Chaotic Time Series Prediction using Improved ANFIS with Imperialist Competitive Learning Algorithm

Maysam Behmanesh, Majid Mohammadi, Vahid Sattari Naeini

Abstract—This paper presents an improved adaptive Neuro-fuzzy inference system (ANFIS) for predicting chaotic time series. The previous learning algorithms of ANFIS emphasized on gradient based methods or least squares (LS) based methods, but gradient computations are very computationally and difficult in each stage, also gradient based algorithms may be trapped into local optimum. This paper introduces a new hybrid learning algorithm based on imperialist competitive algorithm (ICA) for training the antecedent part and least square estimation (LSE) method for optimizing the conclusion part of ANFIS. This hybrid method is free of derivation and solves the trouble of falling in a local optimum in the gradient based algorithm for training the antecedent part. The proposed approach is used in order to modeling and prediction of three benchmark chaotic time series. Analysis of the prediction results and comparisons with recent and old studies demonstrates the promising performance of the proposed approach for modeling and prediction of nonlinear and chaotic time series.

Index Terms—chaotic time series, Gradient based, imperialist competitive algorithm, Fuzzy systems, ANFIS, least square estimation.

I. INTRODUCTION

Time series prediction is one of the most important prediction that collect past observations of a variable and analysis it to obtain the underlying relationships between observations and finding out a descriptive example. Then this model will be applied to extrapolate future time series. Prediction of time series has widespread in the areas of science, technology, medicine and Econometrics, among others. In general, time series can have some properties such as nonlinearity, chaotic, non-stationary and cyclic. Among the various types of time series chaotic time series can be typically found in natural phenomena [1], [2]. Anticipation of the behavior of chaotic time series as a nonlinear dynamic system is the case of a comparatively new research that has drawn the attention and efforts of many scientists. In the expanse of time series has been given many classical and statistical approaches, but those methods were often complex and not accurate and efficient in the encounter of chaotic and very large data [3], [4]. In recent years, many new prediction approaches, such as the wavelet networks [5], [6], neural networks [7], [8], fuzzy [9], [10] and Neuro-fuzzy systems [11]-[15] and evolutionary algorithms [16]-[19] have emerged.

One technique for modeling nonlinear systems is the integration of fuzzy logic and neural networks known as Neuro-fuzzy. In Neuro-fuzzy approach the capability of fuzzy-rules based systems in handling uncertain and noisy data and the learning capability of neural networks is aggregated to form better estimators. Many different structures for fuzzy neural networks (FNNs) have been suggested. A special form of Neuro-fuzzy systems is ANFIS, which has demonstrated significant effects in modeling nonlinear functions. ANFIS is an adaptive neural network based fuzzy inference system that learning processes are performed by interleaving the optimization of the antecedent and conclusion parts parameters. Many types of Neuro-fuzzy systems have been proposed along with their learning algorithms, such as gradient based learning algorithms [20], genetic algorithms (GAs) and evolutionary based [21]. But, most current studies are not hybrid and there are some troubles during training. Training this network in antecedent part is more difficult than consequent part, because most methods for training antecedent part are based on gradient that gradient computation is really difficult in each stage and also may be falling in a local optimum. In this paper a new hybrid learning algorithm based on an imperialist competitive algorithm (ICA) and least square estimation (LSE) has been employed to train ANFIS and adjust its antecedent and consequent parameters. This algorithm is free of derivation which is really difficult to calculate for training antecedent part parameters, also complexity of this algorithm is less than other training algorithms. This new learning algorithm has been used in order to improve ANFIS performance and increasing its accuracy in prediction of chaotic time series. The simulation results of the new method for three types of chaotic time-series and models such as Mackey-Glass, Lorenz and Rossler models are compared with previous methods based on the Hyken criteria [48] to prove the effectiveness of the new method in prediction accuracy. The remainder of the article is organized as follows. In section II imperialist competitive algorithm (ICA) is discussed. Section III describes the ANFIS and its learning algorithms. Section IV describes the determination of ANFIS parameters using new learning method and uses this proposed model for time series prediction. The prediction results of three benchmark time series, including the Mackey–Glass time series, Lorenz and Rossler model are provided in section V. Finally, conclusions are provided in the final section.

