Bulletin of the *Transilvania* University of Braşov • Vol 12(61), No. 2 - 2019
Series III: Mathematics, Informatics, Physics, 469-480
https://doi.org/10.31926/but.mif.2019.12.61.2.24

CHAOTIC TIME SERIES PREDICTION WITH MULTI–LAYER PERCEPTRON AND META–HEURISTIC METHODS: A REIVEW

Marjan KUCHAKI RAFSANJANI[*1] **and Meisam SAMAREH GHASEM**[2]

Abstract

Predicting chaotic time series is an applicable issue, so that many scientists have introduced different methods to predict their behavior. Artificial neural networks are a tool that forecasts system behavior. These tools should be trained and back propagation algorithms used as learner. But this training process may be falling into trap of local optimum. Heuristic methods are introduced to solve this challenge.

2000 *Mathematics Subject Classification:* 34C28, 92B20, 90C31, 68T20
Key words: Chaotic time series prediction, Multi–layer Perceptron, Optimization, Meta–heuristic methods, Takens theorem.

1 Introduction

Time series are created from a set of data points that occurred over time. There are two types of time series, including continuous and discrete. In continuous time series, time is continuous and between any two points there are an infinite number of points. In discrete time series, observations occur at certain time points and often time intervals are equal [1, 2]. Chaotic time series have complex behavior, and their analysis is difficult and these commonly used for modeling natural behaviors [3, 4]. Chaotic time series used in different science, such as financial, economic, traffic and weather [4, 5, 6, 7].
Many traditional methods have been tested for time series prediction [8, 9]. Intelligent methods have been used in chaotic time series prediction and have had good results. Artificial neural networks (ANNs) are a class of intelligent methods that have been inspired from human brain. Multi–layer perceptron (MLP) is one

[1*] *Corresponding author*, Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran, e-mail: kuchaki@uk.ac.ir
[2] Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran, e-mail: meisam.samareh@gmail.com

of famous ANN that has been used in some researches [7], some of the other papers have used radial basis functions (RBFs) [10] and recurrent neural network (RNN) [11, 12]. Meta–heuristic algorithms, support vector machine (SVM) and fuzzy inference system (FIS) have been used as predictor [3, 13, 14, 15, 16, 17].
ANNs are power classifiers, but these methods have also special problems. One of the problems of ANNs is training process, gradient based methods have been proposed to solve it. Gradient based methods usually trap in local optimum so that training process becomes a big challenge [4, 18]. Meta–heuristic algorithms are used to train ANNs that means tuning of weights and biases of ANNs are introduced as an optimization problem [18, 19, 20, 21, 22, 23].
This paper is organized as follows: a review on artificial neural networks and especially on multi–layer perceptron is in section 2. Section 3 describes two heuristic methods, including particle swarm optimization and genetic algorithm. Section 4 reviews chaotic time series prediction. Section 5 explains how a neural network trained with heuristic methods. In section 6 dataset, tools and evaluation criteria are introduced and the results compared and finally conclusion is in section 7.

2 Artificial Neural Networks (ANNs)

An ANN includes simple components which have a local behavior, but a set of them have a global act, this global model is very practical. For example, these act as function approximation, classification and clustering, modeling and prediction [24, 25, 26, 27, 28] and etc.

2.1 Multi–Layer Perceptron

Multi–layer perceptron (MLP) is very popular because they capable to classify non–linearly separable problems and this feature increases their generalization. The architecture of MLP is feed–forward and learning algorithm is back–propagation. This architecture has three general units. First, input layer receives problem inputs where each input is called a neuron. Next module is known hidden layer that is a bridge between input and output layer. The last module is named output layer which gives outputs. Each neuron in first module should be connected to the next module (first hidden layer) and similarly to the end this process continues. Each connection has a weight and all layers have biases that learning process is the adjustment of weights and biases. MLP has some free parameters such as the number of hidden layers and the number of their neurons that should be resolved with trial and error. Methods based on Gradient descent (GD) are used to learning process, but they suffered from a big challenge. These methods fall in local optimum [27, 28, 29, 30, 31].

