A New Bootstrapped Hybrid Artificial Neural Network Approach for Time Series Forecasting

Erol Eğrioğlu[a,b,*], Robert Fildes[b]

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

[b]Department of Management Science, Management Science School, Marketing Analytics and Forecasting Research Center, Lancaster University, UK

[*] Corresponding Author

Abstract

In this study, a new bootstrapped hybrid artificial neural network is proposed for forecasting. This new neural network provides input significance, linearity and nonlinearity hypothesis tests in a unique network structure via a residual bootstrap approach. The network has three parts: linear, non-linear and a combination with associated weights and biases. These weights are used to test the input significance, linearity and nonlinearity hypotheses with this new method providing empirical distributions for forecasts and weights. The proposed method employs a bagging approach to obtain forecasts. It is then applied to real-time series including the M4 Competition data set and stock exchange time series where its performance is compared with appropriate benchmark methods including other popular neural networks. The proposed method results are less affected than other neural networks by initial random weights, which means that the results of the proposed method are more stable and precise. The new method provides improvements in forecasting accuracy over the established benchmarks.

Key Words: Artificial Neural Networks, Deep Learning, Forecasting, Input Significance, Interval Forecast, Bootstrap

1. Introduction

Artificial neural networks (ANNs) can be used to obtain forecasts for linear or non-linear time series. Many types of artificial neural networks have been proposed in the literature. The findings of studies about the performance of ANNs for forecasting purpose vary from study to study. There is no consensus about the reasons behind the success or failure of ANNs performance on forecasting problem. In early research, Gorr et al. (1994) stated that ANN can (1) automatically

transform and represent complex and highly non-linear relationships and (2) automatically detect different states of phenomena through independently variable data patterns and switch on/off model components as appropriate. Besides these good properties, Gorr et al. (1994) emphasised that ANNs have several limitations, mostly noticeable in 'explanation research' (causal modelling and hypothesis testing) but not when used in forecasting. This occurs because ANN models are non-linear in the model coefficients and the normal probability models are not applicable. As a result of this, they do not have parametric statistical properties based on the t and *F* distributions. In this study, these problems are focused on and a new method proposed to solve them. Additional to these problems, the mean square error function used as the loss function in estimating the weights in ANNs is multi-modal, so the optimization algorithms suffer from the local optimum traps. The outputs of optimization methods are therefore not stable. The problem has been partially alleviated by using an artificial bee colony algorithm as an artificial intelligence optimization technique. In time series analysis, it is expected that a forecasting method provides forecasts, prediction intervals for forecasts, and hypothesis tests such as input significance, linearity and nonlinearity. These are important aspects of modelling to provide as simple a model as possible to conform to the data.

Many studies have not considered input significance tests, model selection or model adequacy. Researchers have focused more on point estimations in ANNs. In an ANN approach, determining inputs, the number of hidden layer nodes, activation function types and network architecture affect network performance. Moreover, inputs to the networks should influence the outputs. Determining relevant inputs have usually been identified by trial and error or from theoretical information about the data from the literature. The alternative is to develop statistical hypothesis tests for an ANN to determine input variables and appropriate functional forms to include. This viewpoint is supported by Anders and Korn (1999) who suggested that statistical analysis as described below should become an integral part of neural network modelling. (The terms given in parenthesis corresponding the meaning in the statistics literature.)

- Input significance test: This test is needed to see which inputs are relevant to produce an output or outputs in ANN (Variable Selection)

- Non-linearity test: This test is needed to decide where to apply an ANN to the data, otherwise a linear alternative modelling method should be used.
- Architecture tests: These tests are needed to establish if the network has a linear part or an additional non-linear part. These tests provide evidence that using the ANN architecture proposed has the potential to be useful in forecasting (Model Selection, e.g. RESET tests)
- Weights significance tests: These tests are needed for pruning the ANN, eliminating unnecessary hidden connections (Parameter Significance Tests).

Applying these tests is no easy task. Moody (1994) and Moody and Utans (1994) developed input selection and architecture selection approaches. Researchers faced some important problems for proposing these tests. Anders and Korn (1999) wanted to carry out parameter inference in a neural network based on an asymptotic normal distribution but they emphasise that the parameters of ANN are at least locally unique. To guarantee this, it is necessary to ensure that a given network model contains no irrelevant hidden units. One solution in the literature is to use model selection criteria to determine network architecture, the inputs and number of hidden nodes. Anders and Korn (1999) stated that criteria are not theoretically justified for over-parameterized networks, e.g. networks with irrelevant hidden units, even if the neural network model encompasses the true structure. Anders and Korn (1999) proposed strategies based on Terasvirta et al. (1993) using hypothesis tests and network information criteria but their method still suffered from the aforementioned problems. To summarize Refenes and Zapranis (1999) stated that the following situations cause unwanted results in ANNs.

- The omission of relevant variables as inputs in the ANN.
- Inclusion of irrelevant variables employed in the ANN
- Measurement errors in inputs and targets
- Incorrect specification of the architecture
- Inadequacies of the model selection and training algorithm, trapping local optimums

Refenes and Zapranis (1999) emphasised that consistent estimators can be obtained from ANNs if they satisfy the properties of convergence and uniqueness. This can be achieved by making a good decision for determining inputs, and the architectures. The second problem requires estimating standard error for the parameters in the ANN. It is not easy to obtain theoretically a formula for the standard errors in ANN. In discussing the second problem Zapranis and References (1999) proposed using a local bootstrap technique to make hypothesis tests in an ANN but because of the presence of local minima and the sensitivity of the training algorithm to initial conditions, resampling schemes tend to overestimate sampling variation. Their approach used derivate based training algorithm and this algorithm can be easily trapped in local optima.

White (1989), Lee et al. (1993), Terasvirta et al. (1993) have proposed a hypothesis test method for the nonlinearity of time series. These studies only focused on a multilayer perceptron. Yolcu et al. (2019) proposed linearity and non-linearity hypothesis test methods by using particular ANN type and forecasting accuracy was improved by using bootstrap methods, suggesting a potential route forward.

Bootstrap methods can be used to develop hypothesis tests in ANNs if we use an efficient learning algorithm by avoiding local optimum traps. Hypothesis tests and other statistical inferences can be made easily for non-linear of non-parametric models by using bootstrap methods as they delivering distributional estimates of components of the ANN. They have been employed to forecasting methods, for example, Masaratto (1990) discussed bootstrap confidence intervals for an autoregressive model and a residual-based approach was employed in the study. Lam and Veall (2002) compared analytic and bootstrap prediction intervals and they found that bootstrap prediction intervals performed better in Monte Carlo experiments. Dantas et al. (2018) proposed a new forecasting method based on bootstrap aggregation. They combined clustering and bagging in exponential smoothing methods. They found that their method outperforms many methods in the literature for M3 and CIF competition data sets. Bootstrap methods also used for artificial neural networks in the literature. Tiwari and Chatterjee (2010a, 2010b) papers use a bootstrap method to improve the forecasting accuracy of MLP. Kourentzes et al. (2014) proposed an ensemble operator for bootstrap approaches in ANNs. They proved that their operator is better

than the mean ensemble operator. Barrow and Crone (2016) proposed a “crogging method” for ANNs: The method is very similar to bagging. In this study, the results on real and simulated series demonstrated significant improvements in forecasting accuracy especially for short time series and long forecast horizons. Politis and Dimitirs (2016) obtained interval forecasts from ANNs by using bootstrap approaches. Yolcu et al. (2017) obtained confidence intervals for forecasts with an SMNM-ANN. Szafranek (2019) proposed bagged artificial neural network method for forecasting inflation data set. Yolcu et al. (2019) proposed linearity and non-linearity hypothesis test methods by using special ANN type and they improved forecasting accuracy by using bootstrap methods.

In summary, ANNs has been shown to produce good forecasting performance for some type of time series but ANNs cannot automatically produce statistical distributional results for forecasts and model coefficients. In contrast, statistical results can be obtained from other linear or non-linear statistical forecasting methods and this is a deficiency of ANNs but ANNs have better forecasting accuracy for some nonlinear time series in the literature. Most recently, the results of the M4 competition (Makridakis, 2019) compared accuracy on some methods and the two winning methods used hybrid combinations of ML methods with statistical models. Such comparisons aim to provide researchers and practitioners: Fildes (2019) commented on the M4 competition results that “the results certainly should guide the short-list” of methods to consider. Out-of-the-box, ML methods did not perform well. As a result of these findings, proposing new ANN methods can be useful for forecasters if they provide statistical distributional results and more accurate forecasts. The main focus of this study is providing this kind of ANN approach. In this study, a new hybrid artificial neural network is proposed. New methods for testing input significance, linearity and non-linearity in this new ANN are proposed. Hypothesis tests are realized by using the residual bootstrap method to take care of time series serial dependency. Moreover, the forecasting accuracy is improved by using bootstrap methods in the new ANN. The new ANN is trained by an artificial bee colony algorithm for avoiding local optimum traps. The proposed method provides the following advantages for users:

- Empirical distributions and confidence intervals for forecasts
- Empirical distributions and confidence intervals for weights of ANN
- A method to test linearity and nonlinearity

- A method to test input significance
- More accurate and confidential forecasts

In the second section, a new hybrid artificial neural network is introduced. In the third section, the training algorithm of the HANN is introduced. In the fourth section, the bootstrapped HANN method is introduced. Moreover, input significance, linearity and nonlinearity hypothesis test methods are introduced in this section. In the fifth section, the performance of the proposed method is investigated by using stock exchange data sets and M4 yearly competition data set. In section six, the obtained results are discussed. the proposed method is shown to perform well on both data sets compared to benchmark approaches.