Manuscript Received on September 2014.

Maysam Behmanesh, Department of Computer Engineering, Shahid Bahonar University of Kerman, Iran.

Majid Mohammadi, Department of Computer Engineering, Shahid Bahonar University of Kerman, Iran.

Vahid Sattari Naeini, Department of Computer Engineering, Shahid Bahonar University of Kerman, Iran.
II. IMPERIALIST COMPETITIVE ALGORITHM

Imperialist Competitive Algorithm (ICA) is a new social-politically motivated global search strategy that has recently been introduced for dealing with different optimization tasks (Atashpaz and Lucas, 2007) [22], [23]. Fig. 1 shows the flowchart of the ICA. Similar to other evolutionary algorithms, this algorithm starts with an initial population and objective function is that calculate for all. Each individual of the population is called a country. Some of the best countries (in optimization terminology, countries with the least cost) are chosen to be the imperialist and the rest form the colonies of these imperialists. Then competition takes place between the colonists for taking colonies. All the colonies of initial countries are divided among the mentioned imperialists based on their power so that strongest imperialist has a greater chance for further colonies. Then each imperialist with their colonies forms an empire.

Fig. 2 shows the initial empires. In this figure imperialist 1 has formed the most powerful empire and consequently has the greatest number of colonies.

![Image of initial empires](image)

Fig. 2 Generating the Initial Empires: The More Colonies an Imperialist Possess, the Bigger is its Relevant Mark

After dividing colonies between imperialists, these colonies approach their related imperialist countries. Fig. 3 represents this movement. Based on this concept, each colony moves toward the imperialist by a unit and reaches its new position. Where $a$ is a random variable with uniform (or any proper) distribution, $\beta$ a number greater than 1, causes colonies move toward their imperialists from different direction and $d$ is the distance between colony and imperialist. An appropriate selection is $\beta = 2$ that $\beta > 1$ causes the colonies to get closer to the imperialist state from both sides.

$$a \sim U(0, \beta \times d)$$  \hspace{1cm} (1)

Fig. 3 Movement of Colonies Toward their Relevant Imperialist

In this paper, in order to increase the convergence speed and accuracy ICA, we used a modified ICA algorithm. To change this algorithm and increase diversity in population and increase the power of searching more area around the imperialist, a random amount of deviation is added to the direction of movement to modify the introduced algorithm. Fig. 4 shows the new direction. Namely instead of moves toward the imperialist by $\theta$ units with the same amount moves toward the imperialist but has $\theta$ deviation in the direction of movement. $\theta$ is a parameter with uniform distribution.

$$\theta \sim U(-\gamma, \gamma)$$  \hspace{1cm} (2)

Increasing $\gamma$ causes search enhanced in the around of imperialist and with reducing, colonies moves near the vector between colony and imperialist. Premature
convergence may occur in different conditions: population will converge to a local optimum or algorithm go ahead slowly. In order to avoid from a fall in a local optimum and premature convergence and increase diversity of the population we use the revolution operator in ICA. In order to implementation of revolution operator and increasing the exploration, we allow each colony to change arbitrarily in some features. With apply this operator colonies obtains completely new position which increases the exploration of the search space. In this paper revolution operator apply randomly for some colonies after assimilation operator, in addition, this operator will also apply randomly to imperialists.

