Turkish Journal of Forecasting vol. 8 no. 2 (2024) pp 1-7

Content list available at JournalPark

Turkish Journal of Forecasting

Journal Homepage: tjforecasting.com

Forecasting of Turkey's Hazelnut Export Amounts According to Seasons with Dendritic Neuron Model Artificial Neural Network

Emine Kolemen[1,*]

[1]Avrasya University, Faculty of Engineering and Architecture, Department of Computer Engineering, Pelitli Campus, 61010,Trabzon Turkey

Abstract

It is seen that artificial neural networks have begun to be used extensively in the literature in solving the time series forecasting problem. In addition to artificial neural networks, classical forecasting methods can often be used to solve this problem. It is seen that classical forecasting methods give successful results for linear time series analysis. However, there is no linear relationship in many time series. Therefore, it can be thought that deep artificial neural networks, which contain more parameters but create more flexible non-linear model structures compared to classical time series forecasting methods, may enable the production of more successful forecasting methods. In this study, the problem of forecasting hazelnut export amounts according to seasons in Turkey with a dendritic neuron model artificial neural network is discussed. In this study, a training algorithm based on the particle swarm optimization algorithm is given for training the dendritic neuron model artificial neural network. The motivation of the study is to investigate Turkey's hazelnut export amounts according to seasons, using a dendritic neuron model artificial neural network. The performance of the proposed method has been compared with artificial neural networks used in the literature.

Keywords: Dendritic Neuron Model, Forecasting, Turkey Hazelnut Export.

1. Introduction

In recent years, artificial neural networks have begun to be used frequently in solving the time series forecasting problem. Multilayer perceptron is the most used artificial neural network for forecasting. However, deep artificial neural networks have recently shown very good performance for forecasting problems. The dendritic neuron model artificial neural network (DNM) offers both similarity to biological neuron structure and a more realistic mathematical model. Therefore, the forecasting performance of DNMs can be promising in forecasting research. DNM has been frequently used in the literature to solve the forecasting problem. Sossa and Guevara (2014), a new training algorithm is proposed for training dendrite morphological neural networks. Yu et.al (2016) proposed an unsupervised learnable DNM to predict China's house price index Zhou et al. (2016) proposed a DNM for forecasting financial time series. Ji et al. (2016) trained the DNM with a genetic algorithm to solve classification problems. Chen et al. (2017) proposed the DNM model for tourism demand forecasting. Jia et al. (2018) proposed a new DNM model for the Istanbul stock and Taiwan futures exchange indices forecast. Gao et al. (2018) proposed new training algorithms for training DNM. Song et al. (2019) proposed an approximate logic DNM to solve classification problems. Qian et al. (2019), the DNM model is proposed to solve the classification problem. Jia et al. (2020) DNM to solve the large-scale classification problem. Zhang et al. (2020) proposed a photovoltaic power forecasting model based on DNM. Song et al.(2020) proposed an evolutionary DNM for wind speed forecasting. Wang et al.(2020) proposed a new DNM model with a differential evolution algorithm and adaptive synapses. Wang et al. (2020) proposed a new median DNM to solve problems where there are outliers in the time series. Yu et al. (2020) proposed a training algorithm for training DNM. Xu et al. (2021) used a whale optimization algorithm for training DNM. He et al.(2021) proposed seasonal trend decomposition based DNM for financial time series forecasting. Tang et al. (2021) proposed a new algorithm for training DNM. Gao et al. (2021) proposed a DNM model for the XOR problem. Yılmaz and Yolcu (2022) DNM trained with modified particle swarm optimization is used to solve the time series forecasting problem. Al-Qaness et al. (2022) used DNM for crude oil production forecasting. Tang et al.(2022) proposed

* Corresponding Author
klmnemine@hotmail.com (Emine Kölemen)

Orcid ID:
Emine Kölemen: 0000-0001-6035-2065

a new DNM model for the forecasting of stock price movements. Li et al.(2023) proposed a new DNM model for COVID-19 forecasting.

