Competitive / Collaborative Statistical Learning Framework for Forecasting Intraday Stock Market Prices: A Case Study

Smaranda BELCIUG[1], Adrian SĂNDIȚĂ[1], Hariton COSTIN[2]*, Silviu-Ioan BEJINARIU[2], Pericle Gabriel MATEI[3]

[1] University of Craiova, 13 Alexandru Ioan Cuza St., Craiova, 200585, Romania
sbelciug@inf.ucv.ro

[2] Institute of Computer Science, Romanian Academy, No. 2 Codrescu Street, Iași, 700481, Romania
hcostin@gmail.com (**Corresponding author*)

[3] Ferdinand I Military Technical Academy Bucharest, 39-49 George Coşbuc Blvd., Bucharest 5, 050141, Romania
pericle.matei@mta.ro

Abstract: This paper presents an intelligent decision system based on statistical learning that regards the tactics of an investor in predicting the next intraday stock price. Significant percentages can be won or lost depending on the tactics applied for buying/selling shares. This paper includes a case study regarding the efficiency of a group of machine learning techniques that work together in a competitive/collaborative manner with a view to achieving an overall price forecast for the next intraday transaction. In order to illustrate the advantages of this intelligent decision system this work provides a concrete example concerning the price forecast for the next intraday transaction for Transilvania Bank (TLV), the stock market at the Bucharest Stock Exchange (BVB), Romania. An important part of the decision system lies in the competitive stage, because only the best competitors are chosen for the ultimate decision-making process. In the collaborative stage of the statistical learning framework one uses a weighted voting system that outputs the final intraday stock price. The results obtained show that this intelligent system outperforms each stand-alone method.

Keywords: Artificial intelligence, Machine learning, Prediction methods, Statistical learning, Stock markets.

1. Introduction

A simplified definition of the bourse or stock exchange is: the market where different shares of stock, bonds and other financial instruments are being sold and bought. The stock market facilitates the money flow between investors and stock issuers. The moment when an investor believes that a certain company can potentially develop, she/he buys stock shares in order (a) to be part of that business or (b) to sell the stock shares at a higher price. An unwritten capital market law states: "Buy low, sell high".

An investor must have a strategy when deciding what needs to be bought/sold and also tactics for buying/selling. In what regards the strategy, an investor would buy stock shares if the issuer has good financial results; has clients/outlets for his products/services; gives dividends to shareholders; the executive management has achieved good results in recent years; the sector in which he operates is expanding. Obviously when the issuer is no longer satisfying these requirements the investor would consider selling.

The stock market produces massive amounts of data. This data needs to be processed by machine learning (ML) techniques. In recent years more and more artificial intelligence algorithms have been utilized to predict the stock market. A hybrid stock selection method that included stock prediction using extreme learning machine and stock scoring was applied on the A-share market of China (Yang et al., 2019). Switching regime, ANFIS and GARCH techniques have been employed in order to design a forecasting model for the stock market risk (Kristjanpoller & Mitchell, 2018). A combined fuzzy system and GARCH model was utilized to forecast stock market volatility (Hung, 2011). A hybrid model that uses neural networks (NNs) and fuzzy interference was used as a forecasting system (Badrul et al., 2015). Neural networks were yet again used to predict the stock market index in (Belciug & Săndiță, 2017), and (Moghaddam, Moghaddam & Esfandyari, 2016). A genetic algorithm-based approach to feature discretization in artificial neural network has been used for the forecast of the stock price index (Kim & Han, 2000). In (Pehlivanli, Asikgil & Gulay, 2016) the support vector machines (SVMs) are combined with four filter methods based on different metrics to obtain filtered features for forecasting stock prices for Istanbul Stock Exchange market. A hybrid model that consists of two linear models (autoregressive moving average model, exponential smoothing model) and a recurrent neural network were used for the prediction of stock returns (Rather, Agarwal & Sastry, 2015). An interesting approach was the analysis of twitter feeds that were found to be correlated with the Dow Jones Industrial Average (Bollen, Mao & Zeng, 2011). A hybrid time-series model was used for forecasting leading

https://doi.org/10.24846/v30i2y202104

industry stock prices in (Tsai et al., 2018). Bao, Lu & Zhang (2004) employed support vector machines regression for stock price prognosis. Khare et al. (2017) predicted the short-term stock price using deep learning. Nei & Xue (2016) applied the interval slope method for long-term forecasting of stock price trends.

This study focuses on a case-study regarding an intelligent decision system based on statistical and machine learning for an investors' tactic in predicting the next intraday stock price. Significant percentages can be won or lost depending on the tactics applied for buying/selling shares. The market prices fluctuate according to the political situation around the globe or in a particular country, extreme natural phenomena (tsunamis, earthquakes), and the behavior of other investors (who buy or sell huge amounts of shares).

Apart from the previously used methods, this paper analyses the efficiency of a group of different ML techniques that work together in a competitive/collaborative way in order to obtain a global forecasting price for the next intraday transaction. In order to illustrate the pluses of this intelligent decision system it was applied on a real dataset obtained from Transilvania Bank (TLV) stock market at the Bucharest Stock Exchange (BVB), Romania. An important part of the decision system lies in the competitive stage, because only the first best competitors are chosen for the ultimate decision-making. The collaborative stage of the framework uses a weighted voting system (WVS) that yields the last intraday stock price. The results of this intelligent system are benchmarked by using statistical tools.

The remainder of this paper is organized as follows. Section 2 presents both the benchmarking stock market dataset and the design of the statistical learning framework. Section 3 sets forth the implementation of the intelligent decision system. Section 4 presents the experimental results and the model assessment, and Section 5 includes the discussion. Section 6 outlines the conclusions and tackles possible future directions.

