Learning Machines For Processing Stock Market Data

Milos Stojanovic[1], Ivana Markovic[2], Jelena Z. Stankovic[3] and Stevica Cvetkovic[4]
[1]College of Applied Technical Sciences Nis
Aleksandra Medvedeva 20, Ng 18000
Serbia
milos.stojanovic@vtsnis.edu.rs
[2] Faculty of Economics at the University of Nis
Trg Kralja Aleksandra Ujedinitelja 11
Ng, Serbia
ivana.markovic@eknfak.ni.ac.rs
[3] Faculty of Economics at the University of Nis
Trg Kralja Aleksandra Ujedinitelja 11, 18000 Nis
Serbia
ielenasAeknfak.ni.ac.rs
[4] Faculty of Electronic Engineering
Aleksandra Medvedeva 14, 18000 Nis
Serbia
stevica.cvetkovic@elfak.ni.ac.rs

ABSTRACT *: For the stock market analysis we have used extreme learning machines. We have deployed the feature extraction with technical parameters. The Extreme Learning Machines that use the single layer neural networks is used for classification. For comparative evaluation we have used the stock market index and further applied the model training and testing.*

Keywords: Stock Market Trend Prediction, Technical Indicators, Extreme Learning Machines

Received: 4 November 2021, Revised 2 March 2022, Accepted 11 March 2022

DOI: 10.6025/jcl/2022/13/2/44-51

1. Introduction

Prediction of stock market movements is a challenging task, taking into consideration the fact that the financial market is a complex, evolving and dynamic system whose behavior is pronouncedly non-linear [1].

Although some researchers have suggested that there is evidence that stock prices are not purely random, the general consensus still is that their behavior is approximately close to the random walk process. Therefore, degrees of accuracy of an approximate 60% hit rate in predictions are often considered satisfactory results in this area [2].

Predicting the direction of movement of the price of financial instruments is a current area in academic research where technical analyses in combination with machine learning have proven to be highly effective [3].

The choice of technical indicators substantially affects the overall performance of the classification system. Assuming that investors trade continuously and those past prices provide sufficient information, the most commonly used technical indicators for securities' price trend prediction are trending, volume and oscillating indicators. Trending indicators identify and monitor the securities trends, while the volume indicators are based on the change in the volume of trading in securities and complete the information which is offered by the trending indicators in forming trading strategies. Oscillating indicators are leading indicators which generate early warning signals of changes in the securities trend and determine the strength of the current trend, as well as the moment when a change in the trend occurs.

Since the prediction of the movement of stock market indices plays an important role in the development of effective market trading strategies, it is important to point out that every increase in precision is considered an exceptional contribution since it leads to an increase in the return and the decrease in the risk involved in trading. However, due to illiquidity, any selected indicators, especially on emerging markets such as Belgrade stock exchange, may provide trading signals that cannot be used to form profitable trading strategies. Market liquidity is an important factor for portfolio managers and large institutional investors and it refers to the ability to execute a trade promptly, at low cost or no cost, risk or inconvenience [4]. Therefore, in this study liquidity risk is in particular considered in the calculation of technical indicators.

The second crucial part of the system is for machine learning technique to be applied for stock market trend prediction. In [5] it was indicated that the Least Squares Support Vector Machines (LS-SVMs), and SVMs - Support Vector Machines outperform other machine learning methods, since in theory they do not require any previous a priori assumptions regarding data properties. In this study we investigate the application of Extreme Learning Machines (ELM) [6, 7] for stock market trend prediction, as an alternative to the commonly used SVM. ELM is a single hidden layer feed-forward neural network (SLFN), which overcomes an important drawback of traditional artificial neural networks (ANNs) - their slow learning speed. It increases training speed by randomly assigning weights and biases in the hidden layer, instead of iteratively adjusting its parameters by gradient-based methods. As well as minimizing training error, ELM finds smallest norm of output weights and hence has a better generalization performance than gradient based training algorithms, such as backpropagation.

In the rest of the paper we first describe the technical indicators selection that will be used as features for the prediction model. The problem of predicting the direction of the stock index movements is then modelled as a problem of a binary classification, after that we give an overview of ELM for classification. Finally, the experimental evaluation and conclusion are presented.