Yilmaz and Yolcu (2023) proposed a robust learning algorithm for training DNM. Gul et al. (2023) proposed statistical learning algorithms for DNMs. Olmez et al. (2023) proposed a new bootstrapped DNM to solve the forecasting problem. Egrioglu et al. (2023) proposed a new DNM. Bas et al.(2023) proposed robust training of median DNM for time series forecasting. Zhang et al. (2023) proposed a DNM model for time series forecasting. Ding et al. (2024) proposed a new multi-output dendritic DNM. This study focuses on forecasting hazelnut export amounts according to seasons in Turkey. Other parts of the article are as follows. In the second part of the study, DNM and the training algorithm used in training DNM are introduced. In the third section, applications of hazelnut export time series and comparison results with other methods in the literature are presented. Finally, the study is concluded with a conclusion section.

2. Dendritic Neuron Model

DNM is proposed by Todo et al. (2014). DNM is a feed-forward artificial neural network model consisting of four layers. There are four different layers in the structure of DNM. These layers are the synaptic layer, dendritic layer, membrane layer and somatic cell layer. In these layers, nonlinear transformations are obtained for all inputs in the DNM. DNM architecture is given in Figure 1.

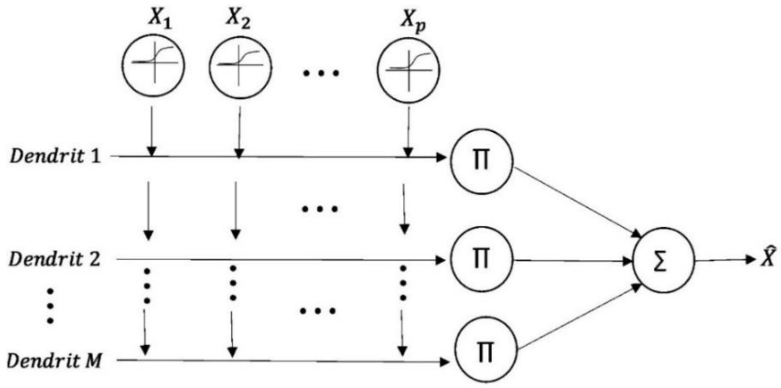

Figure 1. Architecture of DNM

Synaptic Layer:

It is the first layer of DNM. In DNM, inputs first come to this layer and are processed in this layer. The information processed in this layer is transferred to the next layer, the dendritic layer. The output of the synaptic layer is calculated with the help of Equation (1).

$$S_{ij} = \frac{1}{1+exp(-k(w_{ij}.x_{ij}+\theta_{ij}))} \quad i = 1,2,\dots,p \qquad j = 1,2,\dots,M \tag{1}$$

In Equation (1), k is the slope parameter for synaptic functions, S_{ij} is the output of the jth synaptic layer from i.synaptic input, x_{ij} is the input signal of i.synapse in j.synaptic layer. w_{ij} represents the weights and θ_{ij} represents the side values.

Dendrite Layer:

It is the layer in which the outputs obtained from each synapse of the synaptic layer are combined by a multiplication process. The output of the dendrite layer is calculated with the help of Equation (2).

$$D_j = \prod_{i=1}^{p} S_{ij} \qquad j = 1,2,\dots,M \tag{2}$$

Membrane Layer:

The membrane layer is the layer where the outputs obtained from the dendrite layer are combined. The output of the membrane layer is calculated with the help of Equation (3).

$$V = \sum_{J=1}^{M} D_j \tag{3}$$

Soma Layer:

It processes the outputs from the membrane layer with another sigmoid function, thus obtaining the output signal of the entire dendritic neuron model. The output of the soma layer is calculated with the help of Equation (4).

$$Output = \frac{1}{1+exp(-k_{soma}(V-\theta_{soma}))} \tag{4}$$

The total number of parameters in a DNM is $2pM + 3$.

