Computer Science and Information Systems 18(3):749–769 https://doi.org/10.2298/CSIS200326044R

# A Multicriteria Optimization Approach for the Stock Market Feature Selection

Dragana Radojičić[1], Nina Radojičić[2] and Simeon Kredatus[1]

[1] TU Wien, Institute of Statistics and Mathematical Methods in Economics
Wiedner Hauptstr. 8/E105-01-05 FAM
1040 Vienna, Austria
{gagaradojicic, simeon.kredatus}@gmail.com
[2] Faculty of Mathematics, University of Belgrade
Studentski trg 16, 11000 Belgrade, Serbia
nina@matf.bg.ac.rs

**Abstract.** This paper studies the informativeness of features extracted from a limit order book data, to classify market data vector into the label (buy/idle) by using the Long short-term memory (LSTM) network. New technical indicators based on the support/resistance zones are introduced to enrich the set of features. We evaluate whether the performance of the LSTM network model is improved when we select features with respect to the newly proposed methods. Moreover, we employ multicriteria optimization to perform adequate feature selection among the proposed approaches, with respect to precision, recall, and $F_\beta$ score. Seven variations of approaches to select features are proposed and the best is selected by incorporation of multicriteria optimization.

**Keywords:** Limit order book, multicriteria optimization, time-series, feature selection, machine learning.

## 1. Introduction

The stock market has been the subject of studies of many research projects and publications, as well as of many financial institutions.

It has already been noted in the literature that the financial markets hold memory properties, see [48], [11], [38]. A wide range of research is done analyzing the financial data and exploring adequate mathematical models for modeling financial phenomena. In [32], the optimal strategy and the dynamic volatility derivatives pricing in an incomplete financial market were established using the expected utility maximization formulation. On the other hand, the authors in [21] proposed a smart market model that reduces peak-demand charges and has a positive final profit in energy systems.

Since there are more and more data on stock markets nowadays, there is a potential in developing algorithms to analyze historical data and take advantage of it. Nowadays, most exchanges have been automatized, because more trades are observed via automated electronic markets (see [6]). Therefore, most exchanges have switched from the traditional stock exchange, which had a physical location, to the electronic stock exchanges.

The electronic marketplace mechanism is comprehensively described in [7].

Various Artificial Intelligence approaches have been used for, usually very challenging, financial forecasting, which has been attracting many researchers. The authors in [37] provided a review on various techniques that perform well in the forecasting of stock markets, as well as they proposed an approach through data discretization by fuzzistics and rule generation by rough set theory.

The authors in [2] investigated the role of sample size and class distribution in credit risk assessments and their results indicate that various factors play a role in determining the optimal class distribution (the performance measure, classification algorithm and data set characteristics). Moreover, seven different machine and deep learning algorithms were proposed in [17] to learn the inherent patterns and predicting future movements in the stock market. Also, in [9] authors have used the weighted support vector machine (WSVM) method above the principal components to predict the stock trading signals. In [43] authors analyzed how Window Size influences the performance of future price direction prediction models that use technical indicators as input. The description of the trading strategy evolution and the strategy valuation is interpreted in [39].

In this research, we tend to cross compare the performance of the Long short-term memory (LSTM) network (initially introduced in [23]) when the model is fed with different groups of features extracted from the market limit order book (LOB) data. The research presented in this paper is unique in the sense that we used the customized data transformation, included the resistance support features and multicriteria optimization is used to select the best option among seven different approaches proposed to perform feature selection. Therefore, the presented research is not comparable to any other work available in the literature. But note that in [41] the authors extracted the Fourier transformation based features from the LOB and evaluated the performance of the GRU based model when fed with different groups of extracted features. Further, note that multicriteria optimization approaches for choosing the set of features for LSTM network models have not been considered in the literature up until now.

The LSTM network model is employed within this research in order to estimate if it is a good time to open a trade or it is better to idle. During the data transformation part, a huge amount of features are extracted from the limit order book and dealing with a large amount of various types of data has become an important challenge in computer science in recent years (see [47] and [27]). These extracted features describe market behavior. In this research, seven groups of features are chosen from the set of all features by employing proposed selection criteria. To evaluate the performance of the proposed LSTM network model when fed with the different groups of features several relevant metrics can be used.

In order to be able to better express and make objective decisions about the efficiency of the proposed method for feature selection, we incorporated multicriteria optimization, which has become used with its software support in various applications (see [16]). Besides the fact that choosing the most reliable set of features is multi-layered problems, the trades are occurring very fast and decisions need to be made quickly. Therefore, an intelligent approach for selection is necessary to make quick and precise decisions in order to capture opportunities. Further, multicriteria optimization enables to improve the quality of decisions, and it allows the decision-maker to point out the preference of one alternative over the other ones compared with several arguments. Therefore, to make a reliable financial decision and to evaluate opportunities and investments, financial intelligence needs to be utilized, see [22] and [8].

We start our paper with a brief introduction of the theory that stands behind the LOB concept, which is important for the research presented in this paper. After that, in Section 2 we describe the dataset which is used for our research. Then, new features based on the support/resistance zones are introduced in Section 3. In Section 4 we introduce criterion with the respect which we group features. Section 5 contains a multicriteria approach needed in this research and basics on the used method. Moreover, the results are presented in Section 6. Final conclusions and prospective work are given in Section 7.

### 1.1. A short review of a mathematical description of the Limit Order Book

The main job of the limit order book is to record all incoming and outgoing orders. The Limit Order Book is defined on a discrete price grid, where every grid's point models a price level, see Fig. 1. The minimum distance between two price levels is called a *tick*. On the ASK side the orders that need to be sold are placed, while on the BID side the orders that need to be bought are placed. The minimum price at the moment $t$ on the ASK side, denoted by $P_t^a$, is named the best ask price, while the maximum price at the moment $t$ on the BID side, denoted by $P_t^b$, is called the best bid price.

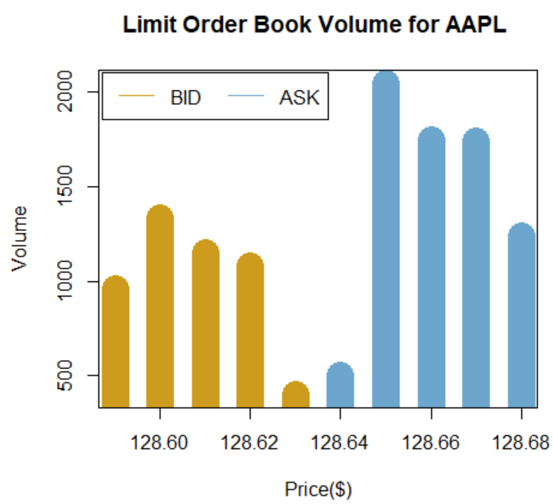



**Fig. 1.** Snapshot of the NASDAQ limit order book for AAPL stock symbol for 5 levels at 09:28:47am (on the 20th Februar 2015). On the bid/ask side are placed outstanding buy/sell orders (gold/blue), and the best bid price is $\$128.63$ with volume of $500$ shares, while the best ask price is $\$128.64$ with volume of $600$ shares.

