Expert Systems with Applications 42 (2015) 1797–1805

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using Volume Weighted Support Vector Machines with walk forward testing and feature selection for the purpose of creating stock trading strategy

Kamil Żbikowski

Institute of Computer Science, Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland

ARTICLE INFO

Article history:
Available online 22 October 2014

Keywords:
Support Vector Machines
Trend forecasting
Walk-forward testing
Stock trading

ABSTRACT

This study aims to verify whether modified Support Vector Machine classifier can be successfully applied for the purpose of forecasting short-term trends on the stock market. As the input, several technical indicators and statistical measures are selected. In order to conduct appropriate verification dedicated system with the ability to proceed walk-forward testing was designed and developed. In conjunction with modified SVM classifier, we use Fishers method for feature selection. The outcome shows that using the example weighting combined with feature selection significantly improves sample trading strategy results in terms of the overall rate of return, as well as maximum drawdown during a trading period.

1. Introduction

Researchers, analysts and investors are familiar with various methods of analyzing financial time series data. Generally, they can be divided into two categories. First one is the group of techniques that relies on the assumption that future prices of assets, their levels or just trends can be forecast by analyzing their past performance. It includes econometric and mathematical modeling as well as technical analysis. On the other hand, the second category consists of methods which analyze particular asset from the wider perspective like economic surrounding, sentiment of investors and other factors that not always can be easily quantified and measured.

For the purpose of enhancing investment performance, there are numerous examples of combining data mining algorithms with traditional investment strategies. One of the earliest works in the field is Hammerbacher and Yager (1981) where the theory of fuzzy subsets were applied to the multiple objective decision problem of stock selection. In Wong (1991) a neural network approach was described for time series forecasting. The most encouraging for the proposed model is a fact that it does not require any assumption to be made about underlying function or model that should be used. It is a significant advantage over traditional methods such as regression and Box–Jenkins model.

In many further studies, an attempt was made to increase models' performance. Refenes, Bentz, Bunn, Burgess, and Zapranis (1997) proposed a modification of least-squares cost function used in error backpropagation. Obtained estimator was biased towards the more recent observations, and the bias was determined by the decay parameter in sigmoid function. Further improvements can be made by applying feature selection procedure. In particular, Hsu (2011) used feature selection procedure based on backward simulation. Many researchers applied other techniques for attribute selection (Dai, Shao, & Lu, 2012; Lee, 2009; Hsu, 2011; Kara, Acar Boyacioglu, & Baykan, 2011; Huang, 2012; Ng, Liang, Li, Yeung, & Chan, 2014; Zhang et al., 2014). In recent years, however, applications of machine learning algorithms in the area of financial time series prediction and trading are focused on combining the knowledge gained from financial markets with the well-defined models. Chavarnakul and Enke (2008) emphasized the role of trading volume for understanding stock price movements and incorporated Adjusted Moving Average and Easy of Movement indicator into generalized regression neural network model which was utilized on past S&P500 index data. Chu, Chen, Cheng, and Huang (2009) stressed the role of transaction volume as well. They utilized a type 2 fuzzy time-series model in a combination with volume technical indicator for forecasting stocks indexes.

Recent studies tend to hybridize SVM with all those techniques – namely robust feature selection, transactional volume incorporation and technical analysis factors. Kara et al. (2011) conducted an experiment in which neural network and plain SVM models were compared for the prediction of stock price index movement with the extensive use of several technical indicators. Rosillo, Giner, and Fuente (2013) used Volatility Index and technical analysis in

E-mail address: kamil.zbikowski@ii.pw.edu.pl

http://dx.doi.org/10.1016/j.eswa.2014.10.001
0957-4174/

order to forecast weekly change in S&P 500. Dai et al. (2012) incorporated MARS splines for attribute selection that are then served as an input for Support Vector Regression model.

Existing studies showed that separately applying feature selection, transaction volume dependency and technical analysis can lead to significant improvement in terms of models performance. Therefore, combining all of them with an SVM classifier can be a natural expansion of current research. In this paper, we focus on the modified version of Support Vector Machines (SVM), Volume-Weighted SVM. Volume dependency is not provided as another predictor but as a structural modification of SVM classifier. The extension makes an assumption that incorporating volume-based weighting into penalty function can lead to significant improvement in classifier accuracy (Zbikowski, 2014). The main goal of presented experiment was to develop a trading strategy which uses VW-SVM in a combination with F-Score feature selection and several technical indicators to make the most accurate predictions about future trends of a particular stock. This study focuses on a selected subset of stocks instead of analyzing stock index. This approach allows to examine far more data points than it was done in previous studies.

The paper is organized as follows: we briefly introduce the concept of VW-SVM in Section 2. Then, in Section 3, we describe in detail the system design which was developed for the purpose of testing different variants of trading strategies. In Section 4, we analyze experiment results. Finally, in Section 5 we make some concluding remarks.

2. Volume Weighted Support Vector Machines

The primary objective of SVM algorithm is to maximize the margin $||\mathbf{w}||$ of separating hyperplanes in n-dimensional feature space. It applies the structural minimization risk principle which is described in detail (Cortes & Vapnik, 1995). In Zbikowski (2014), the concept of volume-based example weighting was introduced. The penalty function for this Volume Weighted SVM (VW-SVM) is defined as follows:

$$\min_{\mathbf{w}\in\mathcal{H},\, b\in\mathcal{R}} \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{m} C_i\xi_i, \tag{1}$$

subject to:

$$y_i(\mathbf{w}\mathbf{x_i} - b) \geqslant 1 - \xi_i, \tag{2}$$

$$\xi_i \geqslant 0, \tag{3}$$

where ξ_i are slack variables, C_i is the penalty parameter for particular input x_i and y_i is the corresponding target value. For each input vector, the penalty term is defined as follows:

$$C_i = v_i C, \tag{4}$$

where

$$v_t = \frac{\sum_{k=0}^{d} V_{t-k}}{\sum_{i=0}^{m} W_{t-i}}, \tag{5}$$

where V_{t-k} denotes the real transactional volume for the moment t with the delay of k periods and d is the length of data over which particular feature is calculated. Problem (1) can be reformulated using Lagrange multipliers α_i and μ_i as follows:

$$L(\mathbf{w},b,\alpha) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\mathbf{x_i} - b) - 1 + \xi_i) - \sum_{i=1}^{m}\mu_i\xi_i. \tag{6}$$