2. A new Hybrid Artificial Neural Network (HANN) Model

Artificial neural networks (ANNs) have been commonly used for forecasting time series in the literature in recent years. ANNs generally create a nonlinear model between output and inputs. ANNs are preferred as a forecasting tool for non-linear time series besides they can also be used for forecasting linear time series. In the literature, many hybrid methods have been proposed for forecasting time series which have linear and nonlinear components. In some time series, the linear component is dominant or vice versa. In the literature, some hybrid methods use artificial neural networks and linear models such as autoregressive integrated moving average (ARIMA) model. Hybrid artificial neural network architectures have linear and non-linear parts are proposed in the literature. In this study, multilayer perceptron and autoregressive method are hybridized in a unique architecture. The architecture of the HANN is presented in Figure 1.

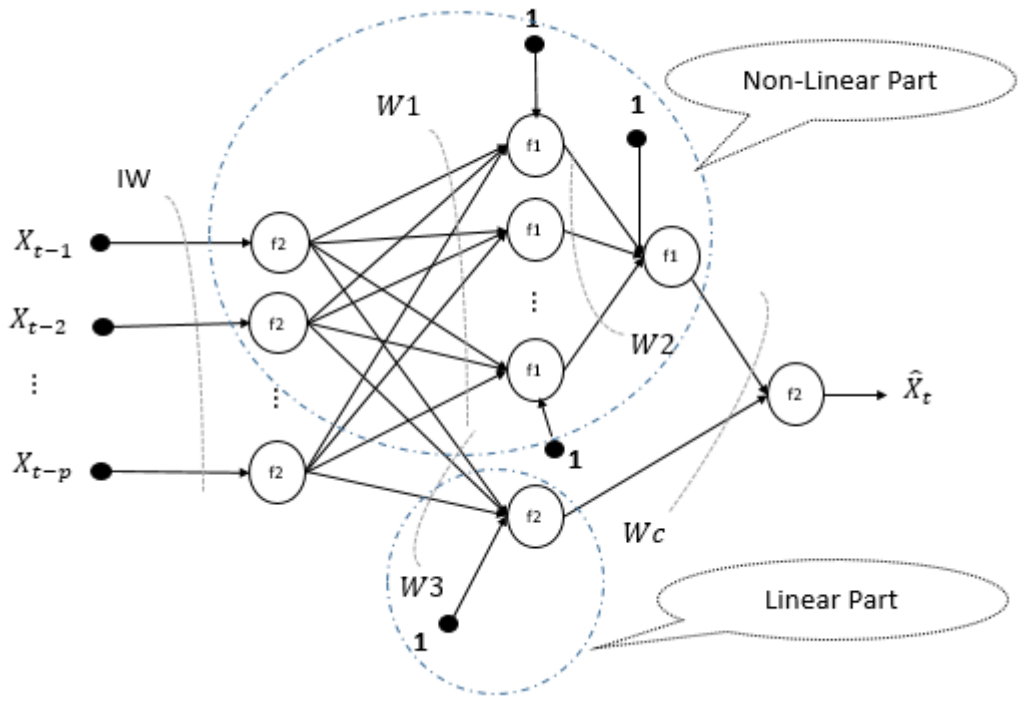

Figure 1. The architecture of the HANN

In HANN, lagged variables $X_{t-1}, \dots, X_{t-p}$ are inputs, $\hat{X}_t$ is output and X_t is target. IW presents input weights vector and has p elements. Each element of IW corresponds to an input.

$$IW = [iw_1\ iw_2\ \dots\ iw_p] \tag{1}$$

W_1 is weight matrix between inputs and hidden nodes for nonlinear part of HANN. The dimension of W_1 matrix is $p \times nh$ and nh is the number of hidden nodes in the nonlinear part. For each hidden nodes, there is a bias term and they collected in a $b_1 = [b_{1,1}\ b_{1,2}\ \dots\ b_{1,nh}]$ vector. W_2 is the weight vector between hidden nodes and output nodes of the nonlinear part. W_2 has nh elements and $W_2 = [v_1\ v_2\ \dots\ v_{nh}]$. The output of the nonlinear part has a bias term $b_2$and it is a scalar. W_3 is weights between inputs and hidden node in the linear part. $W_3 = [\,w_{3,1}\ w_{3,2}\ \dots\ w_{3,p}]$ has p elements. The output of the linear part has a bias term b_3 and it is a scalar.

W_c is combination weight vector and it has two elements $W_c = [wc_1\ wc_2]$. Elements of the W_c vector are weights of nonlinear and linear parts. In the HANN, f_1 is logistic and f_2 is linear activation functions.

$$f_1(x) = \frac{1}{1+\exp(-x)} \quad (2)$$

$$f_2(x) = x \quad (3)$$

Elements of IW are used to make input significance tests, elements of W_c are used to test linearity and linearity.

If any element of the IW statistically equals to zero, its corresponding lagged variable is insignificant. If the first element of W_c statistically equals to zero, time series has not a nonlinear component or it can be said time series is not nonlinear. If the second element of W_c statistically equals to zero, time series has not a linear component or it can be said time series is not linear. The output of the HANN can be calculated with the following algorithm.

Algorithm 1. Calculation output of HANN

Step 1. Calculate the output of the first layer with the following formula.

$$\boldsymbol{o_i^1} = f_2(X_{t-i} \times iw_i) = X_{t-i} \times iw_i \; ; i = 1,2, \dots, p \quad (4)$$

Step 2. Calculate outputs of the hidden layer in the nonlinear part.

$$\boldsymbol{o_j^{nl-h}} = f_1\left(\sum_{i=1}^{p} o_i^1 \times w_{i,j} + b_{1,j}\right) ; j = 1,2, \dots, nh \quad (5)$$

Step 3. Calculate the output of the nonlinear part.

$$\boldsymbol{o^{nl}} = f_1\left(\sum_{j=1}^{nh} o_j^{nl-h} \times v_j + b_2\right) \quad (6)$$

Step 4. Calculate the output of the linear part.

$$o^l = \sum_{i=1}^{p} X_{t-i} \times w_{3,i} + b_3 \quad (7)$$

Step 5. Calculate the output of HANN.

$$\hat{X}_t = wc_1 \times o^{nl} + wc_2 \times o^l \quad (8)$$

3. Training of the HANN

Training of artificial neural networks has been managed with various training methods. Some methods were based on derivate of the specific error functions and these methods have been commonly used in literature. Moreover, artificial intelligent optimization methods have been used for training of the ANNs in recent years. Genetic algorithm, differential evaluation algorithm and particle swarm optimization are the most preferred for the training of ANNs. Although it is not the most preferred training algorithm, artificial bee colony (ABC) algorithm has good results for training ANNs in the literature. Artificial bee colony algorithm was firstly proposed in Karaboga (2005). Karaboga et al. (2007) discussed the training of feed-forward neural networks by using the ABC algorithm. The ABC algorithm does not need computation derivate of any error function. The ABC algorithm can use any error function, it only needs for computing values of the error function in its stages. The flowchart of the ABC algorithm for general optimization problems is given in Figure 2. For training of HANN step by step algorithm is proposed and presented in Algorithm 2.

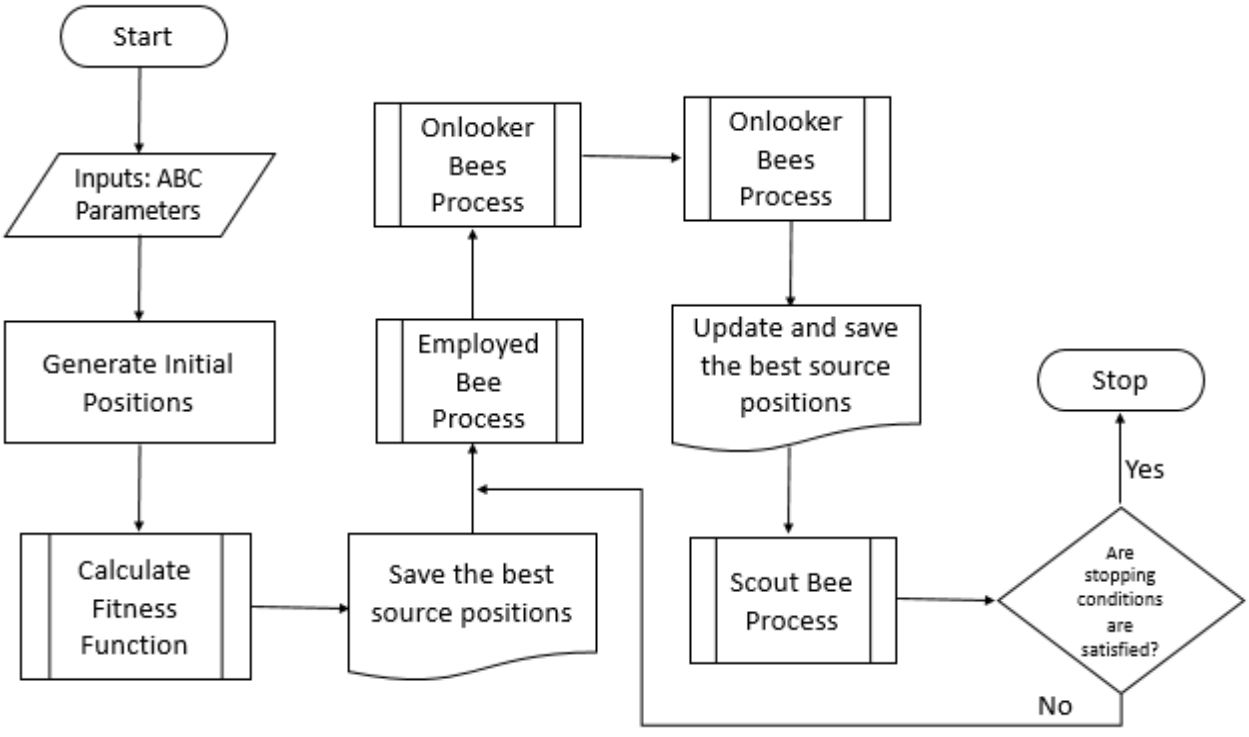

Figure 2. Flowchart of ABC Algorithm

Algorithm 2. Training of HANN by using ABC Algorithm

Step 1. Parameters of the ABC algorithm is selected according to the size of HANNs. These parameters are defined and listed below:

The number of sources (SN): The value of this parameter can be selected as 30 or 50. If the size of the network is too big, the parameter can be selected as 100 or more.

