Neural Computing and Applications (2023) 35:2855–2865
https://doi.org/10.1007/s00521-022-07753-w

ORIGINAL ARTICLE

A new hybrid recurrent artificial neural network for time series forecasting

Erol Egrioglu[1] · Eren Bas[1]

Received: 10 November 2021 / Accepted: 18 August 2022 / Published online: 6 September 2022

Abstract

The forecasting problems can be effectively solved by using artificial neural networks. Classical forecasting methods are not sufficient to forecast nonlinear and complex time series structures such as a stock exchange time series. In this study, a new hybrid recurrent artificial neural network is proposed for nonlinear time series forecasting. The proposed network is a combination of simple exponential smoothing and the single multiplicative neuron model. The combination weights are also weights in the proposed network, and they are automatically estimated in the training processes. The training of the proposed network is achieved by using a particle swarm optimization-based training algorithm. The training algorithm uses restarting and early stopping strategies to prevent overfitting problems. The proposed network is applied to S&P500, Dow Jones stock exchange data sets, minimum temperature data and wind speed data. The performance of the proposed method is superior to two popular deep artificial neural networks and a high-order artificial neural network.

Keywords Forecasting · Artificial neural networks · Exponential smoothing · Particle swarm optimization

1 Introduction

Time series forecasting plays an important role in many real-world systems. Forecasts can help to improve the performance of the various systems. Exponential smoothing methods are one of the most used simple forecasting methods. The exponential smoothing methods use dynamic equations to estimate different components of time series such as level, trend, and seasonality. The exponential smoothing method needs a smaller number of the parameters if they compare autoregressive integrated moving average (ARIMA) models or other complex methods. Simple exponential smoothing is the simplest and very effective forecasting method. The forecasts of the method are calculated by the following formula.

$$\widehat{X}_t = \alpha X_{t-1} + (1-\alpha)\widehat{X}_{t-1} \quad (1)$$

The forecasts of the simple exponential smoothing are the weighted mean of previous observation and forecasting. The parameter (α) is a smoothing parameter, and its values are bounded zero and one interval. The model is recurrent because it uses lagged variables of the predictions.

The most important disadvantage of the exponential smoothing methods is that they cannot forecast complex and nonlinear time series. For example, the exponential smoothing methods produce very similar forecasts to the random walk process for stock exchange data sets like the ARIMA models. Artificial neural networks (ANN) can forecast nonlinear and complex time series structures because of their nonlinear and soft structure. Many different kinds of artificial neural networks have been used to solve forecasting problems. Jaddi [1] proposed an algorithm for rainfall forecasting by using a multilayer perceptron based on a kidney-inspired algorithm. Maté [2] used a multilayer perceptron for forecasting interval-valued time series. While multilayer perceptron artificial neural networks are the most preferred artificial neural network types, deep artificial neural networks are the most preferred architectures these days. In particular, long short-term memory deep artificial neural networks have better performance the others in many of the studies and forecasting competitions. Long short-term memory artificial neural network (LSTM-ANN) was proposed by Hochreiter [3]. [4–6] used LSTM for forecasting problems, and they are also made some contributions to LSTM literature.

✉ Eren Bas
eren.bas@giresun.edu.tr

[1] Department of Statistics, Faculty of Arts and Science, Giresun University, 28200 Giresun, Turkey

Moreover, some other deep artificial neural networks have been used for forecasting problems in recent years in the literature. Wang et al. [7] used an echo state network with a backtracking search optimization algorithm for time series forecasting. Gao et al. [8] used a gated recurrent unit artificial neural network, and Bas et al. [9] used a deep simple recurrent artificial neural network for forecasting problems. Peng et al. [10] used a double-reservoir echo state network for forecasting electricity load. Fang and Yuan [11] investigated the forecasting performance of deep learning models. Mohammed, Al-Bazi [12] proposed an adaptive backpropagation algorithm for forecasting purposes. Michell et al. [13] used LSTM for electricity forecasting problems. Zouaidia et al. [14] used a hybrid network of LSTM and CNN and GRU by optimizing with the grey wolf algorithm for wind speed forecasting. Readshaw, Giani [15] used CNN for forecasting stock market fluctuations. Oliveira et al. [16] used LSTM and MLP for forecasting traffic flow.

The high-order artificial neural networks are another class of neural networks for solving forecasting problems. These neural networks have their different architectures, and they generally employ multiplicative neuron models in their architectures. Yadav et al. [17] proposed a single multiplicative neuron model artificial neural networks (SMNM-ANN) for the aim of forecasting and classification. Cui et al. [18, 19] proposed recurrent SMNM-ANN models for time series prediction. Pan et al. [20] used SMNM-ANN for time series interval prediction. Pan et al. [21] proposed a data-driven time series prediction based on SMNM-ANN. Another high-order neural network is pi-sigma artificial neural network (PS-ANN) proposed by [22]. [23–25] employed PS-ANN for forecasting problems. The dendritic neuron model artificial neural network is a popular high-order neural network in recent years. The dendritic neuron model presents a model which is closer to a biological neuron than others. Todo et al. [26] proposed a dendritic neuron model. Ji et al. [27] modified the dendritic neuron model. Sha et al. [28] proposed a new dendritic neural network for breast cancer classification. A new neural network method based on the dendritic neuron model has been proposed in Zho et al. [29].

Hybrid forecasting methods are good alternatives to classical and machine learning algorithms. Hajirahimi, Khashei [30] gave a comprehensive literature review on hybrid methods. Zhang [31] firstly proposed a hybrid method for ARIMA and multilayer perceptron ANN. Aladag et al. [32] proposed a hybrid method for ARIMA and Elman neural networks. Mohammadi et al. [33] proposed a hybrid method of genetic algorithm, particle swarm optimization and radial basis artificial neural networks. Panigrahi, Behera [34] proposed a hybrid method of ETS and multilayer perceptron. Kouziokas [35] combined ANN and SVM for GDP forecasting. Hybrid methods are generally two-staged methods, and they need different optimization processes in their algorithms. Some studies combined methods in a joint optimization process. [36–38] combined fuzzy, intuitionistic fuzzy and picture fuzzy sets with pi-sigma artificial neural networks. [39, 40] combined the autoregressive model with the multiplicative neuron model and adaptive network fuzzy inference systems, respectively. Shohan et al. [41] proposed a hybrid of LSTM and neural prophet model. Hadwan et al. [42] proposed a hybridization of the classical time series method and a neural network model. Hu et al. [43] proposed a hybrid deep belief neural network for wind power forecasting.

