Cemile ÖZGÜR, PhD Candidate
E-mail: ozgurcemile@yahoo.com
School of Business, Department of Finance
Istanbul University
Professor Vedat SARIKOVANLIK, PhD
E-mail: vedsari@istanbul.edu.tr
School of Business, Department of Finance
Istanbul University

FORECASTING BIST100 AND NASDAQ INDICES WITH SINGLE AND HYBRID MACHINE LEARNING ALGORITHMS

Abstract. *The aim of this paper is to investigate stock market return forecasting performance of single and the developed novel hybrid machine learning (ML) algorithms. Daily returns of BIST100 and NASDAQ indices are predicted by series specific GARCH and ARMA-GARCH as well as three different ML algorithms that are Random Forest, XGBoost and Artificial Neural Networks (ANN). New hybrid ML models incorporating forecasts of the traditional (ARMA-)GARCH and the three ML algorithms are developed. Accuracy of the out-of-sample predictions of the methods are reported both for the single and hybrid models including pre-COVID-19, post-COVID-19 and the full sample test periods. Moreover, a simple trading strategy is applied in order to assess the economic impact of employing a specific forecasting model. According to the obtained accuracy metrics and the results of the trading strategy, developed novel hybrid models suggest quite promising results compared to the forecasts of the other models, especially (ARMA-)GARCH.*

Keywords*: Stock Markets, Random Forests, XGBoost, Artificial Neural Networks, Machine Learning, Hybrid Models.*

JEL Classification: C02, C45, C63, G17

1. Introduction

The research on the predictability of stock markets is not a new phenomenon that dates back as far as to 1900s when Louis Bachelier argued that stock prices follow a Random Walk (Bachelier, 1964). Until 1980s, albeit the exceptions, the main stream of research supporting the Random Walk and Efficient Market Hypothesis (Fama, 1970) was growing on the impossibility of predicting individual stock prices/returns and stock markets as a whole. However, the works of Campbell (1987), Fama and French (1988) and the others showed that prices

DOI: 10.24818/18423264/56.3.22.15

have temporary and permanent components that can be used in prediction at least to some extent. Since then, the research on the predictability of stock markets has developed a new stream focusing on the predictive ability of varying econometric tools and variables using past price/return observations, fundamental valuation ratios or even some macroeconomic variables.

Following the latest stream of research and the recent advances in technology, this paper investigates stock market return forecasting performance of single and hybrid models derived from the forecasts of the traditional ARMA-GARCH and three ML algorithms. In forecasting prices or returns of stock markets, the traditional time series models such as ARIMA and GARCH (Bollerslev, 1986) are frequently applied. From the machine learning algorithms, Support Vector Machines and Neural Networks are the ones commonly employed. In this paper, additional to the ARMA, GARCH and Artificial Neural Networks (ANN), two nonlinear tree-based ensemble learning methods are also employed in order to forecast daily returns of two indices, BIST100 and NASDAQ composite. From the tree-based algorithms, Random Forest (Breiman, 2001) and XGBoost (Chen & Guestrin, 2016) are among the top two recently developed highly randomized and effective ensemble learning methods that can be applied over a wide range of prediction tasks with multiple data sets. Even though, the literature on the applications and forecasting performance of Random Forest is growing, there is still very few papers evaluating the financial forecasting performance of XGBoost algorithm.

Forecasting models that combine the residuals or outputs of the traditional statistical models and/or the algorithms of different ML methods are called hybrid models. The purpose of developing and employing hybrid models in forecasting tasks is to be able to improve the performance of the single models. A hybrid algorithm can model varying linear and/or nonlinear patterns of a data at the same time. This property of hybrid algorithms makes them an excellent candidate for stock market forecasting tasks. As a result, in this paper the developed hybrid models employ a combination of the return forecasts of a) an ARMA-GARCH-ML hybrid algorithm that models the remaining patterns left in the residuals of ARMA-GARCH and b) a ML algorithm employing various features, such as technical indicators, exchange rates and commodity prices. These hybrid models that incorporate the forecasts of the single as well as the hybrid models are developed in order to be able to capture the remaining patterns of the data and generate more accurate return forecasts. Moreover, the main contribution of this research can be summarized as follows: First of all, stock market forecasting literature is mainly focused on the forecasts of either price or the direction of the stock or stock markets. On the other hand, to be able to form an expectation on the magnitude of returns is also very important since one can allocate scarce resources among different alternatives depending on the magnitude of the expected returns. In this paper, instead of price or the direction of movement, daily returns of BIST100 and NASDAQ Composite indices are forecasted. Second, as also mentioned above,

additional to the most frequently applied time series forecasting tools, in this research new hybrid models combining the forecasts of the previously mentioned single and the hybrid ML models are developed. Moreover, most of the studies employ a static approach in forecasting financial time series and either use k-fold cross-validation in order to tune the model hyper-parameters or directly apply the default parameters of the algorithms without taking into account time order of the data. On the other hand, this paper evaluates stock market return forecasting performance of the single and the developed hybrid ML methods by applying time series cross-validation to train and tune the parameters of the algorithms and employ a sliding-windows approach to obtain full sample return forecasts of the indices. Third, a simple trading strategy is also developed in order to assess economic gains obtained by applying a specific forecasting method.

2. Methodology

This section briefly introduces the algorithms applied to forecast returns of the stock indices and the metrics used for evaluating the accuracy of the predictions. Detailed information can be obtained from the references supplied in each subsection.

2.1 ARMA-GARCH

In finance literature ARMA-GARCH, a combination of ARMA and GARCH processes, are one of the most commonly applied statistical models in predicting financial price and/or return series. The ARMA(P,Q) process can be written as:

$$r_t = \sum_{i=1}^{P} \phi_i r_{t-i} + \sum_{j=1}^{Q} \theta_j \epsilon_{t-j} + \epsilon_t \quad (1)$$

where r_t is the dependent variable at time *t*, ϵ_t is the residual term and ϕ_i and θ_j are the coefficients of Autoregressive (AR) and Moving Average (MA) components of the equation. On the other hand, in case of time varying volatility and autocorrelation in the squared residuals of the series, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Engle, 1982; Bollerslev, 1986) processes are applied to model the conditional heteroskedasticity and the heavy-tailed distributions. The GARCH(p,q) model is:

$$\epsilon_t = \sigma_t \varepsilon_t \quad \wedge \quad \sigma_t^2 = \omega + \sum_m^p \alpha_m \epsilon_{t-m}^2 + \sum_k^q \beta_k \sigma_{t-k}^2 \quad (2)$$

where ϵ_t is the residuals of ARMA(P,Q), σ_t^2 is the conditional variance, ω is the intercept and α_m and β_k are the model parameters.