The difference between the best ask and the best bid price is called the quoted spread, i.e.:

$$\mathit{QuotedSpread} = P_t^b - P_t^a.$$

The mid-price is defined as a arithmetic average of the best bid and best ask price, i.e.:

$$\mathit{MidPrice} = \frac{1}{2}(P_t^b + P_t^a).$$

An detailed mathematical explanation of the limit order book (LOB) can be found in Section II in [19].

The study of the limit order book dynamic is crucial for the financial world. The modeling of the next price-flip event in limit order book using the ability of the RNN is presented in [15]. In [12] a continuous-time stochastic model for the limit order book dynamics, which extracts key empirical properties of the LOB is established. Considering the volume of the LOB at different distances to the best ask price authors in [1] derived the mid price diffusion limit. In [24] authors introduce a two-sided limit order book stochastic model and prove the limit theorem when the tick size is converging to zero. A framework for manipulation in a computational model of a limit-order book, which emphasis information asymmetry between buy and sell makes profitable manipulation possible, was presented in [46].

Our research is based on the real market data from the past, more precisely on the high-quality online limit order book data tool Lobster[3], which replicates the entire NASDAQ Stock Market (the second largest exchange in the world) from the 27th of June 2007 up to the two days ago from the current day. Although in [25] the LOBSTER reconstruction approach is presented, in order to extract essential structures from the limit order book for our research we have used our customized data processing reconstructor which is presented in the next section. Moreover, since we are working with a huge data set, to process data we have used a processing engine supported by an advanced technology, which is explained in the [40].

## 2. The market data summary and data processing reconstructor

For each trading day and for each selected ticker (e.g. AAPL which stands for Apple, MSFT which stands for Microsoft Corporation, etc.) LOBSTER has 'orderbook' and 'message' file. The 'orderbook' file has information on the Price and Volume up to the requested number of levels. The 'message' file contains the following columns: time (represents the time when an event has occurred), type of the event (submission of a new order, cancellations, deletions of an order, execution of a visible order, execution of a hidden order, trading halt), order ID, size of the order, price, direction of the trade (sell/buy).

We define a market data vector $x_t$ at a time point $t$, which contains market data features, i.e.:

$x_t$ = ($bidLevel_1$, $bidVolume_1$, $askLevel_1$, $askVolume_1$, $bidLevel_2$, $bidVolume_2$, $askLevel_2$, $askVolume_2, \ldots,$ $bidLevel_n$, $bidVolume_n$, $askLevel_n$, $askVolume_n$, $Time$, $EventType$, $OrderId$, $Size$, $Price$, $Direction$).

Note that the limit order book is continuously evolving, so the vector $x_{t+1}$ is derived from the type of the event that is compressed in the vector $x_t$.

Stock markets are producing a huge amount of data and there are over a million events for each stock symbol data in one trading day. Therefore, to avoid working with a vast amount of data, and in order to extract relevant features from the limit order book, we employ the data aggregation with respect to the 3min interval. During the 3min interval aggregation, more features are extracted, such as a number of canceled limit orders, a number of executed limit orders, open price (price at the beginning of the interval), close price, maximum price (during that interval), etc.

[3] Lobster academic research data. https://lobsterdata.com.

Furthermore, we compute the set of standardly known technical indicators in order to get more features describing market behavior. Since, there are various factors that have an influence on the stock market, using adequate technical indicators is important in achieving good forecasting results, see for example [49]. For a comprehensive description of the technical indicators and their use see [34], [10] and Table 2 in [49]. Note that these technical indicators are computed using an open source library ta-lib[4]. At this stage, for each timestamp $t$ the data vector $\tilde{x}_t$ (enhanced vector $x_t$) contains features extracted from the LOB shape (e.g. Volume at each price level, number of executed trades, Open Price, Maximum Price, etc.) and technical analysis indicators (e.g. Bollinger Bands, moving average). Now for a particular trading day $d$, we have a dataset $\mathcal{D}_d = \{\tilde{x}_t | 0 \leq t \leq \lfloor \frac{duration\ of\ a\ trading\ day\ d\ in\ s}{180s} \rfloor\}$.

The goal is to classify vector of market data $\tilde{x}_t$ into the one of the labels from the set $S_+ = \{buy, idle\}$, if we consider semi-strategy emitting buy signals, or from the set $S_- = \{sell, idle\}$, if we consider semi-strategy emitting sell signals. Considering a semi strategy emitting buy signals, we examine whether from given vector $\tilde{x}_t$ every subsequent vector $\tilde{x}_{t+1}$, reach certain profit until the end of the day, with only exposing ourselves to a certain risk. Thus, the main idea is to see if each point reaches the desired risk-reward ratio (RRR), and to assign the label to each vector with respect to the Boolean function: 1 if the buy order should be issued, and 0 if it is better to idle. The particular algorithm is presented in the Appendix (Algorithm 1). Note that it is based on the same idea as Algorithm 1 in [40] and Algorithm 2 in [41]. In order to label our training set for this research, we have run the Algorithm 1 with the following values: $REWARD = 0.08$ and $RISK = 0.04$. The aforementioned data transformations extract features of interest and prepare data set as needed of our research. Note that [40] contains plain, but similar, concepts of the presented data transformation reconstructor. More precisely, the data lapsing transformations are employed in [40] and therefore other features are extracted. Also, in [41] a similar data reconstructor is used and further features based on the Fourier transformations are extracted.

## 3. Support-Resistance zones based indicators

In this section, we propose an algorithm that estimates support-resistance zones.

The support and resistance levels are often applied for short-term financial market forecasting. For example, the predictive power of support and resistance levels for intraday exchange rates are presented in [36].

Our algorithm takes any of the prices as the input. Be it open price, close price, or eventually the price level of a limit order book, such as the best bid price, the value of the third bid level, etc.

Let the window $\mathcal{W}_{index}$ be the window containing all the points found in the daily price series between the indexes *index - WINDOW_SIZE* to *index*, where *index* is the current position of the iterator in a daily price series. We define the *upper / lower bound line* as a line which approaches the data contained in the window $\mathcal{W}_{index}$ from the top/bottom, such that the absolute distance between points contained in the $\mathcal{W}_{index}$ and the line itself is minimal and there is no point touching the line. Each of the respective lines can be

[4] Ta-lib: Technical analysis library. https://www.ta-lib.org/

described by two parameters slope and offset, i.e.:

$$y_{upperbound,x} = slope_{upperbound,x} * x + offset_{upperbound,x},$$

$$y_{lowerbound,x} = slope_{lowerbound,x} * x + offset_{lowerbound,x},$$

where x stands for the index within the window $\mathcal{W}_{index}$ and $y_{upperbound,x}$ and $y_{lowerbound,x}$ are the predicted bounding lines at that index. For a given window size, for each level and for amount of points $N$ in a single trading day series, we extract slope and offset in order to describe the behavior of the support/resistance over time. However, we also need to identify the strength of such a zone. We define this to be a percentage of total points contained in a window $\mathcal{W}$ which falls into the distance no more than $d > 0$ from a respective line. Let $i$ be the current index in the interval $[$*index* - *WINDOW_SIZE*, *index*$]$. A point $w_i \in \mathcal{W}_{index}$ is called a challenger if a distance between the point and the bounding line is less than $100 * C$ percents of the bounding line value, where $C \in [0, 1]$. So to define the resistance strength $R_s$ as the relative proportion of challengers when compared to the whole window size, i.e.:

$$R_{s,index} = \frac{|\{w_i \in \mathcal{W}_{index} : w_i \geq ((slope_{upperbound} * i + offset_{upperbound}) \times (1 - C)\}|}{|\mathcal{W}_{index}|},$$

where $C$ is the constant controlling the size of the band which if exceeded shall be considered as a challenger of the resistance zone.