In this paper, the hybrid methods have one stage and one optimization process.

In this paper, a new hybrid recurrent ANN is proposed by combining simple exponential smoothing and multiplicative neuron models. The proposed hybrid method uses particle swarm optimization to estimate parameters in the hybrid model.

The main contributions of the paper are listed below:

- A new recurrent artificial neural network architecture and its mathematical calculations are proposed.
- A training algorithm is proposed for the new artificial neural network. The training algorithm uses restarting and early stopping strategies.
- The proposed artificial neural network is also a new automatic hybridization method.

The proposed method has the following advantages:

- The proposed hybrid network has simple exponential smoothing and multiplicative neuron model ANN advantages.
- The proposed method is a kind of hybrid method, but it does not need a few parameters estimation stages like other hybrid methods.
- The proposed hybrid network has a recurrent connection, and it can adjust its forecasts by using the feedback of the model.
- The training algorithm does not need derivatives of the objective function and restarting and early stopping strategies prevent overfitting problems of the network.

In the second Sect. 2, the proposed new ANN and its training algorithm are given by using algorithms and flowcharts. In the Sect. 3, the applications of the proposed method are given by using Figures and Tables. In the Sect. 4, conclusions are given. The discussions are given in the Sect. 5. The proposed method is introduced with step-by-step algorithms, a flowchart, a figure and a table.

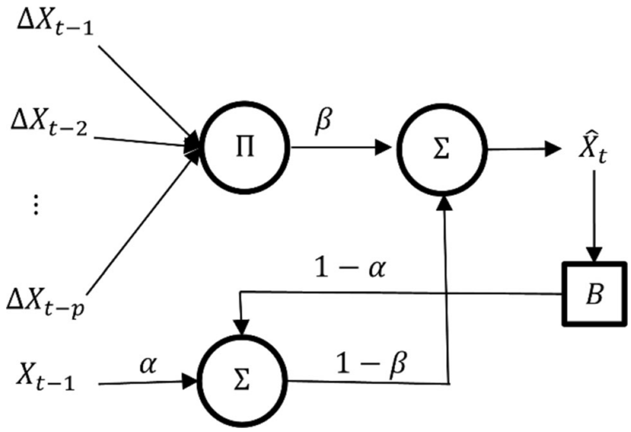

Fig. 1 The architecture of ES-SMN-ANN

2 The proposed method: a hybrid recurrent artificial neural network of simple exponential smoothing and multiplicative neuron model artificial neural network

In this study, a new hybrid artificial neural network (ES-SMN-ANN) is proposed. The new artificial neural network has a new architecture. The architecture contains an exponential smoothing mechanism and multiplicative neuron model. The inputs of the network are lagged variables of the first differenced time series and one-step lagged variable of time series. Nonlinear combinations of lagged variables of the first differenced time series and one-step lagged variable of time series produce outputs of the network. The targets of the network are time series observations.

While the many time series can be forecasted well with simple forecasting models such as simple exponential smoothing, some time series need high-order nonlinear models. Forecasting problem sometimes needs to use simple and complicated model structures in a balance together. The ANN models have generally complex high-order model structures. The proposed network has a combination of simple and high-order nonlinear model structures. The balance of the combination is automatically determined by the training process. The network has a recurrent connection for adding exponential smoothing structure to the new architecture. The recurrent correction creates a new input to the network as $B\widehat{X}_t = \widehat{X}_{t-1}$ (one-step lagged forecast). α smoothing parameter is a weight in the network, and it takes values in $[0, 1]$ interval. If $\alpha = 1$, the model is a combination of random walk processes and multiplicative neuron models. The β parameter is a combination weight for outputs of exponential smoothing and multiplicative neuron model and $0 \le \beta \le 1$. If $\beta = 0$, the proposed network transforms into simple exponential smoothing. If $\beta = 1$, the proposed network transforms into a single multiplicative neuron model artificial neural network. If $0 < \beta < 1$, the output of the network is a combination of exponential smoothing and a multiplicative neuron model. The architecture of the proposed neural network is given in Fig. 1.

The outputs of the proposed network can be calculated by using formulas (2–4) for the given inputs.

$$\text{output}_t^M = \left(1\Big/\left(1+\exp\left(-\prod_{i=1}^{p}\left(w_i\Delta X_{t-p}+b_i\right)\right)\right)\right) \quad (2)$$

$$\text{output}_t^E = \alpha X_{t-1} + (1-\alpha)\widehat{X}_{t-1} \quad (3)$$

$$\text{output}_t^C = \beta \times \text{output}_t^M + (1-\beta) \times \text{output}_t^E \quad (4)$$

Because the logistic activation function is used in the multiplicative neuron model, the inputs and target of the neural network are normalized to $[0, 1]$ interval. The training of the neural network can be achieved by minimizing the least square function.

$$\min_{\theta} \sum_{t=1}^{n} (\text{output}_t - \text{target}_t)^2 \quad (5)$$

θ is a joint parameter vector, and its elements are the weights and biases of the network. The elements of the parameter are given in Eq. (6).

$$\theta = \left(w_1, w_2, \ldots, w_p, b_1, b_2 \ldots, b_p, \alpha, \beta\right) \quad (6)$$

Equation (5) minimization problem is constrained nonlinear optimization problem because $0 \le \beta \le 1$ and $0 \le \alpha \le 1$. This problem can be solved by using particle swarm optimization. Particle swarm optimization can solve numerical optimization problems, and it has many advantages. Particle swarm optimization does not need derivate