The dimension of the problem (D): The dimension depends on the network size. The value of this parameter is selected automatically based on the number of inputs and hidden nodes in the HANN.

The number of onlooker bees (NOB): This parameter is generally selected as 30 or 50.

Limit value ($LIMIT$): This parameter controls whether a source is exhausted or not. The selection of this parameter can be done by the user but there are some formulas for selecting the parameter in the literature. The parameter is generally selected as 200.

The maximum number of iterations ($MAXITR$): The parameter is selected as 50, 100, 200 or 500. The value of this parameter should be more than the number of required iterations.

Allowed number of the consecutive failure steps (ANFS): The parameter can be selected as 5, 6, 10 and 50. If you want to stop the algorithm earlier, the value of ANFS can be selected as a small value.

Step 2. Randomly initialization of sources

Initial values of source positions are generated randomly from a uniform distribution with (0,1) parameters. Decision variables are presented with $x_1, x_2, \dots, x_D$ and the value of D is the total number of weights and biases in the HANN and $D = 2p + (p + 2)nh + 4$.

$$x_j \in [0,1] \ , j = 1,2,..,D \tag{9}$$

x_{ij} is the value of the j[th] position of the i[th] source and it is randomly generated from $x_{ij} \sim Uniform(0,1)$.

Failure indexes are defined for each source, they are taken as zero in the initial step.

$failure_i = 0\ ; i = 1,2, \dots, SN$ (10)

Moreover, the number of the consecutive failure steps ($NFS = 0$) parameter is taken as zero.

Step 3. Fitness function values ($f_i\ ;\ i = 1,2, \dots, SN$) are calculated for each source. The number of the best source is saved.

The fitness function is taken as the mean square error (MSE) for the training set and it is calculated with the following formula:

$$MSE = \frac{1}{ntrain-p} \sum_{i=p+1}^{ntrain} \left(X_t - \hat{X}_t\right)^2 \quad (11)$$

In the MSE formula, $\hat{X}_t$ values are needed and these values are calculated by using Algorithm 1. The inputs of the algorithm 1 are weights and biases values and they are taken from the corresponding source positions.

Step 4. Employed bee phase

Employed bees are assigned to corresponding sources. Let employed bee number and neighbour bee number are $i1$ and $i2$, respectively. Each employed bee is made similar following operations. The operations are given below for source $i1$.

- Select a neighbour source number $i2$ different from $i1$.
- Select a position $i3$.
- Generate $\emptyset$ number from $(-1,1)$ interval.
- Calculate a new position value by using (12) equation.

$$x_{i1,i3}^{new} = x_{i1,i3} + \emptyset(x_{i1,i3} - x_{i2,i3}) \quad (12)$$

- A new source is created by using the new position value. Fitness function value for the new source (f_{new}) is calculated by using Algorithm 1 and positions of the new sources. f_{new} and fitness value for $i1$ source (f_{i_1}) are compared. Following rules are applied.

If $f_{new} \leq f_{i_1}$ then the new source is accepted and $\text{failure}_{i_1} = 0$.

If $f_{new} > f_{i_1}$ then the new source is rejected and $\text{failure}_{i_1} = \text{failure}_{i_1}+1$.

Step 5. Onlooker bee phase

Onlooker bees are sent to sources with corresponding probabilities. These probabilities are calculated as below:

$$P_i = \frac{1/f_i}{\sum_{t=1}^{SN} 1/f_i} \quad (13)$$

Following steps are repeated for each onlooker bee

- $i1$ is randomly selected with calculated probabilities.
- A neighbour source number $i2$ is randomly selected different from $i1$.
- A position $i3$ is randomly selected.
- A new position value is computed by using (12) equation.
- A new source is created by using the new position value. Fitness function value for the new source (f_{new}) is calculated by using Algorithm 1 and positions of the new sources. f_{new} and fitness value for $i1$ source (f_{i_1}) are compared. Following rules are applied.

If $f_{new} \leq f_{i_1}$ then the new source is accepted and $\text{failure}_{i_1} = 0$.

If $f_{new} > f_{i_1}$ then the new source is rejected and $\text{failure}_{i_1} = \text{failure}_{i_1}+1$.

Step 6. The best source of the swarm is updated (or determined in the first step). If the fitness value of the best source is changed then $NFS = 0$ otherwise $NFS = NFS + 1$.

Step 7. Scout bee phase

A scout bee is sent to all source.

If $failure_i > LIMIT\ ; i = 1,2, \dots, SN$ then the source is excluded from the swarm and a new source is generated by using $Uniform(0,1)$ distribution.

Step 8. The stopping condition is checked. If $NFS > ANFS$ then the algorithm was stopped.

4. B-HANN Method

In this paper, bootstrapped hybrid artificial neural network (B-HANN) method is introduced. In the B-HANN, the residual bootstrap method is preferred to use. Residual bootstrap is a model-based bootstrap method and it can preserve the autocorrelation structure of the training time

series. In B-HANN, bootstrapped training samples are generated and the test set is taken as fixed for each bootstrap iteration. The forecasts for the test set and estimated weights and biases are obtained in each bootstrap iteration. The algorithm of the B-HANN is given step by step in Algorithm 3. This study presents B-HANN as a forecasting method and HANN provides network structure for B-HANN.

Algorithm 3. Bootstrapped Hybrid Artificial Neural Network Method

Step 1. Determine the parameters of B-HANN

Number of inputs or number of lagged variables (p): Inputs of the B-HANN can be selected as $[X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}]$ $(p = 4)$ or $[X_{t-1}, X_{t-12}]$ $(p = 2)$. If we prefer inputs in the second form, Algorithm 1 needs a simple correction in some formulas.

Maximum lag number of inputs (m): This parameter is 4 for the $[X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}]$ and it is 12 for the $[X_{t-1}, X_{t-12}]$. The value of the m is determined by inputs.

The number of Hidden nodes (nh): This parameter should be selected as a small number to prevent memorization problem of the ANN.

The number of bootstrap repetitions (nbst): The parameter can be selected as 200 or more to obtain confident inference results.

The length of the test set$(ntest)$: The test set is selected end of the time series to see performance of the method for recent observations. The number of observations is presented by n and the length of the training set is $ntrain = n - ntest$.

Step 2. H-ANN is trained for original training time series by applying Algorithm 1 and Algorithm 2. The forecasts $\left(\hat{X}_t ; t = \mathrm{m}+1, \mathrm{m}+2, \ldots, ntrain\right)$ are calculated. The forecasts are saved as $\hat{X}_t^{initial} = \hat{X}_t ; t = \mathrm{m}+1, \mathrm{m}+2, \ldots, ntrain$. The standard deviation of the training forecasts are calculated by the following formula:

$$v = \sqrt{\frac{1}{ntrain-1}\sum_{i=p+1}^{ntrain}(\tilde{e}_t)} \tag{14}$$

$$\tilde{e}_t = \left(X_t - \hat{X}_t\right) - \frac{1}{ntrain-p}\sum_{t=m+1}^{ntrain}\left(X_t - \hat{X}_t\right) \tag{15}$$

Step 3. Bootstrap time series training set is generated by using step 3.1 – 3.2.

Step 3.1. Artificial residuals are randomly generated from the normal distribution

$e_t^* \sim N(0, v), t = m+1, m+2, \dots, ntrain$

Step 3.2. Bootstrap training time series ($\tilde{X}_t$) are calculated as follow:

$$X_t^* = \hat{X}_t^{initial} + e_t^* ; t = m+1, m+2, \dots, ntrain \quad (16)$$

$$\tilde{X}_t = [X_1, X_2, .., X_m, X_{m+1}^*, X_{m+2}^*, \dots, X_{ntrain}^*] \quad (17)$$

Step 4. H-ANN is trained for the bootstrap training series ($\tilde{X}_t$) by applying Algorithm 1 and Algorithm 2. The forecasts are calculated for the test set. The forecasts from HANN for the i^{th} bootstrap time series at time t is represented by F_t^i. Estimated weights and biases are saved and they presented in $IW^i, Wc^i, i = 1,2, \dots, nbst$.

Step 5. Step 3 and Step 4 are repeated $nbst$ times.

Step 6. Forecasts of B-HANN are calculated from bootstrap samples.

The obtained forecasts from bootstrap repetitions and their statistics are presented in Table 1.