2. Materials and Methods

Bucharest Stock Exchange Dataset

The competitive/collaborative statistical learning framework was applied to real data from the BVB. The data concerns the transactions made by TLV from 01/03/2016 until 30/01/2019. The dataset contains 106306 records with 5 predictive attributes: best buy value, best buy price, best sell value, best sell price, and closing price. The dataset is not publicly available. The dependent variable that needs to be predicted is the next transaction price.

This case-study focuses on presenting a statistical learning framework that may help an investor to choose a tactic for minimizing the risks asSoCiated with selling and/or buying shares. Significant percentages can be won or lost depending on the strategy that has been applied in buying / selling the shares. For a better understanding of the TLV's share evolution a graphic from December 19, 2018 is illustrated in Figure 1.

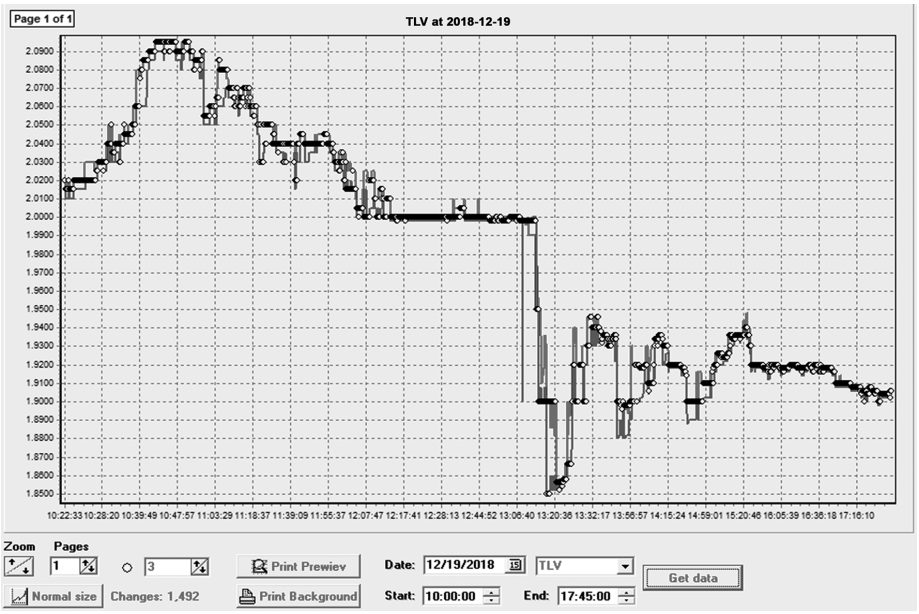

Figure 1. TVL stock prices for December 19, 2018

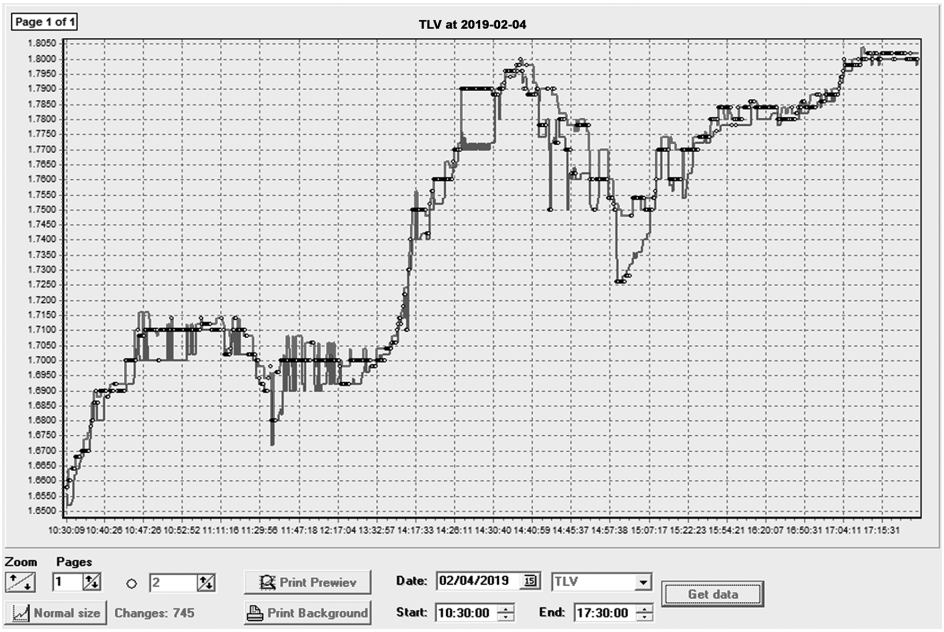

Figure 2. TLV stock prices for February 4, 2019

It can be seen that the price increased by 7 bani (Romanian monetary unit, 100 bani = 1 RON), from 2.02 RON to 2.09 RON on a day that appeared to be normal, after which it suddenly dropped by 24 bani to 1.85 RON.

On the other hand, on February 4, 2019, the price rose by 16 bani, as it is shown is Figure 2.

In both cases, the price variation is linked to rumors about government decisions. On December 19, 2018, the government announced that they were going to tax the bank assets, whereas on February 4, 2019, the government announced that they had not decided yet whether they would tax the bank assets or not.

A potential investor who wants to buy shares has the following level 1 information (as it is illustrated in Figure 3): minimum, maximum, average and best purchase price, best selling price, and the last transaction price.