Fig. 4 Movement of Colonies Toward their Relevant Imperialist in a Randomly Deviated Direction

After moving toward the imperialist, a colony might reach to a place with better position than the imperialist (when cost in the location of the colony is lower than the cost in the location of imperialists). In this instance, the imperialist and the colony change their positions, then the algorithm will continue by the imperialist in the new location (intergroup competition). After this step total power of empire is calculated. The total power of an empire is mainly affected by the power of imperialist country. However the power of the colonies of an empire has an effect, albeit negligible, on the total power of that empire. Total power of empire $n$ is defined by equation 3.

$$Cost(Emp_{i}) = Cost(Imperialist_{i}) + \xi \times \text{mean}\{Cost\{Colonies\ of\ empire_{i}\}\}$$

That $\xi$ is a positive small number and generally considered to be between zero and one and near zero. In the following imperialistic are competing with based on the criteria of total power of empire. In the imperialistic competition process, all empires try to take the possession of colonies of other empires and control them. The imperialistic competition gradually brings about a reduction in the power of weaker empires and an increase in the power of more powerful ones, which is modeled by just picking some (usually one) of the weakest colonies of the weakest empires and making a competition among all empires to possess these (this) colonies. Also when an empire loses all of its colonies, it is assumed to be collapsed. In this method for selecting an empire for getting colony we use a roulette wheel method. Also in this paper, the collapsed empire also competed with other empires. After a while, all the empires except the most powerful one will collapse and all the colonies will be under the ascendancy of this unique empire. Ideally, all the colonies have the same positions and the same costs and they are controlled by an imperialist with the same position and costs as themselves, which means the algorithm converges to the best solution.

III. THE CONCEPT OF ANFIS

A. ANFIS Structure

In this section, type III ANFIS topology and the learning method that used for this Neuro-fuzzy networks are presented. Both neural network and fuzzy logic are model-free estimators and share the mutual ability to deal with uncertainties and noise. The ANFIS combines two approaches: neural networks and fuzzy systems. If both these two intelligent approaches are combined, good reasoning will be achieved in quality and quantity. In other words, both fuzzy reasoning and network calculation will be available simultaneously. The ANFIS is composed of two parts. The first is the antecedent part and the second is the conclusion part, which are connected to each other with the fuzzy rules base in network form. The type III ANFIS with tow inputs structure shown in Fig. 5. As shown in this figure, it is a five layer network that can be described as a multi-layered neural network [24].

Fig. 5 The Equivalent Structure of ANFIS (Type III ANFIS) with Two Inputs and One Output

The first layer executes a fuzzification process, the second layer executes the fuzzy AND of the antecedent part of the fuzzy rules, the third layer normalizes the MFs, the fourth layer executes the conclusion part of the fuzzy rules, and the last layer computes the output of the fuzzy system by summing up the outputs of layer four. The feedforward equations of the ANFIS structure with two inputs and two labels for each input shown in Fig. 5 according to type III rules are as follows:

$$w_i = \mu_{a_i}(x) \times \mu_{b_i}(y), i = 1, 2$$

$$w_i^* = \frac{w_i}{w_1 + w_2}, i = 1, 2$$
ANFIS has high ability of approximation that will depend on the resolution of the input space partitioning, which is determined by the number of MFs in the antecedent part for each input. In this paper, the MFs are used as Gaussian MF that \( m \) represents the center and \( \sigma \) determines the width of the MF respectively. Such as:

\[
\mu_a(x) = e^{-\frac{(x-a)^2}{2\sigma^2}}
\]  

B. Learning Algorithms

The learning algorithm of an ANFIS is to determine the parameters \((a_i, b_i, c_i)\) such that the error between the ANFIS output and the actual data can be minimized. Subsequent to the development of ANFIS approach, a number of methods have been proposed for training its parameters. For example, Mascioli et al. [25] have proposed to merge Min–Max and ANFIS model to obtain Neuro-fuzzy network and determine the optimal set of fuzzy rules. Jang and Mizutani [26] have presented the application of Levenberg–Marquardt method, which is essentially a nonlinear least squares technique, for training the ANFIS network structure. In another paper, Jang [27] has presented a system for input selection and Kumar and Garg [28] have used the Kohonen’s map for training. Jang in his famous article [24] introduced four methods to update the parameters of the ANFIS structure, as listed below according to their computation complexities:

1. GD only: all parameters are updated by the GD.
2. GD only and one pass of LSE: the LSE is applied only once at the very beginning to get the initial values of the conclusion parameters and then the GD takes over to update all parameters.
3. GD only and LSE: this is the hybrid learning.
4. Sequential LSE: using extended Kalman filter to update all parameters.