In order to represent (6) as a dual problem following partial derivatives need to be calculated:

$$\frac{\partial L(\mathbf{w},b,\alpha,\mu)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i}^{m}\alpha_i y_i \mathbf{x_i}, \tag{7}$$

$$\frac{\partial L(\mathbf{w},b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{m}\alpha_i y_i = 0, \tag{8}$$

$$\frac{\partial L(\mathbf{w},b,\alpha,\mu)}{\partial \xi_i} = 0 \Rightarrow C = \mu_i + \alpha_i. \tag{9}$$

Based on constraints (7)–(9) Eq. (6) the dual problem has the following form:

$$Q(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \varphi(\mathbf{x_i})\varphi(\mathbf{x_j}) \tag{10}$$

and has to be maximized according to the following conditions:

$$\sum_{i=1}^{m}\alpha_i y_i = 0, \tag{11}$$

$$\alpha_i \geqslant 0. \tag{12}$$

As it was shown in Zbikowski (2014) optimization Problem (10) has a similar form as in base SVM and can be solved with the application of Sequential Minimal Optimization algorithm.

3. System design

3.1. Architecture

In order to appropriately verify capabilities of using VW-SVM for trading on the stock market, we developed dedicated system which architecture is presented in Fig. 1. Its main objective is to make the most possible accurate predictions of future short-term trends. It consists of several modules that are responsible for specified tasks during backtesting procedure. Both, stocks quotations and trading results are stored in a relational database. Simulation

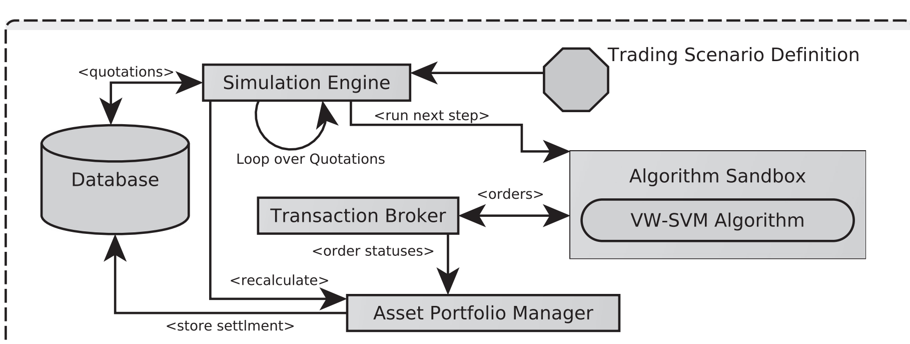

Fig. 1. Backtesting system architecture.

Engine (SE) is responsible for properly iterate over quotations of the stocks defined in the Trading Scenario Definition (TSD). It ensures that the walk-forward procedure described in detail in Ladyzynski, Zbikowski, and Grzegorzewski (2013) is conducted to avoid look-ahead bias. Basically, it goes through the entire data set with the moving window of the length m on which the model is optimized. Next it enables trading on the subsequent l samples. When the end of this subset is reached the optimization window is shifted forward by l.

SE supplies Algorithm Sandbox (AS) with quotations in order in which they occur during the walk-forward procedure. An algorithm is injected in AS where it performs necessary computations and makes a decision about opening or closing position in a particular asset. Those orders are then sent to Transaction Broker (TB) module where they are processed, stored and forwarded to analyze by next module – Assets Portfolio Manager (APM). When SE triggers the recalculation on APM, several things happen. Firstly, APM looks for new orders received from TB and open proper positions. Secondly, for each position APM recalculate current profits, margins and equity, which are held by each instance of the algorithm. This data is available to the algorithm when next iteration occurs.

3.2. Selected technical indicators

In the previous section, we have briefly described the foundations of technical analysis. In the following paragraphs, we will provide more in-depth information about several indicators that were used for the purpose of presented experiment. It should be emphasized that all of them are used commonly with corresponding trading rules which are derived from the knowledge and experience of traders. In the case of using artificial intelligence algorithms, those rules are extracted automatically by particular models and are not the subject of further analysis.

All indicators are derived from OHLC data where P_t^O, P_t^H, P_t^L and P_t^C denotes respectively open, high, low and close prices for particular moment t which spans time period $\langle t-1; t\rangle$.

3.2.1. Average True Range

Average True Range measures volatility of price changes over the specified period of time. It has simple construction that bases on an auxiliary indicator True Range which measures maximum difference between subsequent prices:

$$TR_t = \max\{P_t^H - P_t^L, P_t^H - P_{t-1}^C, P_t^L - P_{t-1}^C\}. \tag{13}$$

Average True Range is n-day arithmetic mean of TR computer over predefined period:

$$ATR_t^n = \frac{1}{n}\sum_{i=0}^{n} TR_{t-i}. \tag{14}$$

ATR is said to be superior to the standard deviation of closing prices for the purpose of quantifying the volatility as it uses also intraday prices fluctuations (Gustafson, 2001).

3.2.2. Vortex Indicator

The Vortex Indicator is a directional movement indicator. The main idea behind is that individual relations between successive quotations provide information about trends directions (Botes & Siepman, 2010). Vortex Indicator actually consists of two values $V^{(+)}$ and $V^{(-)}$ which denote positive and negative movements of trend and are defined as follows:

$$V_t^{n(+)} = \frac{\sum_{k=0}^{d}(P_{t-k}^H - P_{t-k}^L)}{\sum_{k=0}^{d} TR_{t-k}}, \tag{15}$$

$$V_t^{n(-)} = \frac{\sum_{k=0}^{d}(P_{t-k}^L - P_{t-k}^H)}{\sum_{k=0}^{d} TR_{t-k}}, \tag{16}$$

where n is the length of the period over which indicator is computed.

3.2.3. On-Balance Volume

In Davies (2004) On-Balance Volume indicator was described as a momentum indicator that relates the price change to the volume. It is defined as follows:

$$OBV_t = \begin{cases} OBV_{t-1} + V_t & \text{if } P_t > P_{t-1}, \\ OBV_{t-1} - V_t & \text{if } P_t < P_{t-1}, \\ OBV_{t-1} & \text{if } P_t = P_{t-1}, \end{cases} \tag{17}$$

where V_t is the transactional volume for the moment t. The idea behind this indicator is straightforward. When OBV rises (declines), it indicates not only that price of asset increases (decreases), but also that serious investors and large capital are involved in this movement. Moreover, even significant changes in the price that are not confirmed by the volume have no influence on the OBV value.

