Neural Computing and Applications
https://doi.org/10.1007/s00521-023-08240-6

ORIGINAL ARTICLE

A robust training of dendritic neuron model neural network for time series prediction

Ayşe Yilmaz[1] · Ufuk Yolcu[2]

Received: 20 June 2022 / Accepted: 6 January 2023

Abstract

Many prediction methods proposed in the literature can be concerned under two main headings: probabilistic and non-probabilistic methods. In particular, as a kind of non-probabilistic model, artificial neural networks (ANNs), having different properties, have been commonly and effectively used in the literature. Some ANNs operate the additive aggregation function in the structure of their neuron models, while others employ the multiplicative aggregation function. Recently proposed dendritic neural networks also have both additional and multiplicative neuron models. The prediction performance of such an artificial neural network will inevitably be negatively affected by the outliers that the time series of interest may contain due to the neuron model in its structure. This study, for the training of a dendritic neural network, presents a robust learning algorithm. The presented robust algorithm is the first for the training of DNM in the literature as far as is known and uses Huber's loss function as the fitness function. The iterative process of the robust learning algorithm is carried out by particle swarm optimization. The productivity and efficiency of the suggested learning algorithm were evaluated by analysing different real-life time series. All analyses were performed with original and contaminated data sets under different scenarios. The R-DNM has the best performance for the original data sets with a value of 2.95% in the ABC time series, while the FTSE showed the best performance in approximately 27% and the second best in 33% of all analyses. The proposed R-DNM has been the least affected by outliers in almost all scenarios for contaminated ABC data sets. Moreover, it has been the least affected model by outliers in approximately 71% of the 90 analyses performed for the contaminated FTSE time series. The obtained results show that the dendritic artificial neural network trained by the proposed robust learning algorithm produces the satisfactory predictive results in the analysis of time series with and without outliers.

Keywords Dendritic neuron model · Robust learning algorithm · Huber's loss function · Time series prediction · Particle swarm optimization

1 Introduction

Events occurring in nature have been a matter of curiosity from the very beginning. People tried to understand these events with their thoughts, minds, and feelings, and they made progress as technology allowed, and they developed many methods for this purpose.

Many of these methods include prediction models that are frequently used in time series analysis, which have a very important place in decision-making, policy, and, strategy determination for the future. These prediction models can be divided into two main groups, probabilistic and non-probabilistic models. Probability models consist of traditional methods that require the provision of strict assumptions such as the number of observations, stationarity, and certain standard models. Thus, in cases of these assumptions are not met, they may be insufficient in predicting real-world time series. In the past few decades, non-probabilistic models, especially computational artificial neural network (ANN) models, have been successfully used to analyse almost all-time series since they do not contain such strict assumptions.

✉ Ufuk Yolcu
ufuk.yolcu@marmara.edu.tr

[1] Department of Statistics, Ondokuz Mayıs Univesity, Samsun, Turkey

[2] Department of Statistics, Marmara University, Istanbul, Turkey

The ANNs' structure consists of neurons. The achieving process of the best values for the weights, which connect the neurons and the processed data, is called training the network. The most important feature of an ANN, perhaps, is that it is trainable and learnable by using a data source. The training process of the network is carried out by a learning algorithm and is based on the principle of a step-by-step approach to the best values of the weights according to a certain criterion.

McCulloch and Pitts introduced the first ANN model [1]. The first ANN model consisted only of input and output layers. These networks, which have multiple inputs and a single output unit, cannot be used to solve problems involving nonlinear relationships since they use a linear output function. Rumelhart et al. developed the multilayer perceptron (MLP) model and realized the solution to nonlinear problems [2]. This model is also known as a backpropagation (BP) network. In addition to having an input and an output layer, MLP also contains one or more hidden layers.

Since then, including time series prediction, numerous studies on various problems have presented different types of ANN structures. Broomhead investigated the effect of "learning" in adaptive layered networks for the process of fitting data to high-dimensional surfaces Hill and Basu and Ho comparatively evaluated the performance of adaptive procedures and linear programming methods, for real-life data sets [3, 4]. Labib proposed a model consisting of a single neuron having recognizing ability of nonlinear structures, such as the XOR problem [5]. Plate worked on the new neuron structure, sigma-pi units [6]. Zhang et al. [7] proposed a programmable single-neuron local rational logic, effectively conducted by presynaptic activities. Afterwards, the generalized mean neuron model (GNM) [8], geometric mean neuron model (GeoNM) [9], and single multiplicative neuron model (SMNM) [10], bronchospastic plastic neuron model [11] were introduced. The ANNs mentioned above, in terms of neuron models, use either of the diverse aggregation functions with the additive, multiplicative or mean-based. ANNs with all these neuron models, in time series prediction, are often successfully used. Also, as a different kind of ANN, the sigma-pi neural network (SPNN), in its architectural structure, operates multiplicative and additive-based aggregation functions together [12]. Subsequently, the studies have been presented, focusing on the training of SPNN [13, 14]. Nonetheless, these models depend on the configuration of a McCulloch-Pitts neuron model that uses weights between synapses [15]. Furthermore, another type of ANN, the dendritic neuron model neural network (DNM) [16], is adapted from a dendritic mechanism. DNM also operates additive and multiplicative neuron models together. Attia et al. suggested a generalized mean single multiplicative neuron model based on a nonlinear aggregation function [17]. Gao et al. [18] paying attention to the case of the nonlinearity of synapses, developed a new dendritic neuron model. Zhou et al. [19] suggested the use of a new dendritic neuron model to analyse some financial time series.

The different kinds of ANNs having different neuron models have been used in the time series prediction literature. The outlier(s) negatively affect the performance of ANNs, just as in other application areas, in time series prediction problems [20]. Different approaches have been created with the aim of reducing the negative effects of outliers on ANNs' performance and obtaining reliable and effective results in cases where the data has outliers or values. The models based on these approaches are called "*robust*" models.

The robust ANN models can be reviewed in two categories: (1) ANN models having robust learning algorithms [20–24], and (2) ANN models with a robust architectural structure containing different neuron models [25–27].

The approach addressed in this study provides a perspective on the first category of robust neural networks. Therefore, a robust learning algorithm is first proposed in the literature in order to use a dendritic neuron model neural network (DNM), having an architectural structure that uses both additive and multiplicative functions together, in the prediction of time series with the outlier(s). The robustness of the learning algorithm is ensured by using Huber's loss function as the algorithm fitness function. In this respect, DNM with the proposed robust learning algorithm can be called a robust-dendritic neuron model neural network (R-DNM). The iterative process of the robust learning algorithm is performed by modified particle swarm optimization (MPSO) which is a population-based stochastic optimization algorithm created for developing solutions to continuous and discrete optimization problems. The performance of the robust learning algorithm was evaluated by analysing the time series, frequently used in the literature. In this direction, the original time series and their contaminated derivatives with an outlier(s) were utilized. Thus, it is aimed to reveal the performance of the R-DNM in the prediction of time series both with outlier(s) and without outlier(s). The obtained comparative findings prove that R-DNM demonstrates competitive predictive performance for both the original and the contaminated time series. In addition, it has been observed that performance in the case of the outlier is little or nothing affected by outliers and produces analysis results very close to the original data analysis.

As for the organization of the rest of the article: the next section summarizes the related literature. Then the third section presents the dendritic neuron model, its general structure, the working principle, and basic features; the proposed RLA for the training of DNM, and the training process is presented in the fourth section in a depth; in the fifth section, by realizing the analysis of the 5-year daily Financial Times Stock Exchange (FTSE) time series covering the years 2010–2014 and new contaminated time series created by injecting outliers of different sizes and numbers into the original FTSE, a comprehensive and comparative implementation study is given, and the obtained findings from them are summarized via some tables and graphs; finally, in the last section, all the results are evaluated and discussed together and some ideas that can shed light on future studies are presented.

2 Related literature

The robust concept in the NN literature is based on the smallest absolute deviations proposed by Boscovich in 1757 [28] or the L1 estimation method. Chen and Jain [21], Lei et al. [29], Hsiao [30], and Bas et al. [20], using the M-type estimator of Huber (1964) as a conformity value in the ANN learning algorithm, established a robust learning algorithm (RLA). El Melegy et al. and Rusiecki proposed an RLA that uses a fitness function based on the least median squares [22, 31]. Also, Thomas et al. developed an RLA for the MLP [32]. Apart from these studies, Chen and Jain created a robust backpropagation learning algorithm having the rejecting ability to gross errors engendering by the noise effects during the learning process. [21]. Connor et al. put forward an RLA for recurrent neural networks [33]. For the robotization of the training process of the radial basis function neural networks, Sanchez presented a new and efficient robust learning method [34]. Neubauer, as a learning tool used for the MLP with discretized synaptic weights, evaluated the performance of the Fletcher–Reeves algorithm and the genetic algorithm comparatively [24]. Wang et al., for a category of nonlinear uncertain state-delayed systems, dealt with the problem of the robust control procedure [35]. Mili and Coakley introduced a new class of robust predictors called D-estimators. The estimators of Mili and Coakley are the generalization of the least median squares and least trimmed squares estimators [36]. Fan and Chow proposed a robust kernel principal component analysis having the ability of high accuracy achievement even in the case of noises [37][37]. Ham and McDowall designed a robust partial least-squares (PLS) regression neural network model for creating a statistical calibration model [23]. Lei et al. introduced the concepts of polarity and quality of training data and proposed two robust learning algorithms based on this [29]. Shi and Shue proposed a robust control approach for a system based on uncertain discrete time [39]. Sinha and Wiens introduced a basic concept related to an approximately nonlinear regression model. They also suggested sequential design methodologies in the case of an incorrect parametric form for the fitted model [40]. Allende et al. used feed-forward neural networks (FFNN) as nonlinear autoregressive (NAR) models [41]. Pernía-Espinoza et al., to deal with the outliers, produced the TAO-RLA by combining the t-estimates of the nonlinear regression model with the backpropagation algorithm [42]. Hans and Udluft suggested using ensembles of networks to produce an RLA and get close-optimal results more reliably [43]. Song et al. [44] presented a sufficient and robust feature-based movement estimation procedure. When the literature is examined as a whole, it is seen that different robust learning algorithms have been developed for different ANN types. Especially in recent years, there are also types of ANNs with robust architectural structures used in time series prediction. Aladag et al. introduced a robust ANN based on the median neuron model [27]. Yolcu et al. proposed an MLP-ANN based on the trimmed mean neuron model [25]. Egrioglu et al. also proposed a robust ANN architecture based on the median neuron model for Pi-Sigma neural network [26]. When the studies in recent years are examined; Liu et al. presented a dimension-reducing layer that preserves high-dimensional neighbourhoods across the entire manifold for investigating the neural network robustness against adversarial attack [45]. Nakamura, and Fukagata, [46] examined the effectiveness of several considerable approaches for model training to gain more robustness. This paper mainly focused on robust adaptive neural networks for a class of time-delay parabolic systems [47]. Shi et al. [48] developed an approach to increase CNN resilience against adversarial attacks on medical images.