Table 1 Forecasts of the trained HANN for bootstrap samples

Time(t) / Bootstrap Sample	**1**	**2**	**...**	***nbst***	**Mean**	**Standard Deviation**
1	F_1^1	F_1^2	...	F_1^{nbst}	$\hat{F}_1$	SE($\hat{F}_1$)
2	F_2^1	F_2^2	...	F_2^{nbst}	$\hat{F}_2$	SE($\hat{F}_2$)
⋮	⋮	⋮	...	⋮	⋮	⋮
Ntest	F_{ntest}^1	F_{ntest}^2	...	F_{ntest}^{nbst}	$\hat{F}_{ntest}$	SE($\hat{F}_{ntest}$)

In table 1, mean statistics presents forecasts of B-HANN. Different statistics can be used such as median, weighted mean but this is a limitation of the study. The other statistics are not preferred to combine forecasts. Moreover, standard error estimations are shown for each forecast in the last column of Table 1.

Step 7. Bootstrap confidence intervals are calculated from empirical distributions of the forecasts.

In this method, percentile confidence intervals are preferred as a bootstrap confidence interval type. Quartiles $\alpha/2$ and $1-\alpha/2$ are calculated for each forecast from its bootstrap replicates. Here, $Q(F_t^i, \alpha/2)$ and $Q(F_t^i, 1-\alpha/2)$ represent the $\alpha/2$ and $1-\alpha/2$ quartiles in the bootstrap distribution of F_t. The bootstrap percentile confidence interval is $[Q(F_t^i, \alpha/2), Q(F_t^i, 1-\alpha/2)]$.

Step 8. Linearity and nonlinearity tests are applied for time series.

The combination weights (W_c) are used to test linearity and non-linearity. In each bootstrap replicates, the estimated values of wc_1 and wc_2 are obtained and they showed in Table 2.

Table 2 Weights of the linear and nonlinear parts for trained HANN for bootstrap samples

Time(t)/Bootstrap Sample	**1**	**2**	**...**	***nbst***	**Mean**	**Standard Deviation**
$\boldsymbol{w_{c1}}$ ***(non-linearity weight)***	w_{c1}^1	w_{c1}^2	...	w_{c1}^{nbst}	$\overline{\boldsymbol{w_{c1}}}$	$s_1 = \mathrm{SD}\{w_{c1}^i, i = 1,2,\dots,nbst\}$
$\boldsymbol{w_{c2}}$ ***(linearity weight)***	w_{c2}^1	w_{c2}^2	...	w_{c2}^{nbst}	$\overline{\boldsymbol{w_{c2}}}$	$s_2 = \mathrm{SD}\{w_{c2}^i, i = 1,2,\dots,nbst\}$

The linearity and nonlinearity are tested using bootstrap samples and t-statistics. Details of the nonlinearity test are given in Table 3. When the normality assumption is violated, nonparametric "sign rank test" is used instead of t-test.

Table 3. Details of the nonlinearity test

	Hypotheses		
	Null Hypothesis	Alternative Hypothesis	Test statistics under $\boldsymbol{H_o}$ is true
Nonlinearity	$H_0^{NL}: \boldsymbol{w_{c1}} = \boldsymbol{0}$	$H_1^{NL}: \boldsymbol{w_{c1}} \neq \boldsymbol{0}$	$t_{NL} = \dfrac{\overline{\boldsymbol{w_{c1}}}}{s_1/\sqrt{nbst}}$
Linearity	$H_0^{L}: \boldsymbol{w_{c2}} = \boldsymbol{0}$	$H_1^{L}: \boldsymbol{w_{c2}} \neq \boldsymbol{0}$	$t_{L} = \dfrac{\overline{\boldsymbol{w_{c2}}}}{s_2/\sqrt{nbst}}$

The decision rules are as follows. If $t < t_{\frac{\alpha}{2};nbst-1}$ or $t > t_{1-\frac{\alpha}{2};nbst-1}$, then H_0 is rejected. Otherwise, H_0 cannot be rejected. When H_0^L (H_0^{NL}) is rejected, it can be said that the time series has a non-linear (linear) component.

Step 9. Input significance tests are applied. These tests are made based on the empirical distribution of $IW = [iw_1\ iw_2\ ...\ iw_p]$ weights. In each bootstrap replicates, the estimated values of elements of IW are obtained and they showed in Table 4.

Table 4 Weights of the inputs for trained HANN for bootstrap samples

Time(t)/Bootstrap Sample	**1**	**2**	**...**	$\boldsymbol{nbst}$	**Mean**	**Standard Deviation**
$\boldsymbol{iw_1}$ ***(1th Input)***	iw_1^1	iw_1^2	...	iw_1^{nbst}	$\overline{\boldsymbol{iw_1}}$	s_1^{inp} =SD$\{iw_1^i, i = 1,2, \dots, nbst\}$
$\boldsymbol{iw_2}$ ***(2th Input)***	iw_2^1	iw_2^2	...	iw_2^{nbst}	$\overline{\boldsymbol{iw_2}}$	s_2^{inp} =SD$\{iw_2^i, i = 1,2, \dots, nbst\}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\boldsymbol{iw_p}$ ***(2th Input)***	iw_p^1	iw_p^2	...	iw_p^{nbst}	$\overline{\boldsymbol{iw_p}}$	s_p^{inp} =SD$\{iw_p^i, i = 1,2, \dots, nbst\}$

The input significance is tested using bootstrap samples and t-statistics. Details of the nonlinearity test are given in Table 5. When the normality assumption is violated, nonparametric "sign rank test" is used instead of t-test.

Table 5. Details of the input significance test

	Hypotheses		
	Null Hypothesis	Alternative Hypothesis	Test statistics under $\boldsymbol{H_o}$ is true
1th Input	$H_0^{1th}: \boldsymbol{iw_1 = 0}$	$H_1^{1th}: \boldsymbol{iw_1 \neq 0}$	$t_1 = \frac{\overline{\boldsymbol{iw_1}}}{s_1^{inp}/\sqrt{nbst}}$
2th Input	$H_0^{2th}: \boldsymbol{iw_2 = 0}$	$H_1^{2th}: \boldsymbol{iw_2 \neq 0}$	$t_2 = \frac{\overline{\boldsymbol{iw_2}}}{s_2^{inp}/\sqrt{nbst}}$
⋮	⋮	⋮	⋮
pth Input	$H_0^{pth}: \boldsymbol{iw_p = 0}$	$H_1^{pth}: \boldsymbol{iw_p \neq 0}$	$t_p = \frac{\overline{\boldsymbol{iw_p}}}{s_p^{inp}/\sqrt{nbst}}$

The architecture selection problem of H-ANN is solved by using Algorithm 4. The strategy is based on dividing training data into two sets. The first set is used to obtain optimal weights of the HANN, the second part is a validation set and it used to select the best values of p and nh by calculating the root mean square error. The algorithm is given below as step by step. The best architecture is selected by using the following algorithm before Algorithm 3 is applied.

Algorithm 4. Model Selection Algorithm for B-HANN

Step 1. The bounds of the intervals p_1, p_2 and n_h^1, n_h^2 are selected for p and nh parameters.

$p \in [p_1, p_2]$, p is integer

$n_h \in [n_h^1, n_h^2]$, n_h is integer

Step 2. $p = p_1$ and $n_h = n_h^1$

Step 3. The algorithm 2 is applied and the RMSE values are calculated for the validation test and it is saved to $RMSE_1$. Let $k = 1$.

Step 4. Set $k = k + 1$

Step 5. Set $p = p_1 + 1$

Step 6. The algorithm 2 is applied and the RMSE values are calculated for the validation test and it is saved to $RMSE_k$.

Step 7. If $p < p_2$ go to Step 4.

Step 8. Set $n_h = n_h^1$+1.

Step 9. If $n_h < n_h^2$, set $p = p_1$-1 and go to Step 4.

Step 10. Find minimum $RMSE_k$ value and take p and nh values corresponding to the minimum $RMSE_k$.

5. Applications and Evaluation

When a new forecasting method is proposed it is incumbent on its developers to thoroughly evaluate its performance. The key features of how this should be done have been laid out, for example in Ord et al. (2017, Chapter 12). Key features as they apply here are the choice of suitable benchmark methods with which to compare the proposed new method, a range of error measures to measure comparative accuracy. Crucially, a large number of data series should be examined for a given forecast horizon. In this section, the performance of the B-HANN method is investigated by using stock exchange time series and M4 yearly competition data set and the results are compared with performance on suitable benchmarks.

5.1 Application to Stock Exchange Data Sets

Forecasting stock exchange data sets have long been important and their analysis attracted by ANNs researcher because linear models are not the good methods for them. Granger (1992) and Timmerman and Granger (2004) examine possible gains from new stock price forecasting methods, mentioning novel non-linear methods as potentially rewarding (if only temporarily). Sarantis (2001) used a SETAR model for forecasting of stock prices of seven major industrial countries. He applied nonlinearity tests to stock price data and linearity was rejected for all stock price time series. Olson and Mossman (2003) investigated the performance of ANNs on Canadian stock returns. They found that ANNs produce accurate point forecasts than statistical models. Bradley and Jansen (2004) used linear and non-linear models for forecasting stock returns and industrial production time series. They found that linear models are better than nonlinear models for stock returns but the situation was reversed for industrial production series. McMillan (2007) investigated four international stock market returns data forecasting by using lagged volume as the threshold in a logistic smooth-transition model. It was found that the model produced better forecasts than simple AR, random walk model and the logistic

smooth-transition model. Nyberg (2011) used dynamic binary probit models for forecasting the direction of the US stock market. Ebrahimpour et al. (2011) proposed a combined neural network approach for forecasting in the Tehran stock exchange. Lohrmann and Luukka (2019) forecasted classes of S&P500 time series by using random forests. Overall the studies support the conjectures of Granger (1992): gains in accuracy from time series methods should be possible but are probably not long-term sustainable.