These methods update antecedent parameters by using GD or Kalman filtering and due to the good performance still applies. Methods that have been proposed were gradient based and due to the proper performance also methods based on least squares as effective ways to optimize the parameters of the consequent part still applies. Training of antecedent parameters is very important. But this technique due to the high complexity in the gradient computation and the nonlinear presence of antecedent parameters in the output is not suitable. To resolve this problem population-based approaches such as genetic algorithm, ACO and other related methods can be raised [29], [30]. Many works have proposed train Neuro-fuzzy networks by combining evolutionary algorithms and gradient descent methods, least squares and Kalman filter. For example, use of hybrid optimization method like PSO for the antecedent part and GD for the conclusion part in [31], [32]. In [33] training of antecedent and conclusion part done by PSO and LSE respectively. This paper proposed new method for determining the parameters of ANFIS that has less complexity and more accuracy and training of antecedent and consequent parameters are done by ICA and LSE algorithms respectively in a repetitive process.

IV. PROPOSED APPROACH

The algorithm that was used for the proposed method is described in this section step-by-step.

**First Step:** Create a matrix with time series data. A chaotic time series generally exhibits stochastic characteristics in time or frequency domain. However, based on Taken theory in phase-space reconstruction in time series and by using the coordinates with appropriate embedding dimension \( D \) and time delay \( \tau \), a quasiperiodic attractor in the phase space can be derived [34]. Consider a time series \( \{x(1),x(2),...,x(N)\} \).

An embedded phase vector \( u(i) \) is:

\[
u(i) = [x(i), x(i - \tau), \ldots, x(i - (D - 1)\tau)]
\]

Where \( \forall i \in [1, (D - 1)\tau, N] \), \( D \) is the embedding dimension, \( \tau \) is the time delay and \( u(i) \) is a vector in the D-dimensional phase space \( R^D \). A trajectory in \( R^D \) is defined as:

\[
U = [u^T(i), u^T(i + 1), \ldots, u^T(i + m)]
\]

In order to extract the behaviors of the time series in an efficient way, optimal values of \( D \) and \( \tau \) have to be determined. In this paper, \( \tau \) and \( D \) are determined by using the average mutual information in [35], and the global false nearest neighbors in [1], respectively.

**Second Step:** Get the time series matrix to the entrance of the ANFIS structure. The inputs of ANFIS is a matrix in equation (9) with the embedding phase vector as the columns and the chaotic time series as the rows. Its dimension is \( D \times (m + 1) \), where \( i \) is an integer, \( i \in [1, (D - 1)\tau, N - m - k] \) and \( k \) is the number of prediction steps.

**Third Step:** Training ANFIS with data obtained in both time domain and phase space in an iterative and hybrid procedure by ICA and LSE algorithms. This combined training process is used to tune the antecedent and consequent parameters in order to achieve accurate prediction. Whenever the number of steps performed or desired training error is reached the process stops. After defining the training data, checking data, number of training epochs and type of membership functions, the optimal values for the antecedent parameters are determined by ICA in each step and after this, consequent parameters of improved ANFIS are estimated by LSE. This algorithm is performed in seven steps.

Step 1: Creating the initial empires.

1. Initializing the membership functions of the input parameters \((m, \sigma)\) randomly and then based on those, initial estimates of the consequent parameters \((a_i, b_i, c_i)\) by LSE.
2. Calculate the fitness countries with MSE criterion in equation (10).
3. Select the best countries as imperialists and the remainder as the colonies of these imperialists. In this algorithm, SUS method used to select the colonies of imperialists.
Step 2: Apply assimilation and revolution operators. On each iteration, one of the parameters of membership function is being updated. In other words, in the first iteration, for example, \( m_i \) is updated, then in the second iteration, \( \sigma_i \) is updated. Then after updating all parameters again, the first parameter update is considered and so on. These parameters are grouped in a vector that is being updated iteration to iteration. With any change in the antecedent parameters, consequent parameters (\( a, b, c \)) are estimated by LSE.

