

# A Comprehensive Learning Polynomial Auto-Regressive Model based on Optimization with the Application of Time Series Forecasting

Nastaran Darjani<sup>1</sup>, and Hesam Omranpour<sup>2, †</sup>

<sup>1,2</sup>Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

**A** Nowadays, time series analysis is an important challenge in engineering problems. This paper proposes a Comprehensive  
**B** Learning Polynomial Autoregressive Model for predicting linear and nonlinear time series. The model is based on the  
**S** autoregressive model but developed in a polynomial aspect to make it more robust and accurate. The model predicts  
**T** future values by learning the weights of the weighted sum of the polynomial combination of previous data. The learning  
**R** process for the hyperparameters and properties of the model in the training phase is performed by the metaheuristic  
**A** optimization method. Using this model, we can predict nonlinear time series, as well as linear time series. The proposed  
**C** method was implemented on eight standard stationary and non-stationary large-scale real-world datasets. It  
**T** outperformed the state-of-the-art methods that use deep learning in seven time series and has better results compared to  
all other methods in six datasets. Experimental results show the advantage of the model's accuracy over other compared  
methods on various prediction tasks based on the root-mean-square error.

## Article Info

### Keywords:

Autoregressive, Forecasting, Machine learning, Optimization, Time series prediction

### Article History:

Received 2021-08-17

Accepted 2021-12-18

## I. INTRODUCTION

Nowadays, time series prediction is used in a wide range of areas including energy [1], traffic [2],[3], economy tourism [4], and health [5]. The methods defined for time series prediction are divided into statistical and non-statistical sections. The autoregressive (AR) and Autoregressive Integrated Moving Average (ARIMA) models are two examples of statistical methods [6]. Models for forecasting commodity prices, including oil or natural gas are already in application with ARIMA [7]–[9]. Also, these methods are used for load forecasting in power systems with acceptable results [10]. These methods are linear models for accurate prediction of time series; however, these are only satisfactory for linear time series and unsuitable for nonlinear time series

[11]. Multiple methods have been extended from classic methods to satisfy this limitation. Multiresolution autoregressive model (MAR) is a wavelet-based AR model presented in [12]. Due to Kalman filter-inspired recurrence scheme, which allows not recomputing all estimations for each new value, this method is a faster and more flexible algorithm than the basic AR model.

Neural networks [13]–[16],[17] and Fuzzy methods [18]–[21] are known as non-statistical and soft computing methods. The advantages of artificial neural networks are their flexible nonlinear modeling capability, strong adaptability, and their learning and extensive parallel computing abilities [22],[23], but the neural network is inadequate in predicting linear time series and also requires a lot of training data to accurately predict future data [24]. In [25], time series is decomposed using multiscale fast wavelet transform, and each scale's wavelet coefficients are learned with a multilayer perceptron to predict the next sample of time series. More recently, more deep learning-based methods have been developed but

<sup>†</sup>Corresponding Author: H.Omranpour@nit.ac.ir  
Tel: 01135501467, Department of Electrical and Computer Engineering,  
Babol Noshirvani University of Technology, Babol, Iran

**Algorithm I.** Fitness Function

**Input:** particles  
**Output:** RMSE  
 /\* decoding particles to matrix of weight matrix and bias \*/  
 1: **for** each number  $j$  of time series window **do**  
 2: **for** each power  $i$  of CLPAR model **do**  
 3: **set**  $i$ -th particle from  $j$ -th part with  $n$  size particles to  $w(i, j)$   
 4: **end for**  
 5: **end for**  
 6: **set** last particle to bias  
 /\* predict last data of windows with weight and bias based on Equ. 3 \*/  
 7: **for** each position  $i$  in time series window **do**  
 8: **set** 0 to  $p$   
 9: **for** each number  $j$  of time series window **do**  
 10: **for** each power  $k$  of CLPAR model **do**  
 11: **set** power  $k$  of  $i$ -th data in  $j$ -th window multiplied by  $w(j, k)$  to  $p$   
 12: **end for**  
 13: **end for**  
 14: **set**  $p +$  bias to  $y_{\text{predict}}(i)$   
 15: **end for**  
 /\* calculate RMSE by applying Equ. 5 \*/  
 16: **set** square sum of  $(y - y_{\text{predict}})^2$  divided by size of  $y$  to RMSE

these methods also have the same advantages and disadvantages. High-order Fuzzy Cognitive Maps (HFCMs) are learned via Convolutional Neural Network (CNN) deep learning networks in [26].