2.2 Random Forests

Breiman (2001) proposed *Random Forests* as one of a randomized decision-tree based ensemble learning methods. Compared to decision trees, Random Forests has two sources of randomization. One of the sources of randomization is that the algorithm uses bootstrapped aggregation for predictions.

Bootstrapped aggregation or bagging is a procedure of repeatedly obtaining a number of separate subsamples from the training set in order to train and produce an average prediction from the predictions of each subsample. Second, Random Forest also randomizes the predictors considered in each node split of a tree by selecting a number of *m* predictors from a total of *p* number of predictors during the bootstrapped aggregation. The default number of predictors considered in each node split of a tree is set to $m = p/3$ for regression and $m = \sqrt{p}$ for classification problems (James et al., 2013).

2.3 eXtreme Gradient Boosting (XGBoost)

XGBoost is one of the another recently developed ensemble learning algorithms proposed by Chen and Guestrin (2016). XGBoost is defined as a high performing, efficient and a highly scalable advanced gradient boosting algorithm. The objective of the algorithm for tree boosting is to minimize the loss function $\mathcal{L}$ defined as a measure of the difference between the real (y_i) and forecasted values ($\hat{y}_i$) of the dependent (respondent) variable including a regularization term to prevent model overfitting. The objective function as an additive model is defined by (Chen & Guestrin, 2016):

$$\mathcal{L} = \sum_i^n \ell(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (3)$$

where K is the number of additive functions used to forecast the respondent variable and each f_k is an independent tree structure. $\Omega(f_k)$ is the regularization term estimated from:

$$\Omega(f) = \gamma T + 0.5\lambda \parallel w \parallel^2 \quad (4)$$

where T is the number of leaves in the tree, γ is the minimum loss reduction required for further division of internal nodes and λ is the coefficient of the ℓ_2-norm of leaf scores (w).

2.4. Artificial Neural Network

Inspired from the biological neural networks, Artificial Neural Network (ANN) is a nonparametric, nonlinear system that is able to learn complex patterns of a given data. In 1940s, McCulloch and Pitts (1943) proposed to apply logic and computation to model neural activities of human brain (Carbonell et al., 1983). Since then, their idea is followed by many researchers that helped to improve and develop varying types of neural networks.

On the other hand, developed by Rosenblatt (1962), the simplest neural networks with a threshold activation function are called *perceptrons* (Bishop, 1995). A perceptron consists two layers, an input and an output layer. In order to improve the flexibility and performance of perceptrons, intermediate layers are added between the input and output layers. A feed-forward ANN with at least three layers is called a *Multilayer Perceptron* (MLP). The estimated function of a MLP consisting three layers (input, hidden and output layers) can be written as:

$$y(x) = f(b_0 + \sum_{i=1}^L w_i \cdot f(b_{0i} + \sum_{j=1}^J \omega_{ji} x_j)) \quad (5)$$

where $f(\cdot)$ is the activation function, L is the number of neurons in the hidden layer, ω_{ji} is the weight of the link (synapse) between the *j*-th input neuron and *i*-th hidden neuron, w_i is the weight of the link between the *i*-th hidden neuron and the output neuron. Moreover, b_0, b_{0i} are the intercepts (bias neurons) of the output and the *i*-th hidden neurons, respectively. During the functioning of a MLP, neurons in the hidden and output layers receive input signals from the preceding neurons among synapses or the weighted links and compute output signals by first combining the weighted inputs and then by processing them with a non-linear activation or transfer function. Sigmoid, logistic or hyperbolic tangent are the commonly employed activation functions. In this paper, the researchers employed a three- and four-layer MLP with the hyperbolic tangent activation function. A diagram of a fully-connected three-layer MLP is given in Figure 1. The first unit in the diagram is called *input layer* in which *p* number of features or explanatory variables are fed into separate nodes. The nodes in each layer are the neurons that are fully connected by synapses and provide outputs to the successive layer. In Figure 1, the internal unit has one hidden layer that consists three neurons. As the number of hidden layers as well as the number of neurons in each layer increase, the complexity of a MLP model also increases. In the diagram, the final unit, the output layer has one neuron since there is only one variable to predict.

2.5 Performance Evaluation

The return forecasting performance of models are evaluated with two accuracy metrics estimated by comparing the predicted values with the realized previously unseen data. The first metric of Root Mean Squared Error (RMSE) is estimated as follows:

$$RMSE = \left(N^{-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right)^{1/2} \quad (6)$$

where y_i and $\hat{y}_i$ are the out of sample realized and predicted values of the respondent variable, respectively and N is the total number of out of sample observations. The second metric of Mean Absolute Error (MAE) is estimated by:

$$MAE = N^{-1}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (7)$$

Additionally, in order to be able to observe the proportion of return forecasts that have the similar sign as the realized values, the Sign Symmetry (SS) statistic is also estimated by:

$$SS = \frac{1}{N}\sum_{i}^{N}\mathbf{1}_i \quad \wedge \quad \mathbf{1}_i = \begin{cases} 1, & sign(y_i) = sign(\hat{y}_i) \\ 0, & sign(y_i) \neq sign(\hat{y}_i) \end{cases} \quad (8)$$

where $\mathbf{1}_i$ is the indicator function taking a value of either one or zero.

2.6. Trading Strategy

Additional to the accuracy metrics, a simple trading strategy is also developed in order to compare the models in terms of final value obtained at the end of the strategy. The trading strategy final value (TSFV) is calculated as:

$$TSFV = P \cdot \prod_{i=1}^{N}(1 + r_i) \tag{9}$$

where P is the principal amount at the beginning of the strategy, N is the total number of out of sample observations and r_i equals to:

$$r_i = \begin{cases} r_i = y_i, & if\ \hat{y}_i > 0 \\ r_i = r_{ty}, & if\ \hat{y}_i \leq 0 \end{cases} \tag{10}$$

where $\hat{y}_i$ is the *i*-th return forecast of a model, y_i is the *i*-th out of sample observation of the realized return series and r_{ty} is the daily yield of 10-year government bonds obtained at the day of the forecasts.