TLV [REGS] - Closed

Buy VOL	Buy PRC	Sell PRC	Sell VOL	Last	LastVOL	Chg
910,202	1.8000	1.8060	25,000	1.8000	400	11.11

min PRC	Avg PRC	High PRC	No TRD	Vol TRD	Val TRD	Ref PRC
1.6260	1.7480	1.8100	1,490	14,016,553	24,488,163....	1.6200

Figure 3. Level 1 information about stock market shares

Beside this data, a broker has access to level 2 information that refers to the prices and quantities of orders that are placed on the market (as it is depicted in Figure 4, where Rem stands for "Remarks").

2019-02-04 17:59:57.033 Orders by price

Rem	BUY Vol	BUY Prc	SELL Prc	SELL Vol	Rem
[12]	910,202	1.8000	1.8060	25,000	[1]
[1]	71,900	1.7980	1.8100	34,430	[4]
[1]	5,000	1.7900	1.8120	40,450	[3]
[1]	100	1.7880	1.8140	700	[3]
[1]	200	1.7860	1.8160	44,950	[5]
[2]	10,100	1.7840	1.8180	100,200	[2]
[2]	3,028	1.7820	1.8200	6,690	[8]
[8]	18,576	1.7800	1.8220	100	[1]
[1]	200	1.7780	1.8240	40,000	[1]
[4]	20,500	1.7700	1.8280	30,000	[2]
[2]	9,500	1.7620	1.8300	100,270	[9]
[2]	30,230	1.7600	1.8360	4,560	[2]
[1]	200	1.7560	1.8380	500	[1]
[1]	5,000	1.7540	1.8400	47,193	[9]
[4]	6,100	1.7520	1.8460	2,000	[1]
[8]	20,110	1.7500	1.8480	1,300	[2]
[1]	1,800	1.7480	1.8500	124,952	[12]
[2]	85,700	1.7420	1.8560	2,500	[1]
[1]	20,000	1.7400	1.8580	160	[1]
[2]	10,500	1.7300	1.8600	75,894	[12]
[1]	1,000	1.7260	1.8620	87,629	[15]
[1]	180	1.7240			
[1]	10,000	1.7220			
[6]	28,461	1.7200			
[3]	3,750	1.7180			
[1]	3,100	1.7160			
[3]	3,435	1.7140			
[2]	10,200	1.7120			
[2]	6,500	1.7100			
[1]	290	1.7060			
[2]	3,420	1.7040			
[3]	6,190	1.7020			
	3,366,729			769,478	

Figure 4. Level 2 information about stock market shares

Competitive / Collaborative Statistical Learning Framework

This study presents a statistical learning framework for forecasting the stock market price for the next transaction having as starting point a model proposed by Gorunescu et al. (2011). This automatic system works in two steps, as follows:

- the *competitive* stage: a statistical benchmarking process analyzes *n* ML techniques that use regression to predict the stock market price for the next transaction, thereby establishing a hierarchy by taking into account different accuracy measures: the *mean square error* (MSE), the *root mean square error (RMSE)*, the *mean absolute error (MAE),* the *root mean squared log error (RMSLE)* of the residuals obtained during this procedure (Mehdiyev et al., 2016). Other accuracy measures such as precision, recall, F-score, Receiver Operating Characteristic curve (ROC curve), Kappa, Matthews's correlation coefficient can be used only when classification tasks are performed, not for regression. *k* models ($k < n$) are chosen. The value of *k* denotes the best ML models and it is selected by the user;

- the *collaborative* stage: after the *k* ML models have been chosen in the competitive stage, they will work together in an ensemble to predict the value of the stock market price for the next transaction. Generally, the output price of the ML group is computed as the weighted average of each ML algorithm's predicted price. Thus the final prediction carries more weight and is more reliable than the sole prediction of each competitor.

- **Competitive phase**

In the competitive phase, the intelligent informatics system applies all the initial ML algorithms on the stock market dataset. Next, the results are statistically assessed and a hierarchical structure is built depending on their value.

Due to the fact that ML algorithms are stochastic algorithms, in order to obtain reliable results as effectiveness and robustness are concerned, they have to be run for a certain number of times. An a priori statistical power analysis (two-tailed type of null hypothesis) was performed to determine the appropriate sample size, i.e. the appropriate number of computer runs for each ML algorithm in order to achieve adequate statistical power (Altman, 1991). This indicates how many times a test should be performed in order to get a correct interpretation of the obtained results. Hence, in this case, a set of 130 runs for every ML algorithm has been decided yielding a statistical power of 95% (for type I error, $\alpha = 0.005$). The standard 10-fold cross-validation was used (Bishop, 1995).

Data screening involved applying the *Kolmogorov-Smirnov* and *Lilliefors* normality tests. The Kolmogorov-Smirnov one-sample test for normality computes the maximum difference between the hypothesized and the sample cumulative distributions (Belciug, 2020). The assumption "the considered distribution is normal" must be eliminated if the corresponding *D* statistics is meaningful.

The statistical analysis involves a benchmarking process that consists in analogy tests that concern the ML's performance:

- the MSE, RMSE, MAE, and RMSLE;

- graphical representation of the target data versus predictions;

- difference in means (*t*-test for independent samples).

The *t*-test for independent samples is based on Student's *t* distribution. It is the most widely used method for evaluating the differences in means between two independent groups of observations. Based on two independent sets of observations, a point of interest is represented by the average difference between the two sets, while also taking into account the variability between observations. The *t*-test can be applied for independent samples providing that the variables have a Gaussian distribution within each set and the variances of the two sets are significantly distinct.