3.2.4. Williams Oscillator

Williams Oscillator expresses aberration of the price series from its maximal value for the given period of time. It indicates moments when the market of an asset is overbought or oversold. It is defined as follows:

$$\%R_t^n = 100\frac{MaxP_t^n - P_t}{MaxP_t^n - MinP}, \tag{18}$$

where

$$MaxP_t^n = \max\{P_t, P_{t-1}, \ldots, P_{t-n}\}$$

and

$$MinP_t^n = \min\{P_t, P_{t-1}, \ldots, P_{t-n}\}.$$

3.2.5. Relative Strength Index

Relative Strength Index is a momentum indicator which can be utilized in buying stock shares near the bottom of its trend (Faber, 1994). Te formula is defined below:

$$RSI_t^n = 100 - \frac{100}{1 + RS_t^n}, \tag{19}$$

where RS_t^n is positive to negative closes ratio defined by:

$$RS_t^n = \frac{\sum_{t=0}^{n} U_t}{\sum_{t=0}^{n} D_t}. \tag{20}$$

U_t and D_t are respectively upward and downward trend indicators with the following definition:

$$U_t = \begin{cases} P_t^C - P_{t-1}^C & \text{if } P_t^C \geqslant P_{t-1}^C, \\ 0 & \text{otherwise}, \end{cases} \tag{21}$$

$$D_t = \begin{cases} P_{t-1}^C - P_t^C & \text{if } P_t < P_{t-1}^C, \\ 0 & \text{otherwise}. \end{cases} \tag{22}$$

3.2.6. Standard deviation and rate of return

Due to the fact that presented model has the feature selection capability additional simple measures are provided. In the case they are irrelevant, we expect that they will not be included in the training stage. Common rates of investment profitability and its risk are simple n-day rate of return R_t^n and standard deviation of those quantities over the specified period of time σ_t^n.

3.3. Data preparation

In the process of determining best investment opportunities, technical indicators are usually analyzed as sequences of successive values. This is not always reflected in designing trading strategies that use data mining algorithms (Teixeira, 2010). In Wen, Yang, Song, and Jia (2010), the concept of introducing delays for each indicator was presented and experiment conducted there showed interesting results. We applied this approach for each input vector. For the particular indicator

$$f \in \{ATR_t^n, V_t^{n(+)}, V_t^{n(-)}, OBV_t, RSI_t^n, \%R_n^t, R_t^n, \sigma_t^n\}, \tag{23}$$

the training subset is defined as following matrix:

$$\mathbb{X}^f = \begin{bmatrix} x_{t-m,1} & x_{t-m-1,2} & \dots & x_{t-m-p^f,p^f} \\ x_{t-m+1,1} & x_{t-m,2} & \dots & x_{t-m-p^f+1,p^f} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t,1} & x_{t-1,2} & \dots & x_{t-p^f,p^f} \end{bmatrix}, \tag{24}$$

where m denotes the optimization window length and p^f is the delay defined for the feature f. In order to provide unified training set for each training phase, matrices $\mathbb{X}^f$ are merged in accordance to the following equation:

$$\mathbb{X} = \left[\mathbb{X}^{ATR_t^n}, \quad \mathbb{X}^{V_t^{n(+)}}, \quad \mathbb{X}^{V_t^{n(-)}}, \quad \mathbb{X}^{OBV_t}, \quad \mathbb{X}^{RSI_t^n}, \quad \mathbb{X}^{\%R_n^t}, \quad \mathbb{X}^{R_t^n}, \quad \mathbb{X}^{\sigma_t^n}\right] \tag{25}$$

As we are dealing with classification problem, target classes need to be properly defined. In our study, particular example is assigned to an upward class when d-day rate of return is greater than or equal 0 and to downward class otherwise. Therefore, for particular moment t we are trying to forecast rate of return for the period from $t+1$ to $t+d$. The corresponding target values vector is defined as:

$$\mathbb{Y} = \begin{bmatrix} y_{t-m+d} \\ y_{t-m+d+1} \\ \vdots \\ y_{t+d} \end{bmatrix}, \tag{26}$$

where ith row of $\mathbb{Y}$ is the target value assigned to ith row of training set $\mathbb{X}$ and is computed as follows:

$$y_t = \begin{cases} 1, & \text{if } R_t^d \geqslant 0, \\ -1, & \text{otherwise}, \end{cases} \tag{27}$$

where d denotes prediction horizon in days.

In each iteration, algorithm receives an information about trend predictions and depending on them makes investment decisions for the particular asset. Partial Input vector for VW-SVM model for the purpose of prediction is defined similarly to Eq. (24):

$$\mathbb{I}^f = [x_{t,1} \quad x_{t-1,2} \quad \dots \quad x_{t-p^f,p^f}]. \tag{28}$$

Accordingly to Eq. (25), the aggregated vector over all features f which is directly used for prediction is defined as follows:

$$\mathbb{I} = \left[\mathbb{I}^{ATR_t^n}, \quad \mathbb{I}^{V_t^{n(+)}}, \quad \mathbb{I}^{V_t^{n(-)}}, \quad \mathbb{I}^{OBV_t}, \quad \mathbb{I}^{RSI_t^n}, \quad \mathbb{I}^{\%R_n^t}, \quad \mathbb{I}^{R_t^n}, \quad \mathbb{I}^{\sigma_t^n}\right] \tag{29}$$

3.4. Fisher score for feature selection

Many researchers emphasize the importance of feature selection for automated stock trading algorithms (Dai et al., 2012; Lee, 2009; Hsu, 2011; Kara et al., 2011). For the purpose of presented experiment we chose Fisher score which is feature ranking method with the ranking function defined as:

$$F^f(t,j) \equiv \frac{SB_{(t,j)}^{f(U)} + SB_{(t,j)}^{f(D)}}{SW_{(t,j)}^{f(U)} + SW_{(t,j)}^{f(D)}}, \tag{30}$$

where

$$SB_{(t,j)}^{f(T)} = \left(\overline{x}_{t,j}^{f(T)} - \overline{x}_{t,j}^{f}\right)^2 \tag{31}$$

and

$$SW^{f(T)} = \frac{1}{m_{(T)} - 1} \sum_{k=1}^{m_T} \left(x_{k,j}^{f(T)} - \overline{x}_{t,j}^{f(T)}\right)^2, \tag{32}$$

where $T \in \{U, D\}$ denotes target class for particular feature which represents either *Upward* or *Downward* trend. Values $\overline{x}_{t,j}^{f(T)}, \overline{x}_{t,j}^{f}$ are the averages of features for the f indicator over all m periods with the delay of jth periods for the moment t computed respectively over target values either from T class or from all examples within the optimization window, m_T indicates the number of examples within $\mathbb{X}^f$ which belong to class T.