Table 1 Positions of a particle in the particle swarm optimization

Positions									
$P_{1,1}$	$P_{1,2}$	$\cdots$	$P_{1,p}$	$P_{1,p+1}$	$P_{1,p+2}$	$\cdots$	$P_{1,2p}$	$P_{1,2p+1}$	$P_{1,2p+2}$
w_1	w_2	$\cdots$	w_p	b_1	b_2	$\cdots$	b_p	α	β

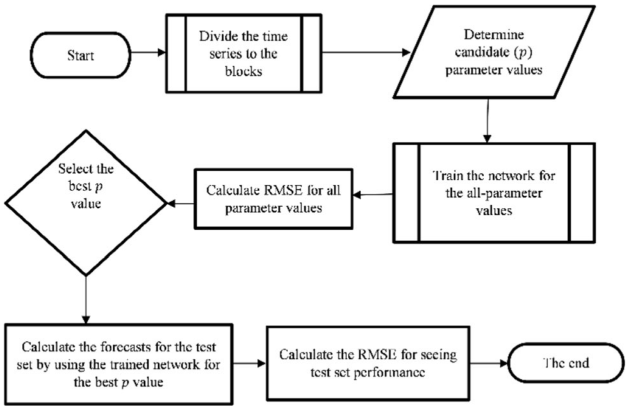

Fig. 2 Flowchart for the application of ES-SMN-ANN for time series forecasting problem

of the objective function like gradient-based methods. The probability of trapping local optimum points is lower than gradient-based learning algorithms. Moreover, two new strategies are used to strengthen the learning algorithm: restarting and early stopping.

An algorithm is given below for the training of the ES-SMN-ANN. The algorithm shows how to apply the training algorithm for a training set.

Algorithm 1. Training of the ES-SMN-ANN.

Step 1. Determine the initial particle swarm optimization parameters.

pn: Number of particles.

$maxitr$: Number of the maximum iterations.

$vmaps$: The limit for the change of particle positions.

$[iw_1, iw_2]$: The interval for the change of inertia weight.

$[c_1^{initial}, c_1^{final}]$: The interval for the change of cognitive coefficient.

$[c_2^{initial}, c_2^{final}]$: The interval for the change of social coefficient.

Rsp : Restarting point.

Fil: Failure index limit.

These parameter values can be taken as $pn = 30, maxitr = 1000, vmaps = 1, [iw_1, iw_2] = [0.4, 0.9]$, $\left[c_1^{initial}, c_1^{final}\right] = [1, 2]$, $\left[c_2^{initial}, c_2^{final}\right] = [1, 2], Rsp = 30, Fil = 20$.

Step 2. Generate initial positions and velocities in the particle swarm optimization.

Positions of a particle are the weight and biases of the ES-SMN-ANN. The positions are presented in Table 1. The number of positions is $2p + 2$.

All positions are generated by uniform distributions with the parameters 0 and 1. The velocities are generated by uniform distributions with the parameters $-vmaps$ and $vmaps$.

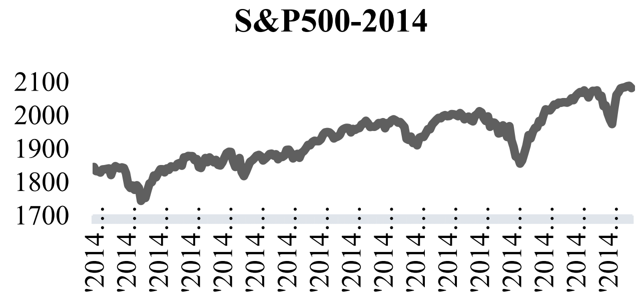

Fig. 3 S&P500 daily time series graph for the 2014 year

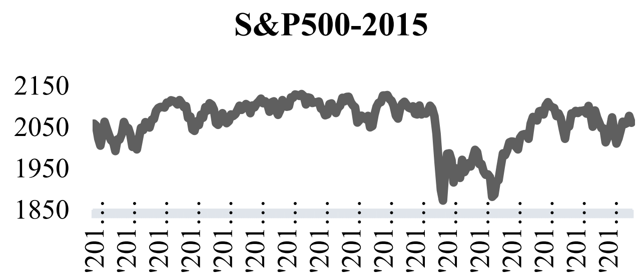

Fig. 4 S&P500 daily time series graph for the 2015 year

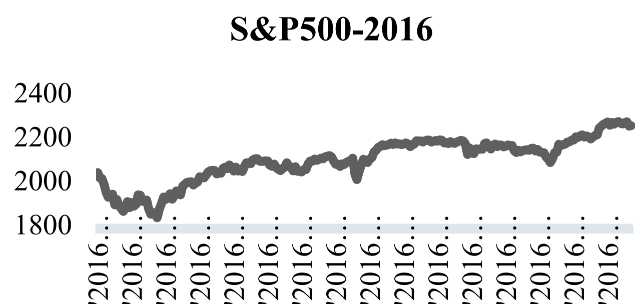

Fig. 5 S&P500 daily time series graph for the 2016 year

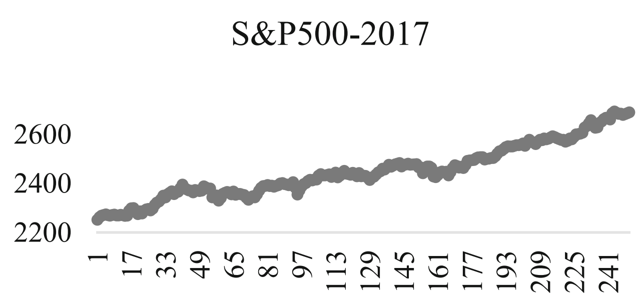

Fig. 6 S&P500 daily time series graph for the 2017 year

This step is similar to determining initial weights and biases in the backpropagation algorithm.

Step 3. Fitness function values are calculated for the particles.