These parameters are: w_{ij} $(i = 1,2, \dots, p;\ j = 1,2, \dots, M)$ θ_{ij} $(i = 1,2, \dots, p;\ j = 1,2, \dots, M)$

k: Slope parameter for synaptic functions
k_{soma}: Slope parameter for output transformation
θ_{soma}: Center parameter for output transformation

2.1. Training Algorithm Used for Training DNM

Step 1. The data set is divided into three parts: training, validation, and testing. Here n refers to the number of observations. The length of the training set is expressed as $n_{training}$ the length of the test set as n_{test} and the length of the validation set as $n_{validation}$ and n is calculated with Equation 5.

$$n = n_{training} + n_{test} + n_{validition} \tag{5}$$

Step 2. Upper and lower limits are determined for the hyperparameter values of DNM. Here, they are expressed as the number of inputs (p) and the number of dendritics (M).

$$p \in [p_{low}, p_{up}]$$

$$M \in [M_{low}, M_{up}]$$

Step 3. The parameters used in particle swarm optimization (PSO) are determined. These parameters are given below.

c_2^i : Initial value of the cognitive coefficient
c_2^f : End value of the cognitive coefficient
w^i : Limit value for speeds
w^f : End value for inertia weight
$vmaps$: Limit value for speeds
$limit1$: Limit value for restart strategy
$limit2$: Limit value for early stopping rule
$maxitr$: Maximum number of iterations
pn: Number of particles

Step 4. $p = p_{low}$, and $M = M_{low}$ are made.

Step 5. In PSO, initial position values and initial velocity values are generated. The total number of positions is $2pM + 3$. The positions of a particle in DNM consist of its weight and side values. The positions of a particle in DNM are as given in Table 1.

Table 1. Positions of a Particle

Weight and bias	w_1	$\cdots$	$w_{p\times M}$	θ_1	$\cdots$	θ_{pxM}	k	k_{soma}	θ_{soma}
Position	1	$\cdots$	$p \times M$	$p \times M + 1$	$\cdots$	$2x(pxM)$	$2x(pxM) + 1$	$2x(pxM) + 2$	$2x(pxM) + 3$

Step 6. Fitness function values are calculated for each particle. The fitness function is calculated with the help of Equation 6.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{6}$$

Step 7. New velocities and positions are calculated using Equation 7-9.

$$v_{i,k}^{(itr)} = w^{(itr)} v_{i,k}^{(itr-1)} + c_1^{(itr)} r_1 \left(P_{best\,i,k}^{(itr)} - P_{i,k}^{(itr)} \right) + c_2^{(itr)} r_2 \left(g_{best\;k}^{(itr)} - P_{i,k}^{(itr)} \right) \tag{7}$$

$$v_{i,k}^{(itr)} = \min\left(vmaps, \max\left(-vmaps, v_{i,k}^{(itr)}\right)\right) \tag{8}$$

$$P_{i,k}^{(itr)} = P_{i,k}^{(itr-1)} + v_{i,k}^{(itr)} \tag{9}$$

In Equation 7, the numbers r_1 and r_2 are randomly generated from the range U(0,1). In Equation 10-12, cognitive, social coefficients and inertia weights are calculated.

$$w^{(t)} = (w^i - w^f)\frac{maxitr - t}{maxitr} + w^f \tag{10}$$

$$c_1^{(t)} = (c_1^f - c_1^i)\frac{t}{maxitr} + c_1^i \tag{11}$$

$$c_2^{(t)} = (c_2^i - c_2^f)\frac{maxitr - t}{maxitr} + c_2^f \tag{12}$$

Step 8. P_{best} and g_{best} matrix is updated.

Step 9. The restart strategy is controlled by incrementing the counter $(rsc = rsc + 1)$.

Step 10. The early stopping rule counter (Esc) increases based on the condition in Equation 13. Here $MSE_{best}^{(t)}$, MSE's tth is the fitness value in the iterations.

$$esc = \begin{cases} esc + 1\,, & \dfrac{MSE_{best}^{(t)} - MSE_{best}^{(t-1)}}{MSE_{best}^{(t)}} < 10^{-3} \\ 0, & otherwise \end{cases} \tag{13}$$

The early stopping rule is esc>limit2. If the rule is met, the algorithm is stopped, otherwise, Step 7 is continued.

Step 11. Error root mean of squares (RMSE) values for the validation set are calculated with the help of Equation 14.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=n_{training}+1}^{n_{training}+n_{validation}} (y_t - \hat{y}_t)^2} \tag{14}$$

Step 12. If $p = p + 1$ and $p \le p_{up}$ and $M = M + 1$ and $M \le M_{up}$ go to Step 5, otherwise go to Step 13.

