Prediction of chaotic time series by using ANNs, ANFIS and SVMs

[1]Ana Dalia Pano-Azucena, [1]Esteban Tlelo-Cuautle, [2]Sheldon X.-D. Tan
[1]INAOE, México and [2]University of California at Riverside, USA
ana.dalia.p.a@gmail.com, etlelo@inaoep.mx, stan@ece.ucr.edu

***Abstract*—Many biological systems and natural phenomena exhibit chaotic behaviors that are saved into time series data, which can be used to predict their future behavior. Unfortunately, chaotic time series prediction is a challenge and more difficult when the data do not have a similar pattern. In this manner, this work shows the prediction of chaotic time series that have different maximum Lyapunov exponent (MLE) values, which are generated by a chaotic oscillator, and by applying three techniques, namely: artificial neural networks (ANNs), adaptive neuro-fuzzy inference system (ANFIS), and least-square support vector machines (LS-SVM). The predicted chaotic data is compared with respect to the root mean squared error. The three prediction techniques have the same conditions and they are compared with statistical metrics to predict chaotic time series with different MLE values.**

I. Introduction

Chaotic time series prediction techniques have the challenge to forecast long horizons of chaotic data that have different entropy and maximum Lyapunov exponent (MLE) values. Those prediction techniques have the goal to provide a compact description of observed data, which must guarantee low error that can be measured as mean squared error (MSE) and root MSE (RMSE). Among the kinds of chaotic signals that require attention to predict their future behavior are for example: epilepsy, pancreatic beta cell, electroencephalogram, electrocardiogram, and so on. In those human diseases pathologies, the chaotic behavior is quite different among humans and then this is the challenge to propose a generic time series prediction technique. In this work, a chaotic oscillator based on saturated nonlinear functions (SNLF) is the case of study in which the coefficients of the mathematical model are varied to generate different chaotic time series data with different unpredictability characteristics provided by their associated MLE, which is evaluated by using the time series analysis (TISEAN) tool.

In the state of the art, several techniques have been proposed in order to solve the problem on chaotic time series prediction considering short-term and long-term prediction horizons, e.g., artificial neural networks (ANNs) [1, 2, 3], fuzzy systems [4, 5, 6], support vector machine (SVM) [7, 8], hybrid systems [9, 10, 11], and so on. Those techniques missed to predict chaotic time series with different MLE values. In this manner, this work applies ANN, an adaptive neuro-fuzzy inference system (ANFIS) and least square (LS)-SVM to predict chaotic time series with different MLE values.

ANNs are mathematical tools inspired by the human brain's ability to process information [12, 13]. In general, an ANN is a set of elementary processing units, named as neurons or nodes, and their processing capacity is stored at neural connections in synaptic weights form [14]. The general structure of an ANN is composed of an input layer that accepts external information, m hidden layers and one output layer that provides the objective value, each of these layers contains one or more neurons [15], and they can be easily implemented within field-programmable gate arrays (FPGAs) [16]. Figure 1 shows the functional structure of an artificial neuron, where x_j represents the input signal, w the synaptic weights, b the bias, and $f(.)$ the activation function that can be hyperbolic tangent function, step. sigmoid, etc. [13, 14]. The final state of a neuron is evaluated by $y = f(u) = f\left(\sum_{j=1} x_j w_j + b\right)$.

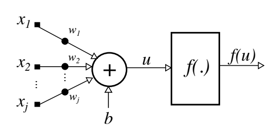


Fig. 1: Artificial neuron functional structure.

ANFIS is a type of system that incorporates ANNs and fuzzy logic. In this case, the neurons of the ANN incorporates TSK (Takagi- Sugeno-Kang) so that the system is adaptive, fault tolerant, distribute, learns from experience and presents generalizability. Assuming that the fuzzy control system has an output f, two inputs (x, y), thus the two fuzzy rules are represented by (1). Figure 2 shows the general structure of ANFIS, where the function of nodes in each layer is: Layer 1 contains input nodes regards of passing external signals to the next layer, Layer 2 performs the parameter adjust of the input membership function, Layer 3 and 4 perform fuzzy operations, Layer 5 and 6 weight and provides the output of the system.