For the support zone, we can proceed similarly:

$$S_{s,index} = \frac{|\{w_i \in \mathcal{W}_{index} : w_i \leq ((slope_{upperbound} * i + offset_{upperbound}) \times (1 + C)\}|}{|\mathcal{W}_{index}|}.$$

A visualization of the bounding lines can be found in the Figure 2. Two green lines represent the currently chosen window. The yellow line represents fitted upper bound and the blue line represents fitted lower bound. Apart from extracting slopes, offsets and strengths for each line we also extract the angle between the upper and lower bound. If the lines are diverging (i.e. they have intercepted one another prior to the currently evaluated index $i$), then the angle is kept positive. If the lines have not intercepted one another before currently evaluated index $i$ (as seen in the Figure 2, the second green vertical line depicts currently evaluated index $i$ - intersection occurs after it) we set the angle to the negative value. In order to compute the angle we compute the direction vectors $\boldsymbol{d_u}, \boldsymbol{d_l}$ for each of the lines ($y_{upperbound}$ and $y_{lowerbound}$). Furthermore, we apply the well known equation to extract the angle itself:

$$cos\alpha = \frac{\boldsymbol{d_u} \cdot \boldsymbol{d_l}}{|\boldsymbol{d_u}| \cdot |\boldsymbol{d_l}|}.$$

We extracted our support/resistance (SR) based features from all the limit order book level prices (ask / bid) - we work with the data of depth five. In Figure 3 we provide mutual information values computed for the SR features extracted from the Ask 1 level prices. As a reader can note, these features provide significant mutual information with the close price. Since there is a low correlation between support/resistance (SR) based features and features extracted in Section 2, it implies the fact they shall bring different information and thus they could nicely encompass one another.

The best performing feature *intersecting_ys* denotes the price value at which the upper and lower bounding line intersect.

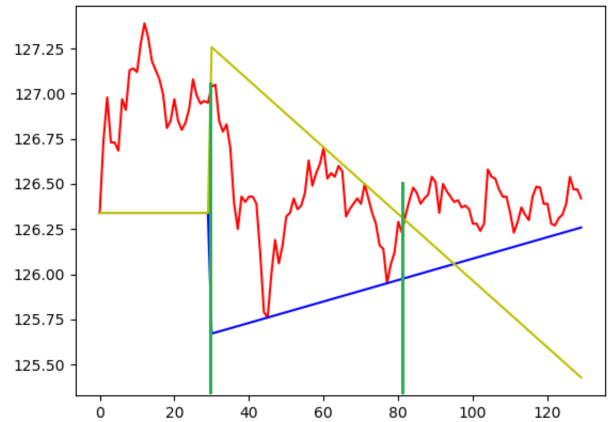



**Fig. 2.** Visualization of the upper and lower bounding lines estimating the S/R zones. Red line depict the price.

Furthermore, we can study the behavior of the same feature when extracted over different limit order book levels to understand which level holds relevant information.

Surprisingly, in some of the extracted features, we experience an increasing portion of mutual information with increasing depth of limit order book. As can be observed in Figure 4, the angle of intersection tends to have more information when extracted over the deeper level and poses linear growth with respect to the depth.

Another feature exhibiting similar interesting behavior is the $'intersecting\_ys'$ feature, which again shows higher importance with increasing depth level 5.

The observed behavior can be explained by a concept that the deeper level of the market is usually where the price lands if a change starts. This means that if there is a significant market movement theoretically the fifth level might become the first level value in the future and thus there is higher mutual information.

## 4. Proposed methodology

### 4.1. Motivation

During the data transformation part, we extracted features from the LOB and we enhanced that set with technical indicators. Furthermore, we included the new features based on the support-resistance zones, and thus there are nearly 3800 features. Since there is a huge number of features for each timestamp, if we plug all of them in the LSTM model, the model tends to over-fit. Therefore, feature selection is important for the model to perform well. But from such a large amount of possible features that could be extracted, how to choose the relevant ones? Considering any feature, the key question here is how to determine its quality.

Different approaches to perform feature selection have been explored in the literature. In [26] four different models, based on different feature selection methods, which use

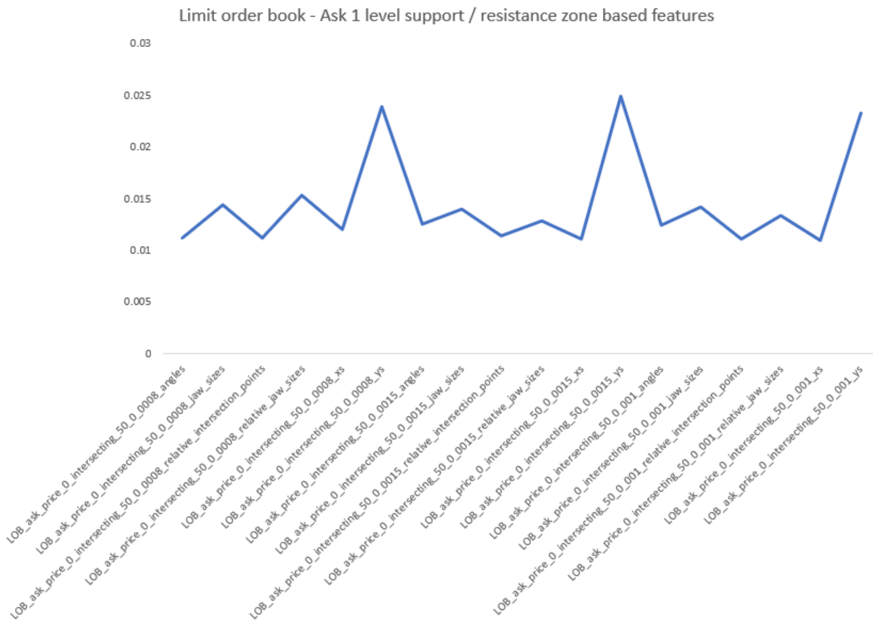



**Fig. 3.** Mutual information of the support / resistance zone based features for the level 1 prices on the ask side.

fifty-five technical indicators as input variables to predict the direction of the price, were compared. Moreover, hybrid Artificial Neural Network (ANN) models were employed in [18] to choose technical indicators that are relevant to determine stock market price direction. Feature selection processes have been applied in various fields. The authors in [4] aimed of this research is to develop a Majority Vote based Feature Selection algorithm (MVFS) to identify the most valuable software metrics and the thorough experiments showed the ability of the proposed method to find out the most significant software metrics that enhance defect prediction performance. On the other hand, the authors in [30] analyzed the correlations among different commodities sales to identify interesting patterns to increase cross-marketing quality.