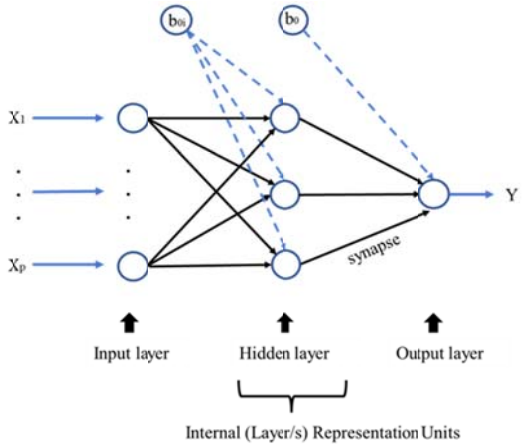

Figure 1. Schematic of a one hidden layer MLP

The principal amount is assumed to be equal to 100 at the beginning of each test period. Moreover, an equally weighted trading strategy is also developed among the single and hybrid models in order to be able to compare the period specific average performance of the models. On the following section, the composition of the single and the two hybrid models (hybrid 1 and hybrid 2) are explained more in detail. On the other hand, equally weighted trading strategies of the single (EWTSs), hybrid 1 (EWTSh1) and hybrid 2 (EWTSh2) models are calculated as:

$$EWTSc = \frac{P}{k} \cdot \sum_{j=1}^{k}\left(\prod_{i=1}^{N}(1 + r_{ij})\right) \tag{11}$$

where *c* defines the class of the models, *c=s* for single ML, *c=h1* for hybrid 1 and *c=h2* for hybrid 2 models. *P* and *N* are as defined above and *k* is the total number of forecasting models in each class. For example, if *c=s* than *k=3* (RF, XGBoost and ANN) and if *c=h2* than *k=4* since four hybrid 2 models are developed (see the 3.4 Hybrid Models subsection). Similar as above, r_{ij} is equal to y_i if the *i*-th return forecast of the *j*-th model belonging to the class *c* is greater than zero ($\hat{y}_{ij} > 0$), otherwise r_{ij} is equal to r_{ty}.

3. Data and Empirical Results

3.1 Data

This research employs daily price series of two stock market indices, BIST100 index of Turkish equity market and NASDAQ Composite index of US equity market. Daily price series of the indices are obtained for the period of October 2014 - May 2021 for BIST100 and December 2014 – May 2021 for NASDAQ. Daily logarithmic returns are estimated by:

$$r_t = ln(P_t/P_{t-1}) \quad (12)$$

Descriptive statistics of the returns are given in Table 1. The analysis period of this research includes pre-COVID-19 and the outbreak of the COVID-19 virus with the worldwide bearish financial market conditions following the announcement of World Health Organization on 11th of March 2020 that classified the spread of the virus as a pandemic. As a result, the data is divided in four test periods in order to be able to compare the forecasting performance of the models in pre- and post-COVID-19 periods. Moreover, as a final step, the aggregate (full sample) performance of the models is also evaluated.

Table 1. Descriptive Statistics of the Index Returns – Full Sample

Index	Mean	Median	Min	Max	SD	Skew.	Kurt.	JB	ADF
BIST100	0.0003	0.0010	-0.1031	0.0581	0.0139	-0.8511	7.6643	0	0.01
NASDAQ	0.0007	0.0012	-0.1315	0.0893	0.0132	-0.9166	15.7653	0	0.01

Note: p values of the Jarque-Bera (JB) and Augmented Dickey-Fuller (ADF) tests are reported.

3.2 Data pre-processing

Obtained time series of the variables are first cleaned from the missing values by removing NAs if there is any. Since returns of the indices are forecasted by single and hybrid models for varying periods, the first 600 observations of the data in each period are employed for fitting / training the models and the remaining observations are reserved for testing. During the train/test split, the order of the data is preserved taking into account time series characteristics of the data.

On the next step, series specific outliers of the train sets are defined according to being in or out of the range of $[(Q_1 - 1.5\,IQR), (Q_3 + 1.5\,IQR)]$. The range is defined by the first quartile (Q_1), the third quartile (Q_3) and *IQR* which is the Inter Quartile Range estimated by $Q_3 - Q_1$. Observations smaller (greater) than $Q_1 - 1.5\,IQR$ $(Q_3 + 1.5\,IQR)$ are replaced by $Q_1 - 1.5\,IQR$ $(Q_3 + 1.5\,IQR)$. Additionally, since some of the data is in different scales, both the respondent variables and the features are standardized. It is also important to note that in order to prevent data leakage from the test sample, standardization parameters are estimated from the outlier corrected train data and applied to both train and test data.

3.3. Single Models

Daily returns of the indices are forecasted by single and hybrid models using the previously explained forecasting algorithms. In the first step, return series are modelled with the traditional (ARMA-)GARCH processes and one-day ahead returns of the indices are forecasted. As mentioned above, in each test period outlier corrected first 600 observations of the data is employed for the first (ARMA-)GARCH fit window in order to obtain one-day ahead return forecasts. On the other hand, the rest of the forecasts of (ARMA-)GARCH is obtained with a one-day ahead rolling windows approach. Every (ARMA-)GARCH fitting window is corrected for outliers and tested for the existence of serial auto-correlation and heteroscedasticity. Moreover, (ARMA-)GARCH processes are re-fitted and their orders and parameters are re-estimated in each fitting window. For this purpose, *forecast* (Hyndman et al., 2019; Hyndman & Khandakar, 2008) and *rugarch* (Ghalanos, 2019) packages of R software (R Core Team, 2019) are employed. This approach enabled the researchers to find and employ the most suitable ARMA-GARCH specification for each window that is able to model series specific characteristics, rather than fitting one ARMA-GARCH specification to all.