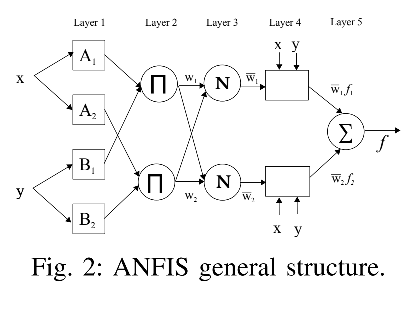


Fig. 2: ANFIS general structure.

$$
\begin{aligned}
R^1 &: \text{IF } x \text{ is } A_1 \text{ AND } y \text{ is } B_1,\ \text{THEN } f_1 = p_1 x + q_1 y + r_1 \\
R^2 &: \text{IF } x \text{ is } A_2 \text{ AND } y \text{ is } B_2,\ \text{THEN } f_2 = p_2 x + q_2 y + r_2
\end{aligned} \tag{1}
$$

The prediction problems can be solved by using classification techniques. For that reason, although SVM has been

978-1-5386-4788-2/18/$31.00 ©2018 IEEE

used in classification and regression tasks, its principles can be easily extended to prediction problems [7]. When the prediction of the time series is developed by SVMs, given a time series data y_t composed by a set of consecutive real values $y_t = \{y_t \in R \mid y_1, y_2, ..., y_n\}$, the prediction is represented by dividing $y_t = \{y_1, y_2, ..., y_n\}$ in windows $w = (y_t, ..., y_{t+p-1})$ of size p, in order to find a function $f(y_t, ..., y_{t+p-1}) = y_{t+p}$.

A characteristic of SVM is the use of Kernel functions, which are used to describe a problem in a characteristic space of higher dimension and applies linear algorithms in nonlinear problems. In this case, a data mapping from the input space X to a wide space of characteristic χ is performed through a function ϕ, $\phi : X \to \chi$, where the function ϕ does not need to be known, since it has a kernel function k that computes the dot product in the features space $k(x, y) = \phi(x) \cdot \phi(y)$. In time series prediction problems, the linear Kernel $k(x, y) = x \cdot y$ has been the most useful, where the decision function takes the form $f(x) = wx + b$ [17].

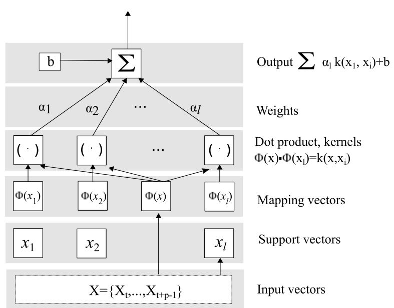


Fig. 3: SVM functional structure [18].

Among the main application that used SVM as prediction technique are control systems, environmental and meteorological prediction, applications that involve non-linear systems, electric charges, medical fields, etc. [19]. In the case of chaotic time series prediction, SVM has proved to be useful in one-step-ahead prediction and multi-step-ahead prediction problems, due to the use of a sliding window that is used to enter the input data to SVM and guarantees a global minimum solution. However, there is no predetermined heuristic for the choice of SVM parameters and designs.

We show the comparison of these three prediction techniques with respect to the root mean squared error (RMSE).

II. Generating chaotic time series

Chaos can be generated from mathematical models that can include piecewise-linear (PWL) functions, as the one one given in (2), where x_1, x_2, x_3 are the state variables, a, b, c, d_1 are positive real coefficients whose values are in the range $[0, 1]$, and $f(x_1)$ is the SNLF described by (2), this and other chaotic oscillators have been already implemented within FPGAs [16].