The numerator in Eq. (30) measures discrimination between two classes, and the denominator indicates variability within each class. According to Chen and Lin (2006), its main disadvantage is that it does not incorporate mutual information between features. In order to reduce the risk of choosing a wrong set of attributes, slightly modified procedure described in Chen and Lin (2006)

Input: Quotation list Q for dates
$t \in < startDate; endDate >$

$l \leftarrow$ trading window length
$m \leftarrow$ optimization window length
$QL \leftarrow$ empty list for incoming quotations
$CP \leftarrow NONE$
for $t \leftarrow startDate, endDate$ **do**
  $QL(t) \leftarrow Q(t)$
  for each f **do**
    $x_t^f \leftarrow$ compute f for current input
    $\mathbb{N} \leftarrow \left[\mathbb{I}^f[p^f-1], \mathbb{X}^f[m, 1:p^f-1]\right]$
    $\mathbb{I}^f \leftarrow \left[x_t^f, \mathbb{I}^f[2:p^f-1]\right]$
    $\mathbb{X}^f \leftarrow \left[\mathbb{X}^f[1:m-1;], \mathbb{N}\right]^T$
  end for
  if size(QL) $< m$ **then**
    continue
  end if
  if size(QL) mod $l = 0$ **then**
    compute weights for $\mathbb{X}$
    $\mathbb{X}' \leftarrow$ feature selection on $\mathbb{X}$
    scale $\mathbb{X}'$
    train VW-SVM classifier on $\mathbb{X}'$
  end if
  $\mathbb{I}' \leftarrow$ features from $\mathbb{I}$ present in $\mathbb{X}'$
  $p \leftarrow$ VW-SVM classifier prediction for $\mathbb{I}'$
  if $p = 1 \wedge CP <> LONG$ **then**
    open $LONG$ position
    $CP \leftarrow LONG$
  else if $p = 0 \wedge CP = LONG$ **then**
    close all open positions
  end if
end for

Fig. 2. Trading strategy algorithm based on VW-SVM classifier.

Table 1
Testing configurations. EW and FS denotes example weighting and feature selection respectively. When EW or FS is activated it is indicated by "+" or by "–" otherwise.

	Configuration value				
	p^f	m	l	EW	FS
Conf.1	1	100	10	–	–
Conf.2	1	100	10	–	+
Conf.3	1	100	10	+	–
Conf.4	1	100	10	+	+
Conf.5	1	500	50	–	–
Conf.6	1	500	50	–	+
Conf.7	1	500	50	+	–
Conf.8	1	500	50	+	+
Conf.9	5	500	50	+	+

was applied. Each training set $\mathbb{X}$ was subdivided into another training and validation sets. Then, ten random thresholds for previously computed Fisher score values for each feature were chosen. On the training subset with each threshold and feature subset new predictor was constructed. Finally, the one with the least validation error was selected for further analysis.

3.5. Scaling procedure

Both indicators *RSI* and *Williams %R* have values from the range 0–100 whereas *R* being a plain rate of return usually vary from -5% to 5% but this range is dynamic and may be different for each walk-forward iteration. According to Theodoridis and Koutroumbas (2008) and Hsu, Chang, and Lin (2010) scaling the input data increases accuracy of SVM classifier. It protects attributes with lower ranges from being dominated by those with higher values. For each feature x_i from the input vector scaling procedure to the range $[L, U]$ where $L, U \in \mathbb{R}$ and $L < U$ is applied:

$$x'^f_{tj} = L + (U - L)\frac{x^f_{tj} - m^f_j}{M^f_j - m^f_j}, \tag{33}$$

Table 2
Backtesting results for Configurations 1–4.

Stock	Conf. 1		Conf. 2		Conf. 3		Conf. 4		Ref	
	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]
ACT	6.17	−61.48	95.59	−48.76	32.16	−57.81	43.50	−64.94	295.15	−57.73
AVB	−48.75	−57.82	−48.26	−69.01	−23.73	−48.35	35.85	−56.12	219.74	−72.07
CCI	86.08	−46.49	177.07	−56.45	310.93	−25.33	336.83	−47.05	882.37	−76.16
GE	−79.92	−86.60	−75.55	−82.11	−54.76	−72.68	−35.88	−53.76	−7.67	−84.19
JPM	43.00	−56.54	−61.95	−83.02	39.20	−55.45	−60.05	−79.65	69.33	−70.11
L	−46.61	−52.66	62.86	−33.29	16.17	−42.56	93.31	−30.41	220.79	−66.01
MCD	−44.34	−45.83	37.51	−29.39	−38.29	−49.65	118.64	−23.19	432.30	−22.88
QCOM	8.85	−58.17	−15.57	−68.83	73.31	−51.96	37.93	−53.48	332.92	−48.18
TER	−10.83	−79.02	−53.66	−80.87	−52.47	−79.84	−40.58	−67.90	5.40	−90.32
USB	−42.19	−61.00	−7.45	−61.51	−32.11	−52.67	−37.07	−68.32	63.69	−76.78
ALL	−79.23	−86.17	−46.13	−78.91	−78.73	−88.13	−52.82	−81.80	52.46	−78.58
BMS	21.47	−53.57	−33.58	−58.07	16.84	−52.31	−25.85	−49.84	71.45	−52.83
EBAY	−34.81	−75.34	14.65	−71.71	12.58	−74.13	−7.21	−76.40	101.61	−82.56
HOT	−72.68	−77.02	−57.17	−79.54	−54.13	−68.03	−33.45	−68.50	212.67	−87.32
LNC	−94.43	−97.04	−76.96	−95.17	−89.02	−94.76	−37.69	−85.11	36.93	−93.27
MAC	−84.19	−95.64	−45.33	−88.40	−81.05	−95.66	62.96	−81.08	80.08	−94.39
NWL	−79.23	−88.59	−45.62	−66.01	−59.43	−79.96	−24.29	−65.12	1.40	−85.79
SYK	17.65	−63.13	71.15	−39.39	56.07	−61.22	22.23	−56.56	119.70	−59.22
TSO	136.36	−90.55	713.46	−62.45	464.85	−85.15	669.16	−71.38	1246.63	−89.45
YHOO	−63.68	−82.50	2.36	−68.74	−55.35	−76.90	−43.11	−83.89	127.29	−79.39
Mean	−23.07	−70.76	30.37	−66.08	20.15	−65.63	51.12	−63.22	228.21	−73.36

Table 3
Backtesting results for Configurations 5–8.