Step 13. Hyperparameter values with the best performance in the validation set are selected

Step 14. By combining the best hyperparameter values obtained from the training and validation sets to calculate the test set performance of the network, the network is retrained 30 times in random starts with a larger training set.

3. Application

In this section, Turkey's hazelnut export amounts according to seasons are used as the data set. The data set was taken from https://www.giresuntb.org.tr/IhracatVerileri. This data set consists of Turkey's hazelnut export amounts according to seasons for the years 2014-2022. The time series was solved using a simple multiplicative neuron artificial neural network (SMN) (Yadav et al. (2007)), Pi-Sigma artificial neural network (Pi-Sigma) (Shin and Gosh (1991)), and gated recurrent unit artificial neural network (GRU) (Cho et al. (2014)) methods. In the study, the number of inputs is taken as 1-5, the number of hidden layers as 1-2, and the number of hidden layer units as 1-s. The data set used in the application is divided into three parts: training, testing and validation. The length of the test and validation data sets is determined as 10. Each method is applied 30 times using random initial weights. Test set forecasting performance was calculated for each iteration using RMSE. Descriptive statistics (mean, standard deviation, minimum and maximum) of RMSE values are calculated and given in Table 2. The averages show the representative values of repeated solutions, and the standard deviations show the changes. Minimum statistics represent the best case, maximum statistics represent the worst case. The graph of the data set is given in Figure 2.

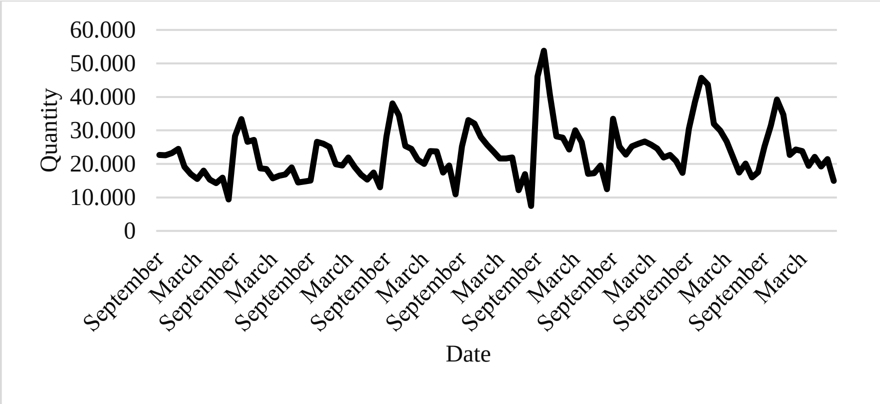

Figure 2. Turkey's hazelnut export amounts by season

Table 2. Statistics of the methods according to RMSE values

Data set	Methods	Mean	Med.	Std. Dev.	Iqr.	Min.	Max.	*p*	*m*	*h*
Export amounts	**SMN**	0.2071	0.2071	0.0000	0.0000	0.2071	0.2072	2	-	-
	PS	0.1986	0.1986	0.0000	0.0000	0.1986	0.1986	2	-	1
	DNM	0.1981	0.1972	0.0033	0.0043	0.1924	0.2074	2	-	-
	GRU	0.2681	0.2497	0.0563	0.0632	0.2022	0.4375	5	2	1

Med.: median, Std.Dev.: Standart Deviation, Iqr: interquartile difference, Min.:Minimum, Max.:Maksimum, p: number of inputs, h:.number of hidden layers, m: number of hidden layer units

When the analysis results for Turkey's hazelnut export data set are examined, it is seen that the DNM method is the best method among all methods according to the average, median and minimum statistics. However, it was determined to be the second-best method according to standard deviation and interquartile difference statistics. Additionally, DNM has been shown to produce better results than the GRU method in all statistics. As a result of the analysis, the best solution in average, median and minimum statistics is obtained when the number of inputs is 2 and the number of hidden layers is 2.