Fuzzy methods predict the time series using a function-estimation approach by one or more fuzzy rules [27]. In these methods, every sample of time series is used to evaluate parameters of the fuzzy model and then the error of test data reported [28],[29]. In [30], the Adaptive-Network-Based Fuzzy Inference System (ANFIS) is employed by using a hybrid learning procedure. In [27], sparse autoencoder (SAE) extracts features from the original time series and feeds them to an HFCM predictor. A combination of HFCMs with SAE output is used to calculate the predicted time series. Another FCM-based method is mentioned in [31], where HFCMs are combined with Haar wavelet transform to manage nonstationary time series.

In the real world, time series are composed of linear and non-linear parts. The abovementioned methods are only appropriate for either linear or non-linear time series. As the nonlinear part of the time series dominates the linear part, the nonlinear models produce a satisfactory result. The same outcome is achieved when the linear part of the time series is greater than the nonlinear part. In either case, one part of the time series is ignored, which reduces the prediction accuracy [32].

## II. PROPOSED MODEL

This paper introduces a polynomial model generalized from the classic AR model in which nonlinear properties, besides linear ones, are considered. The method provides nonlinearity with the possibility of having different powers of previous data points. Section II introduces the proposed

model with two phases of training and testing algorithms. In section III, the RMSE value of the proposed method is compared with the other methods

**TABLE I**

RMSE IN TESTING AND TRAINING DATA REPRESENTED BY THE BEST

Dataset	POLYNOMIAL DEGREE AND WINDOW SIZE				
	R-squared	Testing RMSE	Training RMSE	Optimal polynomial degree	Optimal window size
Sunspot	0.839	16.947	13.06	1	13
Mackey-Glass	0.996	0.007	0.0077	2	6
S&P 500 stock index	0.988	10.247	10.69	1	9
Monthly milk production	0.992	6.839	5.91	1	14
Monthly closings of the Dow-Jones	0.993	22.549	21.46	2	37
Monthly critical radio frequencies	0.945	0.320	0.32	3	49
CO2 (ppm)	0.996	0.357	0.37	1	17
Monthly Lake Erie levels	0.913	0.336	0.24	3	28

. Section IV presents the concluding points. AR-based methods make accurate predictions of linear time series but they are unable to predict nonlinear real-world problems [12]. The proposed model can be considered an extension of the AR model. Equ. (1) is used in AR-based models from a prediction that is the linear sum of the previous amounts [33].

$$Y_t = \sum_{i=1}^p \varphi_i Y_{t-i} + \varepsilon_t \quad (1)$$

But, the behavioral pattern of the series might not be entirely linear, so this paper presents an equation that does not have this limitation. A separate coefficient is assigned for each power of  $y_{(t-i)}$  data. An  $\varepsilon_t$  bias value is also considered. Equ. (2) is achieved by simplifying Equ. (1). The sum of  $\varepsilon_i$  is replaced by  $\varepsilon$  in Equ. (2).

$$Y_t = \varepsilon + \sum_{i=1}^p \varphi_i Y_{t-i} \quad (2)$$

On the other hand, if the different powers of the  $y_{t-i}$  data are used to predict  $y_t$ , this sum will reach a large amount, so in addition to inserting the coefficient, data is also normalized. We consider an equation consisting of two sigmas. One is on the powers of  $y_{t-i}$  and the other on the sum of these values on the preceding  $p$  data. The sum of the two sigmas, which were mentioned above, is considered for every number in the time series. Finally, this equation is presented for modeling in this paper. Equ. (3)(6) need to learn the coefficients. The extended polynomial capability enables the method to satisfy nonlinear time series prediction although, with a much bigger search space, it can be challenging to estimate the parameters. Given that a window with the length  $p$  is used for prediction and also the upper limit of polynomial degree is  $n$ , learning is done in a space