3 Dendritic neuron model

A realistic computational form of a single neuron with a synaptic nonlinear structure in dendritic structures has been demonstrated [19]. DNM can calculate linearly inseparable functions by modelling the nonlinear structure with the sigmoid function. Thus, DNM approximates it to any complex continuous function. Nonlinear interaction in dendritic structure is expressed by Boolean logic. AND (together), OR (discrimination), or NOT (deficiency) is used instead of complex functions [19].

The first artificial neural network model, also known as the linear threshold gate model, has a specific mathematical description of the McCulloch–Pitts neuron model. However, this model is fairly easy and can only create a two-element mixed threshold value. Koch et al. [16] built the most primitive form of the dendritic mechanism based on the nerve cell. By using logical operators, this model transfers the input signals at the roots to the soma. Then,

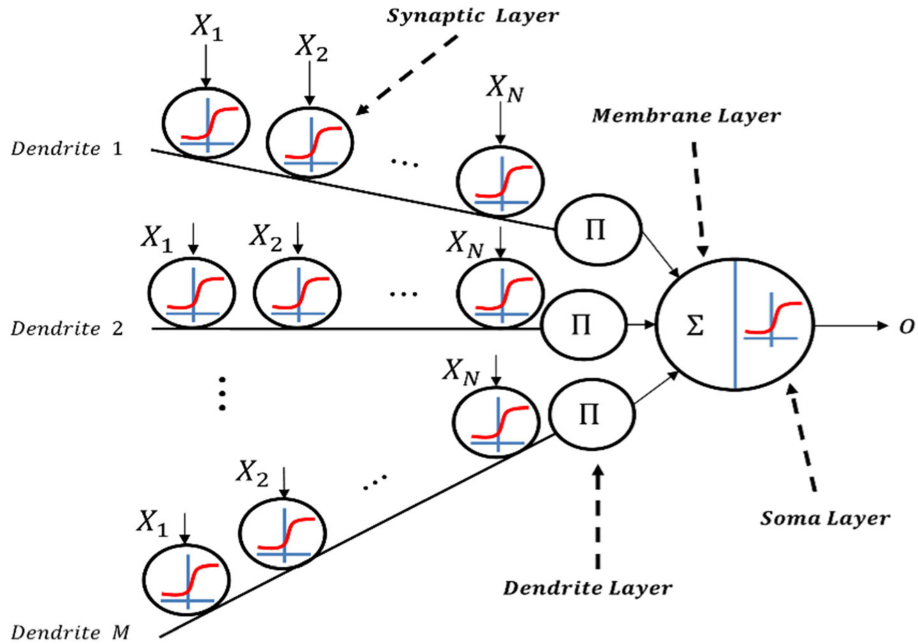


Fig. 1 A hypothetic architecture of DNM

from there, it gets a logical operator-dependent result. Based on the traditional Koch model, the logical operation of a neuron can be generalized with the steps given below. [19].

1. The input and the output classes are represented by values 0 or 1.
2. The shape of the dendrite is random. Dendrites, in the beginning, are determined by an arbitrary decision.
3. In the first stage, there is a fundamental interaction between all synapses.
4. A logic network can express nonlinear interactions in the dendrite.
5. For a learning period, the shape of the dendrite and the types of interaction between synapses can be determined in an appropriate situation [16].

The main features of DNM can be summarized as follows [18]:

- Its architectural structure is similar to both Pi-Sigma and multiplicative neuron model. It has a multilayered architectural structure. The signals are transmitted forward
- The multiplicative function is simple and widely used in nonlinear neural networks
- Uses the sigmoid function
- Depending on the values of the parameters in the synapse, the synapse output, which is useful for determining the morphology of a neuron, successfully represents stimulating, inhibitory, 0-constant, and 1-constant.

The architectural structure of the DNM is composed of four layers. A sigmoid function processes the signals incoming into the synaptic layer. Then the signals processed by the sigmoid function are transmitted to the dendrite layer. In the dendrite layer, the output produced for each input drives into the multiplicative function. In the next layer, the membrane layer, the outputs of the dendrite layer are processed using an additive function. The processed signals, coming to the soma layer, are handled by the sigmoid activation function [18]. A hypothetic architecture of DNM is illustrated in Fig. 1.

3.1 Synaptic layer

Synapses connect two dendrites or between a dendrite and a soma. Signals are fed forward. A synaptic layer is expressed in Eq. (1).

$$Y_{ij} = \frac{1}{1 + \exp\left(-k\left(w_{ij}x_i - \theta_{ij}\right)\right)} \tag{1}$$

Here, Y_{ij} represents the output corresponding to ith ($i = 1, 2, \cdots, N$) synaptic input in the jth $j = 1, 2, \cdots, M$) synaptic layer. k is a positive constant. x_i is ith input of a synapse and$x_i \in [0, 1]$. w_{ij} and θ_{ij} represent the weight and threshold, respectively.

3.2 Dendritic layer

In the dendritic layer, the outputs of the synapses in various synaptic layers are subjected to the multiplication function. It is expressed by Eq. (2)

$$Z_j = \prod_{i=1}^{N} Y_{ij} \tag{2}$$

Multiplication operations are successful in nonlinear synapses. For this reason, the model uses multiplicative operators in the dendrite layer. Since the input and output of the dendrite are associated with 0 or 1, the multiplication is logically similar to "and".

3.3 Membrane layer

The membrane layer connects the signals from the dendrite branches. Signals received from the dendrite branches are calculated in the additive function. Summation is logically similar to "*or*". The resulting output passes to the other layer to activate the soma layer. It is formulated by Eq. (3).

$$V = \sum_{j=1}^{N} Z_j \tag{3}$$

3.4 Soma layer

Finally, the current output of the network enters a sigmoid (logistic) activation function processed in the soma layer. If the final output is above the threshold value, transmission occurs. The sigmoid function used can be expressed by Eq. (4).

$$o = \frac{1}{1 + \exp(-k_s(V - \theta_s))} \tag{4}$$

Here, k_s is a positive constant and θ_s denotes a threshold value changed in the range of $[0, 1]$.

4 Robust-dendritic neuron model trained by PSO

Different ANN types, as a prediction tool, have been often used in time series prediction problems. MLP, SMNM-NN, and PS-NN have an in-depth working area in the time series prediction literature. Just a few studies using DNM, however, are available. Whereas all these ANNs produce successful results in time series problems, they also contain some problems within themselves to be solved. One of the most important of these is the negative effects of the outlier(s) that the time series may contain on the performance of the prediction tool. To eliminate such a problem, two different approaches have been discussed in the literature. The first approach is to use a robust training algorithm for network training. In this approach during the training of the neural network, a robust function is used as the fitness function. Thus, it is aimed to reduce the negative effect of the outlier(s) on the fitness function values. In the other approach, neuron models using robust aggregation functions are used to eliminate or at least reduce the effects of the outlier(s). This approach will create a robust ANN architecture. In the literature, although both approaches have been introduced for MLP, SMNM-ANN, and PS-ANN, a robust learning algorithm or architectural structure has not yet been revealed for DNM. This study focuses on the first approach and proposes a robust learning algorithm for DNM in the time series prediction problem. In this context, Huber's loss function is used as a fitness function during the training process of the network, thereby the negative effect of an outlier error that may occur due to the outlier value(s) on the training algorithm is reduced. From this perspective, the main contributions and distinctive features of this study are; (1) the recommendation of a robust learning algorithm for the training of DNM for the first time and also (2) Performing a DNM training by MPSO for the first time in the time series prediction literature.

The robust learning algorithm, developed for DNM, can be given step by step as follows.

Algorithm:

Step 1. Establish the parameters of the training process.

pn: # of particle of the swarm.

c_1: Cognitive coefficient.

c_2: Social coefficient.

max*itr*: Maximum number of repetitions.

w: Inertia weight.

(c_{1i}, c_{1f}): The interval which contains the possible values of c_1

(c_{2i}, c_{2f}): The interval which contains the possible values of c_2

(w_1, w_2): The interval which contains the possible values of w

N: # of synaptic input of synaptic layer.

M: # of dendrite.

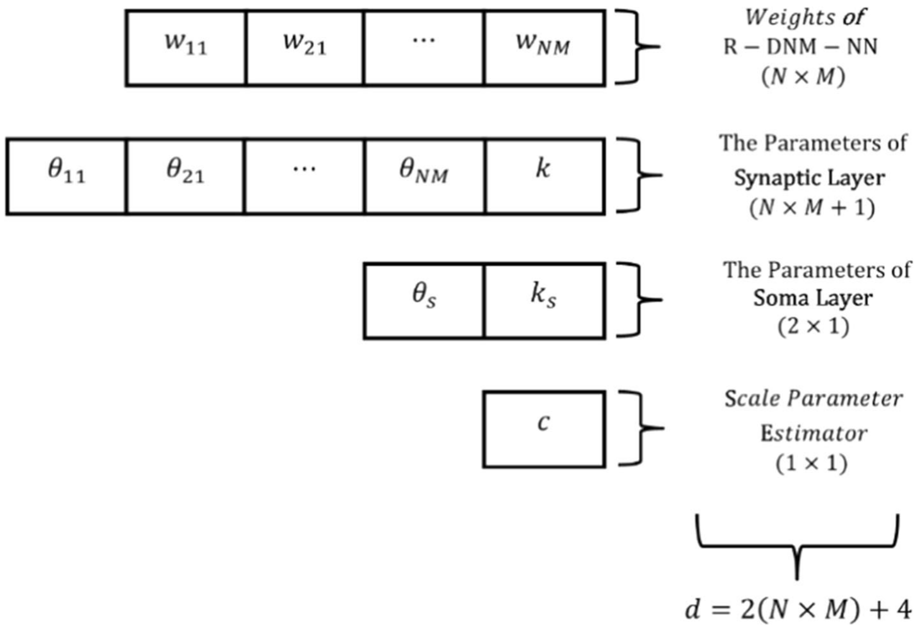
ntest: # of observations in the out-of-sample.

n: # of observations of time series.