4. Conclusion

ANNs have been frequently used in recent years to solve forecasting problems. In this study, the forecasting problem of Turkey's hazelnut export amounts according to seasons is discussed. Time series analyses were performed using DNM, PS, SMN and GRU. However, a training algorithm based on PSO was used in the training of DNM. The DNM method gives the best results in mean, median and minimum statistics. In standard deviation and interquartile difference statistics, the DNM method is the second-best method. It has also been seen that the DNM method produces better results than GRU in all statistics. In the future study, different artificial intelligence optimization algorithms can be developed for training the network.

Funding

No funding was received for this work.

Credit authorship contribution statement

Emine Kölemen: Conceptualization, Methodology, Software, Data curation, Writing, Original draft preparation, Visualization, Investigation, Supervision, Validation, Reviewing, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

- The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Shin, Y., & Ghosh, J. (1991, July). 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* (Vol. 1, pp. 13-18). IEEE.

[2] Yadav, R. N., Kalra, P. K., & John, J. (2007). Time series prediction with single multiplicative neuron model. *Applied soft computing*, *7*(4), 1157-1163.

[3] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[4] Todo, Y., Tamura, H., Yamashita, K., & Tang, Z. (2014). Unsupervised learnable neuron model with nonlinear interaction on dendrites. *Neural Networks*, *60*, 96-103.

[5] Sossa, H., & Guevara, E. (2014). Efficient training for dendrite morphological neural networks. *Neurocomputing*, *131*, 132-142.

[6] Yu, Y., Song, S., Zhou, T., Yachi, H., & Gao, S. (2016, December). Forecasting house price index of China using dendritic neuron model. In *2016 International Conference on Progress in Informatics and Computing (PIC)* (pp. 37-41). IEEE.

[7] Zhou, T., Gao, S., Wang, J., Chu, C., Todo, Y., & Tang, Z. (2016). Financial time series prediction using a dendritic neuron model. *Knowledge-Based Systems*, *105*, 214-224.

[8] Ji, J., Song, Z., Tang, Y., Jiang, T., & Gao, S. (2016, December). Training a dendritic neural model with genetic algorithm for classification problems. In *2016 International Conference on Progress in Informatics and Computing (PIC)* (pp. 47-50). IEEE.

[9] Chen, W., Sun, J., Gao, S., Cheng, J. J., Wang, J., & Todo, Y. (2017). Using a single dendritic neuron to forecast tourist arrivals to Japan. *IEICE TRANSACTIONS on Information and Systems*, *100*(1), 190-202.

[10] Jia, D., Zheng, S., Yang, L., Todo, Y., & Gao, S. (2018, November). A dendritic neuron model with nonlinearity validation on Istanbul stock and Taiwan futures exchange indexes prediction. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)* (pp. 242-246). IEEE.

[11] Gao, S., Zhou, M., Wang, Y., Cheng, J., Yachi, H., & Wang, J. (2018). Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, *30*(2), 601-614

[12] Song, S., Chen, X., Tang, C., Song, S., Tang, Z., & Todo, Y. (2019). Training an approximate logic dendritic neuron model using social learning particle swarm optimization algorithm. *IEEE Access*, *7*, 141947-141959.

[13] Qian, X., Wang, Y., Cao, S., Todo, Y., & Gao, S. (2019). Mr2DNM: a novel mutual information-based dendritic neuron model. *Computational intelligence and neuroscience*, *2019*.

[14] Jia, D., Fujishita, Y., Li, C., Todo, Y., & Dai, H. (2020). Validation of large-scale classification problem in dendritic neuron model using particle antagonism mechanism. *Electronics*, *9*(5), 792.

[15] Zhang, T., Lv, C., Ma, F., Zhao, K., Wang, H., & O'Hare, G. M. (2020). A photovoltaic power forecasting model based on dendritic neuron networks with the aid of wavelet transform. *Neurocomputing*, *397*, 438-446.

[16] Song, Z., Tang, Y., Ji, J., & Todo, Y. (2020). Evaluating a dendritic neuron model for wind speed forecasting. *Knowledge-Based Systems*, *201*, 106052.

[17] Wang, Z., Gao, S., Wang, J., Yang, H., & Todo, Y. (2020). A dendritic neuron model with adaptive synapses trained by differential evolution algorithm. *Computational intelligence and neuroscience*, *2020*.