3 Meta–heuristic methods

There are two main approaches in optimization problems, deterministic algorithm and non–deterministic or stochastic algorithms. Mainly difference between them is: in non–deterministic, there are not guarantee to find optimal solution, but they solve complex problem with accepting the amount of errors such that deterministic algorithms cannot solve them. A heuristic method is a non–deterministic algorithm and it finds an approximation of solution in an iterative process [32, 33, 34].

3.1 Particle Swarm Optimization (PSO)

Particle swarm optimization is a method based on swarm intelligence. This is an imitation of nature, such as birds flock or fish schooling that looking for food. In this technique, each of the particles are not intelligent alone, but particle collective is intelligent.
PSO is created by the particles. Each particle is a problem solution and they move to optimal solution based on velocity parameter. Each particle moves based on best own position and global best position between all particles. Initially a swarm of particles is broadcasted in space problem and all particle position is selected as best personal position (pbest) and global solution is determined based on the merit all particle (gbest) and in each iteration the position of particles is updated according p_{best} and g_{best}. In each iteration the new positions of the particles are evaluated by fitness function again, if the current position of each particle is better than p_{best} then p_{best} is changed to the current position and if the best p_{best} or current g_{best} is better than g_{best} then the g_{best} is adjusted to the current g_{best} as g_{best} too, this process continues until algorithm achieves to the stop criteria [4, 22, 33, 35]. Velocity and position of each particle is shown in equations 1 , 2.

$$v_i(t+1) = wv_i(t) + c_1 r_1(t)[p_{best}(t) - x_i(t)] + c_2 r_2(t)[g_{best}(t) - x_i(t)] \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

In equation 1 $v_i(t)$, $x_i(t)$ is velocity and position of particle i in iteration t and c_1, c_2 are two positive constant parameters, r_1, r_2 are random number of uniform distribution between 0, 1 and w is inertia weight for better convergence of the algorithm.

3.2 Genetic Algorithm (GA)

Genetic algorithm (GA) is an evolutionary algorithm which is similar to the process of nature evolution. GA creates an initial population as the primal solution that are called chromosomes and chromosomes have been made of bits (genes). After evaluating solutions with objective function and these are sorted merit. In each iteration needs to generate a new population which done by three operations; selection, crossover and mutation. Two chromosomes (parents) generate at least two chromosomes (children), the selection process (selection of parents) is commonly a probability function that acts based on chromosomes merit. Then

GA simulates biological reproduction with combination two parents and children with crossover process. In each iteration a gene of children is changed that known as mutation [14, 19, 32, 33, 35].

4 Chaotic time series prediction

For modeling the behavior of a chaotic system should be determined system evolution over the time, state (phase) of chaotic system contains information of evolutionary of system [1]. When time be discreet in chaotic system and its evolution is defined by equation 3.

$$\mathbf{X}(n+1) = \mathbf{F}[\mathbf{X}(n))] \tag{3}$$

$\mathbf{F}[.]$ is a non–linear function and $\mathbf{X}(n) \in \mathbf{R}^n$ is a state (phase) of the system at time n. If the observable output of a system is defined in the term phase $\mathbf{X}(n)$ then a time series $X(n), n = 1, 2, \cdots, N$ is created, that can be formulated following to equation 4.

$$X(n+1) = g[\mathbf{X}(n))] + \xi(t) \tag{4}$$

$g[.]$ is a non–linear function and ξ describes uncertainty and noise. Since that only information recourse is observable output $X(n)$ so to improve accuracy of prediction, reconstructed phase space of a dynamic system is needed that done based on Takens and embedding theorem [28]. Based on Takens theorem, phase space of a chaotic system is reconstructed by equation 5.