- **Collaborative stage**

In the second stage the weighted collaborative mode is performed. Every chosen ML algorithm is applied to novel data and the weighted average of their results is computed as the next transaction stock price determined by the weighted voting system (WVS). The WVS computes the final price by the following formula:

- let us consider that the hierarchy of the ML algorithms is $P_1, P_2, ..., P_k$;
- the weights of each ML are established indirectly proportional to their place in the hierarchy, that is: P_1 has $w_1 = k$ votes, P_2 has $w_2 = k-1$ votes, ..., P_k has $w_k = 1$ vote. If two or more ML algorithms occupy the same place in the related hierarchy, the vote weights are also equal:

$$WVS = \frac{P_1 \cdot w_1 + P_2 \cdot w_2 + ... + P_k \cdot w_k}{w_1 + w_2 + ... + w_k} = \\ = \frac{P_1 \cdot k + P_2 \cdot (k-1) + ... + P_k \cdot 1}{k + (k-1) + ... + 1}. \quad (1)$$

Formula (1) is for the computation of the weighted voting system, and many practitioners are using it.

Algorithm 1 explains the statistical learning framework broadly.

Algorithm 1.

Input. Training dataset

Step 1. Run each ML algorithm on the dataset 120 times using 10-cross validation.

Step 2. Compute MSE, RMSE, MAE, RMSLE.

Step 3. Plot graph for a better visualization of target versus predictions.

Step 4. Rank the ML algorithms according to their MSE, or RMSE, MAE, and RMSLE, and assign the vote weights.

Step 5. Apply the normality tests: Kolmogorov-Smirnov & Lilliefors.

Step 6. Apply *t*-test to see if there are significant differences in means between the results obtained by using the ML algorithms. In case there are no significant differences update ranks and weights.

Step 7. Compute final intraday stock market price using the WVS formula above.

Output: intraday stock market price.

3. Calculation

In this study six ML algorithms were used: a multilayered perceptron neural network (MLP), a radial basis function (RBF), support vector machines (SVM), multivariate adaptive regression splines (MARSplines), boosted trees (BT) and random forests (RF).

(a) *Multilayer perceptron* is one of the most widely used neural networks, which consists of 3 or 4 layers: an input layer, one or two hidden layers of neurons and an output layer. The backpropagation algorithm is used as a training method. The output values of the network are confronted with the target values and the global error is calculated by means of an error function *E*. Taking into account the result, the error is propagated backwards into the network in order to adjust the weights so that the error is minimized. The gradient vector of the error function $E = E(w_1, w_2, ..., w_p)$ is computed.

Mathematically speaking, if the input x_i produces the output y_i which is different from the ground truth d_i, then the error function should be minimized, which can be defined as:

$$E = \frac{1}{2}\sum_i |y_i - d_i|^2 . \quad (2)$$

E can be minimized through the iterative process of the gradient descent method:

$$\Delta E = \left(\frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_p} \right). \quad (3)$$

Each weight of the network will be updated afterwards using the increment:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

In this work the three-layer MLP with 5 input neurons, 7 hidden neurons and 1 output neuron was used.

(b) *Radial basis function (RBF)* is a neural network that contains a hidden layer of radial units, which model a Gaussian response surface. The training of a RBF is much faster than that of a MLP. As a drawback, the RBF is more susceptible to the *curse of dimensionality*.

In a RBF the activation of a hidden neuron is computed using the distance between the input vector and a prototype vector. A RBF uses

hyperspheres characterized by radii and centers to divide up the space. Mathematically speaking, *M* basis functions are considered that map the network as follows:

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} w_{kj}\,\phi_j(\mathbf{x}), \tag{4}$$

where ϕ is the classical Gaussian basis set, expressed by:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mu_j\|^2}{2\sigma_j^2}\right), \tag{5}$$

where **x** is the input vector, μ_j is the vector that determines the center of the basis function ϕ_j, and σ_j is the width parameter.

The training of a RBF is performed in a two-stage procedure. In the first stage the training dataset is employed in order to determine the parameters of the basis function μ_j and σ_j, while in the second stage the weights w_{kj} are determined. The sum of squared errors needs to be minimized in the training process (Gorunescu, 2011).

In this study one used a RBF with 5 input neurons, 7 hidden neurons and 1 output neuron.

(c) *Support Vector Machine - Regression (SVR)* is a ML algorithm that apart from being used in pattern recognition is also utilized for non-linear regression. Given a regression problem represented by the equation

$$d = f(\mathbf{x}) + v, \tag{6}$$

where *d* is the scalar dependent variable, **x** is a vector that contains the predictors, f is a scalar-valued non-linear function, and v is the 'noise', the dependence of *d* upon **x** should be evaluated (that is f, also the distribution of v).

In order to do so, let's denote by *y* the estimate of *d* and $g(\mathbf{x}) = \{g_j(\mathbf{x}), j = 0,1,2,...,m\}$ a set of non-linear basis functions. The expansion of *y* shall be considered in terms of $g(\mathbf{x})$ as:

$$y = \sum_{j=0}^{m} \mathbf{w}_j \cdot g_j(\mathbf{x}) = \mathbf{w} \cdot g^T(\mathbf{x}), \tag{7}$$

where **w** is the weight vector.

The function of a SVR that estimates *d* is an ε-insensitive loss function, given by the following formula:

$$L_\varepsilon(d,y) = \begin{cases} 0, & |d-y| < \varepsilon \\ |d-y|, & |d-y| \geq \varepsilon \end{cases}. \tag{8}$$

Hence, one should minimize the empirical risk, R:

$$R = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(d_i, y_i), \tag{9}$$

with the constraint: $\|\mathbf{w}\| \leq c$, where c is a constant. For more details see (Drucker et al., 1997), and (Vapnik, Golowich & Smola, 1996). In this study 25 support vectors were used.