Step 2 Create the positions and the velocities of each particle.

The positions of each particle consist of the weights $(w_{ij}, i = 1, 2, \cdots, N; j = 1, 2, \cdots, M)$ of the network and the constants in the synaptic (k and θ_{ij}) and the soma (k_s andθ_s) layers, respectively. In addition, the scale parameter estimator (c) of Huber's loss function is another position of the particle. Figure 2 presents the structure containing the positions of a particle. Each particle has $d = 2NM + 4$ positions. $N \times M$ of these positions are wights of neural network (w_{ij}). $N \times M + 1$ of them are θ parameters. Two of them are k parameters of synaptic and soma layers, Finally, one of them is the estimator of scale parameter (c). The

Fig. 2 The positions of a particle



initial values of the first $2NM + 3$ positions are randomly generated from the uniform distribution with the parameters of $(0, 1)$. For the last position, by considering the studies available in the literature, the initial value is randomly generated from the uniform distribution with the parameters of $(6, 10)$. Moreover, the initial values of velocities are produced from a uniform distribution with the parameters of $(-\text{vmaps}, \text{vmaps})$.

Step 3 Calculate the fitness function values for each particle.

The produced outputs of the network by utilizing the weight and parameter values represented by the positions of each particle, and the target values are used to get the Huber's loss function value. In this stage, first, the error value for tth learning sample e_t, and the beta parameter can be obtained as in Eqs. (5) and (6), respectively.

$$e_t = d_t - o_t \tag{5}$$

$$\text{beta} = c \times \underset{t}{\text{median}}\left(e_t - \underset{t}{\text{median}}(e_t)\right) \tag{6}$$

Here, d_t and o_t are the target and output values for tth learning sample, respectively. By using these values, Huber's error function can be calculated as in Eq. (7).

$$\widetilde{e_t} = \begin{cases} e_t^2/2, & \text{if } |e_t| \leq \text{beta} \\ \text{eta}\left|e_t - {\text{beta}^2}/{2}\right|, & \text{otherwise} \end{cases} \tag{7}$$

Therefore, for ith particle, Huber's loss function as the fitness function is obtained by using Eq. (8).

$$\text{fitness}_i = \sum_{t=1}^{n} \widetilde{e_t} \tag{8}$$

Step 4 Determine the Pbest and the Gbest

Pbest depicts the position values of the best individual performance of the relevant particle among the existing repetitions. Gbest represents the positions of the best performing particle among the available iterations across the entire swarm. The measure of the best performance is the fitness function with the lowest value.

Step 5 Update the positions and the velocities.

Since velocities and positions can be updated depending on cognitive and social coefficients and inertia parameters, updating these values is performed as given in Eqs. (9–11).

$$c_1^t = (c_{1f} - c_{1i})\frac{t}{\text{maxi}tr} + c_{1i} \tag{9}$$

$$c_2^t = (c_{2f} - c_{2i})\frac{t}{\text{maxi}tr} + c_{2i} \tag{10}$$

$$w^t = (w_2 - w_1)\frac{\text{maxi}tr - t}{\text{maxi}tr} + w_1 \tag{11}$$

Thus, updating the velocities and the positions can be given as in Eqs. (13) and (14)

$$V_{l,r}^{y+1} = \left[w^y \times V_{l,r}^y + c_1^y \times \text{rand}_1 \times \left(Pb_{l,r}^y - P_{l,r}^y\right) + c_2^y \times \text{rand}_2 \times \left(Pg_r^y - P_{l,r}^y\right)\right] \tag{12}$$

$$V_{l,r}^{y+1} = \min\left(\text{vmaps},\ \max\left(-\text{vmaps},\ V_{l,r}^{y+1}\right)\right) \tag{13}$$

$$P_{i,r}^{y+1} = P_{i,r}^y + V_{i,r}^{y+1} \tag{14}$$

Here, $y = 1, 2, \ldots, \text{maxi}tr$; $l = 1, 2, \ldots, pn$; $r = 1, 2, \ldots, d$ and.

$P_{l,r}^y$; rth position of lth particle in yth iteration.

$Pb_{l,r}^y$; rth position of *Pbest* which is determined for lth particle in yth iteration.

Table 1 Summary information of the analyses for the data sets

Time series	# of synaptic layer unit (changing from to)		# of dendrite (changing from to)		# of observations	Size of the training set	Size of testing set
ABC	2	12	2	12	153	137	16
FTSE-2010	2	5	2	5	252	242	10
						232	20
						212	40
FTSE-2011	2	5	2	5	251	241	10
						231	20
						211	40
FTSE-2012	2	5	2	5	252	242	10
						232	20
						212	40
FTSE-2013	2	5	2	5	253	243	10
						233	20
						213	40
FTSE-2014	2	5	2	5	253	243	10
						233	20
						213	40

Pg_r^y; rth position of *Gbest* which is determined for lth particle in yth iteration.

Step 6. Check the stop condition.

In the algorithm, there are two stopping criteria: (1) reaching a pre-determined maximum number of iterations and (2) being less than a predetermined value of the loss function values for the Gbest in two consecutive iterations. When the algorithm is stopped, the positions of the Gbest obtained in the last iteration are the optimum weight and parameter values. Otherwise, by returning to Step 3, depending on the new positions, the loss function for each particle is calculated and the algorithm is continued until the stop condition is satisfied.

5 Implementations

In this study, the performance of the proposed robust learning algorithm for R-DNM in predicting time series with and without outliers has been investigated by analysing two different real-world time series. The first of these time series is the Australian Beer Consumption (ABC) time series, which was quarterly observed between 1956 and 1994. The second is the London Stock Exchange Index (FTSE-Financial Times Stock Exchange Index) daily observed between 2010 and 2014. The predictive ability of the robust learning algorithm has been first evaluated over the original time series that did not contain any outliers. After that, the outliers have been created by taking 5 and 10 times the highest valued observation in the relevant time series. Artificially contaminated time series have been created by injecting outlier(s) into the original time series. The number of outliers injected in the contamination process of the time series has been determined as 1, 2, and 3. Thus, six different contaminated time series with different characteristics and numbers of outliers for each data set have been created and analysed. A summary of the information about the analyses for each data set is given in Table 1.

For each data set, there are seven scenarios given below:

- *Scenario 1*: Analysis of original time series.
- *Scenario 2*: Analysis of time series contaminated with a single outlier generated by 5 times the largest observation.
- *Scenario 3*: Analysis of time series contaminated with two outliers generated by 5 times the largest observation.
- *Scenario 4*: Analysis of time series contaminated with three outliers generated by 5 times the largest observation.
- *Scenario 5*: Analysis of time series contaminated with a single outlier generated by 10 times the largest observation.
- *Scenario 6*: Analysis of time series contaminated with two outliers generated by 10 times the largest observation.
- *Scenario 7*: Analysis of time series contaminated with three outliers generated by 10 times the largest observation.

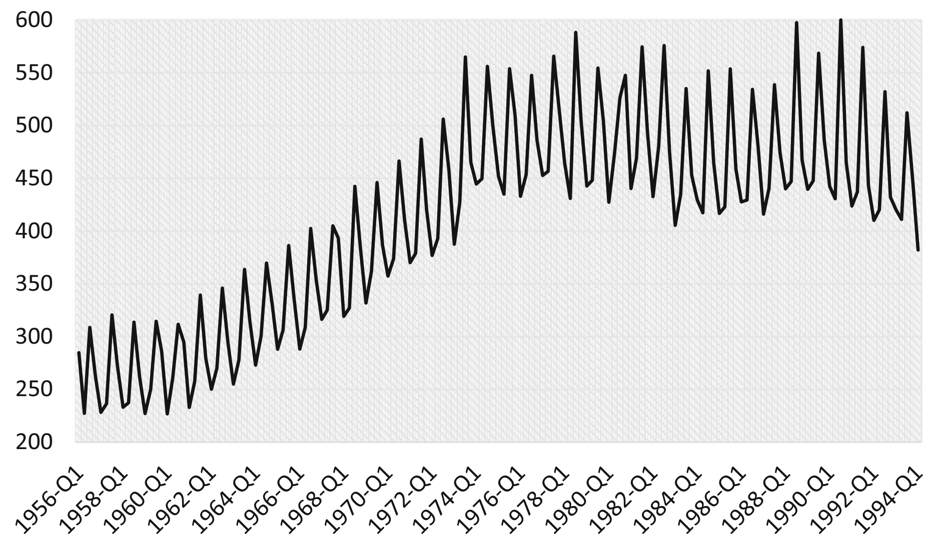


Fig. 3 The ABC time series

5.1 Australian beer consumption time series analysis

The predictive ability of the robust learning algorithm has been first evaluated in the ABC time series containing 153 observations, given in Fig. 3. A summary of the information on the analysis for each scenario can be given as follows:

pn: 30 (# of particle of the swarm).

vmapss_1: 10 (The upper limit of speeds for weight and network parameters).

vmapss_2: 0.1 (The upper limit of speeds for the scale parameter).

maxitr: 100 (Maximum number of repetitions).

(c_{1i}, c_{1f}): $(1, 2)$ (The interval which contains the possible values of c_1).

(c_{2i}, c_{2f}): $(1, 2)$ (The interval which contains the possible values of c_2).

(w_1, w_2): $(0.4, 0.9)$ (The interval which contains the possible values of w).

N: changing from 2 to 12 (# of synaptic input of synaptic layer).

M: changing from 2 to 12 (# of dendrite).

ntest: 16 (# of observations in the out-of-sample).

n: 153 (# of observations of time series).

The prediction performance has been evaluated over seven analyses, one original case (Scenario 1) and six contaminated cases (Scenarios 2–7).

5.1.1 ABC: scenario 1-original time series

The prediction performance of the proposed robust training algorithm has been evaluated together with two traditional time series models and seven different ANN models. These models are.

WMES: Winters' multiplicative exponential smoothing.

SARIMA: Seasonal autoregressive moving averages.

MLP: Multilayer perceptron neural network.

MLP-PSO: Multilayer perceptron neural network trained by PSO.

SMNM: Single multiplicative neuron model neural network.

SMNM-PSO: Single multiplicative neuron model neural network trained by PSO.

RBF: Radial basis function neural network.

ENN: Elman neural network.

R-SMNM: Robust single multiplicative neuron model neural network.