$$\mathbf{X_l}(n) = [X(n), X(n-\tau), \cdots, X(n-(d-1)\tau)] \tag{5}$$

$X(n)$ is a sample of time series in time n and τ is embedding delay and d is embedding dimension, equation 5 expresses delay embedding theorem. Based on delay embedding, time delay values can be used to help predict the phase space [1]. Embedding theorem uses the relationship between current phase $\mathbf{X_l}(n)$ and the next value of time series $X(n+1)$ that determined by equation 6 for chaotic time series forecasting.

$$X(n+1) = g[\mathbf{X_l}(n))] \tag{6}$$

τ and d are determined before and $g[.]$ is a nonlinear function and it's capable to estimate the next value of time series $X(n+1)$ following equation 7.

$$\bar{X}(n+1) = \bar{g}[\mathbf{X_l}(n))] \tag{7}$$

$\bar{X}(n+1)$ is an approximation of $X(n+1)$ and $\bar{g}[.]$ is a simulator of $g[.]$, hence the error is calculated by difference between the value of $X(n+1)$ and $\bar{X}(n+1)$ [12, 27, 36].

5 Methodology

There are two main groups of learning algorithm: supervised and unsupervised. Supervised algorithms have input and output data and they try to train ANN based on this dataset, but unsupervised learning only has input data and they try to discover the knowledge of input [32].
Time series prediction is a supervised problem and this paper uses heuristic methods (GA and PSO) to train MLP. Since learning algorithm to be considered as an optimization problem so heuristic methods can solve it. The generic process of the training process as follows [18, 22, 23]:

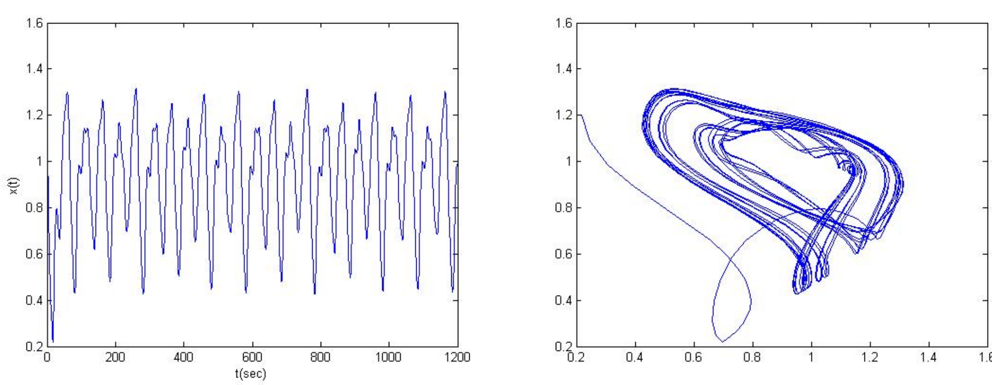
- Constructing basic ANN with inputs and outputs.
- Creating an optimization problem based on weights and biases.
- Consign this problem with heuristic algorithm.
- Evaluating performance of ANN.

6 Tests and details

The experiments have been done with MATLAB software and Mackey–Glass (MG) time series has been considered as a chaotic time series that shown in 8.

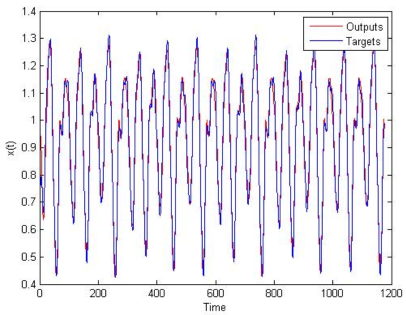
$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x(t-\tau)^{10}} - 0.1x(t) \tag{8}$$

$x(0) = 1.2$ and $\tau = 17$ and this diagram is shown in figure 1(a) and its phase diagram is shown in figure 1(b) that known as an attractor. This problem has been challenged in very researches [1, 3, 12, 27, 37, 38].