### 4.2. Basic measures used in the paper

Here, we mention two measures used in the following of the paper.

*Mutual Information* The mutual information (MI), also known as the information gain, of two random variables is a quantity that measures the amount of information about one random variable which is communicated with another one.

Formally, the mutual information of a pair of two random variables $X$ and $Y$ with values in $\mathcal{X}$ and $\mathcal{Y}$ respectively, with the joint distribution $p_{X,Y}$ and the marginal distri-

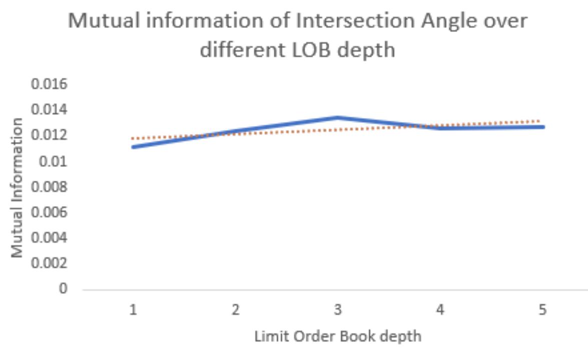



**Fig. 4.** Mutual information of the angle of 'intersection' feature when computed over different limit order book levels.

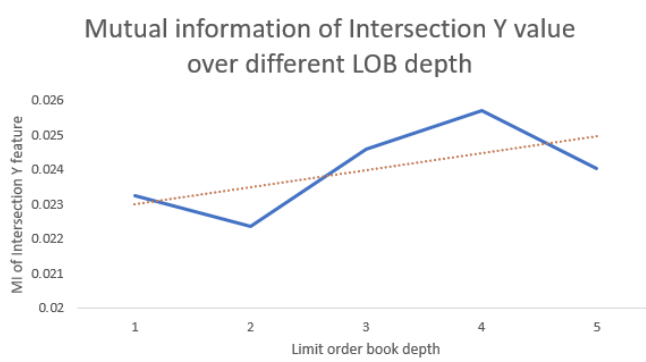



**Fig. 5.** Mutual information of the 'intersection Y' feature plotted over different limit order book depth levels.

butions $p_X$ and $p_Y$, is calculated by the formula:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)}.$$

For a deeper theoretical overview on a mutual information we refer the reader to [44]. Furthermore, a comprehensive overview of feature selection methods based on mutual information is stated in [45].

*Autocorrelation* Autocorrelation measures a correlation of a signal with its lagged version over different successive time periods. Informally, it is a degree of similarity between a signal's present value and its former values.

### 4.3. Feature groups

In this paper, two assumptions have been made in order to perform feature selection. We assume that a feature with higher autocorrelation is more powerful when it comes to classifying the market data vector into the label than a feature having lower autocorrelation. Another assumption is that a feature that has higher mutual information with close price contains more information than a feature with lower mutual information.

Therefore, we construct two lists with features, which are sorted descended, one with respect to feature's autocorrelation and one with respect to the feature's mutual information with the close price. Moreover, we use also the initial list of all features. These three different lists correspond to the column named list of features Table 1, while in the last column the selection method was presented. The first group of features was selected completely randomly from the set of all features. For other groups, we employ more interesting selection criteria: if two features A and B have the absolute value of cross-correlation higher than $\alpha = 0.7$, we choose one which is higher in the list. More precisely, from the set of all features in the lists, sorted descending, we choose features with the highest value (autocorrelation or mutual information) such that each pair of selected features have the absolute value of cross-correlation less than 0.7. We measure cross-correlation with respect to Pearson's and Spearman's correlation coefficient, which measures the degree of linear and monotonic correlation between two signals respectively. The list of features for the construction of group 7 was randomly sorted before the selection method was applied.

To summarize, we constructed seven groups of 200 features each and the construction method of each group is presented in Table 1. We compare the performance of the model fed with the features from the aforementioned groups.

For this research, seven different datasets (one per each group of features) were prepared. In order to perform LSTM, each dataset is split into three separate parts: for the training, for the validation and for the testing phase. The first one needs 60% of the data, while validation and testing use 20% each. The data for the testing phase is not visible during the training phase and then the confusion matrix is calculated.

**Table 1.** The lists of features and selection methods for all seven considered groups of features

| | list of features | selection method |
|---|---|---|
| Group 1 | ALL | Random |
| Group 2 | mutal information | cross corr Pearson |
| Group 3 | autocorrelation | cross corr Pearson |
| Group 4 | ALL | cross corr Pearson |
| Group 5 | autocorrelation | cross spearman |
| Group 6 | mutal information | cross spearman |
| Group 7 | ALL | cross spearman |

## 5. Multi-criteria optimization for choosing optimal group of features

Please note that none of these groups leads to high classification accuracy. That is completely expected due to the specific behavior of the stock market, i.e. a lot of noise present in the data.

However, we aim to compare the performance of the model when fed with different groups of features. The confusion matrix (see [34]) is one of the most used metrics to measure the performance of classification based supervised learning models. Two widely used metrics (e.g. see Section 4 in [49]) are precision and recall, defined as presented with equations (1) and (2).

$$precision = \frac{truepositive}{truepositive + falsepositive} \tag{1}$$

$$recall = \frac{truepositive}{truepositive + falsenegative} \tag{2}$$

These two metrics are different in a way that precision calculates how accurate the model is in a sense how many of the predicted positive are actual positive, while recall measures how many of the actual positives are predicted to be positive. Thus, precision is an adequate measure to use when the cost of false positive is high, while recall shall be used when the cost of false negative is high. In this research we also use an interesting measure, a function of precision and recall, called $F_\beta$ score (Eq. (3)). In addition to the widely used $F_1$ score, which is employed when both precision and recall are equally important, commonly used ones are $F_{0.5}$ and $F_{1.5}$. $F_1$ score is a very convenient measure if we want a balance between precision and recall (see [34]). Note that the more we decrease $\beta$, the more we prefer precision over recall. Conversely, the more we increase x, the more we favor recall over precision. Note that as more we decrease $\beta$ we prefer more precision over recall, and vice versa. The $F_\beta$-measure with a smaller $\beta$ value, such as the $F_{0.5}$-measure, is utilized when more weight is put on precision, and less weight is put on recall, i.e. when minimizing false positives is more important than minimizing false negatives. Therefore, the $F_{0.5}$-measure is used in order to increase the importance of precision and decrease the importance of recall. On the other hand, $F_\beta$-measure with $\beta$ value higher than 1, such as the $F_{1.5}$-measure, is utilized when slightly more attention is put on recall, and less is put on precision. Thus, the $F_{1.5}$-measure is used when minimizing false negatives is more relevant than minimizing false positives, i.e. when more attention is put on raising the importance of false positives rather than false negatives. An example, usually found in the literature, is the $F_2$-measure. However, in this research, the focus was to just slightly increase the importance of recall over precision and thus, the $F_{1.5}$-measure is used.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{3}$$

There is no alternative optimizing all the criteria (precision, recall, $F_1$, etc.) at the same time, but there are many approaches within multicriteria optimization that can be used for problems having multicriteria nature. More on multicriteria decision analysis can be found in [20]. The enhanced concept of a confusion matrix, which evaluates the performance of a trading strategy was established in [14].