In this study, the first data set was downloaded from (https://finance.yahoo.com). The first data set is called "S&P 500 (GSPC), SNP - SNP Real-Time Price. Currency in USD". This data set is constituted from daily opening prices for years 2014-2018. In the experimental study, 10 different sub-time series of length 500 observations are randomly taken from the whole time series. The observations of random time series are summarized in Table 6.

Table 6. The random time series observation dates and numbers

Series No	Starting date	Ending Date	Number of Observations
1	'2016-06-15'	'2018-06-08'	500
2	'2016-09-22'	'2018-09-17'	500
3	'2014-05-22'	'2016-05-16'	500
4	'2015-11-25'	'2017-11-17'	500
5	'2014-04-21'	'2016-04-13'	500
6	'2014-11-03'	'2015-10-29'	250
7	'2015-08-25'	'2016-08-19'	250
8	'2016-11-16'	'2017-11-13'	250
9	'2017-11-09'	'2018-11-06'	250
10	'2014-08-20'	'2015-08-17'	250

The time series have been modelled by the proposed approach, long short term memory deep artificial neural network (LSTM: Hochreiter and Schmidhuber; 1997) and pi-sigma artificial neural network (PSGM: Shin and Ghosh; 2001). All methods are applied our program codes in MATLAB and Matlab Neural Networks and Deep Learning Toolboxes. The length of the validation set and test set is taken as 50. The data dividing strategy is summarized in Figure 3. In the figure, *n* represents the total number of observations (here 500) and forecasts are made for 1 period ahead.

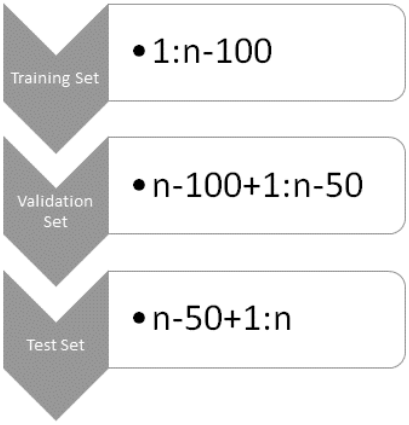

Figure 3. The data dividing strategy

The best model configuration is determined by using the validation set. The forecasts for the best configuration are calculated for the test set. First-order differencing is applied to each time series as a pre-processes method for all methods except Holt method. The error metric, the root of mean square error, is calculated for the test set by using the original time series. In all method applications, the number of inputs is varied from 1 to 5. The number of hidden layers is also varied from 1 to 5. After Model Selection for BHANN (algorithm 4) applied to the time series for each method, the best model configuration is replicated 50 times by using different initial weights and biases. The root of mean square errors metric values is calculated for the test set from all methods by using the following formula. The calculations are repeated for the 50 replications.

$$RMSE_j = \sqrt{\frac{1}{ntest}\sum_{t=n-netest+1}^{n}\left(X_t - \widehat{X_t}\right)^2} \quad ; j = 1,2,\dots,50 \tag{18}$$

RMSE is the standard error measure used for stock prices despite its sensitivity to outliers. The mean and standard deviation statistics are calculated for RMSE values from all methods and they summarized in Table 7. The random walk is used as the standard benchmark for forecasting stock price indices and also, Holt's linear exponential smoothing method has been included in the comparisons. Holt's smoothing parameters are estimated in MATLAB by using particle swarm optimization in. The best results are given in bold style in Table 7.

Table 7. Statistics for RMSE values for S&P500 Time Series from the methods

Data Sets	Methods	Mean	Standard Deviation	Number of Inputs	Number of Hidden Layer Nodes	Random Walk	Holt Linear Trend
1	**LSTM**	26,7972	6,9135	1	5	22,2968	**22,1581**
	PSGM	22,2869	**0,0223**	1	3		
	B-HANN	**22,1675**	0,0723	1	4		
2	**LSTM**	13,5936	1,6621	2	2	13,1351	12,9066
	PSGM	12,9026	**0,0174**	1	4		
	B-HANN	**12,8986**	0,0534	2	4		
3	**LSTM**	11,7483	0,0386	1	1	11,6595	11,6601
	PSGM	11,8692	0,1516	5	4		
	B-HANN	**11,6573**	**0,0264**	2	3		
4	**LSTM**	7,1527	2,4158	5	2	6,5195	**6,17604**
	PSGM	**6,1907**	0,0423	5	3		
	B-HANN	6,2591	**0,0257**	2	3		
5	**LSTM**	16,9119	0,0342	1	1	16,8722	16,8789
	PSGM	17,1041	0,1728	4	3		
	B-HANN	**16,8556**	**0,0495**	4	4		
6	**LSTM**	28,9444	0,1245	3	1	27,6093	27,6125
	PSGM	**27,2991**	0,1948	2	2		
	B-HANN	27,6056	**0,0353**	4	2		
7	**LSTM**	16,1913	0,2616	2	3	15,5547	15,5074
	PSGM	15,5377	0,0334	1	5		
	B-HANN	**15,5015**	**0,0157**	1	1		
8	**LSTM**	7,8493	0,5879	3	4	6,4085	6,1294
	PSGM	**5,9579**	0,0434	4	5		
	B-HANN	6,1072	**0,0220**	5	2		
9	**LSTM**	28,5454	1,0378	2	2	**25,0223**	**25,1678**
	PSGM	**25,2009**	**0,0157**	1	4		
	B-HANN	25,2341	0,0622	5	1		
10	**LSTM**	13,7945	0,4205	1	1	**13,548**	13,5711
	PSGM	13,6109	0,0408	1	5		
	B-HANN	**13,5671**	**0,0084**	3	1		

The best model configurations for LSTM, PSGM and B-HANN methods are given in Table 7. The model configuration is not needed for a random walk model and Holt's exponential smoothing methods. According to Table 7, the proposed method has more accurate and stable

forecasts from the LSTM and PSGM. Moreover, it produces better results than Holt's linear trend and the random walk. The best model configurations are different and values depend on the time series. ANN methods are better than the random walk method on 80% of the time series. Similarly, ANN methods are better than Holt's linear trend method on 80% of the time series.

In table 8, the mean of RMSE statistics for LSTM, PSGM and B-HANN methods and RMSE values for a random walk and Holt's exponential smoothing methods are given. Moreover, percentages of success are given and the percentage of success means that the method produced the best results in the mentioned percentage of all experiments, as measured by RMSE. The rank statistics are calculated for all methods.

Table 8. Mean statistics of RMSE for LSTM, PSGM and BHANN and RMSE values for a random walk and Holt's linear trend method in S&P500 Random Time Series.

Random Data	LSTM	PSGM	BHANN	Random Walk	Holt's Linear Trend
1	26,7972	22,2869	22,1675	22,2968	**22,1581**
2	13,5936	12,9026	**12,8986**	13,1351	12,9066
3	11,7483	11,8692	**11,6573**	11,6595	11,6601
4	7,1527	6,1907	6,2591	6,5195	**6,1760**
5	16,9119	17,1041	**16,8556**	16,8722	16,8789
6	28,9444	**27,2991**	27,6056	27,6093	27,6125
7	16,1913	15,5377	**15,5015**	15,5547	15,5074
8	7,8493	**5,9579**	6,1072	6,4085	6,1294
9	28,5454	25,2009	25,2341	**25,0223**	25,1678
10	13,7945	13,6109	13,5671	**13,5480**	13,5711
Percentage of Success	0%	20%	**40%**	20%	20%
Mean of Rank	4,8000	2,9000	**1,9000**	2,9000	2,5000

In Table 8, B-HANN method has the best statics with %40 percentage of success and 1,90 mean of rank. LSTM couldn't give the best results and its performance looks the worst.

The second data set is the Financial Times Stock Exchange 100 Index, also called the FTSE 100 Index. The data set was daily recorded opening prices between 01/01/2014 and 31/12/2018.

In the experimental study, 10 different sub-time series are randomly taken from the whole time series. The observations of random time series are summarized in Table 9.

Table 9. The random time series observation dates and numbers

Series No	Starting date	Ending Date	Number of Observations
1	‘2016-06-20’	‘2018-06-11’	500
2	'2016-09-27'	'2018-09-18'	500
3	'2014-05-22'	'2016-05-12'	500
4	'2016-10-04'	'2018-09-25'	500
5	'2015-11-27'	'2017-11-17'	500
6	'2014-05-27'	'2015-05-20'	250
7	'2015-02-13'	'2016-02-09'	250
8	'2016-03-10'	'2017-03-06'	250
9	'2017-11-01'	'2018-10-26'	250
10	'2014-08-20'	'2015-08-14'	250

The same methods are applied to the FTSE 100 Index random time series in the same condition with the previous application. The mean and standard deviation statistics are calculated for RMSE values from all methods and they summarized in Table 10. Moreover, the best model configurations for ANN methods and results for a random walk and Holt method are given in Table 10.