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= x_3 \\ \dot{x_3} &= -ax_1 - bx_2 - cx_3 + d_1 f(x_1; \alpha, k_1, h_1, p_1, q_1) \end{aligned} \qquad (2)$$

If one varies the values of the coefficients a, b, c, d_1 and $f(x_1)$ in (2), the chaotic time series can have different MLE values, as already shown in [20], where the MLE is optimized by evolutionary algorithms. For instance, Table I lists the coefficient values associated to eight different MLE values.

TABLE I: Optimized MLE for generating a 2-scrolls attractor and their associated (a, b, c, d_1) coefficient values.

a	b	c	d_1	MLE
0.7000	0.7000	0.7000	0.7000	0.1117
0.5610	0.9470	0.3460	0.6810	0.2225
1.0000	0.7000	0.7000	0.2542	0.3425
1.0000	0.7000	0.6780	0.1069	0.3437
0.7746	0.6588	0.5846	0.4931	0.3460
0.8661	1.0000	0.3934	0.9903	0.3607
1.0000	0.7884	0.6435	0.6665	0.3713
1.0000	1.0000	0.4997	1.0000	0.3761

III. Time series prediction using ANNs

A methodology that allows establishing the number of neurons and layers in an ANN is the geometric pyramid rule, where the hidden layers have fewer neurons than the input layer [21]. Three ANNs are used in this work, two of them are obtained from the geometric pyramid rule with $16\times8\times4\times2\times1$ neurons and $32\times16\times8\times4\times2\times1$ neurons. The third ANN has been proposed in [22] with $4\times7\times4\times8\times5\times1$ neurons, and it has been applied in [23]. These ANNs are feedforward ones, the input weights have a time delay line (TDL) to provide a dynamic response to the time series data, a linear activation function is used in the output layer and in the remaining layers a hyperbolic tangent activation function is used, the levenberg-Marquardt learning algorithm is used.

The ANNs training was performed with the time series of x_1 normalized in the range of [1] and with the optimized MLEs from Table I. In this case, a $x_1(t+6)$ prediction is performed by using $x_1(t), x_1(t-6), x_1(t-12), x_1(t-18)$ data as past values. The RMSE described in (3) is used to measure the prediction performance of the ANNs. Considering that the parameters of the ANNs must be adjusted, the ANN training is initialized many times with with different random assigned weights and bias.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{y_i}]^2} \qquad (3)$$

Figure 4 shows the box plots that represent the RMSE value obtained from the validation process. These plots allow to know the prediction error distribution during the ANN parameters adjustment, to detect the outliers due to an over-adjustment or sub-adjustment occurred during the training stage of the network, and to observe the difference between the weights and bias values, which provide different predictions behaviors. From Fig. 4, one can appreciated that by using an ANN with 6-layers, the RMSE distributions are the least. Once the ANNs has been trained and the parameters have adjusted, the weights and bias with lower prediction error are selected.

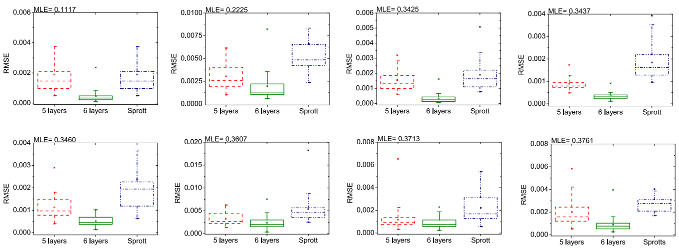


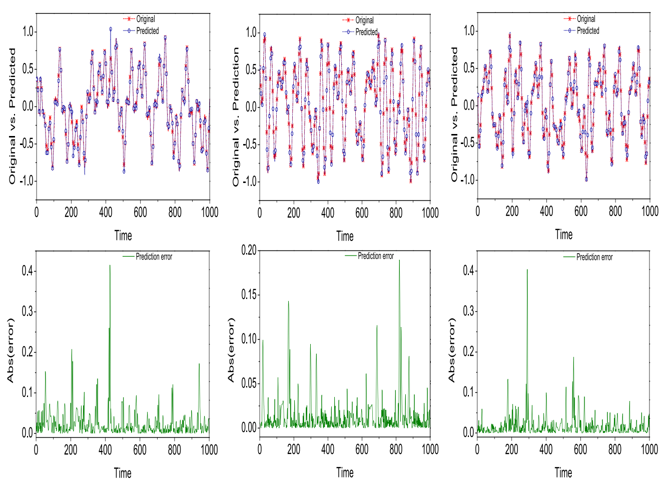
Fig. 4: RMSE box-plots for the validation process by using 8-chaotic time series with different MLE values.