Stock	Conf. 5		Conf. 6		Conf. 7		Conf. 8		Ref	
	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]	R [%]	DD [%]
ACT	29.76	−56.85	233.09	−46.10	174.14	−52.75	125.76	−57.96	355.99	−43.29
AVB	10.89	−51.21	56.40	−54.16	154.07	−28.17	206.98	−26.57	84.08	−72.07
CCI	−0.47	−75.10	202.31	−77.57	69.31	−68.07	171.67	−55.54	357.20	−76.16
GE	−57.97	−74.66	−36.42	−61.42	−15.73	−38.52	10.72	−30.14	−28.52	−84.19
JPM	15.28	−55.45	−38.92	−66.49	2.89	−48.83	−29.73	−51.57	39.12	−70.11
L	23.25	−61.23	59.67	−54.10	150.63	−26.12	151.98	−26.51	111.04	−66.01
MCD	16.42	−25.08	89.59	−22.85	102.36	−21.81	137.99	−22.86	194.21	−22.88
QCOM	53.14	−37.15	187.89	−36.83	152.40	−36.77	207.79	−36.89	58.31	−48.18
TER	−13.16	−71.16	81.58	−61.95	26.48	−57.81	13.09	−58.60	3.48	−83.77
USB	−49.48	−76.19	0.64	−58.37	−3.54	−37.02	52.73	−29.41	21.71	−76.78
ALL	−41.39	−75.50	−31.31	−67.49	−18.72	−63.80	−22.71	−68.87	5.84	−78.58
BMS	33.17	−52.99	−30.85	−58.75	36.72	−52.39	−5.52	−47.67	36.06	−52.83
EBAY	−64.65	−79.96	−24.66	−64.95	−49.24	−76.81	22.42	−59.52	−7.96	−82.56
HOT	−65.96	−83.85	−51.74	−78.07	−16.93	−59.28	66.64	−43.53	48.68	−87.32
LNC	−53.92	−88.67	−36.84	−88.45	15.28	−61.86	62.72	−55.66	−2.51	−93.27
MAC	−73.72	−89.31	16.65	−52.83	65.41	−35.92	101.76	−28.31	1.52	−94.39
NWL	−28.14	−70.86	18.23	−45.09	56.54	−48.11	42.15	−48.77	21.26	−85.79
SYK	−21.72	−52.07	−32.82	−62.85	0.02	−54.89	−6.96	−43.13	55.40	−59.22
TSO	29.34	−69.31	50.74	−77.67	112.30	−65.66	451.12	−54.18	215.81	−89.45
YHOO	−52.70	−75.99	2.52	−59.67	8.92	−59.40	97.91	−41.75	−9.75	−79.39
Mean	−15.60	−66.13	35.79	−59.78	51.16	−49.70	92.93	−44.37	78.05	−72.31

where the M_i and m_i denotes respectively maximum and minimum from the training set defined as:

$$M^f = \left[\max_i(x^f_{i1}) \quad \max_i(x^f_{i2}) \quad \dots \quad \max_i(x^f_{ip^f}) \right] \tag{34}$$

and

$$m^f = \left[\min_i(x^f_{i1}) \quad \min_i(x^f_{i2}) \quad \dots \quad \min_i(x^f_{ip^f}) \right]. \tag{35}$$

3.6. Trading strategy

In order to verify whether presented VW-SVM classifier has the capability to successfully forecast future trends on the stock market, we constructed trading strategy presented in 2. It employs symbols defined in Section 3.3. Main loop iterates over all days. This simulates behavior of SE module described in Section 3.1. In every iteration new value for each indicator f is computed. It is then added at the beginning of the input vector $\mathbb{I}^f$. To preserve its constant size, last element has to be removed from $\mathbb{I}^f$. This value is then added as the first element along with the appropriate shift to the matrix $\mathbb{X}^f$. Every l quotations, classifier is retrained on the current training data stored in matrix $\mathbb{X}$. During this and next $l-1$ iteration, this model is used for making predictions about the forthcoming trend.

4. Empirical study

For the purpose of presented experiment, twenty stocks were randomly selected to evaluate the performance of the trading strategy presented in Section 3.6. Market data consisted of daily quotations from January 1, 2003 to October 21, 2013.

Several configurations were tested. All are presented in Table 1. In order to investigate the influence of introducing both, example weighting (EW) and feature selection (FS), we decided to run tests in all possible configurations using one, both or neither of them. The impact of extending lengths of optimization and trading windows was also examined. For the first four configurations, we set them to 100 and 10 days respectively and for further four experiments to 500 and 50 days. Last but not least, we also conducted a separate test for the purpose of verifying delay factor p^f