The fitness function is the mean square error, and it can be calculated with the following equation. In this equation, n is the number of learning samples, x_t is target, and $\widehat{x}_t$ is

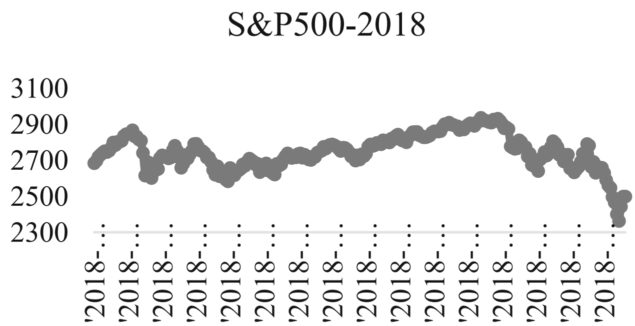

Fig. 7 S&P500 daily time series graph for the 2018 year

the output of the ES-SMN-ANN. $\widehat{x}_t$ values are calculated for each particle by using its positions.

$$f_i = \frac{1}{n}\sum_{t=1}^{n}(x_t - \hat{x}_t)^2 i = 1, 2, \ldots, pn \tag{7}$$

The fitness function is the same as the cost or energy function in the backpropagations algorithm.

Step 4. *pbest* and *gbest* are firstly created according to fitness values. In the first iteration, the *pbest* vector is constituted by the initial positions. The positions of *gbest* are positions of the particle that has the minimum mean square error value. These values are the memory of the training algorithm.

Step 5. Inertia weight, cognitive and social coefficients are calculated by using the following equations.

$$w^{(t)} = (iw_1 - iw_2)\frac{\text{maxitr} - t}{\text{maxitr}} + iw_2 \tag{8}$$

$$c_1^{(t)} = (c_1^{\text{final}} - c_1^{\text{initial}})\frac{t}{\text{maxitr}} + c_1^{\text{initial}} \tag{9}$$

$$c_2^{(t)} = (c_2^{\text{initial}} - c_2^{\text{final}})\frac{\text{maxitr} - t}{\text{maxitr}} + c_2^{\text{final}} \tag{10}$$

Step 6. New velocities and positions are calculated by the following formulas for $i = 1, 2, \ldots, pn; j = 1, 2, \ldots, 2p + 2$

$$\begin{aligned} V_{i,j}^{(t)} = {} & w^{(t)} V_{i,j}^{(t-1)} + c_1^{(t)} r_1 \left(Pbest_{i,j}^{(t)} - P_{i,j}^{(t)}\right) \\ & + c_2^{(t)} r_2 \left(gbest_j^{(t)} - P_{i,j}^{(t)}\right) \end{aligned} \tag{11}$$

$$V_{i,j}^{(t)} = \min\left(v\text{maps}, \max\left(-v\text{maps}, V_{i,j}^{(t)}\right)\right) \tag{12}$$

$$P_{i,j}^{(t)} = P_{i,j}^{(t-1)} + V_{i,j}^{(t)} \tag{13}$$

Moreover, the positions for α and β parameters are restricted to $[0, 1]$ intervals by using the following equation.

$$\begin{aligned} P_{i,j}^{(t)} & = \min\left(1, \max\left(0, P_{i,j}^{(t)}\right)\right) i = 1, 2, \ldots, pn; j \\ & = 2p + 1, 2p + 2 \end{aligned} \tag{14}$$

Table 2 The performance comparison for the proposed method: ES-MNM-ANN by using the S&P500 data set

Year	Methods	Mean	St. dev	Min	Max	Sig. values
2014	LSTM	17.77	1.96	16.47	27.52	< 0.001
	PSGM	18.52	0.19	18.24	19.11	< 0.001
	DSR-ANN	18.28	0.16	17.86	18.62	< 0.001
	ES-MNM-ANN	6.24	0.05	6.15	6.31	
2015	LSTM	30.04	6.21	22.48	41.71	< 0.001
	PSGM	22.81	0.39	21.97	23.69	< 0.001
	DSR-ANN	22.41	0.36	21.64	22.92	< 0.001
	ES-MNM-ANN	15.04	0.13	14.81	15.30	
2016	LSTM	11.23	0.75	10.39	12.44	< 0.001
	PSGM	10.88	0.02	10.83	10.93	< 0.001
	DSR-ANN	10.76	0.05	10.67	10.89	< 0.001
	ES-MNM-ANN	12.73	0.11	12.59	12.95	
2017	LSTM	10.14	0.66	8.76	12.22	< 0.001
	PSGM	22.70	1.16	20.88	25.42	< 0.001
	DSR-ANN	9.66	0.10	9.47	9.91	< 0.001
	ES-MNM-ANN	6.89	0.00	6.88	6.89	
2018	LSTM	57.80	3.76	51.45	66.79	< 0.001
	PSGM	48.14	0.45	47.42	49.22	< 0.001
	DSR-ANN	47.57	0.58	46.45	49.68	< 0.001
	ES-MNM-ANN	31.90	0.55	30.72	32.26	

Table 3 The hyperparameter values of the methods for the S&P500 data set

Years	Methods	Number of inputs	Number of hidden layer nodes	Number of hidden layers
2014	LSTM	5	5	1
	PSGM	5	3	1
	DSR-ANN	4	2	2
	ES-MNM-ANN	2	–	–
2015	LSTM	5	3	1
	PSGM	2	5	1
	DSR-ANN	3	2	1
	ES-MNM-ANN	2	–	–
2016	LSTM	1	4	1
	PSGM	2	5	1
	DSR-ANN	5	5	1
	ES-MNM-ANN	2	–	–
2017	LSTM	5	3	1
	PSGM	5	4	1
	DSR-ANN	5	3	1
	ES-MNM-ANN	2	–	–
2018	LSTM	3	4	1
	PSGM	3	5	1
	DSR-ANN	5	5	2
	ES-MNM-ANN	2	–	–