2. Technical Indicator Analysis and Feature Construction

Feature construction is an essential step for defining an accurate prediction model. The arbitrary application of a large number of explanatory features to ELM or any other machine learning based algorithm could lead to low prediction accuracy. A proper feature selection procedure on the other hand would lead to higher method accuracy. This process is of great importance, but there is no general rule that can be followed.

This study represents a continuation of our prior work [8, 9] with new results and a more in-depth analysis.

Firstly, in order to adjust the technical analysis to the specifics of trading in the observed emerging market, technical indicators are calculated using liquidity risk adjusted returns:

$LAr_t = c_t - r_t,$ r_t - logarithmic return, c_t - illiquidity cost.

The usual approach for liquidity risk modeling in emerging and frontier markets uses proxies. Since detailed transaction data on bid-ask spreads are not available, we employ Amihud's illiquidity measure [10] that can be calculated using daily data on price and trading volumes:

$c_t = \frac{|r_t|}{V_t}$, V_t - Trading volume in ten millions of monetary V_t units.

Considering the fact that returns are negatively correlated with illiquidity [10], illiquidity cost decreases return (Figure 1). Involving liquidity risk that is significantly high in emerging markets [11] should provide higher accuracy of technical indicators reflected in the decrease in the number of trades according to the realistic conditions on the market.

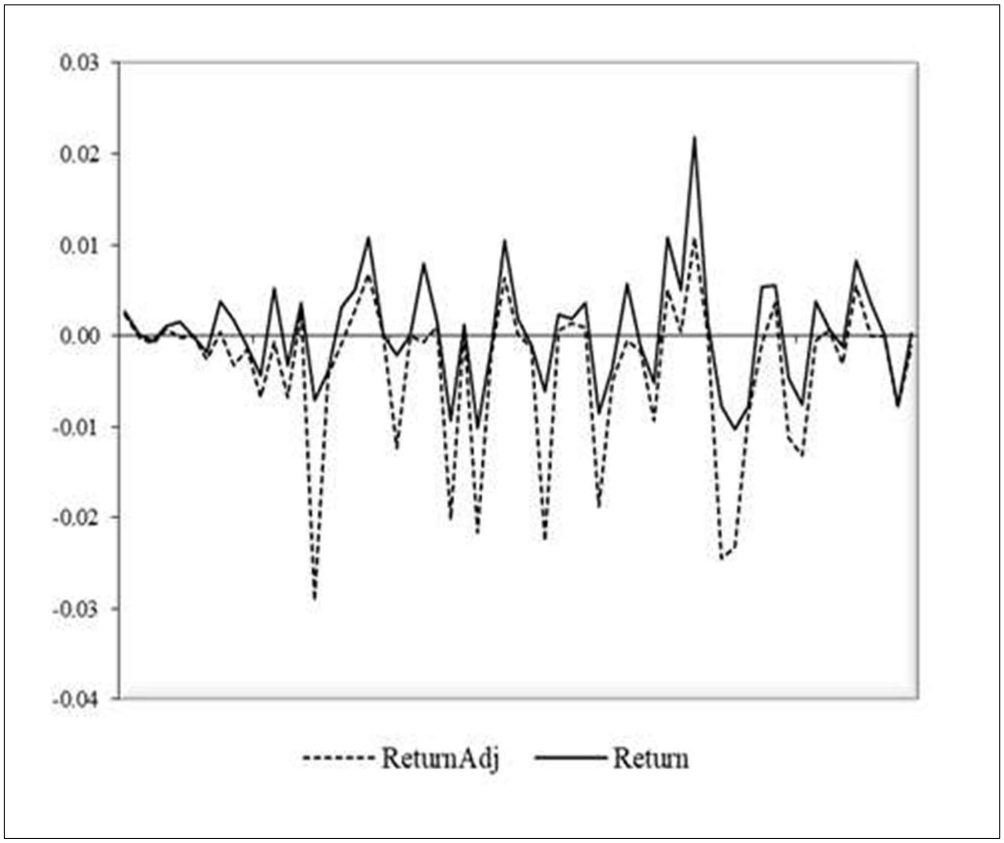

Figure 1. The relation between the logarithmic return (Return) and liquidity risk adjusted return (ReturnAdj) on the BELEX15 index

In our prior studies different technical indicators were analyzed based on their common use in the literature [1-3]. Briefly, it can be summarized in the following way. Considering that the response variable predicts the stock market trend (either an increase or decrease), the explanatory features need to measure changes as well. In effect, observing feature changes over time is more significant for prediction than the absolute value of each feature. In this study a more extended analysis is conducted regarding modeling stock market tendencies with comprehensive indicators which are able to more accurately model dynamic changes on the stock market.