IV. Time series prediction using ANFIS

The architecture shown in Fig. 2 is used to perform a prediction with $x_1(t+6)$ pass values of the time series generated by the chaotic oscillator with different MLE values. The format of the training data is:

$$x_1(t+6) = x_1(t), x_1(t-6), x_1(t-12), x_1(t-18) \quad (4)$$

where 4 inputs and $2^4 = 16$ rules are used. 2000 samples for training and 1000 samples for validation are used. Also, $genfis1$ is used to generate an initial membership function matrix (generalized bell function) from the training data. Figure 5 shows the obtained prediction after training.



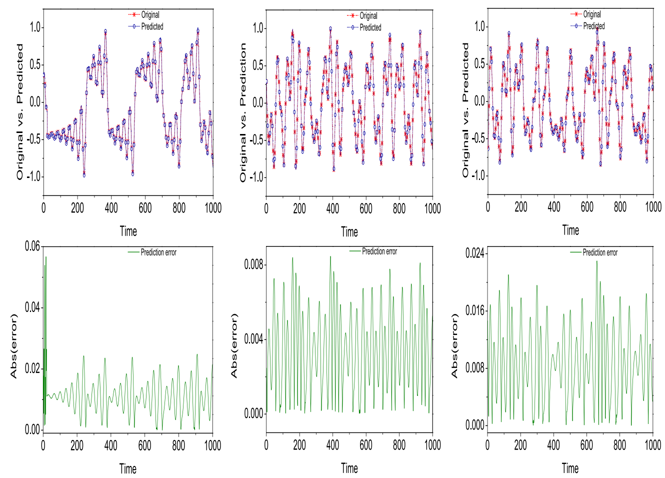
(a) (b) (c)

Fig. 5: ANFIS time series prediction with: (a) MLE=0.2225, (b) MLE=0.3425, (c) MLE=0.3761.

V. Time series prediction using LS-SVM

The chaotic time series prediction using least squares support vector machine (LS-SVM) is done through the historical value set of the time series as input and one output as objective value. The LS-SVM characteristic space has the form $f(\boldsymbol{x}) = \boldsymbol{w}^T\varphi(\boldsymbol{x}) + b$, where $\boldsymbol{x}$ is a weight matrix. The LS-SVM technique is used to predict $x_1(t+6)$. It uses 2000 training samples and 1000 validation samples. The radial base function is used as a kernel function since it provides a favorable performance under general softness assumptions.

Figure 6 shows the time series prediction results and the error. The LS-SVMlab toolbox has been used [24, 25, 26].



(a) (b) (c)

Fig. 6: LS-SVM time series prediction with: (a) MLE=0.1117, (b) MLE=0.3437, (c) MLE=0.3460.

VI. Discussion

The previous section showed the prediction behavior of the three techniques ANNs, ANFIS, LS-SVM, for the same chaotic time series and with three different MLE values. A more suitable comparison is given in Table II, which lists the statistical metrics considering the errors prediction and Fig. 7 shows a comparison of the RMSE value obtained by using different MLE values. As one can see, ANNs produce the lower prediction error. Comparing the three ANNs architectures, it is observed that the 6-layers ANNs and 5-layers ANNs topologies, both designed from the geometric pyramid rule, provide the least prediction error. However, considering that ANNs must be initialized several times with different weights and bias, it represents a limitation in terms of computer time to adjust the ANNs parameters, so it is more useful an ANN with fewer neurons like the architecture proposed in [22].