Table 10. Statistics for RMSE values for FTSE 100 Random Time Series from the methods

Random Data	Methods	Mean	Standard Deviation	Number of Inputs	Number of Hidden Layer Nodes	Random Walk	Holt Linear Trend
1	**LSTM**	52,3928	0,5579	4	1	52,19	52,0963
	PSGM	52,9689	0,1510	2	5		
	B-HANN	51,8214	0,1903	5	4		
2	**LSTM**	53,3868	4,2900	3	2	50,2549	50,3696
	PSGM	50,0872	0,2783	4	4		
	B-HANN	50,4826	0,0827	5	1		
3	**LSTM**	56,0447	15,4767	1	4	49,3421	49,4277
	PSGM	49,4790	0,4062	5	3		
	B-HANN	49,4319	0,0856	4	1		
4	**LSTM**	53,6121	2,0794	1	5	50,2108	50,3992
	PSGM	50,1893	0,0816	3	1		
	B-HANN	50,3465	0,0748	4	3		
5	**LSTM**	36,8277	5,8923	2	5	33,7235	33,9079
	PSGM	34,6996	0,4375	2	5		

	B-HANN	33,7895	0,0578	4	2		
6	**LSTM**	64,0471	5,3453	5	3	60,6946	60,6925
	PSGM	61,5636	0,8713	4	4		
	B-HANN	60,6778	0,0564	3	4		
7	**LSTM**	88,3087	2,1842	1	5	84,6748	84,2412
	PSGM	84,3469	1,1589	5	5		
	B-HANN	84,1721	0,2520	3	5		
8	**LSTM**	41,3384	1,1785	5	1	33,8047	33,2034
	PSGM	34,3204	0,6011	5	4		
	B-HANN	33,3129	0,0942	3	1		
9	**LSTM**	67,6733	4,3137	5	5	51,5091	52,7195
	PSGM	52,8866	0,6958	2	5		
	B-HANN	51,4240	0,1577	3	5		
10	**LSTM**	57,5474	1,2997	5	1	58,1974	58,3895
	PSGM	59,1550	0,5876	4	3		
	B-HANN	58,3040	0,2229	4	4		

When table 10 is examined, B-HANN has smaller means and standard deviations from LSTM and PSGM. The B-ANN has the smallest standard deviation almost all situations. This shows B-HANN produce forecasts has smaller variance. In table 11, the mean of RMSE statistics for LSTM, PSGM and B-HANN methods and RMSE values for a random walk and Holt's exponential smoothing methods are given. Moreover, percentages of success are given and the percentage of success means that the method produced the best results in the mentioned percentage of all experiments. The rank statistics are calculated for all methods.

Table 11. Mean statistics of RMSE for LSTM, PSGM and BHANN and RMSE values for a random walk and Holt's linear trend method in FTSE100 Random Time Series.

Random Data	LSTM	PSGM	BHANN	Random Walk	Holt Linear Trend
1	52,3928	52,9689	**51,8214**	52,1900	52,0963
2	53,3868	**50,0872**	50,4826	50,2549	50,3696
3	56,0447	49,4790	49,4319	**49,3421**	49,4277
4	53,6121	**50,1893**	50,3465	50,2108	50,3992
5	36,8277	34,6996	33,7895	**33,7235**	33,9079
6	64,0471	61,5636	**60,6778**	60,6946	60,6925
7	88,3087	84,3469	**84,1721**	84,6748	84,2412
8	41,3384	34,3204	33,3129	33,8047	**33,2034**
9	67,6733	52,8866	**51,4240**	51,5091	52,7195

10	**57,5474**	59,1550	58,3040	58,1974	58,3895
Percentage of Success	0%	20%	**40%**	30%	10%
Mean of Rank	4,5000	3,5000	**2,1000**	2,3000	2,6000

When table 11 is examined, B-HANN method has the best performance according to the percentage of success and mean of rank statistics. The ANN methods have %60 success for all situations. Moreover, B-ANN has the best results %70 of the situations among ANN methods.

The third data set is the Borsa Istanbul Stock Exchange 100 Index, also called the BIST 100 Index. The data set was daily recorded opening prices between 01/02/2014 and 09/02/2018. In the experimental study, 10 different sub-time series are randomly taken from the whole time series. The observations of random time series are summarized in Table 12.

Table 12. The random time series observation dates and numbers

Series No	Starting date	Ending Date	Number of Observations
1	'2014-01-31'	'2015-12-31'	500
2	'2015-11-12'	'2017-10-11'	500
3	'2016-01-20'	'2017-12-19'	500
4	'2015-06-30'	'2017-05-29'	500
5	'2015-09-01'	'2017-07-31'	500
6	'2016-05-06'	'2017-04-20'	250
7	'2015-03-31'	'2016-03-14'	250
8	'2016-01-27'	'2017-01-10'	250
9	'2014-07-18'	'2015-07-02'	250
10	'2016-03-24'	'2017-03-08'	250

The applications are made in the same design with previous stock exchange data sets. Similar application results are summarized in Table 13 and 14.

Table 13. Statistics for RMSE values for BIST 100 Random Time Series from the methods

Random Data	Methods	Mean	Standard Deviation	Number of Inputs	Number of Hidden Layer Nodes	Random Walk	Holt Linear Trend
1	**LSTM**	1223,6421	15,1430	2	2	1207,08	1210,35

	PSGM	1214,5322	8,1635	4	4		
	B-HANN	1212,3883	1,8548	4	5		
2	**LSTM**	1151,4959	14,1011	1	4	1137,14	1141,48
	PSGM	1168,0600	12,4945	5	3		
	B-HANN	1142,9581	3,0053	5	3		
3	**LSTM**	1270,5354	6,8289	4	2	1294,22	1287,21
	PSGM	1295,6321	5,5221	2	5		
	B-HANN	1288,6971	2,4888	5	1		
4	**LSTM**	922,5593	310,0753	1	5	761,156	758,495
	PSGM	773,3504	6,9704	5	5		
	B-HANN	759,4848	3,6537	4	3		
5	**LSTM**	1010,8165	285,6847	1	5	725,734	716,068
	PSGM	713,0735	2,2901	5	3		
	B-HANN	711,9066	4,4957	2	2		
6	**LSTM**	795,0413	12,7961	1	2	761,709	760,249
	PSGM	740,9551	9,2555	3	5		
	B-HANN	759,6025	1,3078	2	1		
7	**LSTM**	939,8823	24,2007	2	2	852,025	867,407
	PSGM	895,2005	2,0108	1	5		
	B-HANN	869,7583	4,9018	3	3		
8	**LSTM**	860,0633	333,0192	4	5	729,921	729,921
	PSGM	746,2750	6,3012	2	5		
	B-HANN	731,1405	2,4780	5	1		
9	**LSTM**	1238,0248	36,2521	5	2	1181,71	1183,01
	PSGM	1183,3946	20,1003	5	3		
	B-HANN	1182,7036	2,9808	3	5		
10	**LSTM**	1098,7635	35,4691	1	1	894,523	902,259
	PSGM	951,8525	4,3548	2	4		
	B-HANN	904,0714	9,5661	4	2		

Table 14. Mean statistics of RMSE for LSTM, PSGM and BHANN and RMSE values for a random walk and Holt's linear trend method in BIST100 Random Time Series.

Random Data	**LSTM**	**PSGM**	**BHANN**	**Random Walk**	**Holt Linear Trend**
1	1223,6421	1214,5322	1212,3883	1207,0800	1210,3548
2	1151,4959	1168,0600	1142,9581	1137,1357	1141,4752
3	1270,5354	1295,6321	1288,6971	1294,2246	1287,2056
4	922,5593	773,3504	759,4848	761,1564	758,4954
5	1010,8165	713,0735	711,9066	725,7341	716,0679

6	795,0413	740,9551	759,6025	761,7089	760,2486
7	939,8823	895,2005	869,7583	852,0251	867,4073
8	860,0633	746,2750	731,1405	729,9209	729,9209
9	1238,0248	1183,3946	1182,7036	1181,7073	1183,0130
10	1098,7635	951,8525	904,0714	894,5232	902,2589
Percentage of Success	10%	10%	10%	50%	20%
Mean of Rank	4,5000	3,7000	2,5000	2,2000	2,1000

When Table 13 is examined, B-HANN is the best ANN method for %80 of the situations. The B-HANN is outperformed the other ANNs. The random walk and Holt methods produced better forecast results than ANN methods for BIST100 stock exchange. B-HANN method produced very close results to a random walk and Holt methods. If B-HANN compared with a random walk, B-HANN is better than random walk and Holt methods in %40 and %30 of BIST100 applications, respectively. Finally, it can be said that B-HANN can be preferred as a forecasting method for BIST100 data set.

5.2 Applications of Input Significance Tests, Linearity and Nonlinearity Tests and Confidence Intervals

In this section, a time series is randomly generated from the FTSE100 time series by using randomly selected started points. The random time series has 60 observations. The random time series are used to apply input significance tests, linearity and non-linearity tests and obtaining confidence intervals. The last 10 observations are separated as a test set. The confidence intervals are constructed for one step ahead forecasts. The proposed method is applied to time series with $p = 5$ and $n_h = 2$ parameters. Holt linear trend, Yolcu et al. (2019) and B-HANN methods are applied to obtain confidence intervals for 1-step ahead forecasts. Obtained confidence intervals are given in Table 15. Moreover, reliability evaluation (RE), sharpness evaluation (SE), lower bound closeness (LBC), upper bound closeness (UBC) and mean of closeness (MC) criteria are calculated and given in Table 15. RE and CE statistics were used in Yolcu et al. (2019). LBC, UBC and MC criteria are firstly considered in this study. These criteria measure the closeness of bounds to real value without taking into considering bounds contain real values. The formulas of RE, SE, UBC, LBC and MC are given below:

$$RE = \left(\frac{\xi^{(1-\alpha)}}{ntest} - (1-\alpha)\right) x100\% \quad (18)$$

$$SE = \frac{1}{ntest}\sum_{t=1}^{ntest}(UB_t - LB_t) \quad (19)$$

$$LBC = \frac{1}{ntest}\sum_{t=1}^{ntest}|LB_t - X_t| \quad (20)$$

$$UBC = \frac{1}{ntest}\sum_{t=1}^{ntest}|UB_t - X_t| \quad (21)$$

$$MC = \frac{LBC+UBC}{2} \quad (22)$$

Where *ntest* is # of testing points, $\xi^{(1-\alpha)}$ is # times that actual target values do indeed lie within the α-level prediction intervals. Ub_t and Lb_t are the lower and upper bound of the α-level prediction interval.

