applications **27**: 883–898.

Guo, X., Y. Sun, and J. Ren, 2020 Low dimensional mid-term chaotic time series prediction by delay parameterized method. Information Sciences **516**: 1–19.

Han, F., S. Yang, and S. Song, 2018 Local volterra multivariable chaotic time series multi-step prediction based on phase points clustering. Journal of Vibroengineering **20**: 2486–2503.

Han, M., S. Feng, C. L. P. Chen, M. Xu, and T. Qiu, 2019a Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction. IEEE Transactıons On Knowledge And Data Engıneerıng **31**: 1809–1821.

Han, M., W. Li, S. Feng, T. Qiu, and C. L. P. Chen, 2021 Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction. IEEE Transactions On Neural Networks And Learning Systems **32**: 2320–2329.

Han, M., R. Zhang, and M. Xu, 2017 Multivariate chaotic time series prediction based on elm-plsr and hybrid variable selection algorithm. Neural Processing Letters **46**: 705–717.

Han, M., S. Zhang, M. Xu, T. Qiu, and N. Wang, 2019b Multivariate chaotic time series online prediction based on improved kernel recursive least squares algorithm. IEEE Transactions On Cybernetics **49**: 1160–1172.

Han, M., K. Zhong, T. Qiu, and B. Han, 2019c Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview. IEEE Transactions on Cybernetics **49**: 2720–2731.

Heydari, G., M. Vali, and A. A. Gharaveisi, 2016 Chaotic time series prediction via artificial neural square fuzzy inference system.