In order to choose the optimal group of features, a multi-criteria decision-making methodology, namely PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) method, was used. The basic version of the PROMETHEE was presented in [5], while a comprehensive literature review on methodologies and applications of the PROMETHEE family can be found in [3]. Various versions, numerous modifications and additions of the PROMETHEE method have been applied to many problems. In [33] authors used PROMETHEE II to employ a multicriteria evaluation of the statistical and machine learning classifiers for financial decision making.

In this paper, calculations regarding the PROMETHEE method are performed by using the academic version of the Visual PROMETHEE software[5]. We formulate a new multicriteria problem with 7 alternatives/actions (which corresponds to the aforementioned groups from Table 4.3) and 5 criteria (precision, recall, $F_1$, $F_\beta$ for $\beta = 0.5$, $F_\beta$ for $\beta = 1.5$). Six different preference functions were proposed in [5]. In this research, the V-shaped preference function was used for all the criteria due to its simplicity and compatibility with the used criteria.

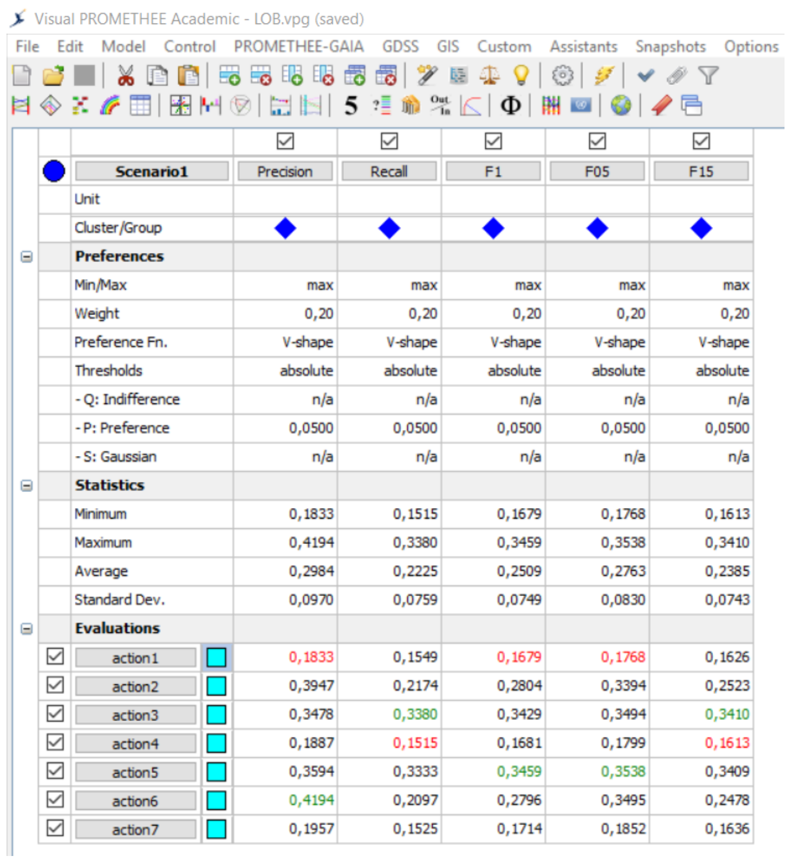
For determinations of weights (relative significance) of each criterion in practical applications of multicriteria optimization various methods can be used depending on the type of the problem, its structuredness as well as the knowledge and experience of the decision maker. A systematic review of the possible methods is given in [35], which can be used as an instruction for an adequate selection of methods for setting the criteria weights in practical applications of multicriteria optimization. One of the presented methods is the Delphi method, which is used in this paper in order to set up the weights of each criterion according to the assessment of the experts in order to obtain the realistic and objective model of the analyzed problem. The Delphi method is based on systematic and organized acquiring and processing of data obtained by the individual predictions given by a small group of experts in order to obtain the mean value from a series of questionnaires (see [13] and [31]). The basic concepts can be found in [42]. Today, this method has many modifications depending on the application, see [28], [29], etc. The simplicity and wide range of application of the Delphi method are based also on the fact that it is suitable to perform this method to tackle the problems that are not structured in a precise way, as well as due to the availability of software support for this method [28]. It can be used for the problems where there is no available data that is relevant and precise enough to perform further research without the usage of experts' assessments, e.g. for setting the weights for multicriteria optimization problems. In this paper, according to the results obtained by using the Delphi method, the system with five different criteria is formulated and the problem is solved for three different variants of weights.

## 6. Results

To perform the PROMETHEE method, the presented seven different groups (actions) and five presented criteria were entered into the Visual PROMETHEE software, as presented in Fig. 6. The values for all criteria for each action were calculated and entered into the table (see Fig. 6), and the software colored the best values for each criterion in green, while the smallest value for each criterion is in red. Moreover, the software calculates basic statistics for each criterion, e.g. minimum, maximum, average and standard deviation. In order for the PROMETHEE method to be performed, the preference functions had to be defined. The V-shaped preference functions are used to represent linear preference between compared alternatives until the threshold value after which preference is set to one.

[5] http://www.promethee-gaia.net/academic-edition.html

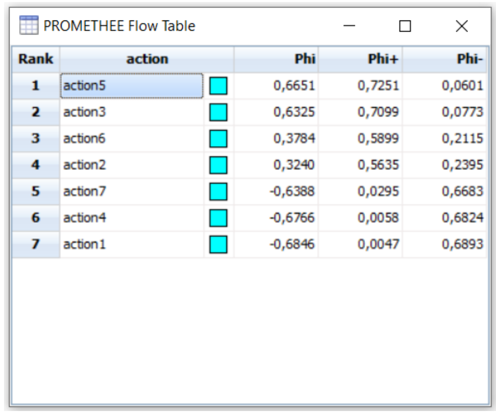
The threshold value is set to 0.05 according to standard deviations of the values for each criterion.