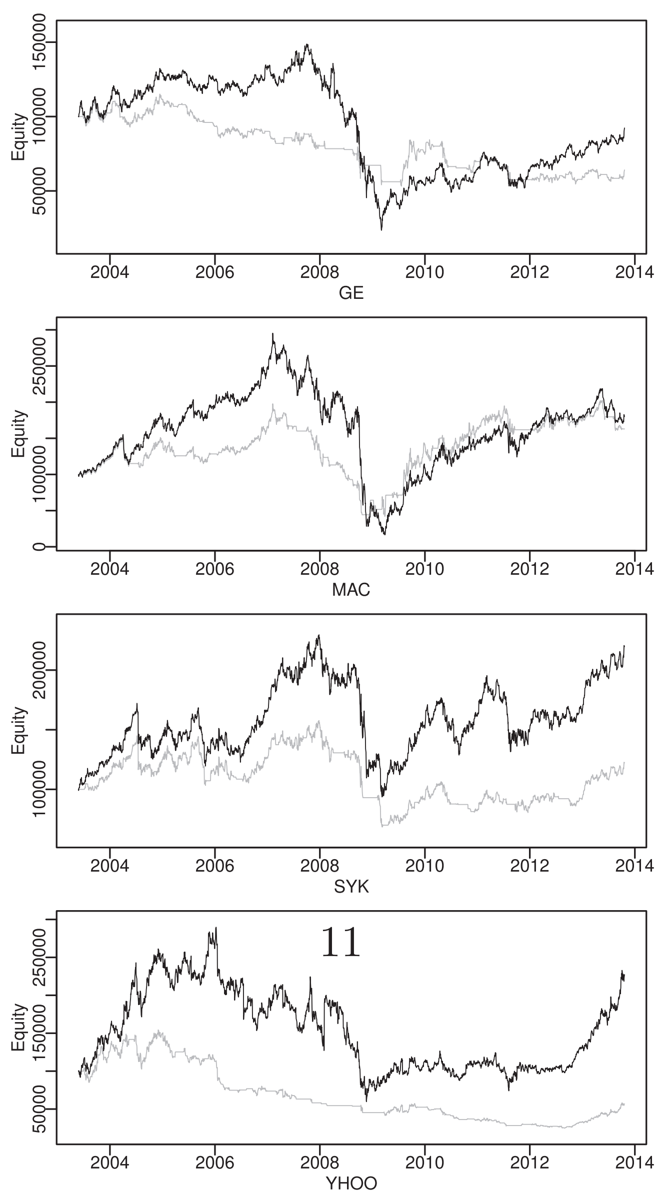

Fig. 3. Trading results sample for Conf. 4. Black – plain stock performance, gray – strategy performance.

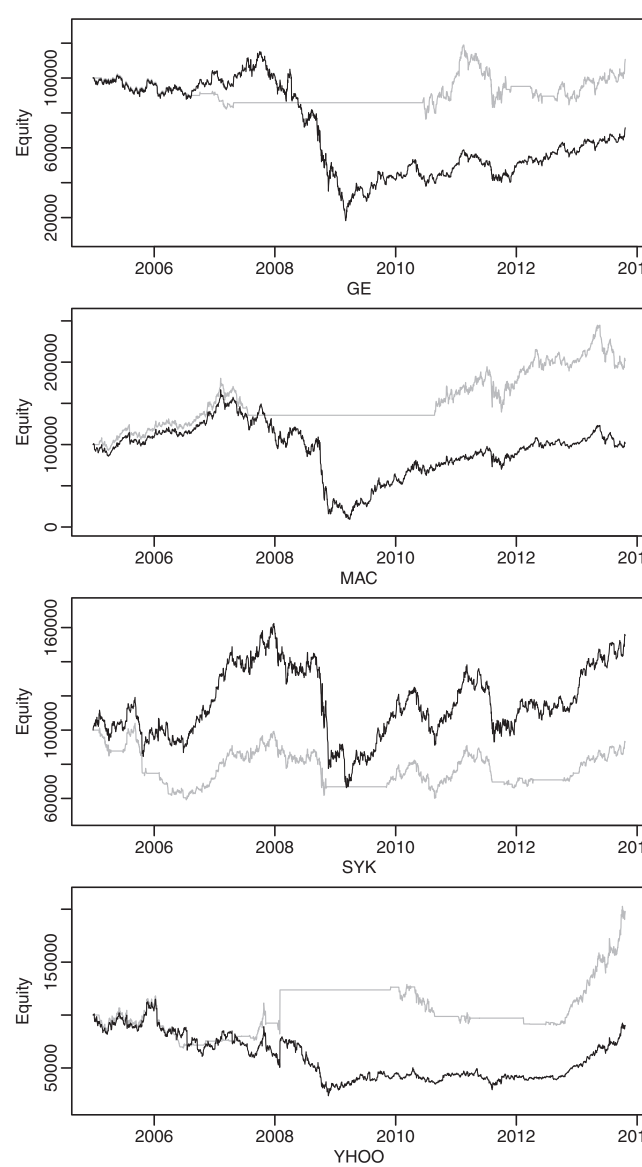

Fig. 4. Trading results sample for Conf. 8. Black – plain stock performance, gray – strategy performance.

introduced in Section 3.3. Overall, 360 different experiments were conducted on the data set which consisted of more than 50,000 quotations.

For Configurations 1–4 results are presented in Table 2, whereas for Configurations 5–8 in Table 3. Two measures were used to evaluate the performance of each of those experiments: rate of return (R) over the whole data set and maximum drawdown (DD) for a particular strategy. Maximum drawdown is often used by the investors as the risk measure. It is preferred over other measures, mainly because when it remains in the certain, a priori defined range, it indicates that the strategy is still reliable. Otherwise, it may be a serious signal of strategy deterioration, for example, as a result of a market regime switch (Magdon-Ismail, Atiya, Pratap, & Abu-Mostafa, 2003). In both tables, additional information about performance of a plain investment in the particular stock is provided. It assumes buying the stock at the beginning of a testing period and selling it at the end.

Results in Table 2 showed that plain SVM classifier (Conf. 1) is not able to forecast short-term trends. In this scenario, the trading strategy lost 23.07% on average with the drawdown of 70.76%. From the perspective of investors, this is an unacceptable result which could lead to the bankruptcy. Performance of the strategy in Configurations 2 and 3 is much better as the mean rate of return for all stocks is greater than 0. The average drawdown remains significant but is slightly better than for Configuration 1. Moreover, those two experiments showed that extensions to the basic classifier applied independently result in improved performance of the proposed strategy. Separately, adding example weighting raises the average rate of return by 43.22%, and applying feature selection with Fisher score by 53.44%. In case they are combined together (Conf. 4) overall improvement in the rate of return is 75.19%. Maximum drawdown decreases on average by 7.74%. Sample results for this case are presented in Fig. 3. This results clearly indicated that there is some potential in the proposed model, though we were not convinced that it can be used as a successful trading strategy.