In the group of technical indicators of oscillations, which are used to discover short-term overbought or oversold conditions, in this study we introduce the TRIX indicator. In order to remove the noise and reveal the market trend for a specific time interval we have facilitated a smoothing process using the TRIX indicator.

The Triple Exponential Moving Average Oscillator (TRIX) is a technical indicator that shows the slope of the security s return trend line. A rising trend line indicates an uptrend and, in that case the TRIX takes positive value. On the other hand, if the TRIX indicator has negative value, it can be concluded that there is downtrend in the concrete security s return, while crossing the signal line indicates a trend-change. Table 1 shows a selected list of the technical indicators.

Indicators	Formula
Closing price	CP_t, t= 1,2, ... N
Lowest price	LP_N – Lowest price in the past N days
Highest price	HP_N – Highest price in the past N days
Logarithmic return	$r_t = \log CP_t - \log CP_{t-1}$
Liquidity risk adjusted return	$LAr_t = r_t - c_t$
Trend indicators	
Exponential Moving Average	$EMA_N = LAr_t * k + EMA_{t-1} * (1-k)$; $K = 2/(N+1)$
Oscillating indicators	
Relative Strength Index	RSI = $100-(100/(1+\frac{\frac{1}{T}\sum_{t=o}^{T} LAr_t^+}{\frac{1}{T}\sum_{t=o}^{T} LAr_t^-}))$
Parabolic "Stop and Reverse" Indicator	$SAR_t = SAR_{t-1} + AF(ELAr_{t-1} - SAR_{t-1})$ AF – acceleration factor (from 0.02 to 0.2), $ELAr_{t-1}$ – extreme return in the previous period
Moving Average Convergence Divergence	$MACD_t = EMA_{12} - EMA_{26}$ Signal Line = Simple 9-day moving average of MACD
Stochastic Oscillator	%K = $100*((LAr_{close}-LAr_{14})/(LAr_{14,high} - LAr_{14,low}))$ %D = EMA_3(%K) Slow %D = EMA_3(%D)
Triple Exponential Moving Average Oscillator	$TRIX = (EMA3_t - EMA3_{t-1}) / EMA3_{t-1}$
Commodity Channel Index	$CCI = (TLAr_t - MA_{20})/(0.015 * MD)$ $TLAr_t=(LAr_{high}+LAr_{low}+LAr_{close})/3$ $MD = \sum_{t=1}^{N}(TLAr_t - MA_{20})/N$
Williams' R Indicator	%R = $-(LAr_{high} - LAr_{close})/(LAr_{high} - LAr_{low})*100$

Table 1. The List of Technical Indicators

For all technical indicators, liquidity risk is considered in the calculation of their values.

Finally, the trend is modeled as a categorical variable used to indicate the movement direction of the BELEX15 index over time *t*. If the liquidity risk adjusted return over time *t* is larger than zero, the indicator is 1. Otherwise, the indicator is -1.

3. Extreme Learning Machines (Elm)

Let us define *N* training examples as (x_j, y_j) where $\mathbf{x}_j = [x_{j1}, x_{j2}, \ldots, x_{jn}]^T \in \mathbf{R}^n$ denotes the j-th training instance of dimension n and $\mathbf{y}_j = [y_{j1}, y_{j2}, \ldots, y_{jm}]^T \in \mathbf{R}^m$ represents the j-th training label of dimension m, where m is the number of classes. The set of features, that is previously explained technical indicators, will further be denoted as x_j, while y_j will denote m dimensional vector of binary class labels with value denoting membership to the class. SLFN with an activation function $g(x)$ and L hidden neurons could be defined as:

$$\sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{f}_j, j = 1, \ldots, N \tag{1}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ denotes the vector of weights which connects the i^th hidden neuron $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector which connects the i^th hidden neuron and all output neurons, and b_i is the bias of the ith hidden neuron. In ELM theory [8], w_i and b_i can be assigned in advance randomly and independently, without a priori knowledge of the input data. The ELM network structure is presented in Figure 2.