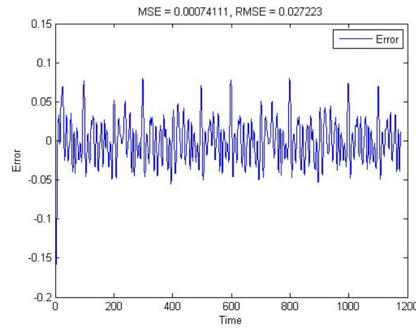


(a) MackeyGlass time series (b) MackeyGlass Phase diagram

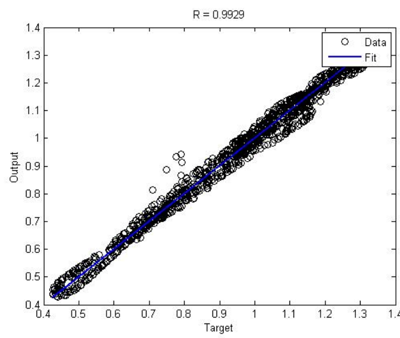
Figure 1: MackeyGlass



(a) Prediction of MG time series



(b) Errors prediction of MG time series



(c) Regression of errors prediction of MG

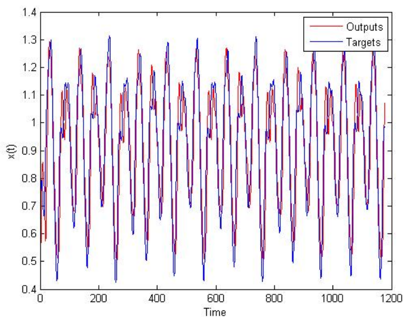
Figure 2: MLP trained with PSO

There are many criteria to compare the results, for example, mean square error (MSE), normalize mean square error (NMSE), root mean square error (RMSE), normalize root mean square error, mean absolute percentage error (MAPE), average mean square error, time elapsed and correlation between observed value and predicted value [3, 12, 27, 37, 39, 40, 41]. MSE and RMSE are used in this paper, see equations 9 and 10; moreover prediction diagram, error and error regression (correlation between output and target) is shown.

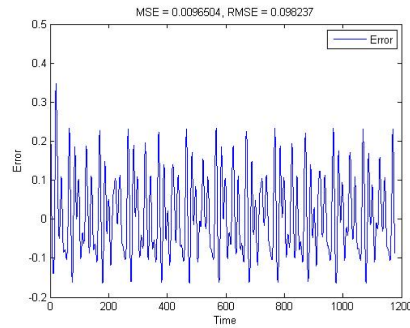
$$MSE = \frac{1}{n}(Y_i - T_i)^2 \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}(Y_i - T_i)^2} \tag{10}$$

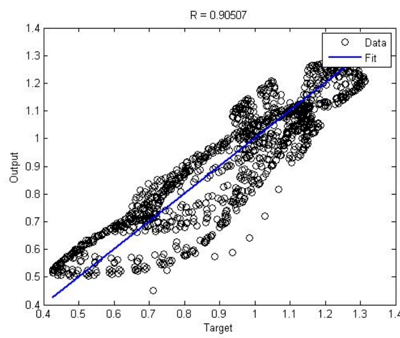
This paper uses MLP as a tool for predicting chaotic time series, this ANN has two input and output layers and a hidden layer with 20 neurons and activation functions of hidden layer and output layer respectively are sigmoid and linear. GA and PSO are used to train MLP so the population size of PSO and GA have been determined 100 and in the PSO $C_1 = C_2 = 2$ and the end the maximum iterations in all of methods are 500. Figure 2(a) shows MG prediction with MLP that is trained by PSO, in this picture the output and target have been determined, and the output is fitted on the target in most points. Figure 2(b) shows the error that