**TABLE II**  
COMPARISON WITH OTHER PREDICTION MODELS IN TERMS OF RMSE

Dataset	CLPAR	CNN-FCM [25] (2020)	RNN [25] (2020)	SAE-FCM [26] (2019)	Wavelet-HFCM [30] (2018)	ANFIS [29] (classic)	AR Model [6] (classic)	Multiresolution AR model [12] (2005)	ScaleNet-Multiscale Neural Network [24] (classic)
Sunspot time series	16.947	17.949	19.292	17.390	18.916	22.753	35.262	19.186	19.901
Mackey-Glass time series	0.007	0.0013	0.0005	0.0009	0.004	0.001	0.035	0.002	0.005
S&P 500 stock index	10.247	20.816	27.896	11.893	16.105	14.935	17.897	16.041	17.696
Monthly milk production per cow	6.839	30.474	29.253	7.931	8.258	9.578	57.717	37838	27.113
Monthly closings of the Dow-Jones industrial index	22.549	25.190	26.232	21.335	23.159	57.526	29.822	26.733	28.532
Monthly critical radio frequencies	0.320	0.567	0.613	0.490	0.547	0.651	0.902	0.662	0.652
CO2 (ppm) at Mauna Loa	0.357	0.731	1.420	0.366	0.56	0.91	1.35	0.812	1.695
Monthly Lake Erie levels	0.336	0.391	0.374	0.359	0.377	0.458	0.638	0.39	0.402

with the size  $p \times n$ . This space reaches  $(p \times n) + 1$  by inserting the epsilon bias value. This search space is bigger than that of the classic method with  $n + 1$  estimation. These coefficients are learned from an optimization perspective. GWO is used to find the best values for these coefficients and epsilon [34].

$$Y_t = \varepsilon + \sum_{j=1}^n \sum_{i=1}^p \varphi_{ij} Y_{t-i}^j \quad (3)$$

#### A. Model Training Level

The training algorithm is as follows:

- Normalizing the data
- Windowing the data: Segmentation into  $p$ -size data per  $y_t$
- Learning the coefficients with GWO: Calculating the output for each window using Equ. (3) and obtaining RMSE as fitness

where the algorithm input is time series and outputs are the best  $\varphi$  and  $\varepsilon$ .

1) *Normalization Phase*: In this phase, all the data is placed in the range  $[0, 1]$  as follows,

$$d_i = \left( \frac{d_i - \min}{\max - \min} \right) \quad (4)$$

where  $\min$  and  $\max$  stands for minimum and maximum data, respectively and  $d_i$  is the  $i$ -th data in time series.

2) *Windowing Phase*: In this section, data is prepared and resized to  $p$  to fit in Equ. (3). For example, for time series with the length  $L$  and window of 3, windowing consists of each data with its previous two data, and the number of rows is  $t = L - p + 1$ .

3) *Learning the Coefficients*: We must obtain the weights for Equ. (3) according to the windowing phase. In the windowing phase, there are  $Tlength - p + 1$  equations with  $(p \times n) - 1$  unknowns (weights) where  $Tlength$  is the time series size. Weights of these equations are achieved by using an optimizer. The optimizer used here is grey wolf optimization [34].

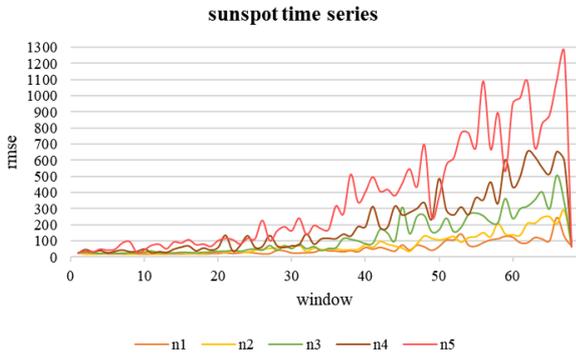
To code the particles existing in optimization search space (wolves in GWO), an  $n \times p$  matrix and an epsilon variable are used. The  $n \times p$  matrix shows the weights of particles (wolves) in Equ. (3).

Inputs of the fitness function are trained windowed data and a particle (wolf). By using the coefficients and inserting them in Equ. (3)(6), the RMSE of the trained window is returned as output. RMSE, which calculates the difference between the original time series and the predicted one, is obtained using Equ. (5) where  $L$  is the length of time series and  $x_i$  and  $y_i$  stand for the predicted time series and the normalized original time series, respectively [30]. Algorithm I shows the pseudo-code of the fitness function.

$$RMSE = \sqrt{\frac{\sum_{i=1}^L (x_i - y_i)^2}{L}} \quad (5)$$

#### B. Model Testing

In this section, outputs of the training data that include best weights and epsilon are used in Equ. (3) to predict the testing data.