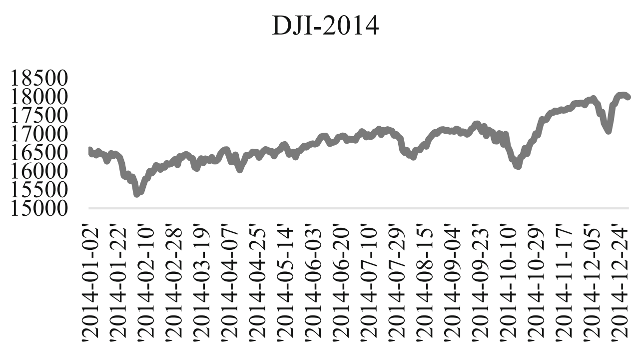

Fig. 8 DJI daily time series graph for the 2014 year

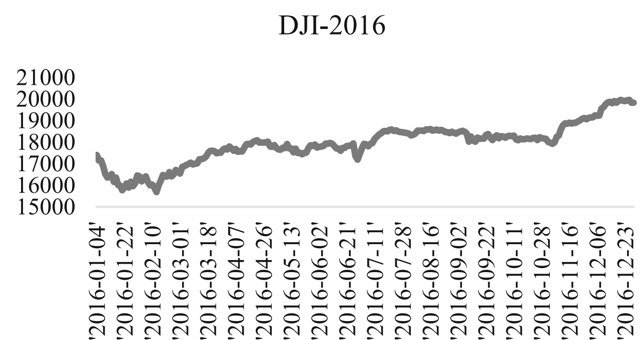

Fig. 10 DJI daily time series graph for the 2016 year

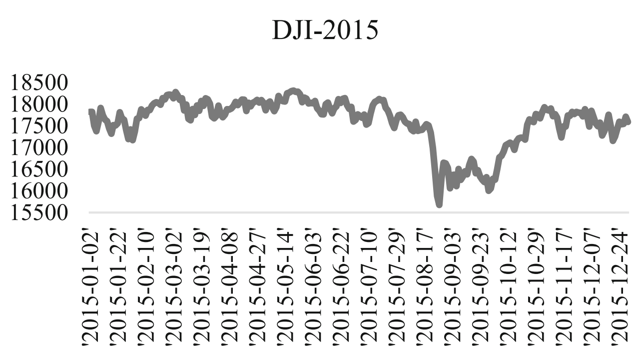

Fig. 9 DJI daily time series graph for the 2015 year

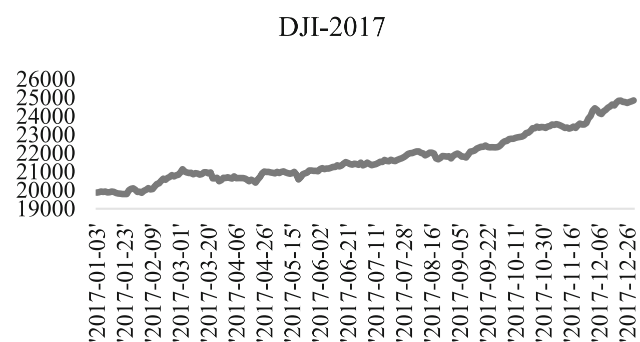

Fig. 11 DJI daily time series graph for the 2017 year

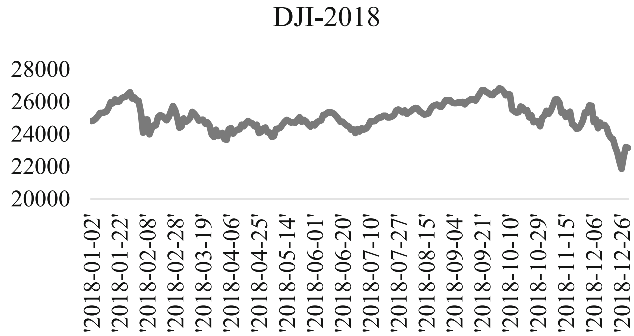

Fig. 12 DJI daily time series graph for the 2018 year

In this step, the solutions are updated like in the back-propagation algorithm.

Step 7. Restarting strategy is checked.

To determine whether the restarting strategy is applied or not, a restarting strategy counter is kept. The restarting strategy counter ($rsc = rsc + 1$) is increased. If the $rsc > Rsp$ then all positions and velocities are generated again and the counter has vanished. *pbest* and *gbest* are not changed, and their values are kept as their previous values. This strategy provides to escape from local optimum points.

Step 8. The fitness values are calculated for the updated particle positions by using (7). Moreover, *pbest* and *gbest* are updated.

Step 9. The early stopping rule is checked as in Eq. (15). *esc* is the counter for the failure. In (15), $fbest^{(t)}$ is the fitness value of *gbest* at iteration t.

$$\text{esc} = \begin{cases} \text{esc} + 1, & \text{if } \dfrac{f\text{best}^{(t)} - f\text{best}^{(t-1)}}{f\text{best}^{(t)}} < 10^{-3} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

If the rule is satisfied ($esc > Fil$), the algorithm is stopped otherwise go to Step 5. This strategy provides to prevent the overfitting problems of the training algorithm.

The proposed neural network is applied to the time series forecasting problem by using the flowchart in Fig. 2. In the applications, the best p value is selected among the candidate.p values by using data set partitions. The ES-SMN-ANN has trained for all candidate p values by using training data set. The best p value is determined according to the validation set performance of the trained networks. The test set performance for the best p value can be used to compare the performance of the proposed method with other methods in the literature.

In the following section, the performance of the proposed method is investigated by using some statistics and hypothesis tests.