DNM: Dendritic neuron model neural network.

For all these alternative ANN models, the input and hidden layer unit numbers have been changed between 2 and 12 in parallel with the proposed model. The comparison of the models has been realized over two error metrics, mean absolute percentage error (MAPE) and root mean square error (RMSE) which are given in Eqs. (15) and (16).

$$\mathrm{MAPE} = \mathrm{mean}\left(\left|\frac{\mathrm{Observed_t} - \mathrm{Predicted_t}}{\mathrm{Observed_t}}\right| \times 100\%\right); \quad t = 1, 2, \ldots, T \tag{15}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{mean}\left((\mathrm{Observed_t} - \mathrm{Predicted_t})^2\right)}; \quad t = 1, 2, \ldots, T \tag{16}$$

The proposed robust training algorithm for the DNM produced the best prediction results in the analysis performed for the case where $N = 4$ and $M = 7$. The results of the first scenario are summarized in Table 2.

When Table 2 is examined, it is seen that the prediction performance of DNM based on the proposed robust

Table 2 The results of the first scenario—the analysis of the original ABC

Models	RMSE	Ranking for RMSE	MAPE (%)	Ranking for MAPE
SARIMA	47.0367	9	9.49	9
WMES	53.3295	10	10.72	11
MLP	24.1052	5	4.76	5
MLP-PSO	44.7780	8	8.56	8
SMNM	74.2551	11	9.83	10
SMNM-PSO	26.7831	6	4.98	6
RBF	41.7000	7	6.86	7
ENN	22.6581	4	4.36	4
R-SMNM	17.7761	2	2.97	2
DNM	17.8554	3	2.99	3
The proposed R-DNM	**16.3609**	**1**	**2.95**	**1**

Values in bold represent the model results with the best performance

learning algorithm is superior to other classical and ANN models, according to both the RMSE (*with the value of 16.3609*) and the MAPE (*with the value of 2.95%*) error metrics. For the test set, the high harmony between the predictions produced by the proposed robust learning algorithm and the actual observations is shown in Fig. 4a. Moreover, this harmony can also be supported by the scatterplot given in Fig. 4b. In such a scatterplot, the measure of the success of the prediction performance is that the actual observations and predictions are on or close to a line. Furthermore, in addition to all of the findings, the regression (β) and the determination (R^2) coefficients of a regression model, to be established as $Y = \beta\widehat{Y}$ (Observation = βPrediction), can also be examined. The basic expectation is that both β and R^2 take values close to 1 or 1. In other words, the distribution of the points on the scatter plot of the Y variable versus the $\widehat{Y}$ variable should be on the $Y = \widehat{Y}$ line or as close to the line as possible. It is also seen that both the regression coefficient and the coefficient of determination have been obtained quite close to 1, as expected from a satisfactory predictive tool.

5.1.2 ABC: scenario 2—contaminated time series—a single outlier—5 times

In this scenario, the contaminated ABC has been formed by replacing the 15th observation with an outlier created by taking 5 times the largest observation. The findings obtained from the analysis of this time series contaminated with a single outlier are summarized in Table 3.

From Table 3, with the RMSE value of 18.5804 and the MAPE value of 3.13%, it is seen that R-DNM produces superior predictions compared to other ANN models for the contaminated ABC prediction. When evaluating these findings together with scenario 1, which is the original case, it is seen that the performance of the proposed robust DNM is adversely affected by approximately 14% according to the RMSE criterion and approximately 6% according to the MAPE criterion. However, it should be emphasized that the proposed R-DNM still has the top performance for both error metrics. These findings are also supported by Fig. 5, which visualizes the high harmony of the proposed robust model's predictions with the actual observations. It is also seen that both β and R^2 have been obtained quite close to 1, as expected from a satisfactory predictive tool.

5.1.3 ABC: scenario 3—contaminated time series—two outliers—5 times

In this scenario, the contaminated ABC has been formed by replacing the 15th and 75th observations with the outliers created by taking 5 times the largest observation. The findings obtained this scenario are summarized in Table 4.

Considering Table 4, with the RMSE value of 16.5414 and the MAPE value of 2.68%, it is said that R-DNM generates outstanding predictions compared to other ANN models for the contaminated ABC prediction in scenario 3. When comparing these findings to scenario 1, which is the original case, it is seen that the proposed R-DNM is almost little or nothing affected in terms of the RMSE criterion. Moreover, an improvement level of 9% is achieved in terms of MAPE. Figure 6 visualizes the high harmony of the predictions with the actual observations supports these findings. Figure 6, again, presents that both β and R^2 have been obtained quite close to 1.

5.1.4 ABC: Scenario 4—contaminated time series—three outliers—5 times

This scenario includes the case where the ABC data set is contaminated with three outliers. These outliers are generated by quintuplicating the largest observation and then

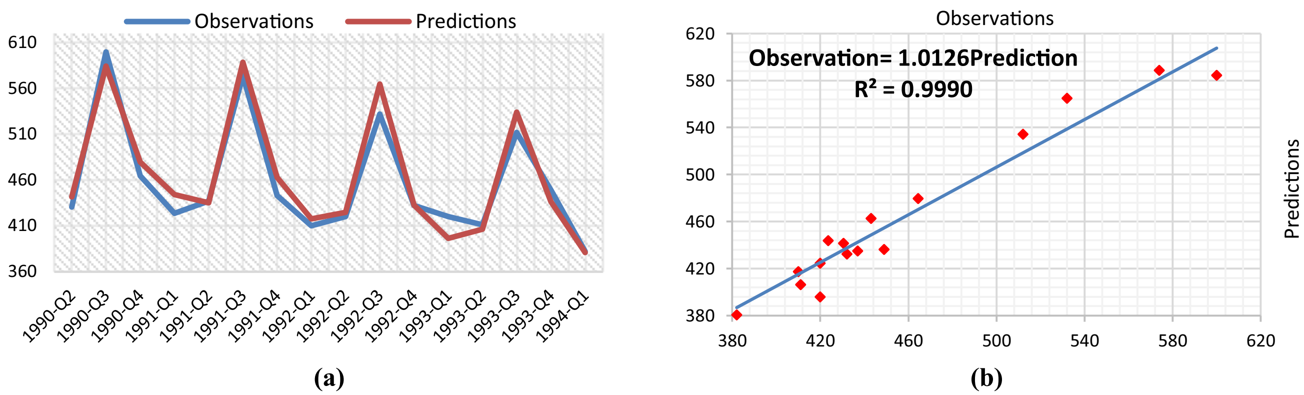


Fig. 4 The prediction and the observation values for the test set of original ACB Scenario 1

Table 3 The results of Scenario 2

Error metric		MLP	MLP-PSO	R-SMNM	The proposed R-DNM
RMSE	Value	176.5654	70.4099	18.8008	**18.5804**
	Percentage change	632%	57%	**6%**	14%
MAPE	Value	38.27%	18.63%	3.15%	**3.13%**
	Percentage change	704%	118%	**6%**	**6%**

Values in bold represent the model results with the best performance

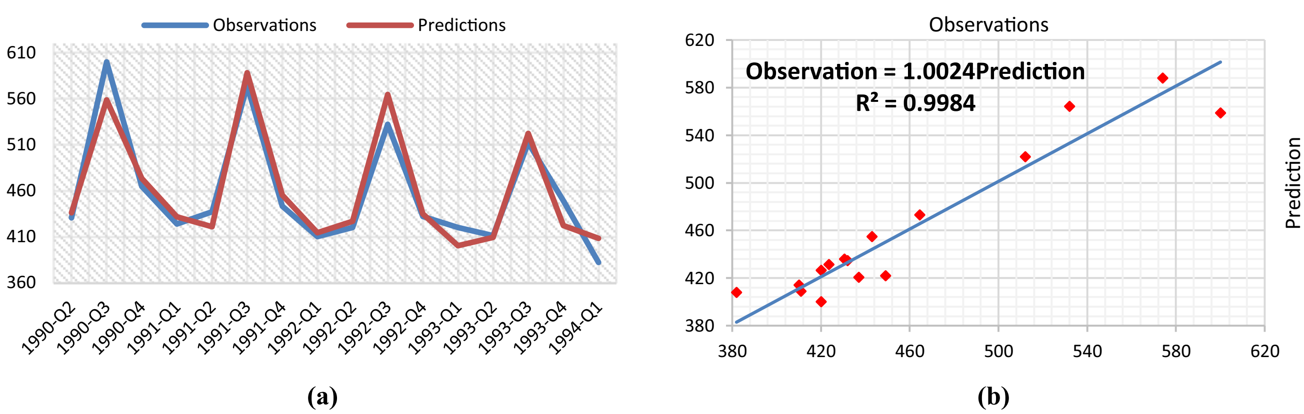


Fig. 5 The prediction and the observation values for the test set of contaminated ACB/Scenario 2

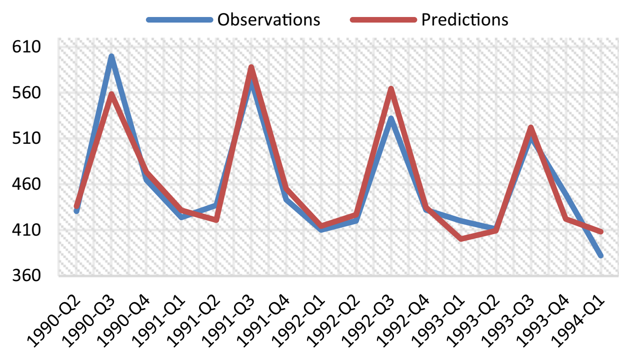
Table 4 The results of Scenario 3

Error metric		MLP	MLP-PSO	R-SMNM	The proposed R-DNM
RMSE	Value	284.1976	79.0784	20.1621	**16.5414**
	Percentage change	1079%	77%	13%	**1%**
MAPE	Value	63.20%	11.40%	3.26%	**2.68%**
	Percentage change	1228%	33%	10%	**− 9%**

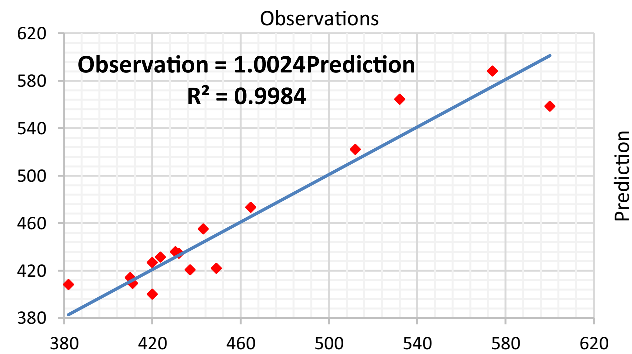
Values in bold represent the model results with the best performance

replaced with the 15th, 75th, and 120th observations. The findings obtained this scenario are summarized in Table 5.