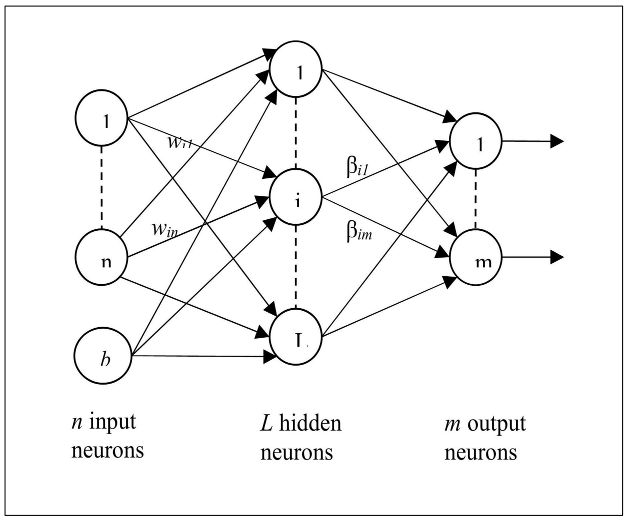

Figure 2. Structure of the ELM network

SLFN in (1) should satisfy $\sum_{i=1}^{L} \|\mathbf{f}_i - \mathbf{y}_i\| = 0$, i.e., there β_i, w_i and b_i such that:

$$\sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{y}_j, j = 1, \ldots, N \tag{3}$$

If we denote as H a hidden layer output matrix of the ELM; column of H represents the i^th vector regard to inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$.

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L} \tag{3}$$

and

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times m} \tag{4}$$

Then the equivalent matrix form of (2) can be represented as:

$$\mathbf{H}\beta = \mathbf{Y} \tag{5}$$

The output weights are then computed by finding the unique smallest norm least-squares solution of the linear system (5) as:

$$\beta = \mathbf{H}^{\dagger}\mathbf{Y} \tag{6}$$

where $\mathbf{H}^{\dagger}$ represents the Moore-Penrose generalized inverse of H.

For a given training set $T=\{(\mathbf{x}_j, \mathbf{y}_j)\}|\ \mathbf{x}_j \in \mathbf{R}^n, \mathbf{y}_j \in \mathbf{R}^m, j = 1, ..., N\}$ with N instances of n-dimensional descriptors, the sigmoid activation function $g(x)$, and a hidden number of neurons L, the ELM algorithm for classification problems can be summarized as follows:

Training:
(a) Assign random input weights w_i, and biases b_i, $i = 1,..., L$.

(b) Compute the hidden layer output matrix H using (3).

(c) Compute the output weights β using (6).

Testing:
(a) Compute the hidden layer output matrix H_{test} for instances from the test set, using (3)

(b) Compute the output matrix Y_{test} according to (5) using the β obtained in step 3 of the training.

(c) For every row in Y_{test} (i.e. every test instance), compute a class label as the index of the maximal value in that row.

4. Experimental Evaluation

To test the proposed method for stock trend prediction, we used data taken from the website of the Belgrade Stock Exchange (www.belex.rs). The available data were divided into two groups. The first group consisted of 2485 records required for the training model, from February 24, 2006 to December 31, 2015. For the second group of data, from January 4, 2016 to December 29, 2016, a total of 252 days of trading were selected.

As a general measure for the evaluation of the prediction, the Hit Ratio (HR) is used, which was calculated on the basis of the number of properly classified results within the test group:

$$HR = \frac{1}{m}\sum_{i=1}^{m} PO_i \ \text{ for } \ PO_i = \begin{cases} 1 & PV_i = AV_i \\ 0 & PV_i \neq AV_i \end{cases} \tag{7}$$

where PO is the prediction output of the i trading day, AV_i is the actual value for the i training day and PI_i is the predicted value for the i trading day and m is the number of data in the test group.

For the tests, we implemented ELM in MATLAB and used it to measure the classification hit rate and speed. Hit rates are measured on both the training and test set, with training and test times, measured in seconds on an Intel Core i5 computer. The results are presented in table 2, where the data in the first column represent the number of neurons in a hidden layer. The sigmoid function is used as the activation.