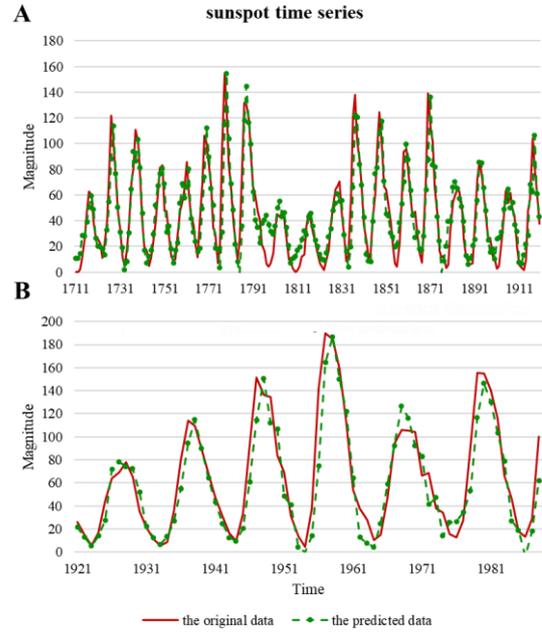
**Fig. 1.** RMSE in different window sizes and polynomial degrees. The horizontal axis is window size and different colors of the line represent the maximum polynomial degree in Equ. (3).

**III. EXPERIMENTAL RESULT**

This section presents the results of the implemented experiments on real data. First, we introduce the datasets. The results of the implementation are then presented along with analysis and comparison with other methods.

**A. Datasets**

To show the efficiency of the suggested model, we use eight benchmark time series with distinct characteristics. The first one is the sunspot time series that has recorded the annual amount of sunspots from 1700 to 1987, which includes a total of 289 observations and has often been used by other researchers [31],[35]–[40]. The second dataset contains a total of 251 recorded observations, used in associated publications [31],[41]–[46] and accessible from the Yahoo finance website [47]. It concerns the day-to-day price of the S&P 500 stock index from 1 June 2016 to 1 June 2017. The third dataset records a total of 168 observations in the monthly manufacturing of milk in pounds from January 1962 to December 1975 [48]. In the fourth dataset, the index of Dow-Jones Industrial monthly closing has recorded a total of 291 observations from August 1968 until August 1981 [49]. The fifth dataset includes 240 observations that reflect the highest radio frequency used in Washington, D.C. between May 1934 and April 1954 in [50]. The sixth dataset relates to 192 observations [51] recording CO<sub>2</sub> (ppm) at Mauna Loa from 1965 to 1980 and the seventh dataset, including 600 observations [52], contains the monthly Lake Erie level from 1921 to 1970. A first-order nonlinear differential delay equation is described by Equ. (6). The last dataset, the chaotic Mackey-Glass time series [53], is derived from that. This time series validates the effectiveness of prediction models in [31],[35],[36],[54]–[57]. The dataset of the Mackey-Glass time series was acquired by solving the fourth-order Runge-Kutta algorithm with 1000 points from t = 124 to t =



**Fig. 2.** Original time series and predicted time series in (A) training data, (B) testing data.

1123 by solving Equ. (6). In addition, the initial conditions  $x(0)$  and  $\tau$  are set to 1.2 and 17, respectively [31].

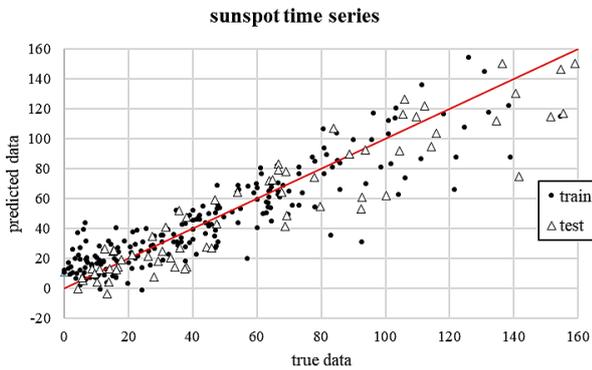
$$\dot{x} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t) \tag{6}$$

Each dataset is divided into testing and training. Given that the time series prediction is applied to forecast future data, the time series in the experiments in this paper is broken into two parts of testing and training at one point. This segmentation causes the next  $k$ -th data to be predicted as well as the next data.