Considering Table 5, the proposed R-DNM produces the second most successful prediction results after R-SNM. The proposed R-DNM still performed satisfactorily and can be said to be reasonably affected by outliers compared to scenario 1. The high harmony of the predictions with the actual observations, visualized in Fig. 7, can be seen as other evidence that the R-DNM produces satisfactory



(a)



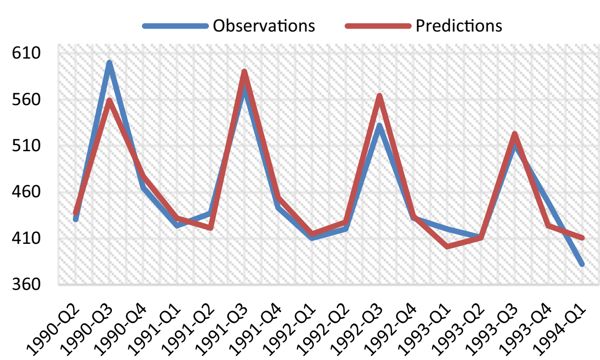
(b)

Fig. 6 The prediction and the observation values for the test set of contaminated ACB/Scenario 3

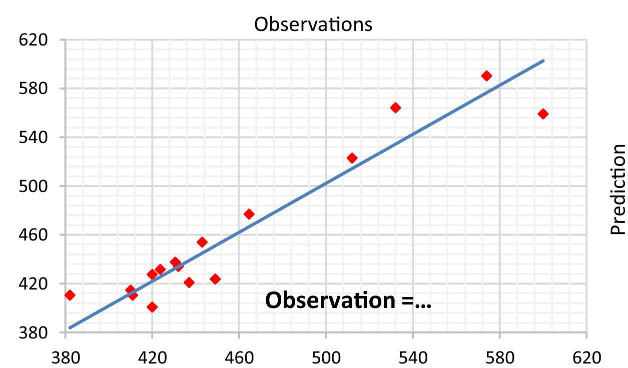
Table 5 The results of Scenario 4

Error metric		MLP	MLP-PSO	R-SMNM	The proposed R-DNM
RMSE	Value	328.3296	67.8867	**18.6250**	18.7551
	Percentage change	1262%	52%	**5%**	15%
MAPE	Value	72.55%	13.51%	**2.94%**	3.20%
	Percentage change	1424%	58%	**− 1%**	8%

Values in bold represent the model results with the best performance



(a)



(b)

Fig. 7 The prediction and the observation values for the test set of contaminated ACB/Scenario 4

predictions since both β and R^2 have been obtained quite close to 1 in this scenario again.

5.1.5 ABC: scenario 5—contaminated time series—a single outlier—10 times

In this scenario, the contaminated ABC has been formed by replacing the 15th observation with an outlier created by taking 10 times the largest observation. The findings obtained from the analysis of this time series contaminated with a single outlier are summarized in Table 6.

From Table 6, with the RMSE value of 16.2590 and the MAPE value of 2.88%, it is seen that R-DNM produces outstanding predictions compared to other models in the scenario 5. When evaluating these findings together with the original case, it is seen that the robust DNM is not adversely affected from outlier. Moreover, it should be emphasized that the proposed R-DNM has the top performance in terms of both error metrics. These findings are also supported by Fig. 8, which visualizes the high harmony of the R-DNMs' predictions with the actual observations. It is also seen that both β and R^2 have been obtained quite close to 1, as expected from a satisfactory predictive tool.

Table 6 The results of Scenario 5

Error Metric		MLP	MLP-PSO	R-SMNM	The Proposed R-DNM
RMSE	Value	188.1847	70.1409	18.4853	**16.2590**
	Percentage change	681%	57%	4%	**− 1%**
MAPE	Value	41.07%	10.12%	3.07%	**2.88%**
	Percentage change	763%	18%	3%	**− 2%**

Values in bold represent the model results with the best performance

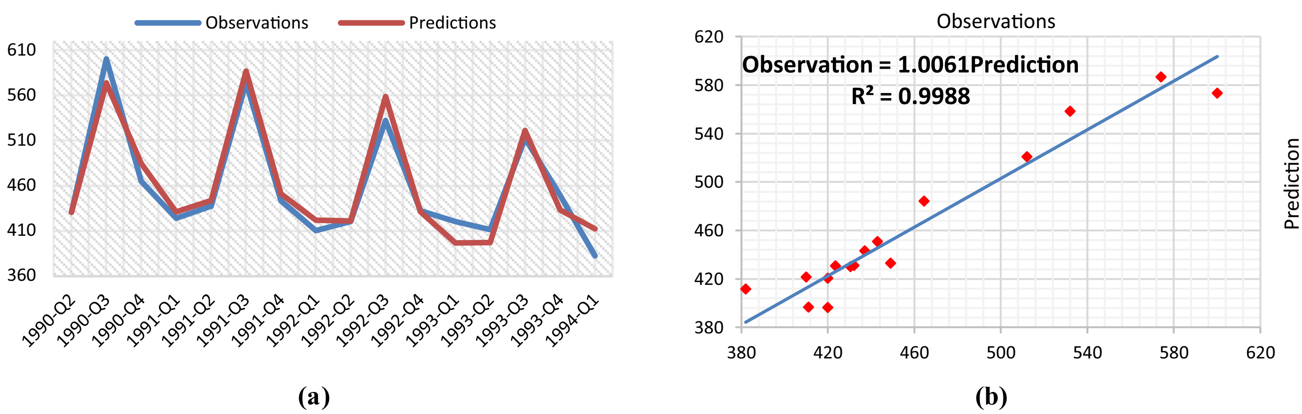


Fig. 8 The prediction and the observation values for the test set of contaminated ACB/Scenario 5

5.1.6 ABC: scenario 6—contaminated time series—two outliers—10 times

In this scenario, the contaminated ABC has been formed by replacing the 15th and 75th observations with the outliers created by taking 10 times the largest observation. The findings obtained this scenario are summarized in Table 7.

Considering Table 7, it is seen that the proposed model makes progress at the level of 7% in terms of MAPE. It is also the second least affected model in terms of RMSE. But still, with the RMSE value of 18.8324 and the MAPE value of 2.73%, R-DNM has supreme prediction performance compared to other models in scenario 6. It can be seen from Fig. 9 the high harmony of the predictions with the actual observations. Moreover, both β and R^2 are quite close to 1.

5.1.7 ABC: scenario 7—contaminated time series—three outliers—10 times

This scenario includes the case where the ABC data set is contaminated with three outliers. These outliers are generated by taking ten times the largest observation and then replaced with the 15th, 75th, and 120th observations. The findings obtained this scenario are summarized in Table 8.

From Table 8, it is seen that although the proposed R-DNM is the second least affected model in terms of both metrics, it still produces the most successful prediction results. The success of the received results can also be seen in Fig. 10, which shows the high harmony of the predictions with the actual observations. Moreover, the R-DNM produces the predictions with both β and R^2 having almost 1.

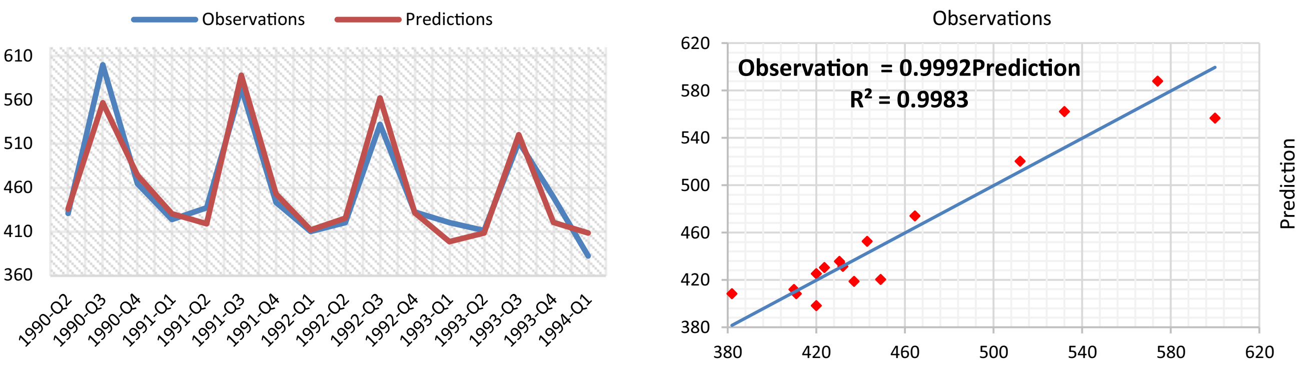
5.2 FTSE analysis

The performance of the proposed robust learning algorithm has been tested second by analysing the FTSE data sets. The FTSE contains time series daily observed in 5 individual years. The graphs of the FTSE time series are given in Fig. 11. Just like ABC, FTSE data sets have been analysed in seven scenarios. These scenarios again contain

Table 7 The results of Scenario 6

Error metric		MLP	MLP-PSO	R-SMNM	The proposed R-DNM
RMSE	Value	224.7867	73.3807	19.3856	**18.8324**
	Percentage change	833%	64%	**9%**	15%
MAPE	Value	49.39%	15.30%	2.95%	**2.73%**
	Percentage change	938%	79%	− 1%	**− 7%**