[18] Wang, S., Yu, Y., Zou, L., Li, S., Yu, H., Todo, Y., & Gao, S. (2020). A novel median dendritic neuron model for prediction. *IEEE Access*, *8*, 192339-192351.

[19] Yu, J., Shi, J., Li, Z., He, H., & Gao, S. (2020, December). Single dendritic neuron model trained by spherical search algorithm for classification. In *2020 IEEE International Conference on Progress in Informatics and Computing (PIC)* (pp. 30-33). IEEE.

[20] Xu, W., Li, C., Dou, Y., Zhang, M., Dong, Z., Jia, D., & Ban, X. (2021, September). Optimizing the weights and thresholds in dendritic neuron model using the whale optimization algorithm. In *Journal of Physics: Conference Series* (Vol. 2025, No. 1, p. 012037). IOP Publishing.

[21] He, H., Gao, S., Jin, T., Sato, S., & Zhang, X. (2021). A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction. *Applied Soft Computing*, *108*, 107488.

[22] Tang, C., Todo, Y., Ji, J., Lin, Q., & Tang, Z. (2021). Artificial immune system training algorithm for a dendritic neuron model. *Knowledge-Based Systems*, *233*, 107509.

[23] Gao, S., Zhou, M., Wang, Z., Sugiyama, D., Cheng, J., Wang, J., & Todo, Y. (2021). Fully complex-valued dendritic neuron model. *IEEE transactions on neural networks and learning systems*, *34*(4), 2105-2118.

[24] Yilmaz, A., & Yolcu, U. (2022). Dendritic neuron model neural network trained by modified particle swarm optimization for time-series forecasting. *Journal of Forecasting*, *41*(4), 793-809.

[25] Al-Qaness, M. A., Ewees, A. A., Abualigah, L., AlRassas, A. M., Thanh, H. V., & Abd Elaziz, M. (2022). Evaluating the applications of dendritic neuron model with metaheuristic optimization algorithms for crude-oil-production forecasting. *Entropy*, *24*(11), 1674.

[26] Tang, Y., Song, Z., Zhu, Y., Hou, M., Tang, C., & Ji, J. (2022). Adopting a dendritic neural model for predicting stock price index movement. *Expert Systems with Applications*, *205*, 117637.

[27] Li, J., Liu, Z., Wang, R. L., & Gao, S. (2023). Dendritic Deep Residual Learning for COVID-19 Prediction. *IEEJ Transactions on Electrical and Electronic Engineering*, *18*(2), 297-299.

[28] Yilmaz, A., & Yolcu, U. (2023). A robust training of dendritic neuron model neural network for time series prediction. *Neural Computing and Applications*, *35*(14), 10387-10406.

[29] Gul, H. H., Egrioglu, E., & Bas, E. (2023). Statistical learning algorithms for dendritic neuron model artificial neural network based on sine cosine algorithm. *Information Sciences*, *629*, 398-412.

[30] Olmez, E., Egrioglu, E., & Bas, E. (2023). Bootstrapped dendritic neuron model artificial neural network for forecasting. *Granular Computing*, *8*(6), 1689-1699.

[31] Egrioglu, E., Bas, E., & Karahasan, O. (2023). Winsorized dendritic neuron model artificial neural network and a robust training algorithm with Tukey’s biweight loss function based on particle swarm optimization. *Granular Computing*, *8*(3), 491-501.

[32] Zhang, Y., Yang, Y., Li, X., Yuan, Z., Todo, Y., & Yang, H. (2023). A dendritic neuron model optimized by meta-heuristics with a power-law-distributed population interaction network for financial time-series forecasting. *Mathematics*, *11*(5), 1251.

[33] Ding, Y., Yu, J., Gu, C., Gao, S., & Zhang, C. (2024). A multi-in and multi-out dendritic neuron model and its optimization. *Knowledge-Based Systems*, *286*, 111442.

[34] Bas, E., Egrioglu, E., & Cansu, T. (2024). Robust training of median dendritic artificial neural networks for time series forecasting. *Expert Systems with Applications*, *238*, 122080.