From table 2 we can note that HR on the training and test set remains stable for the number of neurons in a hidden layer in the range from 10 to 100. When increasing the number of neurons further, HR on test set increases significantly, while HR on test set

Number of neurons	HR Training Set	HR Test Set	Training time (s)	Test time (s)
10	65 %	62 %	0.001	0.001
25	66 %	62 %	0.09	0.001
50	66 %	61 %	0.07	0.001
100	66 %	61 %	0.1	0.001
500	72 %	59 %	0.6	0.06
1000	77 %	58 %	1.5	0.07
10000	97 %	58 %	217	0.3

Table 2. A Comparison of the Models

decreases slightly, which implies possible overfitting. We can observe that increasing the number of neurons above a certain value does not improve classification results. Nevertheless, the obtained values of the hit rates for all ELMs, regardless of the number of neurons in the hidden layer are within the expected range of precision and comparable to the results obtained in other studies [1], [2], [12].

Training time for all 2458 instances in the training set is only 0.1 seconds with 100 neurons in a hidden layer, while classification for all 252 instances is done instantly (< 1ms). With an increase in the number of neurons of up to 1000, these times slightly increase. Only with 10000 neurons is training time increased significantly, while test time increases slightly (< 1s). These results demonstrate high performances in terms of training and test speed on this dataset.

In order to compare the results of the ELM with other common classification techniques, we measured the accuracy of the Linear SVM and kernelized RBF SVM [13], on the same dataset. Both Linear SMV and RBF SVM reached approximately a 62 % hit rate. Thus, it can be noted that ELM reaches results comparable with Linear SVM, as well as to the kernelized SVM while operating significantly faster during the training and testing. Regarding the results obtained using the random walk model (RW) as a benchmark that used current value to predict the future value, assuming that the latter in the following period (y_{t+1}) will be equal to the current value (y_t), ELM significantly outperformed the obtained results of 51.19%.

5. Conclusion

In this paper we presented the results of our research in the field of stock market trend prediction. A standard set of technical indicators is used as features in combination with a fast and powerful ELM classifier. A hit rate of around 60% is the expected range in this area. It can be concluded that a combination of technical indicators with an ELM classifier is reasonable choice for stock trend prediction applications. The ELM classifier could be used as an alternative to the commonly used SVM. In the future, we plan to investigate the performance of other types of features combined with an ELM classifier, particularly integration of trading strategies.

References

[1] Huang, W., Nakamori, Y. & Wang, S.Y. (2005) Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*, 32, 2513–2522 [DOI: 10.1016/j.cor.2004.03.016].

[2] Lahmiri, S. (2011) A comparison of PNN and SVM for stock market trend prediction using economic and technical information. *International Journal of Computers and Applications*, 29.

[3] Chsherbakov, V. (2010) Efficiency of use of technical analysis: Evidences from Russian stock market. *Ekonomika a Management*, 4.

[4] Dowd, K. (2002). Measuring Market Risk. John Wiley & Sons Ltd: Chichester, UK.

[5] Phichhang, O., Wang, H. (2009) Prediction of Stock Market Index Movement by Ten Data Mining Techniques. *Modern Applied Science*, 3, 28–42.

[6] Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2006) Extreme learning machine: Theory and applications. *Neurocomputing*, 70, 489–501 [DOI: 10.1016/j.neucom.2005.12.126].

[7] Huang, G., Huang, G.B., Song, S. & You, K. (2015) Trends in extreme learning machines: A review. Neural Networks, 61, 32–48 [DOI: 10.1016/j.neunet.2014.10.001] [PubMed: 25462632].

[8] Markovia, I., Stankovic, J. & M. (2014) Stojanovia. M. Bone. Prediction of the Stock Market Trend Using LS-SVMs Based on *Technical Indicators. presented at the 49th international conference*. ICEST: Nis, Serbia.

[9] Stojanove, M., Cvetkovie, S. & Staneie, G. (2017) Poredenje metoda maginskog ueenja za predikciju Trenda promene finansijskih vremenskih serija. Simpozijum INFOTEH Jahorina, 2017, 398–401.

[10] Amihud, Y. (2002) Illiquidity and stock returns: Cross-section and time-series effects. Journal of Financial Markets, 5, 31–56 [DOI: 10.1016/S1386-4181(01)00024-6].

[11] Stankovic, J.Z., Petrovic, E. (2018) Liquidity risk implications for market risk assessment in emerging markets. *Journal of Contemporary Economic and Business Issues*, 5, 5–23.

[12] Yuling, L., Guo, H. & Hu, J. (2013) An SVM-based approach for stock market trend prediction. *Neural Networks (IJCNN). IEEE Publications*, 1–7.

[13] Chang, C.-C. Lin, C.-J. (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2, 1–27 [DOI: 10.1145/1961189.1961199], p. 27:1-27:27.