In the second step, using the single ML algorithms (Random Forest (RF), XGBoost (XG) and Artificial Neural Networks (ANN)), daily return forecasts of the indices are obtained. For this purpose, each single ML algorithm employed seventeen features. A list of the features that are used for predicting one-day ahead index returns is given in Table 2. In order to be able to define the forecasting methodology more formally, let **R** be the vector of observations of the respondent (outcome) variable *r* and $f()$ define an unknown function mapping features to *r*:

$$\boldsymbol{R} = \begin{pmatrix} r_{t+1} \\ \vdots \\ r_{t+n} \end{pmatrix} = f\left(\begin{matrix} r_t & r_{t-1} & x_{t,p} \\ \vdots & \vdots & \vdots \\ r_{t+n-1} & r_{t+n-2} & x_{t+n-1,p} \end{matrix} \right) \tag{13}$$

Forecasts of a single ML (RFs, XGs or ANNs) model can be written in the general form of:

$$\hat{r}_{s,t+1} = f(r_t\,,\ r_{t-1}\,,\ x_{t,p}) + \epsilon_{t+1} \tag{14}$$

where *s* is for the single ML model, *n* is the total number of observations, *t* is the day of the observation, ϵ is the residual term and $p = 1,2,\ldots,15$ is the number of the features other than the lagged values of the respondent variable. Following the data pre-processing, each of the three ML models (RF, XG and MLP) are trained on the first 600 observations and the rest of the data of the first test period (test sample: 108 observations till 11th of March 2020) is employed for testing the models. Forecasts of the following periods are obtained with a sliding windows approach. The train sample of TP1 is rolled forward by 100 days and the train sample of the test period 2 (TP2) that also includes 600 observations is obtained. Similarly, the rest of the unseen data of the TP2 is employed for testing the forecasting performance of the models (test sample: 92 observations beginning from 12th of March 2020). Train samples of TP3 and TP4 that also include 600 observations are obtained by the same 100 days sliding windows approach (see

Figure 2). Moreover, full sample performance of the models is also evaluated by comparing all the model specific out-of-sample forecasts (TP1 + TP2 + TP3 + TP4) with the realized values.

Table 2. Features Employed in the Predictions of the Index Returns

Features	Description / Source
r_t	Index returns at time t
r_{t-1}	Index returns at time t-1
R_{vix}	Return of the CBOE Volatility index (VIX) at time t $[(C_t - C_{t-1})/C_{t-1}]$ / finance.yahoo.com
C_t- SMA20	Close price of the respondent variable (index) at time t minus 20-day simple moving average
$R_{eur-usd}$	One day rate of change* in the EUR/USD exchange rate (only for NASDAQ) / investing.com
$R_{cny-usd}$	One day rate of change in the CNY/USD exchange rate (only for NASDAQ) / investing.com
range	High price minus low price of the index at time t (H_t - L_t)
R_{vol}	One day rate of change in the daily trading volume $[(Vol_t - Vol_{t-1})/Vol_{t-1}]$ / investing.com
SMA5 - SMA20	5-day simple moving average minus 20-day simple moving average of the index close prices
ROC(5)	Rate of Change / Momentum of the index over 5 days
MACD Histogram	MACD - Signal: Moving Average Convergence Divergence Oscillator minus the signal
$R_{eur-try}$	One day rate of change in the EUR/TRY exchange rate (only for BIST100) / investing.com
$R_{usd-try}$	One day rate of change in the USD/TRY exchange rate (only for BIST100) / investing.com
CCI(n)**	The Commodity Channel Index (n=20)
UOs(7,14,28)	The Ultimate Oscillator developed to capture momentum across different time periods
R_{gold}	One day rate of change in the close prices of gold / finance.yahoo.com
WPR	William's %R
R_{oil}	One day rate of change in the close prices of crude oil / finance.yahoo.com
R_{gy}	One day rate of change in the 10-year government bond yields (Turkey and US) / investing.com

Notes: *One day rate of change of a variable is calculated from: $[(P_t - P_{t-1}) / P_{t-1}]$ where P_t is the value of the variable at time t obtained from the specified source. **n : number of days for moving average.

3.4 Hybrid Models

Forecasts of one day ahead index returns are also obtained from two types of hybrid models derived from the combinations of the single models explained in the previous sections. The first type of hybrid model (h1) develops over the predictions of (ARMA-)GARCH by re-modelling and forecasting its residuals with ML algorithms. More formally, let r_{t+1} be the realized (observed) returns and $\hat{r}_{t+1}$ be the one-day ahead return forecasts of (ARMA-)GARCH:

$$r_{t+1} = \hat{r}_{t+1} + \epsilon_{t+1} \quad (15)$$

The residuals in Equation 15 that could not be modelled by (ARMA-)GARCH processes, are fitted by one of the three ML models and one-day ahead residual forecasts are obtained. The final return forecasts of the first hybrid models ($\hat{r}_{h1,t+1}$) are the sum of the (ARMA-)GARCH return forecasts and the residual forecasts of the chosen ML model.

$$\hat{\epsilon}_{t+1} = f(\epsilon_{t-i}) + \varepsilon_{t+1}, \quad \text{for } i \in \{0,1,2,3,4\} \quad (16)$$

$$\hat{r}_{h1,t+1} = \hat{r}_{t+1} + \hat{\epsilon}_{t+1} \quad (17)$$

where ε is the residual term and t is the day of the observation. If the RF algorithm is chosen in order to forecast one-day ahead residuals ($\hat{\epsilon}_{t+1}$), the final forecasts of the first hybrid model is named as RFh1 (XGh1 and ANNh1 are for the other two ML methods). Residual re-modelling is not a new approach in finance, see for example the work of Pai and Lin (2005).

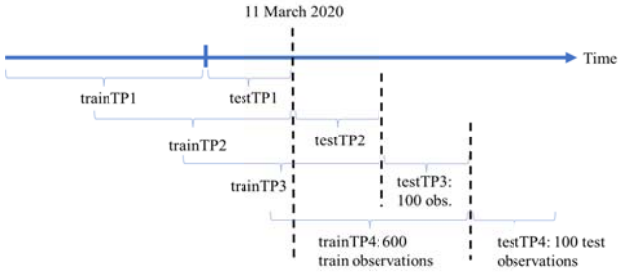

Figure 2. A Schematic Illustration of the Sliding Windows Approach

The second type of hybrid model (h2), as far as we know is novel and developed over the predictions of hybrid model 1 (h1) and the predictions of the single ML models defined in the previous subsection by also employing one of the three ML models. Let $\hat{r}_{h2,t+1}$ be one-day ahead return forecasts of hybrid 2 model and ν be the residual term:

$$\hat{r}_{h2,t+1} = f(\hat{r}_{h1,t+1}, \hat{r}_{s,t+1}, r_{t-k}) + \nu, \quad \text{for} \quad k \in \{0,1,2,3,4,5\} \qquad (18)$$

If the return forecasts of a hybrid 2 model (h2) are obtained with the ANN algorithm by employing return forecasts of RFh1 ($\hat{r}_{h1,t+1}$) and XGs ($\hat{r}_{s,t+1}$) as the model inputs (features) additional to the lagged returns (r_{t-k}), the model is named as RFh1-XGs-ANN. Overall, four variations of the h2 models are developed that are: RFh1-XGs-ANN, XGh1-ANNs-RF, ANNh1-XGs-RF and XGh1-RFs-ANN. One-day ahead return forecasts of the second hybrid models (h2) are also obtained with the same 100 days sliding windows approach as explained in the 3.3. Single Models subsection. Step by step, a schematic explanation of the overall forecasting methodology can be found in Figure 3.