Step 3: Intergroup competition. If after applying operators, prediction error in ANFIS with parameters of one of the colonies is lower than the ANFIS with parameters of imperialist, then the imperialist and the colony change their positions and this algorithm will continue by the imperialist in the new location.

Step 4: Calculate the total power of the empires. The total power of the empire is calculated based on equation (3). In this paper, our purpose is minimizing the difference between ANFIS output and the actual output, thus the cost function is specified for each empire as the objective function with an equation (10).

\[
\text{Cost (colony)} = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 
\]

Step 5: Imperialist competition. Empires are competing based on total power of each empire. This step begins by picking some of the weakest colonies of the weakest empires and making a competition among all empires to possess these colonies. With this method the collapsed empires also compete with other empires.

Step 6: Choice best imperialist of empires as solution of antecedent parameter in ANFIS and estimate consequent parameters with LSE. In this method, in order to avoid overfitting is used validation data that is independent of the training data.

Step 7: The algorithm terminates. If the termination condition is satisfied or the prediction error is obtained, then the algorithm will terminate.

V. EXPERIMENTAL RESULTS

In order to assess the proposed improver ANFIS-based ICA and LSE hybrid learning algorithm, comprehensive experiments and simulations based on three nonlinear and chaotic benchmark time series were conducted. It must be noticed that, in all case studies, amount the last 30% of the training data is utilized as the validation data to select best model structure. The validation data are not utilized during the training, and the model with lowest validation error is selected as the best model, which is applied to predict test data. For numerical assessment of the prediction accuracy, the following error criteria are applied.

1. Root-mean-square error (RMSE)
2. Normalized mean-square error (NMSE)

\[
RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y(n) - \hat{y}(n))^2}
\]

\[
NMSE = \frac{1}{N} \sum_{n=1}^{N} \frac{(y(n) - \hat{y}(n))^2}{(y(n) - \bar{y})^2}, \quad \bar{y} = \frac{1}{N} \sum_{n=1}^{N} y(n)
\]

A. Prediction of Mackey–Glass Time Series

Mackey–Glass time series is generated using the following time-delay differential equation, introduced as a model for white blood cell production:

\[
\frac{dx}{dt} = \frac{0.2}{1 + x^m} (t - \lambda) - 0.1 x(t)
\]

The time series is very sensitive to the initial conditions and this time series is chaotic for \( \lambda \geq 16.8 \) and exhibits no clearly defined period. The standard input variables in this case for phase-space reconstruction are \( \tau = 6 \) and \( D = 4 \) are determined by using the average mutual information [35], and the global false nearest neighbors [1], respectively. Therefore, the phase space is reconstructed as \( x(t - 18), x(t - 12), x(t - 6) \) and \( x(t) \) for predicting \( x(t + 6) \) (i.e., this is a case of six-step-ahead prediction). The proposed model for prediction this time series trained with the parameters listed in Table I. For this case study, convergence diagram shown in Fig. 6. The actual and predicted test series is shown in Fig. 7 and prediction error in test time series is shown in Fig. 8.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Type or Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clusters</td>
<td>20</td>
</tr>
<tr>
<td>Countries</td>
<td>50</td>
</tr>
<tr>
<td>Empires</td>
<td>10</td>
</tr>
<tr>
<td>Epochs</td>
<td>200</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>2</td>
</tr>
<tr>
<td>( \beta )</td>
<td>2</td>
</tr>
<tr>
<td>Revolution Probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Revolution Rate (( \mu ))</td>
<td>0.05</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Fig. 6 Convergence Diagram for the Mackey–Glass Time Series
In order to numerical evaluation and for the purpose of comparison, the training and test RMSEs of the proposed approach and a number of methods available in the literature are presented in Table II.