Values in bold represent the model results with the best performance



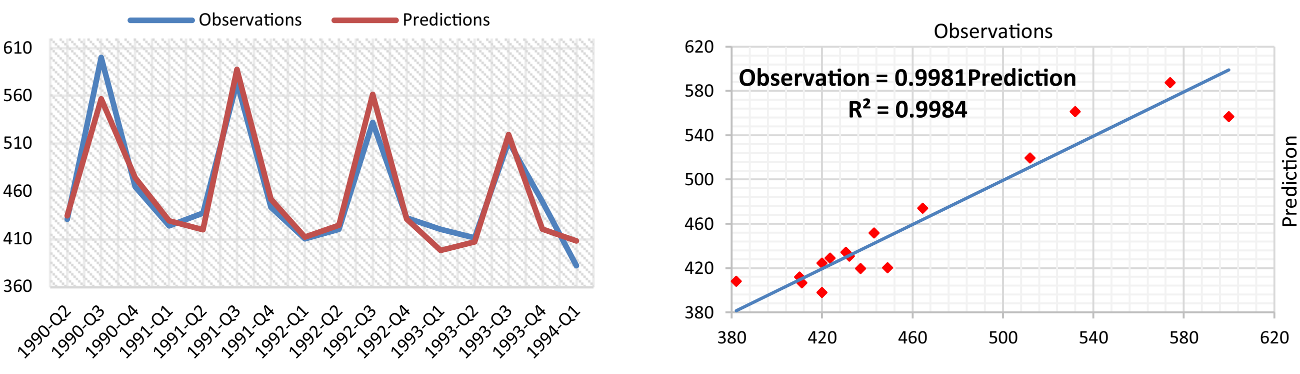
(a) (b)

Fig. 9 The prediction and the observation values for the test set of contaminated ACB/Scenario 6

Table 8 The results of Scenario 7

Error metric		MLP	MLP-PSO	R-SMNM	The proposed R-DNM
RMSE	Value	244.2329	104.9389	18.5871	**18.5290**
	Percentage change	913%	134%	**5%**	13%
MAPE	Value	53.47%	22.33%	2.88%	**2.79%**
	Percentage change	1023%	161%	**− 3%**	− 5%

Values in bold represent the model results with the best performance



(a) (b)

Fig. 10 The prediction and the observation values for the test set of contaminated ACB/Scenario 7

one original case and six contaminated cases. For each scenario, the test set has been taken in three different sizes 10, 20, and 40. Thus, 105 analyses have been carried out in total. A summary of the information on the analysis for each scenario can be given as follows:

pn: 30 (# of particle of the swarm).

vmaps_1: 100 (The upper limit of speeds for weight and network parameters).

vmaps_2: 0.1 (The upper limit of speeds for the scale parameter).

max*itr*: 100 (Maximum number of repetitions).

(c_{1i}, c_{1f}): $(1, 2)$ (The interval which contains the possible values of c_1).

(c_{2i}, c_{2f}): $(1, 2)$ (The interval which contains the possible values of c_2).

(w_1, w_2): $(0.4, 0.9)$ (The interval which contains the possible values of w).

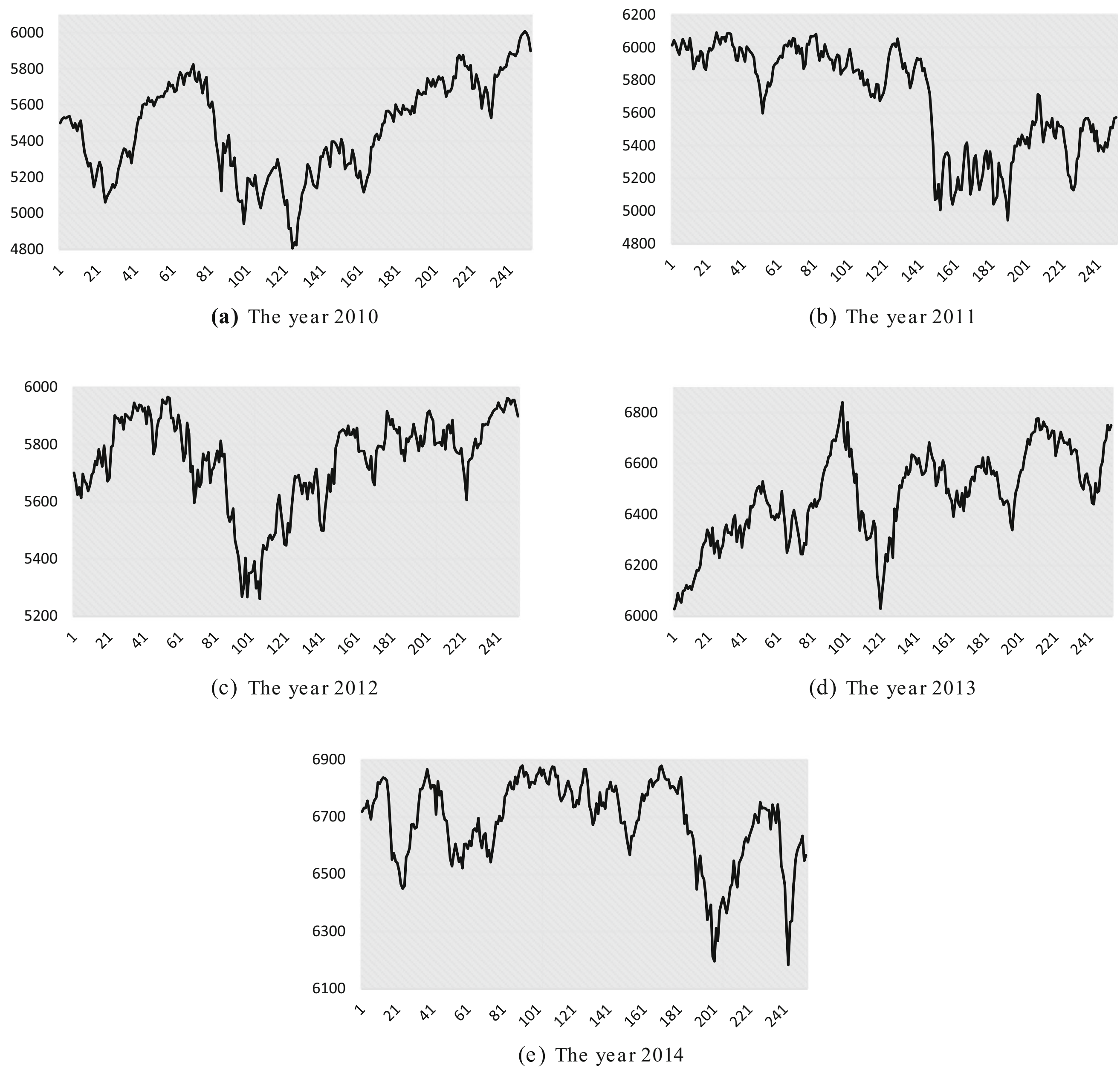
N: changing from 2 to 5 (# of synaptic input of synaptic layer).

M: changing from 2 to 5 (# of dendrite).

ntest: 10, 20, and 40 (# of observations in the out-of-sample).

The predicting performance of the proposed R-DNM has been evaluated together with ten other ANN models given below.

L-NL: Linear and nonlinear neural network [49]



(a) The year 2010 (b) The year 2011

(c) The year 2012 (d) The year 2013

(e) The year 2014

Fig. 11 The FTSE time series

MLP: Multilayer perceptron neural network [2]

MdNM: Median neuron model-based neural network [27]

MdPNM: Median and multiplicative neuron model-based neural network [50]

PS: Multiplicative (*Pi*) and additive (*Sigma*) neural network [6]

RBF: Radial basis function neural network [51]

SMNM: Single multiplicative neuron model neural network [10]

R-SMNM: Robust single multiplicative neuron model neural network [20]

TrMNM: Trimmed mean neuron model-based neural network [25]

DNM: Dendritic neuron model-based neural network [16]

Supplementary Tables 1–10 summarize the findings for the RMSE value for analyses containing all original and contaminated data. These tables also present how much the prediction results produced by each neural network for the contaminated data differ from the results produced for the original data. If the performance of the proposed R-DNM in FTSE estimation is considered in detail, the reached findings are:

Data of 2010.

- For the original data, when the sizes of the test set are 10 and 40, the R-DNM has the best prediction performance (RMSE = 31.3873; RMSE = 52.6651), while it has the third-best prediction performance for the test set size of 20 (RMSE = 37.7129).
- *In scenario 2, for the test set size of 20*, the R-DNM performs best as the ANN least affected by outlier observation, with a change of 0.76% (RMSE = 37.9996) compared to its performance on the original data. *In the case of test set sizes of 10 and 40*, it was relatively more affected by the outlier compared to other ANN models and had the third-best performance.
- *In scenario 3*, while it performs the best in all different cases of the test set size, *for the test set sizes of 10 and 20*, the R-DNM is the least affected by outliers compared to its performance in the original data and is the second least affected *for the test set size of 40*.
- *In scenarios 4 and 7*, the R-DNM both performs the best and is least adversely affected by outliers in all different cases of the number of observations of the test set.
- *In scenario 5*, while the R-DNM performs the best *for the test set sizes of 20 and 40*, it has the second-best performance *for the test set size of 10*. Similarly, it is the least and the second least affected by outliers *for the same cases*.
- *In scenario 6*, *for all different cases of the test set sizes*, the R-DNM is the least affected by outliers compared to its performance in the original data. Moreover, the R-DNM has the best and the second-best predictive performances in cases where the test set sizes are 10 and 40, and 20, respectively.

Data of 2011.

- *In scenario 1*, when the sizes of the test set is 40, the R-DNM has the best prediction performance (RMSE = 65.8484), while it has the second-best prediction performance for the test set sizes of 10 and 20 (RMSE = 40.3531; RMSE = 53.1080).
- *In scenario 2, for the test set sizes of 10 and 40*, the R-DNM performs best as the ANN least affected by outlier observation compared to its performance on the original data. *For the 20 test set size*, The R-DNM is the second least affected model by the outlier but still has the best predictive performance.
- *In all remaining scenarios*, *for all test sizes*, both the least affected by outliers and performing the best performance model is the R-DNM.

Data of 2012.

- *In scenario 1*, even if R-DNM does not produce the best predictions for all three test set sizes, it is still a competitive prediction tool with satisfactory performance.
- *In scenario 2, for the test set sizes of 10 and 20*, the R-DNM is least affected by outlier observation compared to its performance on the original data The R-DNM is least affected by outlier observation compared to its performance on the original data. Thus, it improves its performance much more compared to the other models.
- *In scenarios 3 and 4, for the test set sizes of 10 and 20*, the R-DNM is the least sensitive model to outliers compared to its performance on the original data. Thus, it moves its performance to the first rank.
- *In scenarios 5, for the test set size of 10*, the R-DNM is the least sensitive to the outlier and the best performing model. *For the test set sizes of 20 and 40*, it has the second-best performance.
- *In scenarios 6 and 7*, *for all test sizes*, both the least sensitive to outliers and performing the best performance model is the R-DNM.

Data of 2013.