Expert Systems with Applications **55**: 461–468.
Hochreiter, S. and J. Schmidhuber, 1997 Long Short-Term Memory. Neural Computation **9**: 1735–1780.
Hua, Q., M. Wen-Tao, Z. Ji-Hong, and C. Ba-Dong, 2013 Kernel least mean kurtosis based online chaotic time series prediction. Chinese Physics Letters **30**.
Huang, W., Y. Li, and Y. Huang, 2020 Deep hybrid neural network and improved differential neuroevolution for chaotic time series prediction. IEEE Access **8**: 159552–159565.
Jaeger, H., 2007 Echo state network. scholarpedia **2**: 2330.
Jian-Ling, Q., W. Xiao-Fei, Q. Yu-Chuan, G. Feng, and D. Ya-Zhou, 2014 An improved local weighted linear prediction model for chaotic time series. Chinese Physics Letters **31**.
Jianshan, L., W. Changming, Z. Aijun, and X. Xiaomin, 2012 Residual gm(1,1) model-based prediction method for chaotic time series. Journal of Grey System **24**: 379–388.
Jingjing, L., Z. Qijin, Z. Yumei, W. Xiaojun, W. Xiaoming, *et al.*, 2018 Hidden phase space reconstruction: A novel chaotic time series prediction method for speech signals. Chinese Journal of Electronics **27**: 1221–1228.
Jokar, M., H. Salarieh, and A. Alasty, 2019 On the existence of proper stochastic markov models for statistical reconstruction and prediction of chaotic time series. Chaos Solitons & Fractals **123**: 373–382.
Kurogi, S., M. Toidani, R. Shigematsu, and K. Matsuo, 2018 Performance improvement via bagging in probabilistic prediction of chaotic time series using similarity of attractors and loocv predictable horizon. Neural Computing & Applications **29**: 341–349.
Lau, K. W. and Q. H. Wu, 2008 Local prediction of non-linear time series using support vector regression. Pattern Recogn. **41**: 1539–1547.
Li, D., M. Han, and J. Wang, 2012 Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions On Neural Networks and Learning Systems **23**: 787–799.
Li, Q. and R.-C. Lin, 2016 A new approach for chaotic time series prediction using recurrent neural network. Mathematical Problems in Engineering **2016**.
Li, T.-Y. and J. A. Yorke, 1975 Period three implies chaos. The American Mathematical Monthly **82**: 985–992.
Li, Y., X. Jiang, H. Zhu, X. He, S. Peeta, *et al.*, 2016 Multiple measures-based chaotic time series for traffic flow prediction based on bayesian theory. Nonlinear Dynamics **85**: 179–194.
Li-yun, S., 2010 Prediction of multivariate chaotic time series with local polynomial fitting. Computers & Mathematiıcs with Applications **59**: 737–744.
Liu, Z., 2010 Chaotic time series analysis. Mathematical Problems in Engineering **2010**: 720190.
Lorenz, E. N., 1963 Deterministic nonperiodic flow. Journal of Atmospheric Sciences **20**: 130 – 141.
Lukoševičius, M., 2012 A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade: Second Edition*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, pp. 659–686, Springer Berlin Heidelberg, Berlin, Heidelberg.
Lv, M., X. Zhang, H. Chen, C. Ling, and J. Li, 2020 An accurate online prediction model for kiln head temperature chaotic time series. IEEE Access **8**: 44288–44299.
Ma, W., J. Duan, W. Man, H. Zhao, and B. Chen, 2017 Robust kernel adaptive filters based on mean p-power error for noisy chaotic time series prediction. Engıneerıng Applıcatıons Of Artıfıcıal Intellıgence **58**: 101–110.
Mackey, M. C. and L. Glass, 1977 Oscillation and chaos in physiological control systems. Science **197**: 287–289.
McCulloch, W. S. and W. Pitts, 1943 A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics **5**: 115–133.
Miranian, A. and M. Abdollahzade, 2013 Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction. IEEE Transactions on Neural Networks and Learning Systems **24**: 207–218.
Mirjalili, S. and A. Lewis, 2016 The whale optimization algorithm. Advances in engineering software **95**: 51–67.
Mirjalili, S., S. M. Mirjalili, and A. Lewis, 2014 Grey wolf optimizer. Advances in engineering software **69**: 46–61.
Na, X., W. Ren, and X. Xu, 2021 Hierarchical delay-memory echo state network: A model designed for multi-step chaotic time series prediction. Engıneerıng Applıcatıons Of Artıfıcıal Intellıgence **102**.
Nguyen, H. M., G. Kalra, T. Jun, and D. Kim, 2020 Chaotic time series prediction using a novel echo state network model with input reconstruction, bayesian ridge regression and independent component analysis. Internatıonal Journal Of Pattern Recognıtıon And Artıfıcıal Intellıgence **34**.
Ong, P. and Z. Zainuddin, 2019 Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. Applied Soft Computing **80**: 374–386.
Pano-Azucena, A. D., E. Tlelo-Cuautle, B. Ovilla-Martinez, L. G. de la Fraga, and R. Li, 2021 Pipeline fpga-based implementations of anns for the prediction of up to 600-steps-ahead of chaotic time series. Journal Of Cırcuıts Systems And Computers **30**.
Price, K., R. M. Storn, and J. A. Lampinen, 2006 *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.
Samanta, B., 2011 Prediction of chaotic time series using computational intelligence. Expert Systems With Applications **38**: 11406–11411.
Shi, Y., X. Liu, T. Li, X. Peng, W. Chen, *et al.*, 2010 Chaotic time series prediction using immune optimization theory. International Journal Of Computational Intelligence Systems **3**: 43–60.
Shinozaki, A., T. Miyano, and Y. Horio, 2020 Chaotic time series prediction by noisy echo state network. IEICE Nonlinear Theory and its Applications **11**: 466–479.
Shoaib, B., I. M. Qureshi, Shafqatullah, and Ihsanulhaq, 2014 Adaptive step-size modified fractional least mean square algorithm for chaotic time series prediction. Chinese Physics B **23**.
Su, L. and C. Li, 2015a Local functional coefficient autoregressive model for multistep prediction of chaotic time series. Discrete Dynamics In Nature and Society **2015**.
Su, L. and C. Li, 2015b Local prediction of chaotic time series based on polynomial coefficient autoregressive model. Mathematical Problems in Engineering **2015**.
Su, L. and F. Yang, 2021 Prediction of chaotic time series based on ben-aga model. Complexity **2021**.
Suykens, J. A. and J. Vandewalle, 1999 Least squares support vector machine classifiers. Neural processing letters **9**: 293–300.
Swinburne, R., 2004 Bayes' theorem. Revue Philosophique de la France Et de l **194**.
Tang, L.-H., Y.-L. Bai, J. Yang, and Y.-N. Lu, 2020 A hybrid prediction method based on empirical mode decomposition and multiple model fusion for chaotic time series. Chaos Solitons & Fractals **141**.
Wang, C., H. Zhang, W. Fan, and P. Ma, 2017 A new chaotic time series hybrid prediction method of wind power based on eemd-se and full-parameters continued fraction. Energy **138**: 977–990.

Wang, H. and J. Lian, 2011 Fuzzy predıctıon of chaotıc tıme serıes based on fuzzy clusterıng. Asian Journal Of Control **13**: 576–581.

Wang, H., F. Sun, Y. Cai, and Z. Zhao, 2010 Online chaotic time series prediction using unbiased composite kernel machine via cholesky factorization. Soft Computing **14**: 931–944.

Wang, R., C. Peng, J. Gao, Z. Gao, and H. Jiang, 2020a A dilated convolution network-based lstm model for multi-step prediction of chaotic time series. Computational & Applied Mathematics **39**.