Table 4 The performance comparison for the proposed method: ES-MNM-ANN by using the DJI data set

Year	Methods	Mean	St. dev	Min	Max	Sig. values
2014	LSTM	159.94	4.70	148.16	168.91	< 0.001
	PSGM	161.56	0.01	161.54	161.61	< 0.001
	DSR-ANN	158.57	0.77	156.83	160.14	< 0.001
	ES-MNM-ANN	38.57	1.28	36.99	41.34	
2015	LSTM	319.94	40.69	244.03	413.62	< 0.001
	PSGM	198.13	2.63	191.80	201.40	< 0.001
	DSR-ANN	198.84	2.37	193.15	203.33	< 0.001
	ES-MNM-ANN	130.40	0.63	129.72	131.79	
2016	LSTM	111.56	6.51	92.97	120.53	< 0.001
	PSGM	92.27	2.11	87.35	96.17	< 0.001
	DSR-ANN	93.81	0.62	92.31	95.32	< 0.001
	ES-MNM-ANN	103.16	1.54	100.63	105.55	
2017	LSTM	107.37	2.51	102.90	113.45	< 0.001
	PSGM	100.03	1.72	96.02	103.52	< 0.001
	DSR-ANN	104.55	1.29	102.38	108.26	< 0.001
	ES-MNM-ANN	80.22	0.19	80.08	81.07	
2018	LSTM	454.50	9.48	425.71	467.42	< 0.001
	PSGM	453.05	19.75	427.29	506.89	< 0.001
	DSR-ANN	447.49	5.43	436.57	459.99	< 0.001
	ES-MNM-ANN	329.54	7.31	317.14	335.43	

Table 5 The best hyperparameter values of the methods for the DJI data set

Years	Methods	Number of inputs	Number of hidden layer nodes	Number of hidden layers
2014	LSTM	4	3	1
	PSGM	3	1	1
	DSR-ANN	2	5	1
	ES-MNM-ANN	3	–	–
2015	LSTM	5	4	1
	PSGM	2	5	1
	DSR-ANN	4	3	1
	ES-MNM-ANN	2	–	–
2016	LSTM	5	5	1
	PSGM	5	5	1
	DSR-ANN	5	5	1
	ES-MNM-ANN	2	–	–
2017	LSTM	5	4	1
	PSGM	5	3	1
	DSR-ANN	4	2	1
	ES-MNM-ANN	2	–	–
2018	LSTM	5	1	1
	PSGM	5	5	1
	DSR-ANN	5	2	1
	ES-MNM-ANN	2	–	–

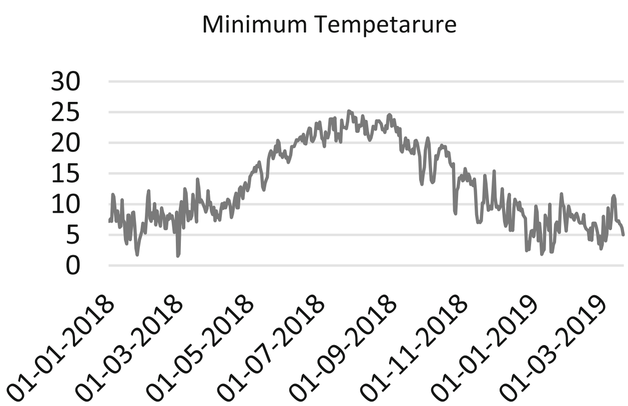

Fig. 13 Giresun city minimum temperature time series graph

Table 6 The performance comparison for the proposed method: ES-MNM-ANN by using the minimum temperature data set

Methods	Mean	St. dev	Min	Max	Sig. values
LSTM	1,77	0,02	1,75	1,77	$p < 0.001$
PSGM	2,32	0,31	1,85	2,32	$p < 0.001$
SRNN	4,30	1,90	2,12	4,30	$p < 0.001$
ES-MNM-ANN	1,43	0,01	1,38	1,43	

3 Applications

The performance of the proposed method is compared with long short-term memory (LSTM), pi-sigma high-order artificial neural network (PS-AN) and deep simple recurrent artificial neural network (DSR-ANN). The stock exchange data sets S&P500 and Dow-Jones, Giresun daily minimum temperature and Giresun wind speed data sets are used in the applications. The applied stock exchange time series' are daily observed in 2014, 2015, 2016, 2017 and 2018 years. The number of examined time series is ten.

In the first part of the applications, S&P 500 stock exchange time series is used. The data set was downloaded from Yahoo Finance Website (https://finance.yahoo.com). The series is called "S&P 500 (GSPC), SNP—SNP Real-Time Price. Currency in USD". The data set is constituted from 5 daily opening prices for the years 2014–2018. Figures of the S&P 500 time series are given in Figs. 3, 4, 5, 6, 7.

Five S&P500 time series are solved by using LSTM, PS-ANN, DSR-ANN and the proposed ANN method. The number of inputs is changed from 1 to 5 by 1 increment in all ANN applications. In the application of LSTM and DSR-ANN, the number of hidden layers is changed from 1 to 5 by 1 increment. In the application of PSANN and the proposed method, the number of hidden layer units is changed from 1 to 5 by 1 increment. Each method is applied 30 times by using random initial weights. In the application, the time series is divided into three parts training, validation and testing data. The validation data were used to select the best model configuration. The test set was used to compare the performance of the methods. The results are given in Table 2. In the last column of Table 2, Wilcoxon signed-rank test significance values are

Table 7 The best hyperparameter values of the methods for minimum temperature data set

Methods	Number of hidden layer nodes	Number of inputs	Number of hidden layers
LSTM	1	4	2
PSGM	5	2	–
DSR-ANN	1	5	1
ES-MNM-ANN	–	5	–

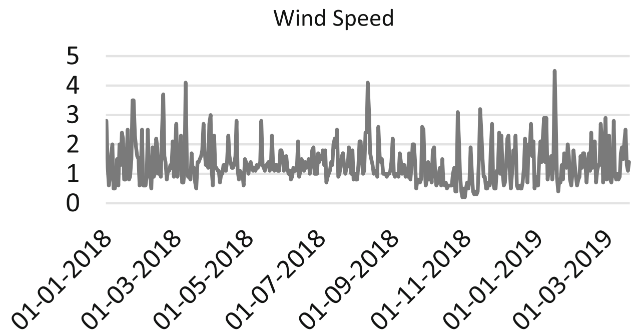

Fig. 14 Giresun city wind speed time series graph

Table 8 The performance comparison for the proposed method: ES-MNM-ANN by using the wind speed data set

Methods	Mean	St. dev	Min	Max	Sig. values
LSTM	0,6724	0,0075	0,6598	0,6724	$p < 0.001$
PSGM	0,7030	0,0096	0,6813	0,7030	$p < 0.001$
SRNN	0,6906	0,0052	0,6848	0,6906	$p < 0.001$
ES-MNM-ANN	0,4881	0,0023	0,4852	0,4881	–

given. Wilcoxon signed-rank test determines whether or not there is a significant difference between the proposed method and others. According to Table 2, the proposed ES-MNM-ANN is better than the other for all calculated statistics except the year 2016. The performance of the ES-MNM-ANN is superior to others except for the year 2016. The performance of the proposed method is the worst in all calculated statistics for the year 2016.