- *In scenario 1*, *for the test set size of 10*, the R-DNM has the best predictive performance while, *for the test set size of 10 and 40*, it has the second-best prediction performance and produced very close results to the best one.
- *In scenarios 2 and 4*, *for all test set sizes except of 20*, the R-DNM is the least sensitive model to outlier(s) compared to its performance on the original data. It also exhibits the best predictive performance *for all test set sizes*.
- *In scenario 3*, *for all test set sizes*, the R-DNM stands out as the superlative model via outlier sensitivity and predictive performance.
- *In all remaining scenarios*, *for the test set sizes of 10 and 40*, the R-DNM is the least sensitive model to outlier(s) compared to its performance on the original data. Thus, it moves its performance to the first rank. *For the test set size of 20*, *in scenarios 6 and 7,* the R-DNM is the second least sensitive model to outliers and also is the model giving the best predictions. Moreover, for the same test set size, it produces the second-best predictions in scenario 5.

Data of 2014.

- *In scenario 1*, even if R-DNM does not produce the best predictions for all three test set sizes, it is still a satisfactory prediction tool with competitive performance.

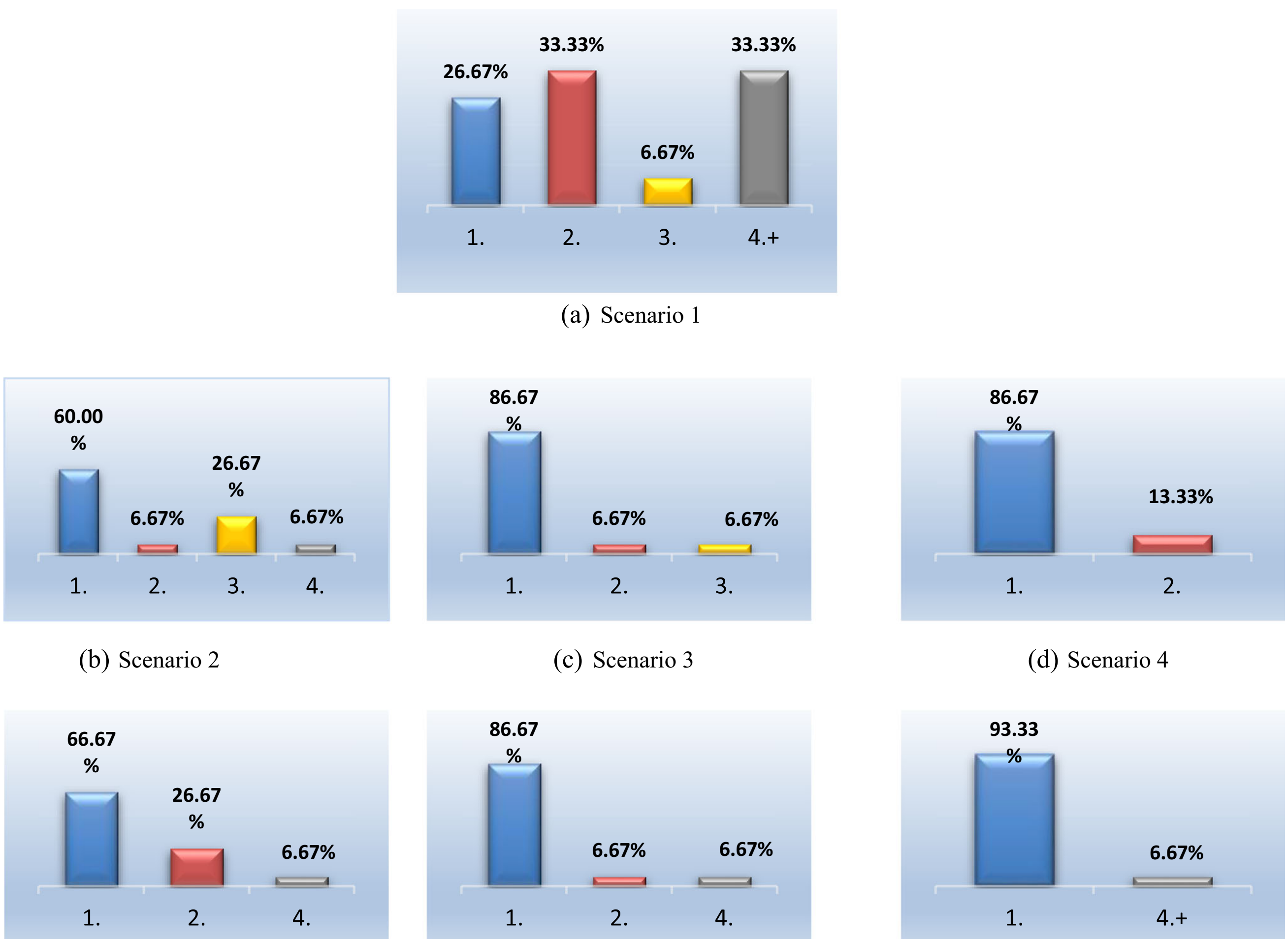
- *In all remaining scenarios*, *for the test set sizes of 20 and 40*, the R-DNM is the least sensitive model to outlier(s) compared to its performance on the original data. Thus, it moves its performance to the first rank. It has been observed that the R-DNM maintains its competitiveness with other ANN models in terms of prediction performance *for test set sizes of 10*.

Another evaluation of the performance of R-DNM can also be realized by considering both predictive performance and sensitivity to outliers according to its ranking among other ANN models. For this purpose, the graphs giving the circumstances related to ranks of prediction performances can be presented in Fig. 12.

In scenario 1, the R-DNM has the first two best prediction performances in *60%* of all analyses, while it increases this rate considerably in other scenarios where outliers have been included. *In scenarios 2* and *5*, the proposed R-DNM produces the best predictions in *60%* or more of the analyses. *In scenarios 3*, *4,* and *6*, the R-DNM has the best performance in about *87%* of all analyses, while it produces the best predictions in over *93%* of all analyses *in scenario 7*.

Furthermore, Fig. 13 presents the sensitivity rankings of the R-DNM to outliers. From Fig. 13, in scenarios 2 and 5, where the data have been contaminated with one outlier, it is observed that the R-DNM has been least affected by the outlier in over 53% of all analyses. In scenarios 3 and 6, where the data have been contaminated with two outliers, the R-DNM has been least affected by the outlier in almost 67% and 87% of all analyses. Similarly, in scenarios 4 and 7, where the data have been contaminated with three outliers, the proposed robust model has been least affected by the outlier in almost 74% and 87% of all analyses.

Considering each scenario for each FTSE time series of the proposed R-DNM, the levels of affection from outliers remained at the lowest level on average when compared to other methods. However, does this indicate statistically significant differences? In this point, it can also be



(a) Scenario 1

(b) Scenario 2 (c) Scenario 3 (d) Scenario 4

(e) Scenario 5 (f) Scenario 6 (g) Scenario 7

Fig. 12 The distribution of the rankings of the prediction performance of the R-DNM in the prediction of the FTSE

(a) Scenario 2 (b) Scenario 3 (c) Scenario 4

(d) Scenario 5 (e) Scenario 6 (f) Scenario 7

Fig. 13 The distribution of the sensitivity rankings of the R-DNM in the prediction of the contaminated FTSE

Table 9 Statistical evaluation of the levels of negatively affect

FTSE	μ_1	μ_2	*t*-stat	*p* value	%95 Confidence Interval of μ_1–μ_2	
					Lower bound	Upper bound
2010	The ProposedRobust DNM	DNM	− 3.0272	**4.70E−03****	− 7.9017	− 1.5539
		Robust SMNM	− 8.7645	**3.05E−10****	− 9.5422	− 5.95
2011		DNM	− 3.8315	**5.23E−04****	− 10.1288	− 3.1079
		Robust SMNM	− 7.5867	**8.19E−09****	− 7.7249	− 4.4607
2012		DNM	− 2.875	**6.90E−03****	− 2.8125	− 0.483
		Robust SMNM	− 3.9754	**3.47E−04****	− 13.4984	− 4.366
2013		Robust SMNM	− 3.7218	**7.13E−04****	− 4.3538	− 1.2784
		RBF	− 3.038	**4.60E−03****	− 11.5991	− 2.3009
2014		Robust SMNM	− 0.5901	0.5590	− 1.6344	0.8989
		DNM	− 1.5831	0.1227	− 5.0597	0.6286

Values in bold represent the model results with the best performance

Values highlighted in bold and with a double asterisk (**) represent the presence of the statistically significant difference at the 1% significance level

statistically demonstrated that the proposed R-DNM is less affected by outliers compared to other models. For this purpose, the R-DNM and the other two models that were least affected by outliers on average among other models were compared statistically with an independent two-sample t-test. The results obtained are summarized in Table 9. The results given in Table 9 clearly show that the proposed R-DNM is statistically less affected on average than other models when all outlier scenarios are taken together. That is, the level of affection of the proposed R-DNM from outlier observations is statistically significantly different and lower than the others. Even for FTS 2014, which is the only data set where the difference is not statistically significant, it is seen that the proposed model is less affected by outliers on average.

6 Conclusions and discussion

Many scientific studies have been put forward to develop different analysis and prediction tools for time series prediction problems faced by decision-makers in many fields. The analysis and prediction tools in time series problems can be divided into two groups probabilistic and non-probabilistic models. While probabilistic models are generally based on statistical approaches, non-probabilistic models are based on fuzzy set theory and computational approaches. Artificial neural networks, in recent years, have been used effectively as non-probabilistic predictive tools based on computational approaches. Among ANNs with different neuron structures, dendritic ANNs use neuron models that include both additive and multiplicative aggregation functions. The fact that DNM has a multiplicative aggregation function in addition to the additive aggregation function in its architectural structure may cause it to be adversely affected by the outlier(s). This may prevent the production of satisfactory predictions and can be seen as a crucial gap in the time series literature related to the usage of DNM. From this point of view, within the scope of this study, a robust learning algorithm, in which Huber's loss function is used as a fitness function, has been proposed for the first time in the literature in DNM training. The iterative process of the robust learning algorithm is carried out by PSO, an artificial intelligence optimization algorithm. This study is the first in the literature with this aspect as well.

To evaluate the performance of the proposed R-DNM, the ABC time series has been first analysed for various scenarios, containing both original data and contaminated data. From the findings obtained in all scenarios of the ABC analyses, it has been observed that the proposed R-DNM produced superior prediction results compared to some other ANN and classical time series prediction models available in the literature. In addition, it has been observed that the proposed R-DNM is almost not adversely affected by outlier(s) in the ABC time series analysis and maintains its outstanding predictive performance in which the analysis of the original time series.