(a) Prediction of MG time series



(b) Errors prediction of MG time series



(c) Regression of errors prediction of MG

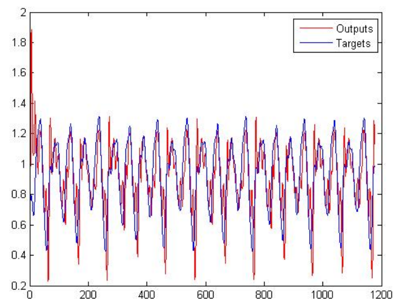
Figure 3: MLP trained with GA

occurs in the predict process, the value of MSE and RMSE are mentioned on top of the picture and the errordiagram variables between 0.05 and 0.05. Also in the figure 2(c) the correlation diagram between output and target is drawn and the value of correlation coefficient as R is mentioned. All figures and values of criteria evaluation show that the performance of this neural network is good.

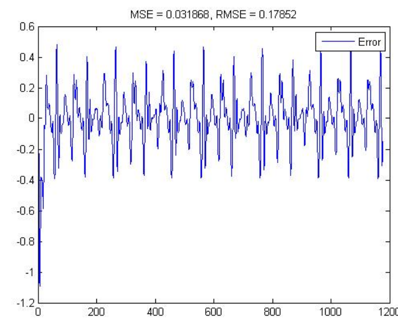
Results of MLP trained with GA shown in figure 3. Comparing these results shows the MLP that trained with GA is weaker than trained with PSO. MSE, RMSE and R are lower than the results of figure 1 and also the value of error is between 0.1 and 0.2.

This paper uses three gradient descent algorithms; gradient descent (GD), gradient descent by momentum (GDM) and gradient descent with adaptive learning rate (GDA) [32]. The results of GD are shown in figure 4 and it has not good performance. For example, error is between 0.4 and 0.4 and also MSE and RMSE values are increased.

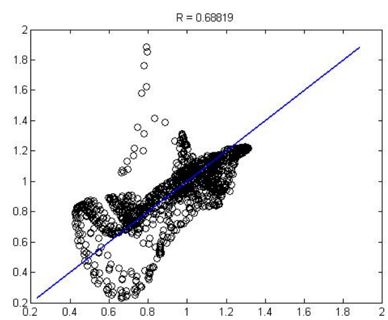
Other results are summarized in table 1 that in this table MSE, RMSE and correlation coefficient value (R) are specified. As table 1 shows the performance of MLP that trained with PSO is best output. In the class of gradient descent the GDA is better than other methods and even it is better than MLP that trained with GA. In table 1 methods have been arranged according goodness. Heuristic methods that train ANN need more CPU time that means these methods need



(a) Prediction of MG time series



(b) Errors prediction of MG time series



(c) Regression of errors prediction of MG

Figure 4: MLP trained with GD

more time to reach for better prediction rather than gradient descent methods.

7 Conclusions

Accuracy predicting chaotic time series is very important. To achieve this goal, great efforts have been made that one of them is intelligent methods. In this paper combination intelligent methods were reviewed, artificial neural networks (ANNs) and meta–heuristic methods. Multi–layer perceptron (MLP) is used as a predictor and genetic algorithm (GA) and particle swarm optimization (PSO) acts as MLP learning algorithm. As the results show heuristic methods with MLP better than other gradient descent methods and forecast accuracy is improved. It should be noted, this needs more time to run. So that between more accuracy and better time spending must choose one. This choice depends on the type of problem, but in this paper the focus was on more accuracy.

Table 1: ANN's performance trained with PSO, GA

criteria / train	MSE	RMSE	R
PSO	0.00074	0.02722	0.9929
GDA	0.00752	0.08671	0.9329
GA	0.00965	0.09824	0.9051
GD	0.03187	0.17852	0.6882
GDM	0.03310	0.18192	0.6633

References

[1] Lau, K.W., Wu, Q.H., *Local prediction of non–linear time series using support vector regression*, Pattern Recognit., **41** (2008) 1539–1547.