**TABLE II. PERFORMANCE COMPARISON FOR MACKAY–GLASS TIME SERIES PREDICTION**

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE Train</th>
<th>RMSE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-regressive model [36]</td>
<td>-</td>
<td>0.19</td>
</tr>
<tr>
<td>Cascade correlation NN [36]</td>
<td>-</td>
<td>0.06</td>
</tr>
<tr>
<td>Back propagation NN [36]</td>
<td>-</td>
<td>0.02</td>
</tr>
<tr>
<td>Linear prediction method [36]</td>
<td>-</td>
<td>0.55</td>
</tr>
<tr>
<td>Product T-norm [37]</td>
<td>-</td>
<td>0.09</td>
</tr>
<tr>
<td>Classical RBF (with 23 neurons) [38]</td>
<td>-</td>
<td>0.0114</td>
</tr>
<tr>
<td>PG-RBF network [39]</td>
<td>-</td>
<td>0.0028</td>
</tr>
<tr>
<td>Genetic algorithm and fuzzy system [40]</td>
<td>-</td>
<td>0.049</td>
</tr>
<tr>
<td>Neural tree model [41]</td>
<td>-</td>
<td>0.0069</td>
</tr>
<tr>
<td>WNN [37]+gradient</td>
<td>0.0067</td>
<td>0.0071</td>
</tr>
<tr>
<td>LLWNN [36]+gradient</td>
<td>0.0038</td>
<td>0.0041</td>
</tr>
<tr>
<td>LLWNN [36]+hybrid</td>
<td>0.0033</td>
<td>0.0036</td>
</tr>
<tr>
<td>Recurrent ANFIS [42]</td>
<td>-</td>
<td>0.0013</td>
</tr>
<tr>
<td>ANFIS [24]</td>
<td>-</td>
<td>0.00156</td>
</tr>
<tr>
<td>RBF network [43]</td>
<td>-</td>
<td>0.0015</td>
</tr>
<tr>
<td>Ensembles of ANFIS [44]</td>
<td>-</td>
<td>0.0025</td>
</tr>
</tbody>
</table>

In order of numerical evaluation and for the purpose of comparison, the training and test NMSE of the proposed approach and a number of methods available in the literature are presented in Table III. Based on the presented results in Table III, the proposed approach has much better performance over all other compared methods for both training and test data.
Rossler model is a system of three ordinary differential equations which define a continuous dynamical system that exhibits chaotic dynamics associated with the fractal properties of the Rossler attractor. This map is depicted by the following differential equations:

\[
\begin{align*}
\dot{x} &= -y - z \\
\dot{y} &= x - ay \\
\dot{z} &= b + z (x - c)
\end{align*}
\]

Where \( x, y \) and \( z \) real functions of time and the attractor are shows chaotic behavior for \( a = 0.2, b = 0.2 \) and \( c = 4.6 \). The standard input variables in this case for phase-space reconstruction are \( \tau = 2 \) and \( D = 3 \). Thus, the phase space is reconstructed as \( x(t-4), x(t-2), x(t) \) for predicting \( x(t+2) \) (i.e., this is a case of tow-step-ahead prediction).

The proposed model for prediction this time series trained with the parameters listed in Table I. Convergence diagram shown in Fig. 12. The target and predicted test data of Rossler model are shown in Fig. 13, and prediction error in test time series are shown in Fig. 14. In order to numerical evaluation and for the purpose of comparison, the training and test NMSE of the proposed approach and a number of methods available in the literature are presented in Table IV. Results that presented in this table demonstrating successful performance of the proposed hybrid approach in capturing and modeling the dynamical behavior of this nonlinear chaotic system.