| Scenario1 | Precision | Recall | F1 | F05 | F15 |
|---|---|---|---|---|---|
| Unit | | | | | |
| Cluster/Group | | | | | |
| **Preferences** | | | | | |
| Min/Max | max | max | max | max | max |
| Weight | 0,20 | 0,20 | 0,20 | 0,20 | 0,20 |
| Preference Fn. | V-shape | V-shape | V-shape | V-shape | V-shape |
| Thresholds | absolute | absolute | absolute | absolute | absolute |
| - Q: Indifference | n/a | n/a | n/a | n/a | n/a |
| - P: Preference | 0,0500 | 0,0500 | 0,0500 | 0,0500 | 0,0500 |
| - S: Gaussian | n/a | n/a | n/a | n/a | n/a |
| **Statistics** | | | | | |
| Minimum | 0,1833 | 0,1515 | 0,1679 | 0,1768 | 0,1613 |
| Maximum | 0,4194 | 0,3380 | 0,3459 | 0,3538 | 0,3410 |
| Average | 0,2984 | 0,2225 | 0,2509 | 0,2763 | 0,2385 |
| Standard Dev. | 0,0970 | 0,0759 | 0,0749 | 0,0830 | 0,0743 |
| **Evaluations** | | | | | |
| action1 | 0,1833 | 0,1549 | 0,1679 | 0,1768 | 0,1626 |
| action2 | 0,3947 | 0,2174 | 0,2804 | 0,3394 | 0,2523 |
| action3 | 0,3478 | 0,3380 | 0,3429 | 0,3494 | 0,3410 |
| action4 | 0,1887 | 0,1515 | 0,1681 | 0,1799 | 0,1613 |
| action5 | 0,3594 | 0,3333 | 0,3459 | 0,3538 | 0,3409 |
| action6 | 0,4194 | 0,2097 | 0,2796 | 0,3495 | 0,2478 |
| action7 | 0,1957 | 0,1525 | 0,1714 | 0,1852 | 0,1636 |

**Fig. 6.** Inserting data into the Visual PROMETHEE software.

An important aspect of multicriteria optimization is stating the importance/weights of each criterion. When all criteria are equally important, weights for all five criteria are set to the same value, i.e. 0.2, the result flow table is presented in Fig. 7. This table also contains the positive and negative outranking flow (denoted as Phi+ and Phi-), as well as calculated net outranking flow (denoted as Phi), which represents the balance between the positive and the negative outranking flows (see [20]). It can be seen that the best option is action number 5, while the worst one is action number 1. Therefore, an illustrative representation of the resulting order of groups can be seen in Figure 8.


PROMETHEE Flow Table

| Rank | action | Phi | Phi+ | Phi- |
|---|---|---|---|---|
| 1 | action5 | 0,6651 | 0,7251 | 0,0601 |
| 2 | action3 | 0,6325 | 0,7099 | 0,0773 |
| 3 | action6 | 0,3784 | 0,5899 | 0,2115 |
| 4 | action2 | 0,3240 | 0,5635 | 0,2395 |
| 5 | action7 | -0,6388 | 0,0295 | 0,6683 |
| 6 | action4 | -0,6766 | 0,0058 | 0,6824 |
| 7 | action1 | -0,6846 | 0,0047 | 0,6893 |

**Fig. 7.** The result flow table for weights (0.2, 0.2, 0.2, 0.2, 0.2)

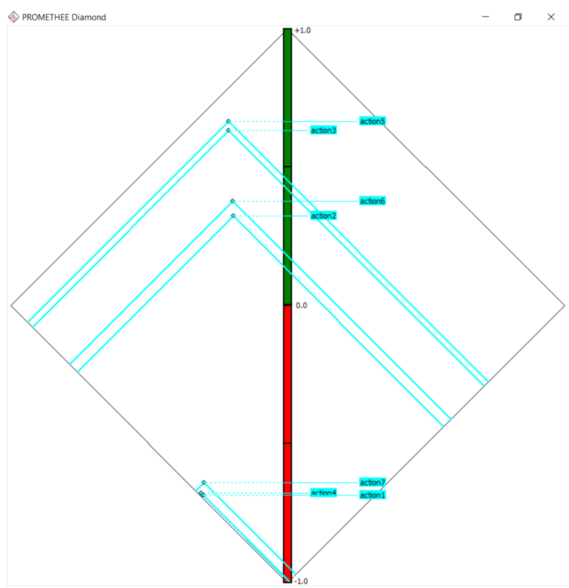



**Fig. 8.** The diamond representation of results for weights (0.2, 0.2, 0.2, 0.2, 0.2)

Moreover, it is interesting to investigate the significance of sensitivity analysis in assessing how different weights (relative significance) of each criterion affect the results. In this research, we were specifically interested in changing weights of two criteria: precision and recall. In order to perform the comparison, the results are found for the situation when the precision has weight 0.4, while others have 0.15. The result flow table for this situation can be seen in Figure 9, while the diamond representation is represented in Figure 10. The best alternative is still the action number 5. Thus, we can say it stays stable when we increase the significance of the precision criteria from 0.2 to 0.4. The second best option is now number 6.

Similarly, the results are found for the situation when the recall has weight 0.4, while others have 0.15. The result flow table for this situation can be seen in Figure 11, while the diamond representation is represented in Figure 12. The action number 5 is still slightly better than the second best option. However, there is a change at the end of the table, and the action number 4 is considered the worst according to these weights of criteria, by being slightly worse than the action 1.

PROMETHEE Flow Table

| Rank | action | Phi | Phi+ | Phi- |
|---|---|---|---|---|
| 1 | action5 | 0,5624 | 0,6785 | 0,1161 |
| 2 | action6 | 0,5127 | 0,6714 | 0,1587 |
| 3 | action3 | 0,5090 | 0,6574 | 0,1484 |
| 4 | action2 | 0,4159 | 0,6161 | 0,2002 |
| 5 | action7 | -0,6296 | 0,0383 | 0,6679 |
| 6 | action4 | -0,6754 | 0,0089 | 0,6843 |
| 7 | action1 | -0,6949 | 0,0036 | 0,6985 |

**Fig. 9.** The result flow table for weights (0.4, 0.15, 0.15, 0.15, 0.15)

Also, we suggest that not only the price as a source of the signal shall be investigated, however respective limit order book levels born significant informativeness too. Furthermore, some of the features extracted out of simple support/resistance zones turned out to have an interesting property - the deeper the limit order book level is, the higher the informativeness (mutual-information) when predicting the target variable.

## 7. Conclusion and future work

In this paper, the performance of the model based on LSTM to predict profitable trades was investigated when fed with different features extracted from the LOB data. To perform that, a good feature selection approach had to be employed. Among presented seven variants of feature selection methodology, in order to choose the most suitable option with respect to the five proposed criteria a multicriteria optimization technique (i.e. PROMETHEE method) was performed. Moreover, it was investigated what happens when different values for the relative significance of the proposed criteria are used. The best feature selection strategy was shown to be the strategy that produced the group that consists of the features having high autocorrelation, which are pairwise not monotonically correlated in the sense that each pair of features has Spearman's correlation coefficient less