Another stage in evaluating the performance of the R-DNM is the analysis of the daily observed FTSE time series of 5 different years. Data sets contaminated with various scenarios have been used in also this analysis process. A total of 105 different analyses have been performed, 90 of which have been for contaminated scenarios. The results show that the R-DNM trained by PSO produces satisfactory and competitive predictions for original data sets. Moreover, it retains satisfactory predictive performance for contaminated data sets with little or no influence from outliers.

In the light of all these findings, it can be said that the proposed R-DNM can be used as a prediction tool to obtain satisfactory results for time series with or without outliers.

Just like in other ANN studies, since the proposed R-DNM is a data-driven model, it is expected to produce similar results in data structures with similar characteristics. However, the possibility of that it may have different performances in different data structures can be seen as the main limitation of this study. In future studies, the performance of prediction tools trained by different optimization algorithms and using different robust function structures can be investigated.

Funding This research received no specific grant from any funding agency.

Data availability FTSE data that support the findings of this study can be obtained from https://www.investing.com/indices/uk-100 or all data are available from the authors upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent for publication All authors consented to the publication of the research.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s00521-023-08240-6.

References

1. McCulloch WSPW (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133
2. Rumelhart DE, Hinton GEWR (1986) Learning representations by backpropagating errors. Nature 323(6188):533–536
3. Hill M (1999) the learning behavior of single neuron classifiers on linearly separable. 1259–1264
4. Basu M, Ho T-K (1999) The learning behavior of single neuron classifiers on linearly separable or nonseparable input. In: Proceedings of the 1999 international joint conference on neural networks
5. Labib R (1999) New single neuron structure for solving nonlinear problems. In: Proceedings of the international joint conference on neural networks vol 1, p 617–620. https://doi.org/10.1109/ijcnn.1999.831569
6. Plate T (2000) Randomly connected sigma-pi neurons can form associator networks. Netw Comput Neural Syst 11:321–332. https://doi.org/10.1088/0954-898x/11/4/305
7. Zhang CN, Zhao M, Wang M (2000) Logic operations based on single neuron rational model. IEEE Trans Neural Netw 11:739–747. https://doi.org/10.1109/72.846745
8. Yadav RN, Kumar N, Kalra PK, John J (2006) Learning with generalized-mean neuron model. Neurocomputing. https://doi.org/10.1016/j.neucom.2005.10.006

9. Shiblee M, Chandra B, Kalra PK (2010) Learning of geometric mean neuron model using resilient propagation algorithm. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2010.04.018
10. Yadav RN, Kalra PK, John J (2007) Time series prediction with single multiplicative neuron model. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2006.01.003
11. Legenstein R, Maass W (2011) Branch-specific plasticity enables self-organization of nonlinear computation in single neurons. J Neurosci 31:10787–10802. https://doi.org/10.1523/JNEUROSCI.5684-10.2011
12. Ghosh J, Shin Y (1992) Efficient higher-order neural networks for classification and function approximation. Int J Neural Syst 3:323–350
13. Gurney KN (1992) Training nets of hardware realizable sigma-pi units. Neural Netw 5(2):289–303
14. Weber C, Wermter S (2007) A self-organizing map of sigma-pi units. Neurocomputing 70:2552–2560. https://doi.org/10.1016/j.neucom.2006.05.014
15. Yilmaz A, Yolcu U (2022) Dendritic neuron model neural network trained by modified particle swarm optimization for time-series forecasting. J Forecast 41:793–809. https://doi.org/10.1002/for.2833
16. Koch C, Poggio T, Torre V (1983) Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing. Proc Nat Acad Sci USA 80(9):2799–2802
17. Attia, MA, Sallam, EA, Fahmy MM (2012) A proposed generalized mean single multiplicative neuron model. In: Proceedings—2012 IEEE 8th international conference on intelligent computer communication and processing, ICCP 2012, art no 6356163 73–78
18. Gao S, Zhou M, Wang Y et al (2019) Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Trans Neural Netw Learn Syst 30:601–614. https://doi.org/10.1109/TNNLS.2018.2846646
19. Zhou T, Gao S, Wang J et al (2016) Financial time series prediction using a dendritic neuron model. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2016.05.031
20. Bas E, Uslu VR, Egrioglu E (2016) Robust learning algorithm for multiplicative neuron model artificial neural networks. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2016.02.051
21. Chen DS, Jain RC (1994) A robust back propagation learning algorithm for function approximation. IEEE Trans Neural Netw. https://doi.org/10.1109/72.286917
22. Rusiecki A (2012) Robust learning algorithm based on iterative least median of squares. Neural Process Lett 36(2):145–160
23. Ham FM, McDowall TM (1997) Robust learning in a partial least-squares neural network. Nonlinear Anal Theory Methods Appl 30:2903–2914. https://doi.org/10.1016/S0362-546X(96)00223-4
24. Neubauer A (1995) Robust learning algorithms for multi-layer perceptrons with discretized synaptic weights. In: IEEE international conference on neural networks - conference proceedings
25. Yolcu U, Bas E, Egrioglu E, Aladag CH (2015) A new multilayer feedforward network based on trimmed mean neuron model. Neural Netw World 25:587–602. https://doi.org/10.14311/NNW.2015.25.029
26. Egrioglu E, Yolcu U, Bas E, Dalar AZ (2019) Median-Pi artificial neural network for forecasting. Neural Comput Appl 31:307–316. https://doi.org/10.1007/s00521-017-3002-z
27. Aladag CH, Egrioglu E, Yolcu U (2014) Robust multilayer neural network based on median neuron model. Neural Comput Appl. https://doi.org/10.1007/s00521-012-1315-5
28. Harter HL (1993) The chronological annotated bibliography of order statistics. Eur J Oper Res. https://doi.org/10.1016/0377-2217(93)90102-s
29. Lei H, Zhang BL, Huang Q (1998) Robust interval regression analysis using neural networks. Fuzzy Sets Syst 97:337–347. https://doi.org/10.1016/s0165-0114(96)00325-9
30. Hsiao C (2012) A panel data approach for program evaluation: measuring the benefits of political and economic integration of Hong kong with mainland China. J Appl Econ 27(5):705–740
31. El-melegy MT, Essai MH, Ali AA (2009) Robust training of artificial feedforward neural networks. Stud Comput Intell 201:217–242. https://doi.org/10.1007/978-3-642-01082-8_9
32. Thomas P, Bloch G, Sirou F, Eustache V (1999) Neural modeling of an induction furnac using robust learning criteria. Integr Comput Aided Eng. https://doi.org/10.3233/ica-1999-6103
33. Connor JT, Martin RD, Atlas LE (1994) Recurrent Neural Networks and Robust Time Series Prediction. IEEE Trans Neural Netw doi. https://doi.org/10.1109/72.279188
34. Sánchez AVD (1995) Robustization of a learning method for RBF networks. Neurocomputing 9:85–94. https://doi.org/10.1016/0925-2312(95)00000-V
35. Wang Z, Huang B, Unbehauen H (1999) Robust reliable control for a class of uncertain nonlinear state-delayed systems. Automatica 35:955–963. https://doi.org/10.1016/S0005-1098(98)00233-7
36. Mili L, Coakley CW (1996) Robust estimation in structured linear regression. Ann Stat 24:2593–2607. https://doi.org/10.1214/aos/1032181171
37. Fan J, Chow TWS (2020) Exactly robust kernel principal component analysis. IEEE Trans Neural Netw Learn Syst 31:749–761. https://doi.org/10.1109/TNNLS.2019.2909686
38. Fan J, Chow TWS (1976) Component analysis. Acta Psychiatr Scand 53:298–302. https://doi.org/10.1111/j.1600-0447.1976.tb02339.x
39. Shi P, Shue S-P (1999) Robust H/sub /spl infin// control for linear discrete-time systems with norm-bounded nonlinear uncertainties. IEEE Trans Automat Contr 44:108–111. https://doi.org/10.1109/9.739084
40. Sinha S, Wiens DP (2002) Robust sequential designs for nonlinear regression. Can J Stat 30:601–618. https://doi.org/10.2307/3316099
41. Allende H, Moraga C, Salas R (2002) Robust estimator for the learning process in neural networks applied in time series. In: Dorronsoro JR (ed) Artificial neural networks—ICANN 2002. Springer, Berlin, pp 1080–1086
42. Pernía-Espinoza AV, Ordieres-Meré JB, Martínez-De-Pisón FJ, González-Marcos A (2005) TAO-robust backpropagation learning algorithm. Neural Netw 18:191–204. https://doi.org/10.1016/j.neunet.2004.11.007
43. Hans A, Udluft S (2010) Ensembles of neural networks for robust reinforcement learning. In: Proceedings—9th international conference on machine learning and applications, ICMLA 2010 401–406. https://doi.org/10.1109/ICMLA.2010.66
44. Song C, Zhao H, Jing W, Zhu H (2012) Robust video stabilization based on particle filtering with weighted feature points. IEEE Trans Consum Electron 58:570–577. https://doi.org/10.1109/TCE.2012.6227462
45. Liu B, Malon C, Xue L, Kruus E (2022) Improving neural network robustness through neighborhood preserving layers. Image Vis Comput. https://doi.org/10.1016/J.IMAVIS.2022.104469
46. Nakamura T, Fukagata K (2022) Robust training approach of neural networks for fluid flow state estimations. Int J Heat Fluid Flow. https://doi.org/10.1016/J.IJHEATFLUIDFLOW.2022.108997
47. Lei Y, Li J (2022) Robust adaptive neural networks control for a class of time-delay parabolic systems with nonlinear periodic time-varying parameter. J Franklin Inst 359:8014–8043. https://doi.org/10.1016/J.JFRANKLIN.2022.07.055

48. Shi X, Peng Y, Chen Q et al (2022) Robust convolutional neural networks against adversarial attacks on medical images. Pattern Recognit. https://doi.org/10.1016/J.PATCOG.2022.108923
49. Yolcu U, Egrioglu E, Aladag CH (2013) A new linear & non-linear artificial neural network model for time series forecasting. Decis Support Syst 54:1340–1347. https://doi.org/10.1016/j.dss.2012.12.006
50. Egrioglu E, Yolcu U, Bas E, Dalar AZ (2019) Median-Pi artificial neural network for forecasting. Neural Comput Appl. https://doi.org/10.1007/s00521-017-3002-z
51. Broomhead D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 2:321–355