3.5 Cross-Validation and Hyperparameter Search

Forecasts of one day ahead index returns of Random Forest, XGBoost and ANN algorithms are obtained by employing *randomForest* (Liaw & Wiener, 2002), *xgboost* (Chen et al., 2021) and *neuralnet* (Fritsch et al., 2019) packages of R Software (R Core Team, 2019), respectively. Moreover, even though, k-fold cross-validation is a frequently applied parameter tuning method, it assumes that the data is independent and identically distributed which ignores the very well documented stylized facts of financial time series. As a result, in order to tune the hyperparameters of each algorithm, time series cross-validation (Hyndman & Athanasopoulos, 2018) as a rolling forecasting origin resampling method is applied with a fixed rolling window of 100 days and h=1day validation samples. From the applied three ML algorithms, the two parameters of Random Forest that can be tuned with cross-validation are the number of trees (ntree) and the number of features considered at each node split (mtry). In this research, the ntree parameter is varied in [100,1000] range by increments of 100 for the single and hybrid models. On the other hand, since the number of features is different in single and

hybrid ML models, mtry is varied in [2,17] range by increments of 1 for the single and in the ranges of [1,5] and [2,8] for the hybrid models, h1 and h2 respectively.

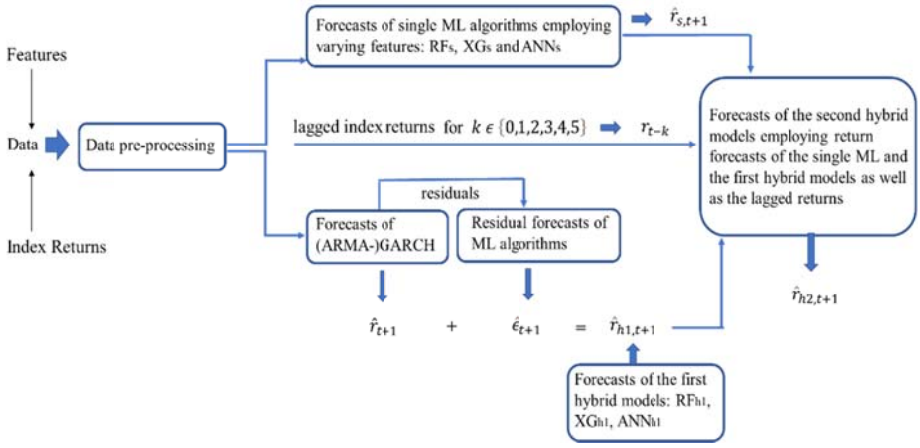

Figure 3. A Schematic Representation of the Forecasting Methodology

The number of hyperparameters of XGBoost algorithm that needs to be tuned is far more compared to Random Forest. The ones that are tuned in this research with their search range and depending on the model type are given in Table 3.

Table 3. Hyperparameter Search Range of XGBoost

Parameter/Model	XGs	XGh1
max_depth	2,3,4,...,16	2,3,4,5
learning rate (eta)	0.1,0.2,...,0.9,1	0.1,0.2,...,0.9,1
Minimum loss reduction (gamma)	0, 0.001, 0.005, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 1.5,	0, 0.001, 0.005, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 1.5,
Maximum number of boosting iterations (nrounds)	50,100,...,450,500	50,100,...,450,500
min_child_weight	1,2,3	1,2,3

On the other hand, the two parameters of the ANN algorithm that are tuned with time series cross-validation are the number of hidden layers and the number of neurons in each hidden layer. The first parameter is searched for the values of $\{1, 2\}$ and the second parameter, the number of neurons in each hidden layer, is allowed to take values from 1 up to 20 by increments of 1. The first 100 observations of the train data of a period are assigned as the first cross-validation window with 1-day validation samples. The fixed sized windows are rolled by 1-day until the last train data. The average of RMSE values obtained from the validation samples are used to determine the optimal number of hidden layers and the number of neurons in each hidden layer of ANN. As a result, in this research the ANN algorithms are designed with period specific number of hidden layers (1 or 2) and number of neurons in each layer (1 up to 20 neurons).

3.6. Empirical Results

Model specific metrics that are RMSE, MAE, sign symmetry (SS) and the final values obtained from the trading strategy (TSFV) are given in Tables 4 and 5. When the metrics reported in Table 4 are evaluated, while in the first test period (TP1) single ML models performed better, it can be seen that the out-of-sample forecasting performance of the hybrid models outperformed the single models in terms of RMSE and TSFV in the rest of the test periods as well as in the full sample. The single models (GARCH, RFs, XGs and ANNs) are not ranked in the top three algorithms in any of the periods except the first. Moreover, one of the hybrid 2 (H2,2) models, XGh1-ANNs-RF, is ranked first in the full sample by having the smallest RMSE and the highest TSFV (see also Figure 4). In terms of RMSE, even though XGh1-ANNs-RF is not ranked first in every test period, it is able to outperform the benchmark GARCH model's accuracy metrics in three out of four periods as well as in the full sample and ranked in the top three algorithms in all the periods except TP1. Similarly, when the accuracy metrics of NASDAQ reported in Table 5 are evaluated, in terms of RMSE there is not any specific model that is ranked first in most of the test periods. The same hybrid 2 model, XGh1-ANNs-RF is ranked first in the full sample and second in the three out of four test periods (excluding TP3) yielding a similar performance as in BIST100.