After analyzing classification accuracy for Configuration 1–4 we decided to expand the length of optimization moving window to 500 days. Accordingly, we increased the size of trading moving window to 50 days. The second change was done primarily because each optimization phase was much longer as it operates on data set that was 5 times larger than before. It also preserves the same ratio between trading and optimization windows sizes. Therefore, we could investigate only the effect of enlarging the training data set solely. However, relation between lengths of both windows and its influence on strategy performance is definitely worth examining, but it is not included in this work.

Results presented in Table 3 for the Configuration 5 show that there was only a minor increase in the rate of return compared to Configuration 1. However, it still remained negative. Bearing in mind findings of influence of example weighting and feature selection, we decided to check their performance with expanded lengths of training data sets. As it can be expected, the experiment showed that both of them, separately, increase the average rate of return. Moreover, applying them together (Conf. 8) resulted in the synergy effect which can be observed in a significant increase in the rate of return and a corresponding decrease of average maximum drawdown. Sample results are presented in Fig. 4. It can be observed that an important property of the proposed model is revealed in this configuration. For three out of four samples, the strategy decided to stay out of the market at the beginning of a downward trend on the underlying asset. As a result, it obtained

Table 4
Backtesting results for Configuration 9.

Stock	Conf. 9	
	R [%]	DD [%]
ACT	423.89	−40.64
AVB	134.37	−24.00
CCI	497.46	−28.58
GE	1.01	−31.64
JPM	13.62	−41.18
L	160.72	−26.10
MCD	140.20	−22.97
QCOM	68.94	−49.14
TER	201.91	−53.01
USB	49.20	−31.80
ALL	16.59	−46.74
BMS	30.89	−28.56
EBAY	18.74	−54.02
HOT	54.19	−43.14
LNC	70.76	−57.10
MAC	48.58	−31.09
NWL	−8.98	−54.05
SYK	77.29	−42.25
TSO	1299.46	−36.90
YHOO	75.45	−42.69
Mean	168.71	−39.28

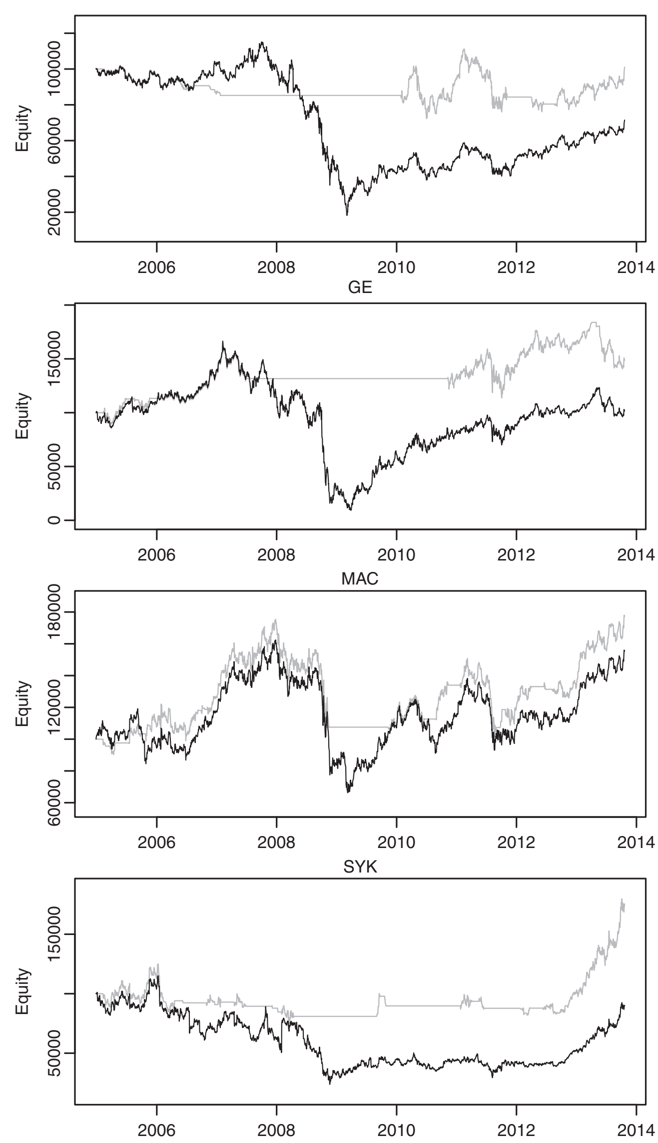

Fig. 5. Trading results sample for Conf. 9. Black – plain stock performance, gray – strategy performance.

much better results in terms of both the rate of return and maximum drawdown than buy-and-hold strategy.

Outcomes of configurations from 1 to 8 lead to the conclusion that enlarging training sample combined with feature selection and incorporating volume to SVM penalty function result in a significant performance improvement of proposed trading strategy. Due to this fact, we conduct another experiment in which we introduce features delays as it was described in Section 3.3. From all presented attributes subsets, the most promising is Configuration 9. In this configuration, we set the delay parameter p^f to 5 days for every feature f from the (23) and the rest of parameters were the same as in Configuration 8. Results for Configuration 9 contain Table 4 and corresponding samples of trading simulations are presented in Fig. 5. They show further improvement in performance of proposed strategy. Average rate of return is 168.71%. Only in one out of twenty simulations strategy did not manage to obtain a positive outcome. Overall return still remains worse than for the buy-and-hold strategy but there is a significant decrease in maximum drawdown, which is 34.08 better. Proposed trading algorithm achieves better results in terms of return to risk ratio. Comparing to previous results, in this configuration strategy even better forecasts downward trends. Presented strategy reaches considerable rate of return and keeps risk at the acceptable level.

It should be noted that applying walk-forward procedure results in using some subsets of previously seen data again in subsequent optimizations. This is quite a computationally expensive task. Presented attributes subsets should be treated as a general direction of further research and additional extensive test should be conducted in order to determine their accurate values that maximize expected rate of return.

5. Conclusion

The aim of this study was to determine whether modified SVM classifier with volume-based example weighting can be successfully applied for the purpose of predicting short-term trends on the stock market which can lead to constructing profitable trading strategy. We examined the influence of applying Fishers feature selection method as well. Further, in order to boost trading results, we extended the training vector by introducing delays of calculated technical indicators. All tests were conducted in accordance with walk-forward methodology for analyzing financial time series.