**B. Survey of the Effect of Window Size and Polynomial Degree**

This section presents the results of the implementation of the proposed method on the eight databases introduced in the previous section with different window sizes and polynomial degrees. Fig. 1 shows the RMSE levels of the testing data for the sunspot time series. Graph row represents the size of the window and each curve represents a polynomial degree. As the window size and polynomial degree increase, the complexity of the model increases. On the other hand, the search space for learning weights also increases so the optimizer may not be able to produce the optimized results in a limited time. Therefore, test errors are more likely to occur for large  $n$  and  $p$  values with the same number of iterations. Table I presents the best values for polynomial degree and window length for each dataset. Due to the unlimited statistics range of RMSE ( $[0, +\infty)$ ), which is not

informative about the quality of regression performance individually,



**Fig. 3.** The original and predicted data in the testing and training datasets.

R-squared values are shown in Table I. It can be determined how well the independent variables can account for the variance of the dependent variables, which is defined as follows [58]:

$$R^2 = 1 - \frac{\sum_{i=1}^L (x_i - y_i)^2}{\sum_{i=1}^L (\bar{y} - y_i)^2} \quad (7)$$

where  $x_i$  is the predicted  $i$ -th value and  $y_i$  and  $\bar{y}$  are the original corresponding time series and their means, respectively.

### C. Implementation Results After Parameter Tuning

In this section, charts are provided using the best values obtained based on Table I. Fig. 2 shows the original data and also the data that has been predicted by the proposed model over time on the sunspot time series. Fig. 2 contains the predicted training (A) and predicted testing (B) data. Also, to determine the deviation rate of the data predicted from the actual data, scatter plots of the sunspot dataset are shown in Fig. 3.

One of the challenges in nonstationary time series is to obtain periodicity. The periodicity presented in Fig. 2 is obtained by averaging the distance between adjacent peaks (valleys). This frequency (periodicity) is not necessarily nonstationary (periodicity). Therefore, the frequency in the time series might be obtained by multiplying the frequency existing in Fig. 3 to a natural number. By comparing the columns of the periodicity table and the window size in Table I, it can be observed that the defined periodicity is a numerical coefficient of the optimal window size. This coefficient in the Mackey-Glass time series, milk production per cow, monthly critical Mauna Loa, and the model can predict these data as a sum of linear coefficients of the pre-data. But, this value is greater than 1 for datasets of Mackey-Glass time series, monthly closings of the Dow-Jones industrial index, monthly critical radio

frequencies, and monthly Lake Erie levels, which means the pattern in the data from the previous window is polynomial.

### D. Comparison against other methods

This section compares the proposed method with other state-of-the-art and classical prediction methods in terms of RMSE value. Table II shows the results of this comparison. The classic statistical AR model [6] is suitable for linear time series. However, this method cannot satisfy nonlinear and nonstationary time series, which are more likely to exist in the real world. Therefore, the RMSE results of more complex time series for this method are not as good as those of the other methods. This method has the worst RMSE value in half of the tested time series against other ones. Multiresolution AR model [12] is an extended method of AR model, which uses wavelet theory to break complex time series into multiple linear time series and AR model to them. This method is slightly better than the AR model. Moreover, it is faster than the classic method because of the recurrence scheme, which removes some recomputing stages of estimations. Artificial neural networks and deep learning methods [25],[26] are functional and flexible for the nonstationary time series. But, these methods need a huge amount of data to train in order to make a reliable prediction and be robust. In Table II, RMSE values for these methods have mixed results. The RNN method [26] has the worst accuracy in the S&P stock index time series and is the second-worst method in the CO2 time series. In contrast, it has the best RMSE for the Mackey-Glass time series against other methods. Moreover, ScaleNet-Multiscale Neural Network [25] has average results compared to other methods except for the CO2 Mauna Loa time series, which has the highest error against other ones. The weak results in these methods are caused by not having enough training data in comparison to lots of estimation parameters. Fuzzy methods [27], [30], [31] use fuzzy rules to estimate the parameters of the given function. Although these methods have the closest results to the proposed method, the function-estimation approach in these models can be complicated and does not have robustness and flexibility for many time series. In [26], HFCMs learn via deep learning networks to have the flexibility of the neural networks in addition to the simplicity of the fuzzy methods. For most time series, the suggested modeling technique can provide maximum accuracy and still obtain remarkable predictions for a further time series. In Dow-Jones industrial index time series, the SAE-FCM method has slightly higher RMSE than the proposed method. All other methods have lower RMSE than CLPAR in the Mackey-Glass time series except for the classic AR model. In fact, in all datasets, the proposed method outperforms the classic AR model, which is the model that CLPAR derives from it. Regardless of the Mackey-Glass time series, CLPAR entirely outperforms ANN-based methods and has lower error against fuzzy methods in six out of the seven

benchmarks.