Wang, Y., Z. Man, and M. Lu, 2020b Prediction of energy-efficient production of coalbed methane based on chaotic time series and bayes-least squares-support vector machine. Internatıonal Journal Of Heat And Technology **38**: 933–940.

Wolf, A., J. B. Swift, H. L. Swinney, and J. A. Vastano, 1985 Determining lyapunov exponents from a time series. Physica D: nonlinear phenomena **16**: 285–317.

Wu, X. and Z. Song, 2013 Multi-step prediction of chaotic time-series with intermittent failures based on the generalized nonlinear filtering methods. Applıed Mathematıcs And Computatıon **219**: 8584–8594.

Xiao, Y., X. Xie, Q. Li, and T. Li, 2019 Nonlinear dynamics model for social popularity prediction based on multivariate chaotic time series. Physıca A-Statıstıcal Mechanıcs And Its Applıcatıons **525**: 1259–1275.

Xin, B. and W. Peng, 2020 Prediction for chaotic time series-based ae-cnn and transfer learning. Complexity **2020**.

Xing, B. and W.-J. Gao, 2014 Fruit fly optimization algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, pp. 167–170, Springer.

Xu, M., M. Han, T. Qiu, and H. Lin, 2019 Hybrid regularized echo state network for multivariate chaotic time series prediction. IEEE Transactions on Cybernetics **49**: 2305–2315.

Yang, H., H. Ye, G. Wang, and T. Hu, 2005 Fuzzy neural very-short-term load forecasting based on chaotic dynamics reconstruction. In *Advances in Neural Networks – ISNN 2005*, edited by J. Wang, X.-F. Liao, and Z. Yi, pp. 622–627, Berlin, Heidelberg, Springer Berlin Heidelberg.

Yang, L., J. Zhang, X. Wu, Y. Zhang, and J. Li, 2016 A chaotic time series prediction model for speech signal encoding based on genetic programming. Applied Soft Computing **38**: 754–761.

Yang, X.-S. and S. Deb, 2010 Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation **1**: 330–343.

Yong, Z., 2013 New prediction of chaotic time series based on local lyapunov exponent. Chinese Physics B **22**.

Yumei, Z., B. Shulin, L. Gang, and W. Xiaojun, 2019 Kernel estimation of truncated volterra filter model based on dfp technique and its application to chaotic time series prediction. Chınese Journal of Electronıcs **28**: 127–135.

Zhang, D., 2019a Wavelet transform. In *Fundamentals of Image Data Mining*, pp. 35–44, Springer.

Zhang, D. and M. Jiang, 2020 Hetero-dimensional multitask neuroevolution for chaotic time series prediction. IEEE Access **8**: 123135–123150.

Zhang, L., 2019b Evaluating the effects of size and precision of training data on ann training performance for the prediction of chaotic time series patterns. Internatıonal Journal Of Software Scıence And Computatıonal Intellıgence-IJSSCI **11**: 16–30.

Zhang, L., F. Tian, S. Liu, L. Dang, X. Peng, *et al.*, 2013 Chaotic time series prediction of e-nose sensor drift in embedded phase space. Sensors And Actuators B-Chemıcal **182**: 71–79.

Zhang, M., B. Wang, Y. Zhou, and H. Sun, 2020 Woa-based echo state network for chaotic time series prediction. Journal of the Korean Physıcal Society **76**: 384–391.

Zhao, C., W. Ren, and M. Han, 2021 Adaptive sparse quantization kernel least mean square algorithm for online prediction of chaotic time series. Cırcuıts Systems And Sıgnal Processıng **40**: 4346–4369.

Zhao, J., Y. Li, X. Yu, and X. Zhang, 2014 Levenberg-marquardt algorithm for mackey-glass chaotic time series prediction. Discrete Dynamics In Nature and Society **2014**.

Zheng, Y., S. Wang, J. Feng, and C. K. Tse, 2016 A modified quantized kernel least mean square algorithm for prediction of chaotic time series. Digital Signal Processing **48**: 130–136.

Zhongda, T., L. Shujiang, W. Yanhong, and S. Yi, 2017 A prediction method based on wavelet transform and multiple models fusion for chaotic time series. Chaos Solitons & Fractals **98**: 158–172.

Zhou, Y.-T., Y. Fan, Z.-Y. Chen, and J.-C. Sun, 2017 Multimodality prediction of chaotic time series with sparse hard-cut em learning of the gaussian process mixture model. Chınese Physıcs Letters **34**.

How to cite this article: Martínez-García, J. A., González-Zapata, A. M., Rechy-Rechy-Ramírez, E. J., and Tlelo-Cuautle, E. On the Prediction of Chaotic Time Series using Neural Networks. Chaos Theory and Applications, 4(2), 94-103, 2022.