Overall, the study showed that plain SVM classifier has the limited capability for accurate trend detection and does not perform well with proposed trading strategy. However, proposed improvements such as feature selection and example weighting significantly enhance trading results. Moreover, enlarging the training data set and introducing delays for technical indicators led to results that were better than for buy-and-hold strategy both in terms of rate of return and maximum drawdown. Proposed strategy has the ability to identify downward trends and, as a consequence, to suspend trading until trend reversal occurs.

This study was inspired by Tay and Cao (2002) where modification to the loss function was made for SVM regression problems. We assumed that problem can be reformulated for the purpose of classification. Moreover, significant improvements, inspired by the knowledge gained from financial market analysis, can be made in its formulation.

Two things are new in the presented study. Firstly, a novel approach of VW-SVM, which incorporates volume information in its loss function, is applied for constructing trading strategy. Secondly, several robust methods were used together for the first time. Fishers feature selection, VW-SVM, input vector delays and technical indicators were employed in combination with walk-forward optimization procedure. This study also produces a novel architectural framework for the purpose of back-testing trading strategies.

Despite the proposed model performance, further research is needed. Further experiments should be conducted in order to verify whether other feature selection methods can improve strategy performance. One of the possible research directions is to apply Random Forests for the purpose of determining feature importance (Breiman, 2001). An example of such an application is described by Chen and Lin (2006).

Another area worth further exploration is the form of the weighting function. It should be investigated whether there exists any other superior to one presented in this study. An approach, in which, instead of plain volume information, volatility would be incorporated as the predictive factor, is one of the possible extensions.

From the perspective of applications, it is worth examining whether proposed algorithm can be deployed on other markets as well as with different financial instruments. It would also be interesting to expand training data set beyond technical analysis indicators and statistical measures and include factors from the economic surroundings of a particular stock.

Last but not least, similar family of weighting function can be proposed for other machine learning algorithms like neural networks or fuzzy sets. It would be appealing to examine and compare their capabilities to predict financial trends with VW-SVM.

References

Botes, E., & Siepman, D. (2010). The vortex indicator. *Technical Analysis of Stocks & Commodities, 28*, 20–30.

Breiman, L. (2001). Random forests. *Machine learning, 45*, 5–32.

Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications, 34*, 1004–1017.

Chen, Y. -W., & Lin, C. -J. (2006). Combining SVMs with various feature selection strategies. In *Feature extraction* (pp. 315–324). <http://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf>.

Chu, H.-H., Chen, T.-L., Cheng, C.-H., & Huang, C.-C. (2009). Fuzzy dual-factor time-series for stock index forecasting. *Expert Systems with Applications, 36*, 165–171. http://dx.doi.org/10.1016/j.eswa.2007.09.037.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Dai, W., Shao, Y. E., & Lu, C.-J. (2012). Incorporating feature selection method into support vector regression for stock index forecasting. *Neural Computing and Applications, 23*, 1551–1561.

Davies, D. (2004). Daytrading with on-balance volume. *Stocks & Commodities, 22*, 20–25.

Faber, B. (1994). The relative strength index. *Stocks & Commodities, 12*, 381–384.

Gustafson, G. (2001). Which volatility measure? *Stocks & Commodities, 19*, 46–50.

Hammerbacher, I. M., & Yager, R. R. (1981). The personalization of security selection: An application of fuzzy set theory. *Fuzzy Sets and Systems, 5*, 1–9.

Hsu, C.-M. (2011). A hybrid procedure with feature selection for resolving stock/futures price forecasting problems. *Neural Computing and Applications, 22*, 651–671.

Hsu, C. -W., Chang, C. -C., & Lin, C. -J. (2010). A practical guide to support vector classification. (pp. 1–16). <https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf>.

Huang, C.-F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing, 12*, 807–818. http://dx.doi.org/10.1016/j.asoc.2011.10.009<http://linkinghub.elsevier.com/retrieve/pii/S1568494611004030>.

Kara, Y., Acar Boyacioglu, M., & Baykan, O. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. *Expert Systems with Applications, 38*, 5311–5319.

Ladyzynski, P., Zbikowski, K., & Grzegorzewski, P. (2013). Stock trading with random forests, trend detection tests and force index volume indicators. *Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7895*, 441–452.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications, 36*, 10896–10904.

Magdon-Ismail, M., Atiya, A., Pratap, A., & Abu-Mostafa, Y. (2003). The maximum drawdown of the Brownian motion. In *2003 IEEE international conference on computational intelligence for financial engineering, proceedings*.

Ng, W. W., Liang, X.-L., Li, J., Yeung, D. S., & Chan, P. P. (2014). LG-Trader: Stock trading decision support based on feature selection by weighted localized

generalization error model. *Neurocomputing, 146*, 104–112. http://dx.doi.org/10.1016/j.neucom.2014.04.066<http://linkinghub.elsevier.com/retrieve/pii/S0925231214008765>.
Refenes, A. N., Bentz, Y., Bunn, D. W., Burgess, A. N., & Zapranis, A. D. (1997). Financial time series modelling with discounted least squares backpropagation. *Neurocomputing, 14*, 123–138.
Rosillo, R., Giner, J., & Fuente, D. (2013). The effectiveness of the combined use of VIX and support vector machines on the prediction of S&P 500. *Neural Computing and Applications, 25*, 321–332. http://dx.doi.org/10.1007/s00521-013-1487-7<http://link.springer.com/10.1007/s00521-013-1487-7>.
Tay, F. E. H., & Cao, L. J. (2002). Modified support vector machines in financial time series forecasting. *Neurocomputing, 48*, 847–861.
Teixeira, L. A. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications, 37*, 6885–6890.
Theodoridis, S., & Koutroumbas, K. (2008). *Pattern recognition*. Academic Press.
Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems with Applications, 37*, 1015–1022.
Wong, F. S. (1991). Time series forecasting using backpropagation neural networks. *Neurocomputing, 2*, 147–159.
Zbikowski, K. (2014). Time series forecasting with volume weighted support vector machines. *Communications in Computer and Information Science, 424*, 250–258.
Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E., & Liu, M. (2014). A causal feature selection algorithm for stock prediction modeling. *Neurocomputing, 142*, 48–59. http://dx.doi.org/10.1016/j.neucom.2014.01.057<http://linkinghub.elsevier.com/retrieve/pii/S0925231214005359>.