(d) *Multivariate Adaptive Regression Splines* (*MARSplines)* is a nonparametric method that makes no supposition about the underlying functional relationship that exists between the predictive variables and the dependent one. This relationship is built by the MARSplines with a set of coefficients and basis functions that are determined by the data involved. The MARSplines use a *divide-et-impera* method that divides the input space into zones, each of them with its own regression. MARSplines is not sensitive to the *curse of dimensionality.*

MARSplines build models using the following equation:

$$f(x) = \sum_{i=1}^{K} c_i B_i(x), \tag{10}$$

where c_i is a constant coefficient, and $B_i(x)$ are the basis functions.

The basis functions can be: (*i*) a constant, the intercept, (*ii*) a *hinge* function of the form: $max(0, x - const)$ or $max(0, const - x)$, or (*iii*) a product of hinge functions.

(e) *Boosted Trees (BT - Stochastic Gradient Boosting)* compute a sequence of simple decision trees, where each tree is built on the prediction residuals of the preceding tree of an independently drawn random sample. The randomness is used in order to prevent overfitting. Using a boosted method, the BT combines trees in an iterative fashion. In BT, the trees are not grown to completion, the size of a tree being specified by

a model parameter. The objective of the BT is to minimize the MSE, RMSE, MAE, and RMSLE. In this study 200 trees have been used.

(f) *Random Forests (RF)* are a "combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all the trees in the forest" (Breiman, 2001). In RF, a bootstrap sample of the training set is chosen. At each step a small random sample of the predictor variables is selected and the best split is made taking into account that particular set of variables. The process continues in this manner until the tree reaches the largest possible size and is left unpruned. The procedure is repeated for a certain number of times. The final response is an average of the predictors of all the trees in a collection. When used for regression, the tree response is an estimate of the dependent variable given the predictors. Each tree produces a numerical response value. In this study 100 trees were used.

The algorithms and benchmarking process have been performed using STATISTICA 12 (StatSoft Inc.) For more details concerning the above-mentioned ML techniques, see the papers (Gorunescu, 2011) for the competitive / collaborative systems; (Bishop, 1995) for neural networks fundamentals; (Breiman, 2001) for random forests ML algorithm; (Chu, 2019) for spline models; (De'ath, 2007) for boosted trees; (Denisko & Hoffman, 2018) for random forests; (Elith, Leathwick & Hastie, 2008) for regression trees; (Grant, Eltoukhy & Asfour, 2014) for neural networks; (Natekin & Knoll, 2013) for gradient boosting machines; (Przybylek, Jelinski & Cysewski, 2019) for committee decision; (Stoean & Stoean, 2014) for SVM; (Jammalamadaka, Qiu & Ning, 2019) for times series approach; (Chinthalapati, Mitra & Serguieva, 2019) for big data and noise.

4. Results

Experimental Results

Each ML competitor from the statistical learning framework has been applied on the TLV dataset. In Table 1 the MSE, RMSE, MAE, and RMSLE of the competitors are presented.

Table 1. Accuracy measures

ML type	MSE	RMSE	MAE	RMSLE
MARSplines	0.000019	0.0043	0.0041	0.0001
MLP	0.00002	0.0044	0.0042	0.0001
SVR	0.00006	0.0077	0.0075	0.0001
RF	0.00023	0.015	0.014	0.0003
BT	0.00136	0.036	0.034	0.0008
RBF	369276.4	192.16	191.78	4.79

It can be seen that irrespective of the accuracy measure used, the hierarchy remains the same (Table 2). It was decided to use MSE because it is the most widely used accuracy measure for regression. The hierarchy according to the MSE obtained by each ML algorithm is displayed in Table 2. Table 2 also features the weight of each ML algorithm according to its ranking. One should keep in mind that this is the initial weight that might be changed during the statistical benchmarking process.

Table 2. Statistical learning framework hierarchy and weight setting

ML type	MSE	Hierarchy ranking	Weight
MARSplines	0.000019	1	5
MLP	0.00002	2	4
SVR	0.00006	3	3
RF	0.00023	4	2
BT	0.00136	5	1
RBF	369276.4	6	0

In Table 2 it can be seen that the weight of the RBF was set at 0, due to the fact that its MSE value is too high in comparison with the others.

In correlation with Table 2, a visual comparison of the forecasting performance of each of the six ML techniques is presented in Figures 5, 6, 7, 8, 9 and 10.

The graphical representation of the performances of the ML algorithms was used, because it is recommended for comparing ML algorithms since it gives insight into the *capabilities of each one of them.*

The forecasting performances of the MLP, SVR, MARSplines, BT and RF will be further considered. Since the RBF performed badly in this