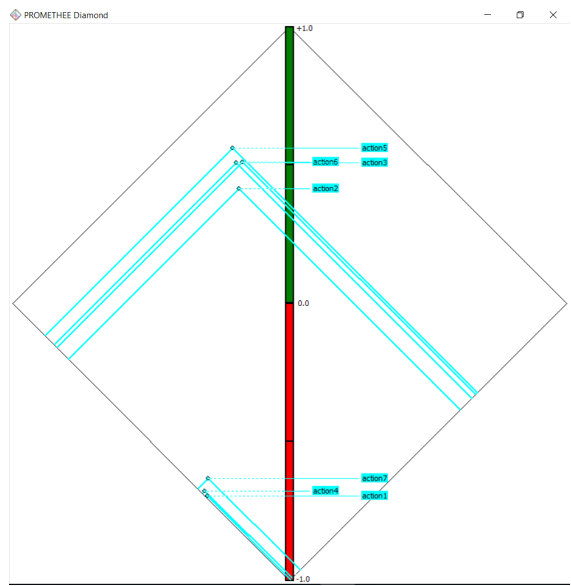
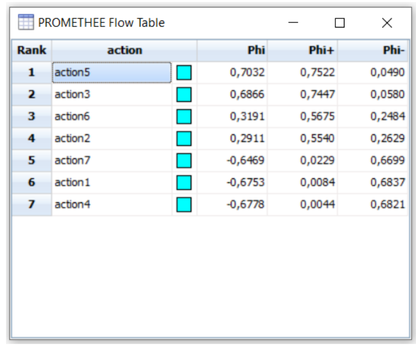



**Fig. 10.** The diamond representation of results for weights (0.4, 0.15, 0.15, 0.15, 0.15)

than $\alpha = 0.7$. This confirms our initial assumption that features that have higher autocorrelation hold more information. Note that group number 5 stayed the best even when we changed the significance of the precision and recall, meaning that this is the best option even when the cost of false positive high and when the cost of false negative high.

Future work will be continued in several directions. We will try to introduce some new features in order to explore how that would affect the proposed model and decision-making method's results. On the other hand, we want to use other multicriteria optimization methods.

**Acknowledgments.** The authors are thankful to Thorsten Rheinländer for providing insightful comments and for his help in doing the LOBSTER data analysis and valuable, constructive and insightful suggestions. This research was supported by TU Wien in Austria, the Faculty of Mathematics, University of Belgrade and the Serbian Ministry of Education, Science and Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts.



PROMETHEE Flow Table

| Rank | action | Phi | Phi+ | Phi- |
|---|---|---|---|---|
| 1 | action5 | 0,7032 | 0,7522 | 0,0490 |
| 2 | action3 | 0,6866 | 0,7447 | 0,0580 |
| 3 | action6 | 0,3191 | 0,5675 | 0,2484 |
| 4 | action2 | 0,2911 | 0,5540 | 0,2629 |
| 5 | action7 | -0,6469 | 0,0229 | 0,6699 |
| 6 | action1 | -0,6753 | 0,0084 | 0,6837 |
| 7 | action4 | -0,6778 | 0,0044 | 0,6821 |

**Fig. 11.** The diamond representation of results for weights (0.15, 0.4, 0.15, 0.15, 0.15)

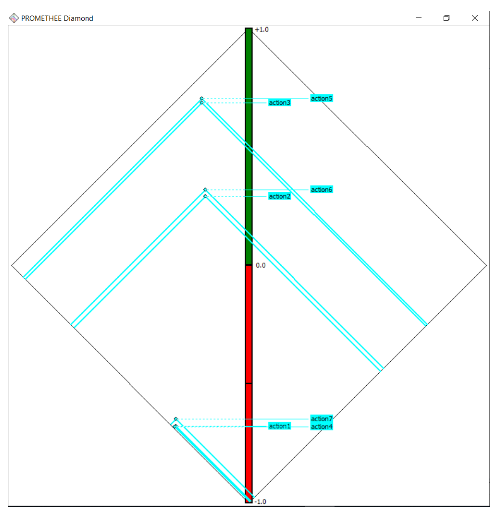



**Fig. 12.** The diamond representation of results for weights (0.15, 0.4, 0.15, 0.15, 0.15)

## 8. Appendix

**Algorithm 1** Sweep forward algorithm labelling the desired trades

**Input:** $\mathcal{D}_d$,$RISK$, $REWARD$
**Output:** $\mathcal{D}_d^*$ - the labelled data set

```
D*_d = None
for t = 0 to |D_day| do
    tPrice = getPrice(D_d[t])
    stopLoss = tPrice · (1 − RISK)
    targetReward = tPrice · (1 + REWARD)
    x_t = D_d[t]
    labelled = False
    for subsequent_t = t to |D_d| do
        currentPrice = getPrice(D_d[subsequent_t])
        tempStopLoss = currentPrice · (1 − RISK)
        if (tempStopLoss > stopLoss) then
            stopLoss = tempStopLoss
        end if
        if (currentPrice < stopLoss) then
            D*_d.append(labelAsIdlePoint(x_t))
            labelled = True
            break
        end if
        if (currentPrice > targetReward) then
            D*_d.append(labelAsBuyPoint(x_t))
            labelled = True
            break
        end if
    end for
    if (labelled == False) then
        D*_d.append(labelAsIdlePoint(x_t))
    end if
end for
return D*_d
```

## References


1. Abergel, F., Jedidi, A.: A mathematical approach to order book modeling. International Journal of Theoretical and Applied Finance 16(05), 1350025 (2013)
2. Andrić, K., Kalpić, D., Bohaček, Z.: An insight into the effects of class imbalance and sampling on classification accuracy in credit risk assessment. Computer Science and Information Systems 16(1), 155–178 (2019)
3. Behzadian, M., Kazemzadeh, R., Albadvi, A., Aghdasi, M.: Promethee: A comprehensive literature review on methodologies and applications. European Journal of Operational Research 200(1), 198 – 215 (2010), http://www.sciencedirect.com/science/article/pii/S0377221709000071