#### IV. CONCLUSIONS

In this paper, we proposed a model for predicting time series. One of the classic models for this purpose is the AR model, which is developed to predict linear time series. The described model is based on the AR model generalized to include nonlinear properties besides linear ones. The classic autoregressive model has a coefficient of previous data. The proposed model also uses the same idea to consider the linear property, and to satisfy the nonlinearity property, we considered coefficients of powers of pre-data. These coefficients learn from training data with a grey wolf optimizer.

The proposed prediction modeling method was applied to several datasets, which yielded satisfactory results. In six out of eight datasets, this model achieved the highest accuracy against other methods. In the remaining dataset, the Mackey-Glass time series, an acceptable error value was achieved in terms of RMSE. It is suggested that other optimizations and learning methods can be used to improve the accuracy of the CLPAR method until better accuracy is achieved in the Mackey-Glass dataset.

#### ACKNOWLEDGMENT

The authors gratefully acknowledge the Babol Noshirvani University of Technology for the financial support for the present research through a research grant.

#### REFERENCES

- [1] V. Ediger, S. Akar, *ARIMA forecasting of primary energy demand by fuel in Turkey*, *Energy Policy*. 35 (2007) 1701–1708.
- [2] M. Khashei, F.M. Rafiei, M. Bijari, Hybrid fuzzy auto-regressive integrated moving average (FARIMAH) model for forecasting the foreign exchange markets, *Int. J. Comput. Intell. Syst.* 6 (2013) 954–968.
- [3] K. Kumar, V.K. Jain, *Autoregressive integrated moving averages (ARIMA) modelling of a traffic noise time series*, *Appl. Acoust.* 58 (1999) 283–294.
- [4] F.-L. Chu, A fractionally integrated autoregressive moving average approach to forecasting tourism demand, *Tour. Manag.* 29 (2008) 79–88.
- [5] H.-K. Yu, N.-Y. Kim, S.S. Kim, C. Chu, M.-K. Kee, Forecasting the number of human immunodeficiency virus infections in the Korean population using the autoregressive integrated moving average model, *Osong Public Heal. Res. Perspect.* 4 (2013) 358–362.
- [6] G. Zheng, J.L. Starck, J.G. Campbell, F. Murtagh, Multiscale transforms for filtering financial data streams, *J. Comput. Intell. Financ.* 7 (1999).
- [7] M.D. Chinn, M. LeBlanc, O. Coibion, The predictive characteristics of energy futures: Recent evidence for crude oil, natural gas, gasoline and heating oil, *Nat. Gas, Gasol. Heat. Oil* (October 2001). UCSC Econ. Work. Pap. (2001).
- [8] C. Morana, A semiparametric approach to short-term oil price forecasting, *Energy Econ.* 23 (2001) 325–338.
- [9] W.K. Buchanan, P. Hodges, J. Theis, Which way the natural gas price: an attempt to predict the direction of natural gas spot price movements using trader positions, *Energy Econ.* 23 (2001) 279–293.
- [10] M.T. Hagan, S.M. Behr, *The time series approach to short term load forecasting*, *IEEE Trans. Power Syst.* 2 (1987) 785–791.
- [11] G.E.P. Box, G.M. Jenkins, *Time series analysis: forecasting and control*, Holden-Day, 1976. <https://books.google.com/books?id=1WVHAAAAMAAJ>.
- [12] O. Renaud, J.-L. Starck, F. Murtagh, Wavelet-based combined signal filtering and prediction, *IEEE Trans. Syst. Man, Cybern. Part B.* 35 (2005) 1241–1251.
- [13] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: *The state of the art*, *Int. J. Forecast.* 14 (1998) 35–62.
- [14] T.-S. Quah, B. Srinivasan, Improving returns on stock investment through neural network selection, *Expert Syst. Appl.* 17 (1999) 295–301.
- [15] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE.* 77 (1989) 257–286.
- [16] J. Roman, A. Jameel, Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns, in: *Proc. HICSS-29 29th Hawaii Int. Conf. Syst. Sci.*, 1996: pp. 454–460.
- [17] S. A. Ghoreishi, and H. Khaloozadeh, "Application of Covariance Matrix Adaptation-Evolution Strategy to Optimal Portfolio," *International Journal of Industrial Electronics, Control and Optimization*, Vol. 2, No. 2, pp. 81–90, 2019.
- [18] A. Setare, O. Hesam, M. Homayun, Application of a fuzzy method for predicting based on high-order time series, in: *2014 Iran. Conf. Intell. Syst.*, 2014: pp. 1–6.
- [19] H.-K. Yu, Weighted fuzzy time series models for TAIEX forecasting, *Phys. A Stat. Mech. Its Appl.* 349 (2005) 609–624.
- [20] S.-M. Chen, Forecasting enrollments based on fuzzy time series, *Fuzzy Sets Syst.* 81 (1996) 311–319.
- [21] K. Huarng, Heuristic models of fuzzy time series for forecasting, *Fuzzy Sets Syst.* 123 (2001) 369–386.
- [22] J.L. Ticknor, A Bayesian regularized artificial neural network for stock market forecasting, *Expert Syst. Appl.* 40 (2013) 5501–5506.
- [23] L. Wang, Y. Zeng, T. Chen, Back propagation neural network with adaptive differential evolution algorithm for time series forecasting, *Expert Syst. Appl.* 42 (2015) 855–863.
- [24] T.C. Jo, The effect of virtual term generation on the neural-based approaches to time series prediction, in: *2003 4th Int. Conf. Control Autom. Proc.*, 2003: pp. 516–520.
- [25] A.B. Geva, ScaleNet-multiscale neural-network architecture