Moreover, the best hyperparameter values for all the methods are given in Table 3.

In the second part of the application, Dow Jones Industrial Average stock exchange time series is used. The data set was downloaded from Yahoo Finance Website (https://finance.yahoo.com). The series is called DJI. The data set is constituted from 5 daily opening prices for the years 2014–2018. Figures of the DJI time series are given in Figs. 8, 9, 10, 11, 12.

Five DJI time series are solved by using LSTM, PS-ANN, DSR-ANN and the proposed ANN method. The parameter configuration is the same as in previous applications. The results are given in Table 4. According to Table 4, the proposed ES-MNM-ANN is better than the other for all calculated statistics except the year 2016. The performance of the ES-MNM-ANN is superior to others except for the year 2016. According to Wilcoxon signed-rank test results in Table 4, the proposed method is significantly different from the other methods.

The best hyperparameter values are given in Table 5.

The third application is made on the minimum temperature data set. The data set is daily observed in Giresun city in Turkey between 01.01.2018 and 19.03.2019. The graph of the time series is given in Fig. 13.

The validation and test set lengths are taken as 30. The hyperparameter selection process is made like in the first application. The application results are given in Table 6. The proposed method produced the best results, and the differences are significant according to Wilcoxon signed-rank test results.

The best hyperparameter values of the methods are given in Table 7.

The last application is made on the wind speed data set. The data set is daily observed in Giresun city in Turkey between 01.01.2018 and 19.03.2019. The graph of the time series is given in Fig. 14.

The validation and test set lengths are taken as 30. The hyperparameter selection process is made like in the first application. The application results are given in Table 8. The proposed method produced the best results, and the differences are significant according to Wilcoxon signed-rank test results.

The best hyperparameter values of the methods are given in Table 9.

Table 9 The best hyperparameter values of the methods for wind speed data set

Methods	Number of hidden layer nodes	Number of inputs	Number of hidden layers
LSTM	2	3	2
PSGM	1	1	1
DSR-ANN	1	2	1
ES-MNM-ANN	–	4	–

4 Conclusions

In this paper, simple exponential smoothing and multiplicative neuron model artificial neural network are combined, and a new recurrent artificial neural network architecture is created. The proposed network uses simple and high-order nonlinear model structures together. Moreover, the balance of these structures can automatically be provided by particle swarm optimization. Moreover, a training algorithm based on particle swarm optimization is proposed for the new network. The proposed training algorithm handled overfitting and local optimum problems by using restarting and early stopping strategies. The proposed network creates a balance between a multiplicative neuron model and simple exponential smoothing by using a weight parameter. The proposed method optimized the weight of the multiplicative neuron model and simple exponential smoothing based on the trained time series data by using particle swarm optimization.

In the application, two important stock exchange data sets, temperature data set and wind speed data set, were used. It is shown that the proposed network produced more accurate forecasts than the most popular ANN forecasting methods such as LSTM. Moreover, the proposed method drastically decreased the RMSE statistics.

5 Discussions

The proposed method has a superior forecasting performance against the selected benchmark methods. The Wilcoxon signed-rank tests are applied to compare the performance of the proposed method with the benchmark methods. The proposed method has a smaller location parameter of RMSE examples than the other methods according to Wilcoxon signed-rank tests results. For the stock exchange data sets, the proposed method can work better with two lagged variables as inputs. For the meteorological time series, the number of inputs is four or five, so the model needs more inputs. All statistics (mean, std.dev. min. and max.) of RMSE values for the proposed method are generally smaller than the other methods.

The proposed method can be used for forecasting non-seasonal time series. This is a limitation of the method. The proposed forecasting method can be improved by using seasonal decomposition techniques or seasonal architecture adding.

In the future studies, new artificial neural network architectures can be created by using other exponential smoothing methods. The new artificial neuron models can be used instead of the multiplicative neuron model for creating a new ANN model. The method can be modified for modelling seasonal time series. MNM-ANN can be hybridized with Holt and Winter's exponential smoothing methods. Moreover, new hybrid networks can be created by using other ANN types such as deep artificial neural networks. Another research direction is that a new robust training algorithm can be proposed for the ANN model against the outliers in outputs and targets. The bootstrap methods can be used to create an ensemble of the proposed ANNs.

Declarations

Conflict of interest The authors declared that they have no conflicts of interest in this study.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s00521-022-07753-w.

References

1. Jaddi NS, Abdullah S (2018) Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting. Eng Appl Artif Intell 67:246–259
2. Maté C, Jimeńez L (2021) Forecasting exchange rates with the MLP: new empirical insight on one multi-layer perceptron for interval time series (ITS). Eng Appl Artif Intell 104:104358
3. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
4. Chung H, Shin KS (2018) Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability 10(10):3765
5. Qiu YY, Zhang Q, Lei M (2020) Forecasting the railway freight volume in China based on combined PSOLSTM model. J Phys Conf Series IOP Publishing 1651(1):012029
6. Gundu V, Simon SP (2021) PSO–LSTM for short-term forecast of heterogeneous time series electricity price signals. J Ambient Intell Humaniz Comput 12:2375–2385
7. Wang Z, Zeng Y-R, Wang S, Wang L (2019) Optimizing echo state network with backtracking search optimization algorithm for time series forecasting. Eng Appl Artif Intell 81:117–132
8. Gao MY, Zhang N, Shen SL, Zhou A (2020) Real-time dynamic earth-pressure regulation model for shield tunneling by integrating GRU Deep learning method with GA optimization. IEEE Access 8:64310–64323
9. Bas E, Egrioglu E, Kolemen E (2022) Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization. Granular Computing 7:411–420
10. Peng L, Lv S-X, Wang L, Wang Z-Y (2021) Effective electricity load forecasting using enhanced double-reservoir echo state network. Eng Appl Artif Intell 99:104132
11. Fang X, Yuan Z (2019) Performance enhancing techniques for deep learning models in time series forecasting. Eng Appl Artif Intell 85:533–542
12. Mohammed NA, Al-Bazi A (2022) An adaptive backpropagation algorithm for long-term electricity load forecasting. Neural Comput Appl 34(1):477–491