[2] Dufour, J.M., *Introduction to time series analysis*, F. econometrics (Economics 763B), McGill University, Winter 2014.

[3] Fu,Y. Y., Wu, C. J., Jeng, J. T., and Ko, C. N., *ARFNNs with SVR for prediction of chaotic time series with outliers*, Expert Syst. Appl., **37** (2010) 4441–4451.

[4] Caia, X., Zhang, N., Venayagamoorthy, G. K., and Wunsch, D. C., *Time series prediction with recurrent neural networks trained by a hybrid PSOEA algorithm*, Neurocomputing, **70** (2007) 2342–2353.

[5] Araujo, R. D. A., *Swarm–based translation–invariant morphological prediction method for financial time series forecasting*, Inf. Sci., **180** (2010) 4784–4805.

[6] Skiadas, C. H., Dimotikalis, I., and Skiadas, C., *Chaos theory modeling, simulation and applications*, World Scientific Publishing co. pte. Ltd, 2011.

[7] Deshpande, R. R., *On the rainfall time series prediction using multilayer perceptron artificial neural network*, Int. J. Emerging Technol. Adv. Eng., **2(1)** (2012) 148–153.

[8] Valencia, A., Alvarez, M., *Short–term time series prediction using hilbert space embeddings of autoregressive processes*, Neurocomputing, **266** (2017) 595–605.

[9] Langkvist, M., Karlsson, L., and Loutfi, A., *A review of unsupervised feature learning and deep learning for time–series modeling*, Pattern Recognit. Lett., **42** (2014) 11–24.

[10] Zhang, Z., Wang, T., and Liu, X., *Melt index prediction by aggregated RBF neural networks trained with chaotic theory*, Neurocomputing, (2014) 368–376.

[11] Gil, P. G., Pedrero, A. G., and Cortes,J. M. R., *Composite recurrent neural networks for long–term prediction of highly–dynamic time series supported by wavelet decomposition*, Soft Computing for Intelligent Control and Mobile Robotics, **318** (2011) 253–268.

[12] Ardalani–Farsa, M., Zolfaghari, S., *Chaotic time series prediction with residual analysis method using hybrid elman–NARX neural networks*, Neurocomputing, **73** (2010) 2540–2553.

[13] Li, K., Liu, L., Zhai, J., Khoshgoftaar, T. M., and Li, T., *The improved grey model based on particle swarm optimization algorithm for time series prediction*, Eng. Appl. Artif. Intel., **55** (2016) 285–291.

[14] Mandal, S. N., Ghosh, A., Roy, S., Choudhury, J. P., and Bhadra, S. R., *A novel approach of genetic algorithm in prediction of time series data*, Int. J. High Perform. C., (2012) 16–21.

[15] Bao, Y., Xiong, T., and Hu, Z., *Multi–step–ahead time series prediction using multiple–output support vector regression*, Neurocomputing, **129**, (2014) 482–493.

[16] Yolcu, O., Lam, H., *A combined robust fuzzy time series method for prediction of time series*, Neurocomputing, **247** (2017) 87–101.

[17] Guo, F., Lin, L., and Wang, C., *Novel continuous function prediction model using an improved takagi–sugeno fuzzy rule and its application based on chaotic time series*, Eng. Appl. Artif. Intel., **55** (2016) 155–164.

[18] Panda, S., Sarangi, A., and Panigrahi, S. P., *A new training strategy for neural network using shuffled frog–leaping algorithm and application to channel equalization*, AEU–Int. J. Electron. C., **68(11)** (2014) 1031–1036.

[19] Jaddi, N. S., Abdullah, S., and Hamdan, A. R., *A solution representation of genetic algorithm for neural network weights and structure*, Inf. Process., **116** (2016) 22–25.

[20] Takahashi, K., Kurokawa, M., and Hashimoto, M., *Multi–layer uantum neural network controller trained by real–coded genetic algorithm*, Neurocomputing **134** (2014) 159–164.