When Table 15 is examined, Holt's linear trend methods' confidence intervals produced better RE and UBC statistics but it is not good in terms of SE, LBC and MC metrics. Yolcu et al. (2019) method are not better than Holt's linear trend method in terms of RE, SE, UBC and MC. B-HANN is better than Holt's linear trend method in terms of SE, LBC and MC. B-HANN is better than Yolcu et al. (2019) method in terms of all metrics except RE. The B-HANN produced competitive results for the simulated data.

Table 15. Confidence Intervals and some metrics for Random FTSE Data

Test Data	Holt Linear Trend Method		Yolcu et al. (2019) Method		B-HANN	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
6746	6627	6749	6719	6856	6776	6830
6710	6682	6807	6756	6892	6786	6837
6785	6646	6771	6718	6896	6779	6843
6738	6717	6848	6743	6930	6794	6853
6749	6670	6803	6723	6888	6790	6851
6728	6682	6813	6722	6890	6791	6853
6795	6661	6792	6705	6869	6784	6850
6798	6720	6854	6803	6886	6798	6849
6821	6730	6863	6779	6898	6802	6856
6792	6753	6886	6796	6901	6807	6854
RE	-15%		-35%		-55%	
SE	129,79		144,33		56,97	
LBC	77,566		31,92		31,49	
UBC	55,659		124,21		81,2	

MC	66,6125	78,07	56,35

The proposed method produced input significance tests, linearity and nonlinearity tests. These test results are given in Table 16. According to Table 6, all inputs are significant in B-HANN. Moreover, the time series has both linear and non-linear components. Similar linearity and nonlinearity test results are obtained from Yolcu et al. (2019).

Table 16. Test Results from B-HANN for FTSE Random Data

Input Significance Test Results				
İnput 1	**Input 2**	**Input 3**	**Input 4**	**Input 5**
p<0.001	p<0.001	p<0.001	p<0.001	p<0.001
Linearity Test	**Non-Linearity Test**			
p<0.001	p<0.001			

The proposed method produced empirical distributions for all weights and forecasts. In figure 4 and 5, histograms for input weights and first forecast are given, respectively. Empirical distributions can give more information about the estimators. For example, Figure 5 gives us that empirical distribution of the first forecast from B-HANN contains real observation value. The forecasts from bootstrap repetitions are generally bigger than the test observation value. It is possible to see 6650 value as a forecast for first observation from B-HANN. Practitioners can use the empirical distribution of the forecast instead of relying on point forecasts or just relying on the prediction intervals.

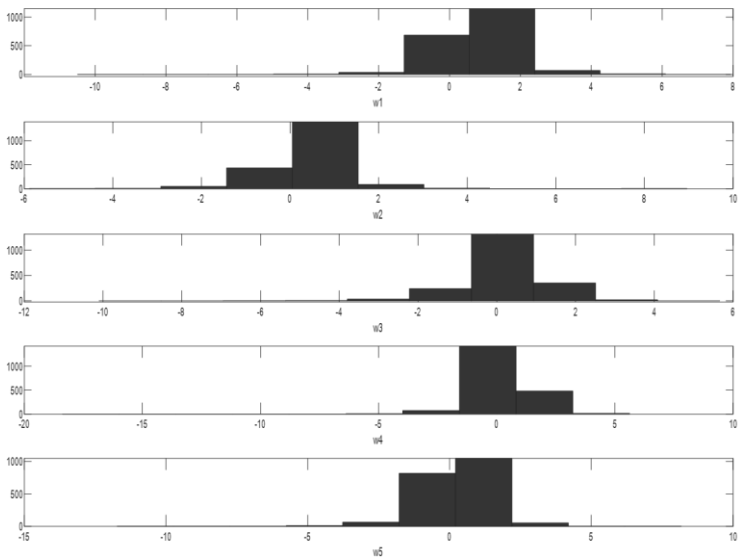

Figure 4. Histogram for Input Weights

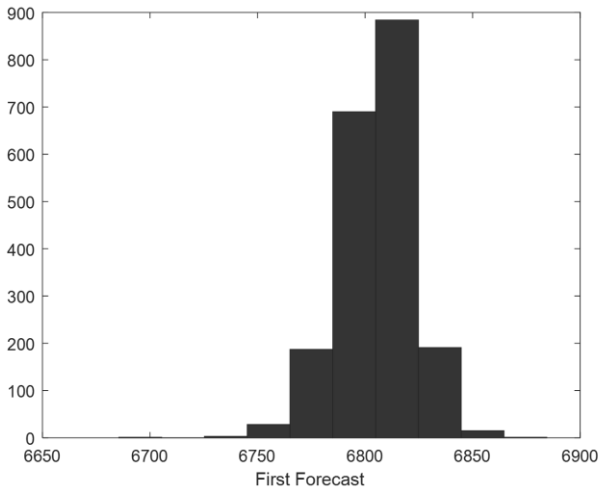

Figure 5. Histogram for 1-step ahead forecast

5.3 Application to M4 Yearly Competition Data Set

The various M-Competitions, as well as other forecast accuracy comparisons (Hyndman, 2020), have proved influential. Most recently Makridakis (2018) published results for competitors in M4 competition which extended the number of series to be analysed to 100K and the number of methods an extended range of ML methods developed by experts. ANNs has been included in some forecasting competitions in the past. ANNs performance generally had not performed well. In the M3 competition, Balkin and Ord (2000) had proposed an automatic neural network modelling tool for univariate time series forecasting. They examined the performance on ANNs and found that ANNs could produce the most successful results for some of the time series. They stated that "results for the M3 and other competitions are generalized over a large number of series. So, simple methods may produce overall better results, but the complex models will perform better for those series with the relevant structure. Such distinctions require that we analyze each series individually". A more detailed study was done by Crone et al. (2014) but overall the performance was disappointing. With an expanded range of ML methods, for the M4 the winner method was a hybrid method of a statistical and an ANN in the competition. Smyl (2020) described the winner method and it is a hybridization of LSTM and an exponential smoothing method. Barker (2020) emphasised that "the M4 competition marked a turning point in the Makridakis competition series, as, for the first time, the winning model was one which would be classified colloquially as machine learning." Barker (2019) classified methods as structured and unstructured. It can be concluded that unstructured ML methods can be

developed to give more successful results Barker (2020). In the M4 Competition, pure ML methods did not demonstrate good forecasting performances. However, ML methods could it seemed to produce good forecast by hybridization of statistical methods.

The B-HANN method proposed in this paper is a pure ML method. Because B-HANN employs bootstrap approaches in its algorithm, someone can comment on it as a hybridization of statistical and ML method. Bootstrap approaches are generally classified into ensemble approaches in the ML literature and are not considered as statistical approaches in the ML literature. To examine the hypothesis implicit in the M4 discussions of the limitations of the ML methods, in this study, we evaluate a pure ML method using the M4 yearly competition data. The obtained results are compared with the best ten methods in the M4 competition. The proposed method is applied with a fixed architecture: The model selection step is omitted to work with a simple model. In the B-HANN, the parameter configuration is $p = 2$, $n_h = 1$, $nbst = 50$, $ntest = 1$ and $MAXITR = 50$. Before applying B-HANN, the series is applied to first-order differencing.

The obtained results are given in Table 17. In table 17, the median of SMAPE values for 23.000-time series are given for different forecast horizons. The other competitor results are taken from Makridakis et al. (2018). The details of other competitor methods can be obtained from Makridakis et al. (2018). These methods are statistical, ML methods and hybrid forecasting methods. When Table 17 is examined, the proposed method has the third rank overall. B-HANN is the best ML approach for the M4 yearly competition data. For 1:2 forecast horizon, B-HANN is better than the winner method and it has the fourth rank. For horizons 3:4, and 5:6 B-HANN has the second rank.

Table 17. M4 yearly competition data results for B-HANN and the best ten method

Submission ID	Method	Median(SMAPE)			
		1:2	3:4	5:6	Total
118	**Syml**	4,9260	7,4807	9,9448	7,8513
260	**Legaki & Koutsouri**	4,7746	7,7020	10,3192	7,9694
New	**B-HANN**	4,9019	7,6545	10,1099	7,9946
69	**Fiorucci & Louzada**	4,8825	7,7369	10,3862	8,0876
245	**Montero-Manso, et al.**	4,8906	7,7547	10,4879	8,1117
36	**Petropoulos & Svetunkov**	4,9996	7,8228	10,5441	8,1434
72	**Jaganathan & Prakash**	5,0079	7,8677	10,6704	8,2192
237	**Pawlikowski, et al.**	4,9643	7,9624	10,7065	8,3280

5	**Spiliotis & Assimakopoulos**	5,1263	8,1022	10,9940	8,4853
39	**Pedregal, et al.**	5,1360	8,0908	10,9929	8,4882
238	**Doornik, et al.**	5,2742	8,3924	11,5181	8,7721

N.B. The error measure used is the symmetric MAPE to make the results comparable to those given in Makridakis (2018).