- for time series prediction, *IEEE Trans. Neural Networks*. 9 (1998) 1471–1482.
- [26] P. Liu, J. Liu, K. Wu, CNN-FCM: System modeling promotes stability of deep learning in time series prediction, *Knowledge-Based Syst.* 203 (2020) 106081. doi:10.1016/j.knsys.2020.106081.
- [27] K. Wu, J. Liu, P. Liu, S. Yang, Time Series Prediction Using Sparse Autoencoder and High-order Fuzzy Cognitive Maps, *IEEE Trans. Fuzzy Syst.* (2019). doi:10.1109/TFUZZ.2019.2956904.
- [28] J.G. Carvalho Jr, C.T. Costa Jr, Identification method for fuzzy forecasting models of time series, *Appl. Soft Comput.* 50 (2017) 166–182.
- [29] O.C. Yolcu, F. Alpaslan, Prediction of TAIEX based on hybrid fuzzy time series model with single optimization process, *Appl. Soft Comput.* 66 (2018) 18–33.
- [30] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, *IEEE Trans. Syst. Man. Cybern.* 23 (1993) 665–685.
- [31] S. Yang, J. Liu, Time-series forecasting based on high-order fuzzy cognitive maps and wavelet transform, *IEEE Trans. Fuzzy Syst.* 26 (2018) 3391–3402.
- [32] E. Bas, E. Egrioglu, C.H. Aladag, U. Yolcu, Fuzzy-time-series network used to forecast linear and nonlinear time series, *Appl. Intell.* 43 (2015) 343–355.
- [33] H. Akaike, Fitting autoregressive models for prediction, *Ann. Inst. Stat. Math.* 21 (1969) 243–247.
- [34] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61. doi:10.1016/j.advengsoft.2013.12.007.
- [35] W. Lu, J. Yang, X. Liu, W. Pedrycz, The modeling and prediction of time series based on synergy of high-order fuzzy cognitive map and fuzzy c-means clustering, *Knowledge-Based Syst.* 70 (2014) 242–255.
- [36] S. Soltani, On the use of the wavelet decomposition for time series prediction, *Neurocomputing.* 48 (2002) 267–277.
- [37] H.S. Lopes, W.R. Weinert, EGIPSYS: an enhanced gene expression programming approach for symbolic regression problems, *Int. J. Appl. Math. Comput. Sci.* 14 (2004) 375–384.
- [38] C. Ferreira, *Gene expression programming: mathematical modeling by an artificial intelligence*, Springer, 2006.
- [39] H. Cao, L. Kang, Y. Chen, J. Yu, Evolutionary modeling of systems of ordinary differential equations with genetic programming, *Genet. Program. Evolvable Mach.* 1 (2000) 309–337.
- [40] Y. Peng, C. Yuan, X. Qin, J. Huang, Y. Shi, An improved gene expression programming approach for symbolic regression problems, *Neurocomputing.* 137 (2014) 293–301.
- [41] S.T.A. Niaki, S. Hoseinzade, Forecasting S&P 500 index using artificial neural networks and design of experiments, *J. Ind. Eng. Int.* 9 (2013) 1.
- [42] R. Majhi, G. Panda, G. Sahoo, A. Panda, A. Choubey, Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique, in: 2008 IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell., 2008: pp. 1276–1282.
- [43] R. Tsaih, Y. Hsu, C.C. Lai, Forecasting S&P 500 stock index futures with a hybrid AI system, *Decis. Support Syst.* 23 (1998) 161–174.
- [44] M. Martens, Measuring and forecasting S&P 500 index-futures volatility using high-frequency data, *J. Futur. Mark. Futur. Options, Other Deriv. Prod.* 22 (2002) 497–518.