[21] Gaxiola, F., Melina, P., Valdez, F., Castro, J. R., and Castillo, O., *Optimization of type–2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO*, Appl. Soft Comput., **38** (2016) 860–871.

[22] Sheikhan, M., Mohammadi, N., *Time series prediction using PSO–optimized neural network and hybrid feature selection algorithm for IEEE load data*, Neural Comput. Appl. **23** (2013) 1185–1194.

[23] Piotrowski, A. P., Osuch, M., Napiorkowski, M. J., Rowinski, P. M., and Napiorkowski, J. J.,*Comparing large number of metaheuristics for artificial neural networks training to predict water temperature in a natural river*, Comput. Geosci. **64** (2014) 136–151.

[24] Lam, H. K., Ekong, U., Liu, H., Xiao, B., Araujo, H., Ling, S. H., and Chan, K. Y., *A study of neural–network–based classifiers for material classification*, Neurocomputing, **144** (2014) 367–377.

[25] Kasabov, N., Capecci, E., *Spiking neural network methodology for modelling, classification and understanding of EEG spatio–temporal data measuring cognitive processes*, Inf. Sci., **294** (2015) 565–575.

[26] Cerri, R., Barros, R. C., and Carvalho, A. d., *Hierarchical multi–label classification using local neural networks*, J. Comput. Syst. Sci., **80(1)** (2014) 39–56.

[27] Wu, X., Song, Z., *Multi–step prediction of chaotic time–series with intermittent failures based on the generalized nonlinear filtering methods*, Appl. Math. Comput., **219** (2013) 8584–8594.

[28] Kuchaki Rafsanjani, M., Samareh, M., *Chaotic time series prediction by artificial neural networks*, J. Comput. Meth. Sci., **16** (2016) 599–615.

[29] Villarrubia, G., Paz, J. F., Chamoso, P., and Prieta, F. D., *Artificial neural networks used in optimization problems*, Neurocomputing, **272** (2018) 10–16.

[30] Wu, X., Wang, Y.,*Extended and unscented kalman filtering based feedforward neural networks for time series prediction*, Appl. Math. Model., **36** (2012) 1123–1131.

[31] Crone, S. F., Kourentzes, N., *Feature selection for time series prediction–a combined filter and wrapper approach for neural networks*, Neurocomputing, **73** (2010) 1923–1936.

[32] Poole, D. L., Mackworth, A. K., *Artificial intelligence foundations of computational agents*, Cambridge University Press (2017).

[33] Boussaid, I., Lepagnot, J., and Siarry, P., *A survey on pptimization metaheuristics*, Inf. Sci., **237** (2013) 82–117.

[34] Gendreau, M., Potvin, J. Y., *Handbook of metaheuristics*, Operat. Res. Manag. Sci., **146** (2010).

[35] Diez, L. and Brownlee, J., *Clever algorithms*, Nature–Inspired Programming Recipes, Lulu Enterprises Incorporated, (2011).

[36] Panella, M., *Advances in biological time series prediction by neural networks*, Biomed. Signal Proces., **6** (2011) 112–120.

[37] Chandra, R., Zhang, M., *Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction*, Neurocomputing, **86** (2012) 116–123.

[38] Yuxia, H., Hongtao, Z., *Chaos optimization method of SVM parameters selection for chaotic time series forecasting*, Physics Procedia, 25 (2012) 588–594.

[39] Wong, W. K., Xia, M., and Chu, W. C., *Adaptive neural network model for time–series forecasting*, Eur. J Oper. Res., **207** (2010) 807–816.

[40] Cherif, A., Cardot, H., and Bone, R., *SOM time series clustering and prediction with recurrent neuran networks*, Neurocomputing, **74** (2011) 1936–1944.

[41] Ali–Ahmadi, M., Ebadi, M., Shokrollahi, A., and Majidi, S., *Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir*, Appl. Soft Comput., **13** (2013) 1085–1098.