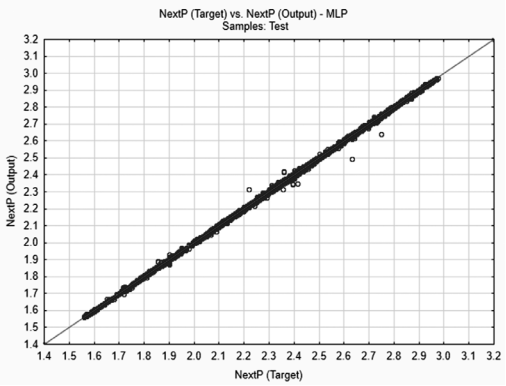

Figure 5. Target vs. Prediction - MLP

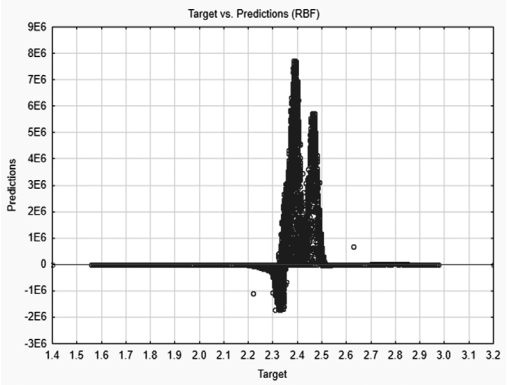

Figure 6. Target vs. Prediction - RBF

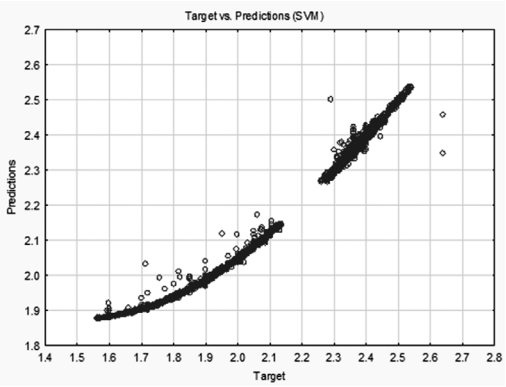

Figure 7. Target vs. Prediction - SVM

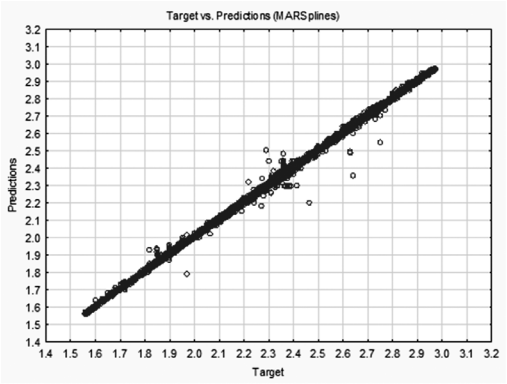

Figure 8. Target vs. Prediction - MARSpline

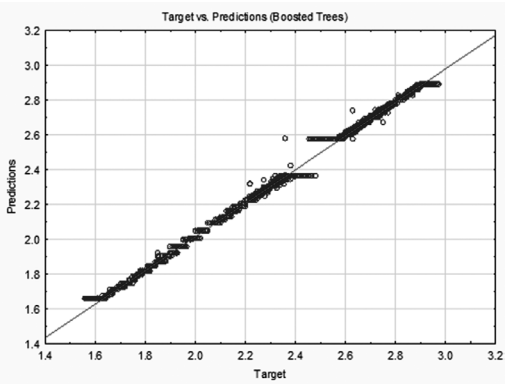

Figure 9. Target vs. Prediction - Boosted Trees

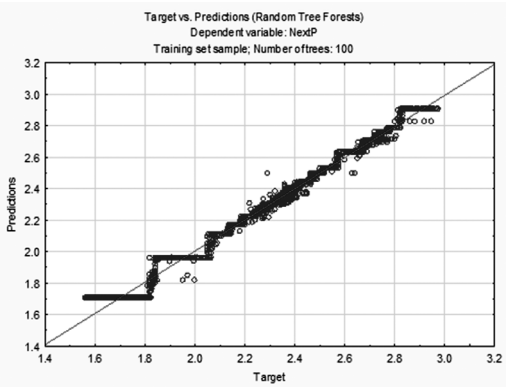

Figure 10. Target vs. Prediction - RF

special case study, its corresponding performance will not be analyzed any further.

Remarks. All the presented algorithms are of stochastic nature and they represent metaheuristic techniques. Therefore, their architectures are chosen accordingly and their parameters are not computed. They are heuristically chosen, based on their performance.

The results above only reflect the performances of the above-mentioned MLs for this special database. It is noteworthy that the performance of each ML strongly depends on the dataset used, consequently the ML hierarchy depends on the problem to be solved, and therefore the statistical learning framework will have to be adjusted every time to the actual case it will be used for.

Statistical Assessment

In order to perform other comparison tests, it is necessary to verify if the sample data has a normal distribution. Hence, in this work the *Kolmogorov-Smirnov* and *Lilliefors* tests were applied. The outcome is depicted in Table 3.

Table 3. Testing the normality of the ML algorithms forecasting performances

Algorithm	Kolmogorov-Smirnov	
	K-S max D	Lilliefors *p*
MLP	0.469	< 0.01
SVR	0.467	< 0.01
MARSplines	0.479	< 0.01
BT	0.340	< 0.01
RF	0.365	< 0.01

In Table 3 it can be seen that regardless of the ML algorithm used, the data is not normally distributed, $p < 0.01$. However, one can presume that the distribution is nearly Gaussian, irrespective of the distribution of the predictions, due to the fact that the sample size is over 100.

If the assumption of the t-test for independent samples was accomplished, one can use it to assess the difference in means between the five ML techniques. The results of the t-test are depicted in Table 4. It can be seen that there are highly significant differences in means (p-level < 0.001) between the five ML algorithms, although the MSE values are close enough, except for the MLP and the MARSplines. Consequently, the weights will be updated for all the methods, thereby assigning equal weights and ranks to the MLP and MARSplines, as it can be seen in Table 5.

Table 4. Compared testing performances (t-test) for the ML algorithms

Variable	*t*-test/*p*-level
MLP vs. SVR	-7.32 / 0.000
MLP vs. MARSplines	-0.89 / 0.36
MLP vs. BT	-60.03 / 0.000
MLP vs. RF	-38.36 / 0.000
BT vs. RF	49.14 / 0.000
SVR vs. MARSplines	6.85 / 0.000

Table 5. Weight assigning in competitive mode

ML type	Hierarchy ranking	Weights
MARSplines	1	4
MLP	1	4
SVR	2	3
RF	3	2
BT	4	1

5. Discussion

In the previous section one could see that the competitive stage has generated the following hierarchy depending on the forecasting prognosis: MARSplines, MLP, SVR, RF and BT. In a more selective competition only the first two or three competitors might have been chosen. Still, in the following analysis all five ML techniques will be considered.