On the other hand, LS-SVM provides an adequate chaotic time series prediction and guarantees a global minimum solution. Nonetheless, the error prediction is larger than that for the ANNs, also there is not an heuristic that can help to determine the selection and construction of the kernel function.

Finally, ANFIS combines the advantages of fuzzy logic and ANNs, and the time required to perform the membership functions adjustment is lower. However, the prediction error is greater than the other techniques, and in some cases, there are useless rules that limit this prediction techniques.

VII. Conclusions

The chaotic time series are highly sensitive to the initial conditions, and a small perturbation of them modify the dynamical system behavior. Despite this, ANNs, ANFIS, and LS-SVM have proven to be effective techniques when used

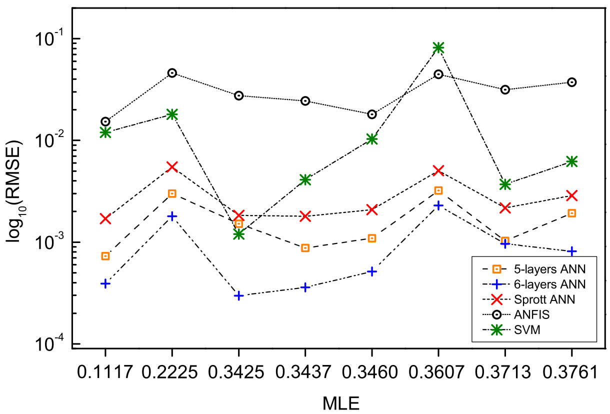


Fig. 7: Comparing the prediction techniques for the 8 different MLE values of the chaotic time series listed in Table I.

individually in chaotic time series prediction. However, choosing a prediction technique is still a challenge, and it basically depends on the prediction problem.

In this work, the simulation results, and the statistical RMSE, demonstrates the suitability of the ANNs, ANFIS, and LS-SVM to predict chaotic time series with different MLE values. In relation to the results, ANNs are the excellent choice, but the computation time must be considered to determine the appropriate topology. However, if the goal is accuracy and computing time, ANFIS and LS-SVM are the good choice.

Acknowledgment

This work is partially supported by CONACyT/Mexico under grant 237991, and UC-MEXUS under grant CN-16-161.

References

[1] N. Stamatis, D. Parthimos, and T. M. Griffith, "Forecasting chaotic cardiovascular time series with an adaptive slope multilayer perceptron neural network," *IEEE transactions on biomedical engineering*, vol. 46, no. 12, pp. 1441–1453, 1999.

[2] M. Ishikawa and T. Moriyama, "Prediction of time series by a structural learning of neural networks," *Fuzzy Sets and Systems*, vol. 82, no. 2, 1996.

[3] S. Aras and İ. D. Kocakoç, "A new model selection strategy in time series forecasting with artificial neural networks: Ihts," *Neurocomputing*, vol. 174, pp. 974–987, 2016.

[4] F. M. Pouzols, A. Lendasse, and A. B. Barros, "Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation," *Fuzzy Sets and Systems*, vol. 161, no. 4, pp. 471–497, 2010.

[5] F. Di Martino, V. Loia, and S. Sessa, "Fuzzy transforms method in prediction data analysis," *Fuzzy Sets and Systems*, vol. 180, no. 1, pp. 146–163, 2011.

[6] P. Singh and B. Borah, "High-order fuzzy-neuro expert system for time series forecasting," *Knowledge-Based Systems*, vol. 46, pp. 12–21, 2013.

[7] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, and L. Buydens, "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35–49, 2003.

[8] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[9] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid elman–narx neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540–2553, 2010.

[10] Y. Bodyanskiy and O. Vynokurova, "Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification," *Information Sciences*, vol. 220, pp. 170–179, 2013.

[11] A. Bagheri, H. M. Peyhani, and M. Akbari, "Financial forecasting using anfis networks with quantum-behaved particle swarm optimization," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6235–6250, 2014.

[12] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on power systems*, vol. 16, no. 1, pp. 44–55, 2001.