13. Michell K, Kristjanpoller W, Minutolo MC (2022) Electrical consumption forecasting: a framework for high frequency data. Neural Comput Appl. https://doi.org/10.1007/s00521-021-06735-8
14. Zouaidia K, Ghanemi S, Rais MS, Bougueroua L, Katarzyna WW (2021) Hybrid intelligent framework for one-day ahead wind speed forecasting. Neural Comput Appl 33(23):16591–16608
15. Readshaw J, Giani S (2021) Using company-specific headlines and convolutional neural networks to predict stock fluctuations. Neural Comput Appl 33(24):17353–17367
16. Oliveira DD, Rampinelli M, Tozatto GZ, Andreão RV, Müller SMT (2021) Forecasting vehicular traffic flow using MLP and LSTM. Neural Comput Appl 33(24):17245–17256
17. Yadav RN, Kalra PK, John J (2007) Time series prediction with single multiplicative neuron model. Appl Soft Comput 7:1157–1163
18. Cui H, Feng J, Guo J, Wang T (2015) A novel single multiplicative neuron model trained by an improved glowworm swarm optimization algorithm for time series prediction. Knowl-Based Syst 88:195–209
19. Egrioglu E, Yolcu U, Aladag CH, Bas E (2015) Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting. Neural Process Lett 41:249–258
20. Pan W, Zhang L, Shen C (2021) Data-driven time series prediction based on multiplicative neuron model artificial neuron network. Appl Soft Comput 104:107179
21. Pan W, Feng L, Zhang L, Cai L, Shen L (2021) Time-series interval prediction under uncertainty using modified double multiplicative neuron network. Expert Syst Appl 184(1):115478
22. Shin Y, Gosh J (1991) The Pi-sigma Network: An efficient higher-order neural network for pattern classification and function approximation. In IJCNN-91-Seattle International Joint Conference on Neural Networks, IEEE 1, pp. 13–18
23. Tawfik H, Liatsis P (1997) Prediction of non-linear time-series using higher-order neural networks. Proceeding IWSSIP'97 Conference, Poznan, Poland
24. Ghazali R, Hussain A, El-Deredy W (2006) Application of ridge polynomial neural networks to financial time series prediction. In: 2006 International Joint Conference on Neural Networks, 913–20
25. Ghazali R, Hussain AJ, Liatsis P, Tawfik H (2008) The application of ridge polynomial neural network to multi-step ahead financial time series prediction. Neural Comput Appl 17(3):311–323
26. Todo Y, Tamura H, Yamashita K, Tang Z (2014) Unsupervised learnable neuron model with nonlinear interaction on dendrites. Neural Netw 60:96–103
27. Ji J, Gao S, Cheng J, Tang Z, Todo Y (2016) An approximate logic neuron model with a dendritic structure. Neurocomputing 173:1775–1783
28. Sha Z, Hu L, Todo Y, Ji J, Gao SC, Tang Z (2015) A breast cancer classifier using a neuron model with dendritic nonlinearity. IEICE Trans Inf Syst E98-D 7: 1365–1376
29. Zhou T, Gao S, Wang J, Chu C, Todo Y, Tang Z (2016) Financial time series prediction using a dendritic neuron model. Knowl-Based Syst 105:214–224
30. Hajirahimi Z, Khashei M (2019) Hybrid structures in time series modelling and forecasting: a review. Eng Appl Artif Intell 86:83–106
31. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
32. Aladag CH, Egrioglu E, Kadilar C (2009) Forecasting nonlinear time series with a hybrid methodology. Appl Math Lett 22:1467–1470
33. Mohammadi R, Ghomi SMTF, Zeinali F (2014) A new hybrid evolutionary based RBF networks method for forecasting time series: a case study of forecasting emergency supply demand time series. Eng Appl Artif Intell 36:204–214
34. Panigrahi S, Behera HS (2017) A hybrid ETS–ANN model for time series forecasting. Eng Appl Artif Intell 66:49–59
35. Kouziokas GN (2020) A new W-SVM kernel combining PSO-neural network transformed vector and Bayesian optimized SVM in GDP forecasting. Eng Appl Artif Intell 92:103650
36. Bas E, Grosan C, Egrioglu E, Yolcu U (2018) High order fuzzy time series method based on pi-sigma neural network. Eng Appl Artif Intell 72:350–356
37. Egrioglu E, Bas E, Yolcu OC, Yolcu U (2019) Intuitionistic time series fuzzy inference system. Eng Appl Artif Intell 82:175–183
38. Egrioglu E, Bas E, Yolcu U, Chen MY (2020) Picture fuzzy time series: Defining, modelling and creating a new forecasting method. Eng Appl Artif Intell 88:103367
39. Yolcu U, Egrioglu E, Aladag CH (2013) A new linear and nonlinear artificial neural network model for time series forecasting. Decis Support Syst 54(3):1340–1347
40. Sarıca B, Eğrioğlu E, Aşıkgil B (2018) A new hybrid method for time series forecasting: AR–ANFIS. Neural Comput Appl 29(3):749–760
41. Shohan MJA, Faruque MO, Foo SY (2022) Forecasting of electric load using a hybrid LSTM-Neural prophet model. Energies 15(6):2158
42. Hadwan M, Al-Maqaleh BM, Al-Badani FN, Khan RU, Al-Hagery MA (2022) A hybrid neural network and box-jenkins models for time series forecasting. Comput Mater Continua 70(3):4829–4845
43. Hu S, Xiang Y, Huo D, Jawad S, Liu J (2021) An improved deep belief network based hybrid forecasting method for wind power. Energy 224:120185