4. Borandag, E., Ozcift, A., Kilinc, D., Yucalar, F.: Majority vote feature selection algorithm in software fault prediction. Computer Science and Information Systems 16(2), 515–539 (2019)
5. Brans, J.P., Vincke, P.: Note—a preference ranking organisation method: (the promethee method for multiple criteria decision-making). Management science 31(6), 647–656 (1985)
6. le Calvez, A., Cliff, D.: Deep learning can replicate adaptive traders in a limit-order-book financial market. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1876–1883. IEEE (2018)
7. Cartea, Á., Jaimungal, S., Penalva, J.: Algorithmic and high-frequency trading. Cambridge University Press (2015)
8. Chen, S.H., Wang, P.P.: Computational intelligence in economics and finance. In: Computational Intelligence in Economics and Finance, pp. 3–55. Springer (2004)
9. Chen, Y., Hao, Y.: Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. Neurocomputing 321, 381–402 (2018)
10. Colby, R.W., Meyers, T.A.: The encyclopedia of technical market indicators. Dow Jones-Irwin Homewood, IL (1988)
11. Cont, R., Kukanov, A., Stoikov, S.: The price impact of order book events. Journal of financial econometrics 12(1), 47–88 (2014)
12. Cont, R., Stoikov, S., Talreja, R.: A stochastic model for order book dynamics. Operations research 58(3), 549–563 (2010)
13. Dalkey, N., Helmer, O.: An experimental application of the delphi method to the use of experts. Management science 9(3), 458–467 (1963)
14. Dixon, M.: A high-frequency trade execution model for supervised learning. High Frequency 1(1), 32–52 (2018)
15. Dixon, M.: Sequence classification of the limit order book using recurrent neural networks. Journal of computational science 24, 277–286 (2018)
16. Eschenauer, H., Koski, J., Osyczka, A.: Multicriteria design optimization: procedures and applications. Springer Science & Business Media (2012)
17. Ghosh, I., Jana, R.K., Sanyal, M.K.: Analysis of temporal pattern, causal interaction and predictive modeling of financial markets using nonlinear dynamics, econometric models and machine learning algorithms. Applied Soft Computing 82, 105553 (2019), `http://www.sciencedirect.com/science/article/pii/S1568494619303333`
18. Göçken, M., Özçalıcı, M., Boru, A., Dosdoğru, A.T.: Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Systems with Applications 44, 320–331 (2016)
19. Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J., Howison, S.D.: Limit order books. Quantitative Finance 13(11), 1709–1742 (2013)
20. Greco, S., Figueira, J., Ehrgott, M.: Multiple criteria decision analysis. Springer (2016)
21. Grgic, D., Vdovic, H., Babic, J., Podobnik, V.: Crocodileagent 2018: Robust agent-based mechanisms for power trading in competitive environments. Computer Science and Information Systems 16(1), 105–129 (2019)
22. Gupta, S.: Financial intelligence. Journal of Global Economy 13(3) (2017)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
24. Horst, U., Paulsen, M.: A law of large numbers for limit order books. Mathematics of Operations Research 42(4), 1280–1312 (2017)
25. Huang, R., Polak, T.: Lobster: Limit order book reconstruction system. Available at SSRN 1977207 (2011)
26. Kumar, D., Meghwani, S.S., Thakur, M.: Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. Journal of Computational Science 17, 1–13 (2016)
27. Kune, R., Konugurthi, P.K., Agarwal, A., Chillarige, R.R., Buyya, R.: The anatomy of big data computing. Software: Practice and Experience 46(1), 79–105 (2016)

28. Lawnik, M., Banasik, A.: Delphi method supported by forecasting software. Information 11(2), 65 (2020)
29. Lee, S., Cho, C., Hong, E.k., Yoon, B.: Forecasting mobile broadband traffic: Application of scenario analysis and delphi method. Expert Systems with Applications 44, 126–137 (2016)
30. Li, H., Wu, Y.J., Chen, Y.: Time is money: Dynamic-model-based time series data-mining for correlation analysis of commodity sales. Journal of Computational and Applied Mathematics 370, 112659 (2020), http://www.sciencedirect.com/science/article/pii/S0377042719306648
31. Linstone, H.A., Turoff, M., et al.: The delphi method. Addison-Wesley Reading, MA (1975)
32. Mi, H., Xu, L.: Optimal investment with derivatives and pricing in an incomplete market. Journal of Computational and Applied Mathematics 368, 112522 (2020), http://www.sciencedirect.com/science/article/pii/S0377042719305278
33. Mousavi, M.M., Lin, J.: The application of promethee multi-criteria decision aid in financial decision making: Case of distress prediction models evaluation. Expert Systems with Applications p. 113438 (2020)
34. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
35. Odu, G.: Weighting methods for multi-criteria decision making technique. Journal of Applied Sciences and Environmental Management 23(8), 1449–1457 (2019)
36. Osler, C.L.: Support for resistance: technical analysis and intraday exchange rates. Economic Policy Review 6(2) (2000)
37. Pal, S.S., Kar, S.: Time series forecasting for stock market prediction through data discretization by fuzzistics and rule generation by rough set theory. Mathematics and Computers in Simulation 162, 18 – 30 (2019), http://www.sciencedirect.com/science/article/pii/S0378475419300011
38. Palguna, D., Pollak, I.: Mid-price prediction in a limit order book. IEEE Journal of Selected Topics in Signal Processing 10(6), 1083–1092 (2016)
39. Pardo, R.: The evaluation and optimization of trading strategies, vol. 314. John Wiley & Sons (2011)
40. Radojičić, D., Kredatus, S., Rheinländer, T.: An approach to reconstruction of data set via supervised and unsupervised learning. In: 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI). pp. 000053–000058. IEEE (2018)
41. Radojičić, D., Kredatus, S.: The impact of stock market price fourier transform analysis on the gated recurrent unit classifier model. Expert Systems with Applications 159, 113565 (2020), http://www.sciencedirect.com/science/article/pii/S0957417420303894
42. Rowe, G., Wright, G.: Expert opinions in forecasting: the role of the delphi technique. In: Principles of forecasting, pp. 125–144. Springer (2001)
43. Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A., Li, Y.: Forecasting price movements using technical indicators: Investigating the impact of varying input window length. Neurocomputing 264, 71–88 (2017)
44. Thomas, J.A., Cover, T.: Elements of information theory. John Wiley & Sons, Inc., New York. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, MPH (2009),"Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," Journal of the Royal Society Interface 6, 187–202 (1991)
45. Vergara, J.R., Estévez, P.A.: A review of feature selection methods based on mutual information. Neural computing and applications 24(1), 175–186 (2014)
46. Withanawasam, R., Whigham, P., Crack, T.: Characterising trader manipulation in a limit-order driven market. Mathematics and Computers in Simulation 93, 43 – 52 (2013), http://www.sciencedirect.com/science/article/pii/S0378475412002212, selected Papers of the MSSANZ 19th Biennial Conference on Modelling and Simulation, Perth, Australia, 2011

47. Zhang, J., Huang, M.L., Meng, Z.P.: Visual analytics for bigdata variety and its behaviours. Computer Science and Information Systems 12(4), 1171–1191 (2015)
48. Zheng, B., Moulines, E., Abergel, F.: Price jump prediction in limit order book. arXiv preprint arXiv:1204.1381 (2012)
49. Zhou, F., Zhang, Q., Sornette, D., Jiang, L.: Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices. Applied Soft Computing 84, 105747 (2019), http://www.sciencedirect.com/science/article/pii/S1568494619305289

**Dragana Radojičić** received her PhD in Mathematics in 2020 from the Department for Financial and Actuarial Mathematics, the Vienna University of Technology, where she has been working as a Teaching assistant. She obtained her MSc degree from the Faculty of Mathematics, Technical University of Berlin in 2016 and her BSc degree from the Faculty of Mathematics, University of Belgrade in 2014. Her research is mainly focused on stochastic analysis models, financial mathematics and machine learning in finance.

**Nina Radojičić** is an Assitant Professor at the Department of Computer Science, Faculty of Mathematics, University of Belgrade. She received her PhD degree in Computer Science from the University of Belgrade, Serbia, in 2018. Her main research interests include Artificial Intelligence, Computational Intelligence, Metaheuristic Optimization algorithms, Mathematical Optimization, etc.

**Simeon Kredatus** holds a Master's degree in Computational Intelligence from Technical University in Vienna and has over 10 years of experience in the machine learning and data processing industry. Data and Artificial Intelligence applications are among his passions as well as a career choice.

*Received: June 23, 2020; Accepted: November 10, 2020.*