Table 4. Test Set Accuracy Metrics of BIST100

Test Period	Metric	GARCH	RFs	XGs	ANNs	RF_{h1}	XG_{h1}	ANN_{h1}	H2,1**	H2,2	H2,3	H2,4
TP1	RMSE	0.01519	0.01510	**0.01475**	0.01534	0.01562	0.01525	0.01512	0.01617	0.01539	0.01507	0.01543
	rank RMSE	5	3	**1**	7	10	6	4	11	8	2	9
	MAE	0.0103	0.0101	**0.0101**	0.0106	0.0112	0.0106	0.0103	0.0110	0.0105	0.0104	0.0106
	SS	0.000	**0.565**	0.546	0.537	0.426	0.463	0.519	0.500	0.454	**0.565**	0.546
	rank TSFV	5	3	**1**	4	9	7	6	10	11	2	8
	TSFV*	103.628	106.829	**110.390**	106.439	95.020	96.819	100.459	93.702	88.032	108.747	96.045
TP2	RMSE	0.01856	0.01944	0.02020	0.01991	0.01862	0.01911	0.01851	**0.01806**	0.01848	0.01935	0.01967
	rank RMSE	4	8	11	10	5	6	3	**1**	2	7	9
	MAE	0.0120	0.0126	0.0129	0.0130	0.0121	0.0122	0.0119	**0.0118**	0.0119	0.0125	0.0131
	SS	0.011	0.402	0.446	0.380	0.554	0.446	0.576	**0.587**	0.533	0.457	0.565
	rank TSFV	6	7	10	11	2	8	3	**1**	5	9	4
	TSFV	103.166	91.441	86.921	86.036	114.741	89.922	109.077	**119.008**	103.298	87.917	104.073
TP3	RMSE	0.01452	0.01490	0.01510	0.01483	0.01455	0.01440	0.01446	0.01450	**0.01418**	0.01462	0.01662
	rank RMSE	5	9	10	8	6	2	3	4	**1**	7	11
	MAE	0.0108	0.0110	0.0111	0.0109	0.0108	0.0107	**0.0107**	0.0107	0.0107	0.0109	0.0115
	SS	0.000	0.490	0.530	0.660	0.580	0.530	**0.670**	**0.670**	0.560	0.510	**0.670**
	rank TSFV	11	10	9	5	8	6	4	3	**1**	7	2
	TSFV	103.663	109.390	118.526	131.362	125.137	130.892	131.869	131.869	**141.665**	128.985	138.448
TP4	RMSE	0.01651	0.01690	0.01668	0.01814	**0.01626**	0.01657	0.01661	0.01654	0.01645	0.01640	0.01755
	rank RMSE	4	9	8	11	**1**	6	7	5	3	2	10
	MAE	**0.0106**	0.0109	0.0106	0.0113	0.0106	0.0107	0.0107	0.0106	0.0108	0.0106	0.0110
	SS	0.270	0.490	0.560	0.530	0.510	0.500	0.530	0.510	0.540	**0.600**	0.530
	rank TSFV	4	10	5	9	3	11	7	6	2	**1**	8
	TSFV	99.396	91.001	97.575	93.349	107.846	90.725	95.986	96.526	116.581	**116.918**	95.986
Full Sample	RMSE	0.01620	0.01660	0.01671	0.01708	0.01627	0.01635	0.01618	0.01633	**0.01614**	0.01637	0.01730
	rank RMSE	3	8	9	10	4	6	2	5	**1**	7	11
	MAE	0.0109	0.0111	0.0111	0.0114	0.0111	0.0110	**0.0108**	0.0110	0.0109	0.0111	0.0115
	SS	0.070	0.490	0.523	0.530	0.515	0.485	0.573	0.565	0.520	0.535	**0.578**
	rank TSFV	9	11	8	7	2	10	5	4	**1**	3	6
	TSFV	110.156	97.242	110.970	112.295	147.136	103.386	138.698	141.941	**150.185**	144.181	132.833

Note: *TSFV is the final value obtained from applying the trading strategy. The best value of each metric is shown in bold. **H2,1, H2,2, H2,3 and H2,4 are the short forms of hybrid 2 models that are RF_{h1}-XGs-ANN, XG_{h1}-ANNs-RF, ANN_{h1}-XGs-RF and XG_{h1}-RFs-ANN, respectively.

Nevertheless, the trading strategy performance of the model is worse compared to its previous performance. Even though, the final trading strategy value of XGh1-ANNs-RF outperformed the benchmark ARMA-GARCH in each test period and is ranked at the top five algorithms out of eleven, another hybrid 2

algorithm, XGh1-RFs-ANN is ranked as the best in the full sample with a final trading strategy value of 183.546. When the full sample is considered, applying the strategy explained in Trading Strategy subsection to the return forecasts of XGh1-RFs-ANN yielded a return of 83.55% (see also Figure 5 for the full sample RMSE and the development of the TSFV of NASDAQ).