**TABLE III. PERFORMANCE COMPARISON FOR LORENZ MODEL TIME SERIES PREDICTION**

<table>
<thead>
<tr>
<th>Method</th>
<th>NMSE Train</th>
<th>NMSE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP-EKF [47]</td>
<td>0.00023</td>
<td>0.00162</td>
</tr>
<tr>
<td>MLP-BLM [47]</td>
<td>0.00033</td>
<td>0.00096</td>
</tr>
<tr>
<td>RNN-BPTT [47]</td>
<td>0.00056</td>
<td>0.00185</td>
</tr>
<tr>
<td>RNN-RTRL [47]</td>
<td>0.00057</td>
<td>0.00172</td>
</tr>
<tr>
<td>RNN-EKF [47]</td>
<td>0.00036</td>
<td>0.00121</td>
</tr>
<tr>
<td>RBLM-RNN [47]</td>
<td>0.00036</td>
<td>0.00090</td>
</tr>
<tr>
<td>ANFIS[24]</td>
<td>0.0026</td>
<td>0.0021</td>
</tr>
<tr>
<td>Fuzzy prediction based on SVD</td>
<td>-</td>
<td>0.0106</td>
</tr>
<tr>
<td>[10]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS-SVM based on PSO [45]</td>
<td>-</td>
<td>0.00018</td>
</tr>
<tr>
<td>[46]</td>
<td>0.00013</td>
<td>0.00029</td>
</tr>
<tr>
<td>Proposed Approach</td>
<td>0.00007</td>
<td>0.00012</td>
</tr>
</tbody>
</table>

**B. Prediction of Rossler Model**

Rossler model is a system of three ordinary differential equations which define a continuous dynamical system that exhibits chaotic dynamics associated with the fractal properties of the Rossler attractor. This map is depicted by the following differential equations:
TABLE IV. PERFORMANCE COMPARISON FOR ROSSLER MODEL TIME SERIES PREDICTION

<table>
<thead>
<tr>
<th>Method</th>
<th>NMSE Train</th>
<th>NMSE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP-EKF [47]</td>
<td>0.00025</td>
<td>0.00193</td>
</tr>
<tr>
<td>MLP-BLM [47]</td>
<td>0.00047</td>
<td>0.00101</td>
</tr>
<tr>
<td>RNN-BPTT [47]</td>
<td>0.00070</td>
<td>0.00311</td>
</tr>
<tr>
<td>RNN-TRL [47]</td>
<td>0.00071</td>
<td>0.00312</td>
</tr>
<tr>
<td>RNN-EKF [47]</td>
<td>0.00060</td>
<td>0.00191</td>
</tr>
<tr>
<td>RBLM-RNN [47]</td>
<td>0.00057</td>
<td>0.00092</td>
</tr>
<tr>
<td>ANFIS[24]</td>
<td>0.0147</td>
<td>0.0118</td>
</tr>
<tr>
<td>LLNF[24]</td>
<td>0.000048</td>
<td>0.000071</td>
</tr>
<tr>
<td>Proposed Approach</td>
<td>0.00083</td>
<td>0.00010</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

This paper proposes an improved version of ANFIS with imperialist competitive learning algorithm for chaotic time series prediction. This version of ANFIS improves the ability of ANFIS in learning parameters with utilize a new hybrid learning algorithm for training its parameters. This hybrid learning algorithm used ICA for updating the antecedent parameters and LSE for estimating consequent parameters in an iterative process. This new learning algorithm has comparable performance with less parameters than gradient based methods which means that the new approach demonstrates high capability in the learning process. Also the complexity of this new algorithm is less than the other training algorithms and free of derivation and with this algorithm the local optimum problem in the gradient based algorithm for training the antecedent part is solved.

The proposed approach was used for modeling and predicting a wide range of different nonlinear time series and chaotic systems and processes. The investigated time series included Mackey–Glass time series, Lorenz model, and Rossler model. The first series are recognized as real-world nonlinear time series, while the last two are known to be dynamical systems with chaotic behaviors. The obtained prediction results and the comprehensive comparisons with some recently published studies revealed the outstanding performance of the proposed approach in modeling and prediction of nonlinear, chaotic, and complex time series and processes.

REFERENCES