According to Table 17, the proposed method is a rival for the best two methods. More results are given to compare the best 2 methods and B-HANN method. For each series, SMAPE values are compared between Syml method and B-HANN, Legaki &Koutsouri and B-HANN, the results are given in Table 18.

Table 18. Comparison of the best two methods with B-HANN on the M4 annual data

Compared Methods	**Success Rate for B-HANN**			
	1:2	**3:4**	**5:6**	**Total**
Syml vs B-HANN	49,35%	47,1700%	48,4800%	48,0000%
Legaki &Koutsouri vs B-HANN	49,62%	50,7900%	50,2000%	50,4300%

According to Table 18, B-HANN has smaller SMAPE than the Syml method 49,35% of the 23000 time series for forecast horizons 1:2. For other forecast horizons, B-HANN is almost as good as the Syml method half of all series. Moreover, B-HANN is better than Legaki&Koutsouri method for 3:4, 5:6 forecast horizons and total.

According to Table 18, B-HANN has smaller SMAPE than the Syml method on 48% of the 23000-time series while the figure is 50.4% when compared to Legaki&Koutsouri, thereby demonstrating the strength of a pure ML method.

6. Conclusions and Discussion

Artificial neural networks can provide powerful forecasting methods for some time series. It is well known that ANNs are useful for stock exchange data sets because results of traditional time series methods produce forecasts that are very close to random walk forecasts. Although ANNs can produce good point forecasts for some kind of time series, it is not easy to apply

hypothesis tests and to obtain empirical distribution for the estimators. In this study, a new hybrid ANN architecture is proposed and it is combined with an identically independent distributed residual bootstrap method to obtain probabilistic results such as empirical distributions for the forecasts, prediction intervals and hypothesis tests. The forecasting performance of the B-HANN method is investigated on FTSE, BIST and SP500 stock exchange data sets with a random choice of data segments. B-HANN produced the best results for FTSE and SP500 but its results were not better than traditional methods for BIST. However, B-HANN produced better results than alternative methods, LSTM and PSGM ANNs for all three stock exchange indices. Moreover, one random data application is used as an example showing how to obtain probabilistic results from the B-HANN. The confidence (prediction) intervals obtained were compared with an ANN and a traditional method on a simulated time series. It was seen that the B-HANN can produce meaningful and good distributional forecasts in terms of most standard metrics. Moreover, B-HANN is the best ML method for M4 yearly competition and B-HANN has the third rank for total forecast horizons. Moreover, B-HANN was compared with the best two methods and it is concluded that B-HANN is a serious rival to them. In future studies, different bootstrap techniques can be used to enhance the performance of the proposed method. Moreover, simulation studies have the potential to investigate hypothesis test performance under different bootstrap techniques. B-HANN can be extended for seasonal or periodic series and tested on the extended M4 data set.

Acknowledgement

This study is supported by Turkish Science and Technological Researches Foundation with Award Number:1059B191800872, Recipient: Erol Egrioglu

References

Anders, U., & Korn, O. (1999). Model selection in neural networks. *Neural Network*, 12, 309-323.

Balkin, S.D., & Ord, J.K. (2000). Automatic neural network modeling for univariate time series, *International Journal of Forecasting*, 16, 509–515.

Barker J. (2020). Machine learning in M4: What makes a good unstructured model?, *International Journal of Forecasting*, https://doi.org/10.1016/j.ijforecast.2019.06.001.

Barrow D.K, & Crone S.F. (2016). Cross-validation aggregation for combining autoregressive neural network forecasts. *International Journal of Forecasting*, 32, 1120–1137.

Bradley M.D., & Jansen D.W. (2004). Forecasting with a nonlinear dynamic model of stock returns and industrial production. *International Journal of Forecasting*, 20, 321– 342.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2014). Advances in forecasting with neural networks? Empirical evidence from the nn3 competition on time series prediction (vol 27, pg 635, 2011). International Journal of Forecasting, 30, 1138-1138.

Dantas T.M., & Oliveira, F.L.C. (2018). Improving time series forecasting: An approach combining bootstrap aggregation, clusters and exponential smoothing. *International Journal of Forecasting*, 34 (2018) 748–761.

Ebrahimpour, R., Nikooc, H., Masoudnia, S., Yousef, M.R., & Ghaem, M.S. (2011). Mixture of MLP-experts for trend forecasting of time series: A case study of the Tehran stock exchange. *International Journal of Forecasting*, 27, 804–816.

Feng Y., & Zhou C. (2015). Forecasting financial market activity using a semi-parametric fractionally integrated Log-ACD. *International Journal of Forecasting*, 31, 349–363.

Fildes R. (2020). Learning from forecasting competitions, *International Journal of Forecasting*, 36, 186–188.

Gilliland, M. (2019). The value added by machine learning approaches in forecasting. *International Journal of Forecasting*, https://doi.org/10.1016/j.ijforecast.2019.04.016.

Gorr W.L., Nagin D., & Szczypula J. (1994). Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, 10, 17-34.

Granger C.W.J. (1992). Forecasting stock market prices: Lessons for forecasters. *International Journal of Forecasting*, 8, 3-13.

Hyndman, R. J. (2020). A brief history of forecasting competitions. International Journal of Forecasting, 36(1), 7-14. doi:https://doi.org/10.1016/j.ijforecast.2019.03.015

Karaboga D., Akay B., & Ozturk C. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: Torra V., Narukawa Y., Yoshida Y. (eds) Modeling Decisions for Artificial Intelligence. MDAI 2007. Lecture Notes in Computer Science, vol 4617. Springer, Berlin, Heidelberg.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. TR-06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Kourentzes N., Barrow D.K., & Crone F.S. (2014), Neural network ensemble operators for time series forecasting. *Expert systems with Applications*, 41, 4235-4244.

Lam J.-P., & Veall, M.R. (2002). Bootstrap prediction intervals for single period regression forecasts. *International Journal of Forecasting*, 18, 125–130.

Lee, T.H., White, H., & Granger, C. (1993). Testing for neglected nonlinearity in time series models. *Journal Econometrics*, 56, 269-290.

Lohrmann C., & Luukka P. (2019). Classification of intraday S&P500 returns with a Random Forest, *International Journal of Forecasting*, 35, 390–407.

Makridakis S., Spiliotis, E. & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward, *International Journal of Forecasting*, 34, 802–808.

Masarotto G. (1990). Bootstrap prediction intervals for autoregressions. *International Journal of Forecasting*, 6, 229-239.

McMillan D.G. (2007). Non-linear forecasting of stock returns: Does volume help?, *International Journal of Forecasting*, 23, 115–126.

Mohammadi, S. (2018). A new test for the significance of neural network inputs.

Neurocomputing, 273, 304-322.

Moody, J. (1994). Prediction risk and architecture selection for neural networks. From Statistics to Neural Networks, *NATO ASI (Series F: Computer and Systems Sciences*), 136, 147-165.

Moody, J., & Utans, J. (1994). *Architecture selection strategies for neural networks: application to corporate bond rating prediction*. Neural networks in the Capital Markets. John Wiley&Sons. New York, 277-300.

Nyberg, H. (2011). Forecasting the direction of the US stock market with dynamic binary probit models. *International Journal of Forecasting*, 27, 561–578.

Olson D., & Mossman C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of Forecasting*, 19, 453–465.

Onkal, D. (2019). M4 competition: What's next?. *International Journal of Forecasting*, https://doi.org/10.1016/j.ijforecast.2019.03.011.

Politis, L.P., & Dimitris, N. (2016). Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions, *Journal of Statistical Planing and Inference*, 177, 1-27.

Refenes, A.-P.N. & Zapranis, A.D. (1999). Neural Model Identi®cation, Variable Selection and Model Adequacy. *Journal of Forecasting*, 18, 299-332.

Sarantis, N. (2001). Nonlinearities, cyclical behaviour and predictability in stock markets: international evidence. *International Journal of Forecasting*, 17, 459–482.

Smyl, S. (2019). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, https://doi.org/10.1016/j.ijforecast.2019.03.017.

Szafranek S. (2019). Bagged neural networks for forecasting Polish (low) inflation. *International Journal of Forecasting*, 35, 1042–1059.

Terasvirta, T., Lin, C.F., & Granger, C. (1993). Power of the neural network linearity in time series models. *J. Time Series Analysis*, 14, 209-220.

Tiwari, M.K., & Chattejee, C. (2010a). Uncertainity assessment and ensemble flood forecasting using bootstrap based artificial neural networks. *Journal of Hydrology*, 382, 20-33.

Tiwari, M.K., & Chattejee, C. (2010b). Development of an accurate and reliable hourly flood forecasting model using wavelet-bootstrap-ANN (WBANN) hybrid approach. *Journal of Hydrology*, 384, 458-470.

White, H. (1989). An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. *Proceedings of the International Joint conference on Neural Networks*, II, New York, IEEE Press, 451-455.

Yolcu U., Egrioglu E., Bas E., Yolcu O.C., & Dalar A.Z. (2019) Probabilistic forecasting, linearity and nonlinearity hypothesis tests with bootstrapped linear and nonlinear artificial neural network. *Journal of Experimental & Theoretical Artificial Intelligence*, https://doi.org/10.1080/0952813X.2019.1595167.

Yolcu, U., Jin, Y., & Egrioglu, E. (2017). An ensemble of single multiplicative neuron models for probabilistic prediction. 2016 IEEE Symposium Series on Computational Intelligence. art. no. 7849975.

Zapranis, A.D., & Refenes, A.-P.N. (1999). Principles of neural model identification, selection, and adequacy: with applications to financial econometrics, Springer-Verlag, London.