- [45] E. Hajizadeh, A. Seifi, M.H.F. Zarandi, I.B. Turksen, A hybrid modeling approach for forecasting the volatility of S&P 500 index return, *Expert Syst. Appl.* 39 (2012) 431–436.
- [46] S.A. Hamid, Z. Iqbal, Using neural networks for forecasting volatility of S&P 500 Index futures prices, *J. Bus. Res.* 57 (2004) 1116–1125.
- [47] Yahoo, GSPC historical prices j S&P 500 stock, (n.d.). <https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC> (accessed August 16, 2019).
- [48] Monthly milk production: pounds per cow. Jan 62 - Dec 75, (n.d.). <https://datamarket.com/data/set/22ox/monthly-milk-production-poundsper-cow-jan-62-dec-75#!ds=22ox&display=line> (accessed August 16, 2019).
- [49] Monthly closings of the Dow-Jones industrial index, Aug. 1968 - Aug. 1981, (n.d.). <https://datamarket.com/data/set/22v9/monthlyclosings-of-the-dow-jones-industrial-index-aug-1968-aug-%0A1981#!ds=22v9&display=line> (accessed August 16, 2019).
- [50] Monthly critical radio frequencies in Washington, D.C., May 1934 - April 1954, (n.d.). <https://datamarket.com/data/set/22u2/monthlycritical-radio-frequencies-in-washington-dc-may-1934-april-1954-these-frequencies-reflect-the-highest-radio-frequency-that-can-be-used-forbroadcasting#!ds=22u2&display=line>.
- [51] Co2 (ppm) mauna loa, 1965-1980, (n.d.). <https://datamarket.com/data/set/22v1/co2-ppm-mauna-loa-1965-%0A1980#!ds=22v1&display=line> (accessed August 16, 2019).
- [52] Monthly Lake Erie levels 1921 - 1970, (n.d.). <https://datamarket.com/data/set/22pw/monthly-lake-erie-levels-1921-%0A1970#!ds=22pw&display=line> (accessed August 16, 2019).
- [53] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, *Science* (80-. ). 197 (1977) 287–289.
- [54] H.J. Song, C.Y. Miao, Z.Q. Shen, W. Roel, D.H. Maja, C. Francky, Design of fuzzy cognitive maps using neural networks for predicting chaotic time series, *Neural Networks.* 23 (2010) 1264–1275.
- [55] C.-F. Juang, Y.-W. Tsao, A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning, *IEEE Trans. Fuzzy Syst.* 16 (2008) 1411–1424.
- [56] B.-T. Zhang, P. Ohm, H. Mühlenbein, Evolutionary induction of sparse neural trees, *Evol. Comput.* 5 (1997) 213–236.

- [57] T. Hastie, R. Tibshirani, J. Friedman, Model assessment and selection, in: Elem. Stat. Learn., Springer, 2009: pp. 219–259.
- [58] Chicco, Davide, Matthijs J. Warrens, and Giuseppe Jurman. "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation." PeerJ Computer Science 7 (2021): e623.



**Nastaran Darjani** received her B.Sc. degree in Computer Science from the Babol University of Technology, Babol, Iran. Her areas of interest are visual perception in the human brain and human vision-inspired neural networks.



**Hesam Omranpur** received his B.Sc. degree in Computer Engineering-Software Engineering from the Iran University of Science and Technology, and his M.S. and Ph.D. degrees in Artificial Intelligence from the Amir Kabir University of Technology, Tehran, Iran in 2006, 2009, and 2016 respectively. He is currently an assistant professor with the Department of Electrical and Computer Engineering, the Babol Noshirvani University of Technology, Iran. His current research interests include optimization, machine learning, pattern recognition, and brain-computer interface.