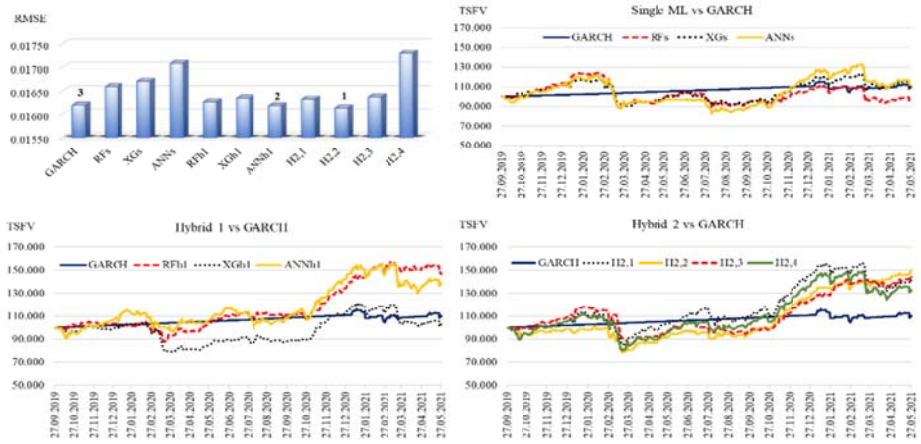

Figure 4. Full Sample RMSE and TSFV of BIST100

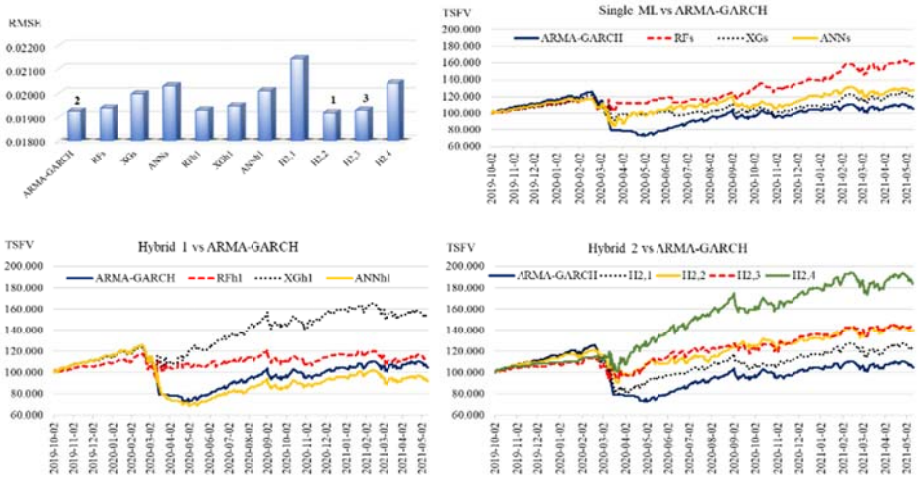

Figure 5. Full Sample RMSE and TSFV of NASDAQ

Moreover, the TSFV of the algorithm is ranked at the top three out of eleven in each test period except one (TP4). According to the accuracy metrics and the results of the trading strategy reported in Table 4 and 5, developed hybrid 2

models suggest quite promising results compared to the forecasts of the other models, especially (ARMA-)GARCH. Additionally, period specific final values of an equally weighted trading strategy applied among the algorithms of the hybrid and the single models are reported in Table 6. This strategy also enables the researchers to compare the period specific average performance of the models. According to Table 6, (ARMA-)GARCH is not ranked first in any of the test periods for both indices. Out of five periods, the equally weighted trading strategy of the developed hybrid 2 (h2) models is ranked first in three (BIST100) and four (NASDAQ) periods. In one test period of each index in which the EWTSh2 is not ranked as the best, the equally weighted trading strategy of the single ML models performed better.

Table 5. Test Set Accuracy Metrics of NASDAQ

Test Period	Metric	ARMA-GARCH	RF_s	XG_s	ANN_s	RF_{h1}	XG_{h1}	ANN_{h1}	H2,1**	H2,2	H2,3	H2,4
TP1	RMSE	0.01556	0.01569	**0.01538**	0.01815	0.01575	0.01567	0.01695	0.01792	0.01550	0.01586	0.01597
	rank RMSE	3	5	**1**	11	6	4	9	10	2	7	8
	MAE	0.0092	0.0093	0.0092	0.0108	0.0093	0.0092	0.0096	0.0098	**0.0091**	0.0095	0.0104
	SS	0.630	0.556	0.546	0.565	0.565	**0.639**	0.630	0.630	0.611	0.528	0.500
	rank TSFV	6	2	5	10	9	3	8	7	4	11	**1**
	TSFV*	101.169	107.182	101.803	100.941	101.057	106.059	101.169	101.169	104.446	94.433	**119.604**
TP2	RMSE	0.02990	0.03031	0.03162	0.03015	**0.02933**	0.02979	0.03112	0.03418	0.02971	0.02975	0.03184
	rank RMSE	5	7	9	6	**1**	4	8	11	2	3	10
	MAE	0.0194	0.0205	0.0217	0.0198	**0.0192**	0.0194	0.0201	0.0232	0.0195	0.0196	0.0206
	SS	0.467	0.424	0.402	0.554	0.511	0.576	0.489	0.500	0.543	0.587	**0.663**
	rank TSFV	10	7	9	6	4	**1**	11	8	5	2	3
	TSFV	92.250	108.770	100.288	109.790	112.933	**131.409**	81.862	103.794	110.254	130.763	129.131
TP3	RMSE	**0.01500**	0.01501	0.01571	0.01634	0.01535	0.01555	0.01504	0.01548	0.01509	0.01505	0.01688
	rank RMSE	**1**	2	9	10	6	8	3	7	5	4	11
	MAE	**0.0115**	0.0115	0.0119	0.0124	0.0117	0.0116	0.0116	0.0120	0.0119	0.0118	0.0120
	SS	0.560	**0.580**	0.560	0.560	0.480	**0.580**	**0.580**	0.540	0.510	0.490	**0.580**
	rank TSFV	7	**1**	10	9	11	5	4	8	2	6	3
	TSFV	108.922	**118.727**	105.345	106.239	102.278	112.319	112.798	108.870	113.665	109.137	113.253
TP4	RMSE	0.01319	**0.01291**	0.01319	0.01389	0.01391	0.01387	0.01378	0.01331	0.01310	0.01322	0.01331
	rank RMSE	4	**1**	3	10	11	9	8	7	2	5	6
	MAE	0.0099	**0.0098**	0.0098	0.0102	0.0107	0.0103	0.0102	0.0102	0.0098	0.0101	0.0101
	SS	0.510	0.500	0.550	**0.560**	0.460	0.500	0.490	0.540	0.540	0.500	0.530
	rank TSFV	8	**1**	2	3	11	10	9	6	4	5	7
	TSFV	103.269	**115.164**	110.946	108.595	97.716	98.405	98.505	105.921	106.838	106.356	104.934
Full Sample	RMSE	0.01926	0.01938	0.01998	0.02032	0.01929	0.01947	0.02011	0.02144	**0.01917**	0.01928	0.02043
	rank RMSE	2	5	7	9	4	6	8	11	**1**	3	10
	MAE	**0.0123**	0.0126	0.0129	0.0131	0.0125	0.0124	0.0127	0.0135	0.0124	0.0125	0.0131
	SS	0.545	0.518	0.518	0.560	0.505	**0.575**	0.550	0.555	0.553	0.525	0.565
	rank TSFV	10	2	8	6	9	3	11	7	5	4	**1**
	TSFV	104.978	159.402	119.326	127.857	114.060	154.042	92.021	121.090	139.841	143.332	**183.546**

Note: *TSFV is the final value obtained from applying the trading strategy. The best value of each metric is shown in bold. **H2,1, H2,2, H2,3 and H2,4 are the short forms of hybrid 2 models that are RF_{h1}-XG_s-ANN, XG_{h1}-ANN_s-RF, ANN_{h1}-XG_s-RF and XG_{h1}-RF_s-ANN, respectively.