By implementing the second stage of the statistical learning framework, an overall automatic forecast of the next transaction price can be performed. A concrete example is presented in Table 6.

Table 6. Concrete example of the WVS

Model	Target	Predicted value	Weights	Residuals
MLP	2.38	2.38067	4	0.0006
MARSplines		2.3825	4	-0.0025
SVR		2.379	3	-0.001
RF		2.3811	2	-0.0011
BT		2.3633	1	0.0167
WVS = (2.38067x4+2.379x3+2.3825x4+ 2.3633x1+2.3811x2)/(4+3+4+1+2) = 2.3796				-0.0003

In Table 6 one can see how the statistical learning framework performs. The next transaction real price (target) is 2.38.

MLP obtained 2.38067 with a residual of 0.0006, MARSplines obtained 2.3825 with a residual of -0.0025, SVR obtained 2.379 with a residual of -0.001, RF obtained 2.3811 with a residual of -0.0011 and BT obtained 2.3633 with a residual of 0.0167. After applying the WVS it can be seen that the final predicted price for the next transaction is 2.3796 with a residual of -0.0003. The residual obtained by the WVS has a smaller value than all the other residuals obtained by each ML in the competitive stage, which confirms the effectiveness of the statistical learning framework in comparison to separate stand-alone models.

The competitive/collaborative system was applied on the whole dataset. The results in terms of MSE are illustrated in Table 7.

Table 7. Competitive/collaborative intelligent system MSE

Competitive/collaborative system	MSE
WVS	0.000009

Besides, a *t*-test analysis was performed for the MSE, MARSplines and MLP methods to see whether the improvement achieved is indeed statistically significant (Table 8).

Table 8. Compared testing performances (*t*-test)

Variable	t-test/p-level
WVS vs. MARSplines	-5.48 / 0.000
WVS vs. MLP	-3.84 / 0.000

In Table 8 it can be seen that there is a statistically significant improvement in terms of MSE, when comparing the competitive/collaborative system with the best two stand-alone competitors (p-level < 0.05).

It can be stated that the aim of this paper was to explore the approach of a competitive/collaborative committee of machine learning techniques. A strong shared belief is that the five above-mentioned algorithms and their combination are enough to prove the viability of this approach.

Since all the algorithms are stochastic, not deterministic, one cannot reproduce the results in an exact manner, even if one had the source code or dataset. That is why data scientists use statistical analysis for validation. 120 computer runs were performed and the 10-fold cross-validation was applied, in order to achieve 95% statistical power.

6. Conclusion

This study investigated the effectiveness of a statistical learning framework based on a competitive / collaborative approach for the task of achieving a confident real-time forecast of the next price transaction for a share on the stock market.

To identify which ML model is best fitted for this task, a profound statistical benchmarking procedure together with a WVS were utilized.

The strong point of this model lies in the fact it does not rely on a single ML technique for predicting the price for the next intraday transaction. The forecast is made by a dynamic adaptable system, which merges the predictions of each standalone algorithm while taking into account its overall performance. The framework's flexibility is due to:

- the trustworthy statistical method of choosing the best competitors for each specific case (database), based on comparison tests;
- the WVS, which enhances the system performance (due to the fact that the performance determines the weights) and thus correcting any possible individual error that may appear.

The experimental results have shown that the statistical learning framework obtained a lower value for the residual, than that of each standalone model, thereby featuring a robust behavior with respect to automatic forecasting reliability.

Possible future works may explore ways of extending the statistical learning framework through the integration of other ML techniques, preferably time-dependent ones, and may explore other WVS models.

REFERENCES

Altman, D. G. (1991). *Practical Statistics for Medical Research*. New York: Chapman and Hall.

Badrul, A. M., Zakir, H., Amjad, H. & Muzahidul, I. (2015). Price Prediction of Stock Market using Hybrid Model of Artificial Intelligence, *International Journal of Computer Applications*, *111*(3), 5-9. DOI: 10.5120/19516-1136

Bao, Y., Lu, Y. & Zhang, J. (2004). Forecasting Stock Price by SVMs Regression. In: Bussler, C. & Fensel, D. (eds), *Artificial Intelligence: Methodology, Systems, and Applications. AIMSA 2004. Lecture Notes in Computer Science*, *3192*, 295-303. DOI: 10.1007/978-3-540-30106-6_30

Belciug, S. (2020). *Artificial Intelligence in Cancer: diagnostic to tailored treatment*. Elsevier.

Belciug, S. & Săndiță, A. (2017). Business Intelligence: Statistics in Predicting Stock Market, *Annals of the University of Craiova, Mathematics and Computer Science, Science Series*, *44*(2), 292-298.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Bollen, J., Mao, H. & Zeng, H. X. (2011). Twitter Mood Predicts the Stock Market, *Journal of Computational Science*, *2*(1), 1-8. DOI: 10.1016/j.jocs.2010.12.007

Breiman, L. (2001). Random Forests, *Machine Learning, 45*(1), 5-32. DOI: 10.1023/A:1010933404324

Chinthalapati, V. L. R., Mitra, S. & Serguieva, A. (2019). Big data and probably approximately correct learning in the presence of noise: implications for financial risk management, *International Journal of Artificial Intelligence*, 17(1), 34-56.