[13] C. Lin and C. Lee, "Neural fuzzy systems: A neural-fuzzy synergism to intelligent systemsprentice-hall," *Englewood Cliffs, NJ*, 1996.

[14] J. D. Rairán Antolines, "Reconstruction of periodic signals using neural networks," *Tecnura*, vol. 18, no. 39, pp. 34–46, 2014.

[15] G. K. Jha and K. Sinha, "Time-delay neural networks for time series prediction: an application to the monthly wholesale price of oilseeds in india," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 563–571, 2014.

[16] E. Tlelo-Cuautle, L. de la Fraga, and J. Rangel-Magdaleno, "Engineering applications of fpgas," 2016.

[17] S. Rüping, "Svm kernels for time series analysis," Universitätsbibliothek Dortmund, Tech. Rep., 2001.

[18] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, May 2009.

[19] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[20] V. H. Carbajal-Gmez, E. Tlelo-Cuautle, and F. V. Fernndez, "Application of computational intelligence techniques to maximize unpredictability in multiscroll chaotic oscillators," in *Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design*. Springer, 2015, pp. 59–81.

[21] T. Masters, *Practical neural network recipes in C++*. Morgan Kaufmann, 1993.

[22] M. Molaie, R. Falahian, S. Gharibzadeh, S. Jafari, and J. C. Sprott, "Artificial neural networks: powerful tools for modeling chaotic behavior in the nervous system," *Frontiers in computational neuroscience*, vol. 8, p. 40, 2014.

[23] A. Pano-Azucena, E. Tlelo-Cuautle, L. de la Fraga, C. Sanchez-Lopez, J. Rangel-Magdaleno, and S. X.-D. Tan, "Prediction of chaotic time-series with different mle values using fpga-based anns," in *14th Int. Conf. on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2017, pp. 1–4.

[24] M. W. Fakhr, "Sparse locally linear and neighbor embedding for nonlinear time series prediction," in *2015 Tenth International Conference on Computer Engineering Systems (ICCES)*, Dec 2015, pp. 371–377.

[25] "Matlab ls-svmlab toolbox for robust least squares svm regression." [Online]. Available: http://www.esat.kuleuven.be/sista/lssvmlab/

[26] Y. Mei-Ying and W. Xiao-Dong, "Chaotic time series prediction using least squares support vector machines," *Chinese Physics*, vol. 13, no. 4, p. 454, 2004.

TABLE II: Statistical metric values obtained during the validation process.

	ANN						ANFIS		LS-SVM	
	5 layers		6 layers		Sprott					
MLE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE
0.1117	7.31E-04	6.18E-07	3.91E-04	2.16E-07	1.70E-03	3.79E-06	0.0153	2.3544E-4	0.0120	1.4488E-4
0.2225	3.00E-03	1.06E-05	1.80E-03	4.57E-06	5.49E-03	3.43E-05	0.0460	0.0021	0.0180	3.2570E-4
0.3425	1.52E-03	2.74E-06	2.98E-04	1.41E-07	1.83E-03	3.92E-06	0.0276	7.6074E-4	0.0012	1.4701E-6
0.3437	8.78E-04	8.36E-07	3.59E-04	1.52E-07	1.80E-03	3.63E-06	0.0244	5.9761E-4	0.0041	1.6982E-5
0.3460	1.09E-03	1.31E-06	5.14E-04	3.05E-07	2.09E-03	5.29E-06	0.0180	3.2487E-4	0.0103	1.0568E-4
0.3607	3.22E-03	1.22E-05	2.29E-03	6.22E-06	5.03E-03	2.95E-05	0.0446	0.0020	0.0816	0.0067
0.3713	1.03E-03	1.19E-06	9.63E-04	1.22E-06	2.17E-03	6.26E-06	0.0315	9.8992E-4	0.0037	1.3424E-5
0.3761	1.93E-03	4.90E-06	8.14E-04	8.01E-07	2.87E-03	9.02E-06	0.0373	0.0014	0.0062	3.8758E-5