4. Conclusions

In this paper, daily returns of BIST100 and NASDAQ indices are forecasted by eleven models consisting single and hybrid machine learning algorithms additional to the traditional ARMA-GARCH for the pre-COVID-19, post-COVID-19 and the full sample test periods. While the single ML models employed various features such as technical indicators and macroeconomic variables to forecast the daily returns of the indices, two different approaches are applied to develop the hybrid models. In the first approach, the residuals of (ARMA-)GARCH are re-modelled by one of the ML algorithms and one-day ahead residual and the final return forecasts are obtained. In the second approach, return forecasts of the single ML algorithms as well as the hybrid models defined

in the first approach are employed as inputs to a different ML algorithm allowing to combine modelling capabilities of the three algorithms. Forecasting performance of the developed models are evaluated for four different test periods and the full sample. Even though, ranks of the models vary among different periods, one of the developed hybrid models, XGh1-ANNs-RF is constantly ranked at the top three algorithms in the three out of four test periods and ranked first in the full sample of both BIST100 and NASDAQ indices in terms of RMSE. The persistently good performance of the model is well found since it is applied on a forecasting task with different index series and test periods including the times of financial turbulence caused by the outbreak of COVID-19.

Table 6. Results of the Equally Weighted Trading Strategy

		TP1		TP2		TP3		TP4		Full Sample	
Index	Model	rank	EWTS*	rank	EWTS	rank	EWTS	rank	EWTS	rank	EWTS
BIST100	GARCH	2	103.63	3	103.17	4	103.66	2	99.40	3	110.16
	$EWTS_s$	1	107.89	4	88.13	3	119.76	4	93.98	4	106.84
	$EWTS_{h1}$	3	97.43	1	104.58	2	129.30	3	98.19	2	129.74
	$EWTS_{h2}$	4	96.63	2	103.57	1	135.24	1	106.50	1	142.28
NASDAQ	ARMA-GARCH	4	101.17	4	92.25	4	108.92	3	103.27	4	104.98
	$EWTS_s$	2	103.31	3	106.28	2	110.10	1	111.57	2	135.53
	$EWTS_{h1}$	3	102.76	2	108.73	3	109.13	4	98.21	3	120.04
	$EWTS_{h2}$	1	104.91	1	118.49	1	111.23	2	106.01	1	146.95

Note: * EWTS is the applied equally weighted trading strategy.

Furthermore, according to the results of the applied first trading strategy, the same model is also ranked as the best in the full sample of BIST100 index and outperformed the benchmark GARCH in all periods except one. When NASDAQ forecasts of XGh1-ANNs-RF applied to the first trading strategy, the final values obtained in each test period outperformed the benchmark ARMA-GARCH in all periods. However, another hybrid model, XGh1-RFs-ANN is ranked as the best in the full sample. On the other hand, according to the second trading strategy, the average performance of the developed hybrid 2 models clearly outperformed the period specific average performance of the single and hybrid 1 models.

REFERENCES

[1] **Bachelier, L. (1964),** ***Theory of Speculation*****.** *P. Cootner* (Ed.), *The Random Character of Stock Market Prices*, 17-78, Cambridge, MA: MIT Press;

[2] **Bishop, C. M. (1995),** ***Neural Networks for Pattern Recognition*****.** *Oxford University Press;*

[3] **Bollerslev, T. (1986),** ***Generalized Autoregressive Conditional Heteroskedasticity***. *Journal of Econometrics*, 31, 307-327;

[4] **Breiman, L. (2001),** ***Random Forests.*** *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324;

[5] **Carbonell, J.G., Michalski, R.S., Mitchell, T.M. (1983),** ***1 - An Overview of Machine Learning***. *R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning*, 3-23. San Francisco: Morgan Kaufmann;

[6] **Campbell, J.Y. (1987),** ***Stock Returns and the Term Structure.*** *Journal of Financial Economics*, 18(2), 373-399;
[7] **Chen, T., Guestrin, C. (2016),** ***XGBoost: A Scalable Tree Boosting System.*** *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794;
[8] **Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y. (2021), xgboost:** ***Extreme Gradient Boosting.*** *R package version 1.3.2.1,* https://CRAN.R-project.org/package=xgboost;
[9] **Engle, R.F. (1982),** ***Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.*** *Econometrica*, 50(4), 987-1007. https://doi.org/10.2307/1912773;
[10] **Fama, E.F. (1970),** ***Efficient Capital Markets: A Review of Theory and Empirical Work.*** *The Journal of Finance*, 25(2), 383–417. https://doi.org/10.2307/2325486 ;
[11] **Fama, E.F., French, K.R. (1988),** ***Permanent and Temporary Components of Stock Prices.*** *Journal of Political Economy*, 96(2), 246-273. http://www.jstor.org/stable/1833108;
[12] **Fritsch, S., Guenther, F., Wright, M.N. (2019),** ***Neuralnet: Training of Neural Networks.*** *R package version 1.44.2,* https://CRAN.R-project.org/package=neuralnet
[13] **Ghalanos, A. (2019).** ***rugarch: Univariate GARCH models.*** *R package version 1.4-1*;
[14] **Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F. (2019),** ***{forecast}: Forecasting Functions for Time Series and Linear Models.*** http://pkg.robjhyndman.com/forecast;
[15] **Hyndman, R. J., Athanasopoulos, G. (2018),** ***Forecasting: Principles and Practice***. Melbourne, Australia:OTexts;
[16] **Hyndman, R. J., Khandakar, Y. (2008),** ***Automatic Time Series Forecasting: the Forecast Package for {R}.*** *Journal of Statistical Software*, 26(3), 1-22. http://www.jstatsoft.org/article/view/v027i03;
[17] **Liaw, A., Wiener, M. (2002),** ***Classification and Regression by RandomForest***. *R News*, 2(3), 18-22. https://CRAN.R-project.org/doc/Rnews/;
[18] **McCulloch, W.S., Pitts, W. (1943),** ***A Logical Calculus of the Ideas Immanent in Nervous Activity.*** *The bulletin of mathematical biophysics*, 5(4), 115-133. https://doi.org/10.1007/BF02478259;
[19] **Pai, P.-F., Lin, C.-S. (2005),** ***A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting***. *Omega*, 33(6), 497-505. https://doi.org/10.1016/j.omega.2004.07.024;
[20] **R Core Team (2019),** ***R: A Language and Environment for Statistical Computing***. *R Foundation for Statistical Computing*. https://www.r-project.org/;
[21] **Rosenblatt, F. (1962),** ***Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms***. Spartan.