Chu, L., Wang, L. J., Jiang, J., Liu, X., Sawada, K. & Zhang, J. (2019). Comparison of Landslide Susceptibility Maps Using Random Forest and Multivariate Adaptive Regression Spline Models in Combination with Catchment Map Units, *Geosciences Journal*, *23*(2), 341-355. DOI: 10.1007/s12303-018-0038-8

De'ath, G. (2007). Boosted trees for ecological modeling and prediction, *Ecology*, *88*(1), 243-251. DOI: 10.1890/0012-9658(2007)88[243:btfema]2.0.co;2

Denisko, D. & Hoffman, M. (2018). Classification and Interaction in Random Forests. In *Proceedings of the National Academy of Sciences of the United States of America*, *115*(8), (pp. 1690-1692). DOI: 10.1073/pnas.1800256115

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. & Vapnik, V. (1997). Support Vector Regression Machines. In: Mozer, M. C, Jordan, M. I. &. Petsche, T. (eds), *Advances in Neural Information Processing Systems*, *9*, 155-161. MIT Press, Cambridge.

Elith, J., Leathwick, J. R. & Hastie, T. J. (2008). A Working Guide to Boosted Regression Trees, *Journal of Animal Ecology*, *77*, 802-813. DOI:10.1111/j.1365-2656.2008.01390.x

Gorunescu, F., Gorunescu, M., Saftoiu, A., Vilmann, P. & Belciug, S. (2011). Competitive / Collaborative Neural Computing System for Medical Diagnosis in Pancreatic Cancer Detection, *Expert Systems*, *28*(1), 33-48. DOI: 10.1111/j.1468-0394.2010.00540.x

Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques*, *12*. Intelligent Systems Reference Library, Springer-Verlag, Berlin Heildelberg. DOI: 10.1007/978-3-642-19721-5

Grant, J., Eltoukhy, M. & Asfour, S. (2014). Short-Term Electrical Peak Demand Forecasting in a Large Government Building Using Artificial Neural Networks, *Energies*, *7*(4), 1935-1953. DOI: 10.3390/en7041935

Hung, J.-C. (2011). Applying a Combined Fuzzy Systems and GARCH Model to Adaptively Forecast Stock Market Volatility, *Applied Soft Computing*, *11*(5), 3938-3945. DOI: 10.1016/j.aSoC.2011.02.020

Jammalamadaka, S. R., Qiu, J. & Ning, N. (2019). Predicting a Stock Portfolio with the Multivariate Bayesian Structural Time Series Model: Do News or Emotions Matter?, *International Journal of Artificial Intelligence*, *17*(2), 81-104.

Khare, K., Darekar, O., Gupta, P. & Attar, V. (2017). Short term stock price prediction using deep learning. In *2nd* IEEE *International Conference on Recent Trends in Electronics, Information & Communication Technology* (RTEIC), (pp. 482-486). DOI: 10.1109/RTEICT.2017.8256643

Kim, K. & Han, I. (2000). Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index, *Expert Systems with Applications*, *19*(2), 125-132. DOI: 10.1016/S0957-4174(00)00027-0

Kristjanpoller, R. W. & Mitchell, V. K. (2018). A Stock Market Risk Forecasting Model Through Integration of Switching Regime, ANFIS and GARCH Techniques, *Applied Soft Computing*, *67*, 106-116. DOI: 10.1016/j.aSoC.2018.02.055

Mehdiyev, N., Enke, D., Fettke, P. & Loos, P. (2016). Evaluating Forecasting Methods by Considering Different Accuracy Measures, *Procedia Computer Science*, *95*, 264-271.

Moghaddam, A. H., Moghaddam, M. H. & Esfandyari, M. (2016). Stock Market Index Prediction using Artificial Neural Network, *Journal of Economics, Finance and Administrative Science, 21*(41), 89-93. DOI: 10.1016/j.jefas.2016.07.002

Natekin, A. & Knoll, A. (2013). Gradient Boosting Machines - a Tutorial, *Frontiers in Neurorobotics*, 7-21. DOI: 10.3389/fnbot.2013.00021

Nei, C. & Xue, J. (2016). The Interval Slope Methods for Long-term Forecasting of Stock Price Trends, *Advances in Mathematical Physics, 6,* 1-7. DOI: 10.1155/2016/8045656

Pehlivanli, A. C., Asikgil, B. & Gulay, G. (2016). Indicator Selection with Committee Decision of Filter Methods for Stock Market Price Trend in ISE, *Applied Soft Computing*, *49*, 792-800. DOI: 10.1016/j.aSoC.2016.09.004

Przybylek, M., Jelinski, T. & Cysewski, P. (2019). Application of Multivariate Adaptive Regression Splines (MARSplines) for Predicting Hanse Solubility Parameters based on 1D and 2D Molecular Descriptors Computed from SMILES String, *Journal of Chemistry,* 1-15. DOI: 10.1155/2019/9858371

Rather, A. M., Agarwal, A. & Sastry, V. N. (2015). Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns, *Expert Systems with Applications*, *42*(6), 3234-3241. DOI: 10.1016/j.eswa.2014.12.003

Stoean, C. & Stoean, R. (2014). *Support Vector Machines and Evolutionary Algorithms for Classification. Single or Together?*, *Intelligent Systems Reference Library, 69*. Springer International Publishing. DOI: 10.1007/978-3-319-06941-8

Tsai, M. C., Cheng, C. H., Tsai, M. I. & Shiu, H. Y. (2018). Forecasting Leading Industry Stock Prices Based on a Hybrid Time-Series Forecast Model, *Plos One,* 1-24. DOI: 10.1371/journal.pone.0209922

Vapnik, V. N., Golowich, S. E. & Smola A. J. (1996). Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In *Proceedings of Advances in Neural Information Processing Systems 9* (NIPS 1996), Denver, USA (pp. 281-287).

Yang, F., Chen, Z., Li, J. & Tang, L. (2019). A Novel Hybrid Stock Selection Method with Stock Prediction, *Applied Soft Computing*, *80*, 820-831. DOI: 10.1016/j.aSoC.2019.03.028