Assistant Professor Ayca Hatice ATLI, PhD (corresponding author)
E-mail: aturkan@aku.edu.tr
Department of Statistics
Afyon Kocatepe University, Turkey

PREDICTION OF STOCK MARKET INDEX MOVEMENT USING PAIRWISE CLASSIFICATION

Abstract. *The prediction of index or stock price movements is an attractive and significant research topic for academia and the business world. In recent years, many approaches based on machine learning have been developed to create an effective prediction model. A substantial part of the articles on movement prediction focuses on predicting up-and-down movements of the stock market index and stock prices. This study focuses on four kinds of price movements and proposes a prediction scheme for the emerging multi-class classification task. The proposed approach is mainly based on pairwise classification. The experiments have been conducted on three data sets, namely, the FTSE 100, KOSPI, and S&P 500 indices, using nine technical indicators as inputs. The prediction performance of the approach is compared with the performance of five traditional techniques, multilayer perceptron, support vector machine, naive Bayes, k-nearest neighbor, and regularised multinomial regression. Experimental results based on 11 years of historical data from the FTSE 100, KOSPI, and S&P 500 indices between 2010 and 2021 demonstrate the effectiveness of the proposed pairwise classification-based scheme. The proposed scheme has achieved an accuracy of more than 84%, higher than other techniques. To our knowledge, this study is the first to include the categories presented and to predict the direction of price movements based on such pairwise classification.*

Keywords*: Machine learning, Pairwise classification, Pairwise support vector machines, Stock price movement prediction*

JEL Classification: C10, C45

1. Introduction

For investors, stock market models and trading strategies that help them make high profits, make the right decisions, and produce consistent and reliable results are important. For both investors and researchers, predicting price or return trends with regard to financial instruments is both a challenging and appealing task because financial time series fluctuate, driven by unpredictable, constantly changing factors, and they have noisy and unstable nature. When the goal is to predict the

DOI: 10.24818/18423264/57.2.23.07

future direction of financial time series, the problem is handled as a classification task. The relevant literature gets deeper depending on the source of the data, the data pre-processing process, and the model used for classification.

Raw historical data of the stock, technical indicators, and macroeconomic indicators are widely used as inputs in price forecasting. The fact that they are generally available to the public and free of charge is an important advantage of transactional data, making them attractive to use. In addition to these, over recent years, the use of financial texts, news, tweets, and sentiment information has increased substantially with the recent developments in social network analysis and text processing techniques. However, the need to transform them into categorical or numerical information before they are used as input complicates the use of textual inputs. In addition, since stock trends are affected by various economic/non-economic factors such as financial conditions, and administrative issues, multi-source information-based forecasting approaches have been suggested in recent years driven by a desire to provide data from different sources (Weng et al., 2017; Zhang et al., 2018b; Chai et al., 2020).

The key requirement for data mining algorithms is an accurate dataset. Data pre-processing, which includes data reduction and data preparation tasks, is performed to meet this requirement (García et al., 2015). Regardless of the source of the data, some studies on stock market predictions have proposed models combined with feature selection techniques that summarise high-dimensional input data without loss of information, reduce the number of input variables, and help improve the performance of algorithms by extracting the most relevant features (Weng et al., 2017; Long et al., 2019). However, feature selection methods also have some limitations (García et al., 2015).

A review of the literature on stock market movement prediction in terms of the models used reveals that the field is dominated by a wide variety of artificial intelligence models. Kara et al. (2011) used the ANN and SVM models, whose inputs are technical indicators, to predict the direction of stock price index movement. Patel et al. (2015) applied the artificial neural network, support vector machine, random forest, and naive Bayes to predict the direction of movement for stocks and stock price indices. They formed the inputs by transforming the continuous values of the technical indicators into discrete values according to their characteristics. Ballings et al. (2015) compared the performance of ensemble methods with single classifier models in predicting the stock price direction. Zhang et al. (2018b) developed a stock price movement prediction framework performed simultaneously for the correlated stocks by integrating multiple sources of information. Bisoi et al. (2019) used robust kernel extreme learning machine, whose inputs are technical indicators, to predict the direction of stock price movement. Dash et al. (2019) proposed an ensemble model that ranks and selects a series of base classifiers for stock index price movement prediction, and estimates the weights of the classifiers. Long et al. (2019) proposed an end-to-end model with integrated feature extraction and classification-based prediction processes. Long et al. (2020)

proposed a deep neural network model to predict the stock price trend, considering that the correlation between the target stock and other related stocks, and the combination of transaction records and public market information are neglected in existing studies. Ismail et al. (2020) presented a hybrid method to predict the next day direction of stock price movement, combining commonly used machine learning methods with persistent homology.

In the literature, most classification algorithms are specialised in two-class classification. The issue of forecasting financial trends is also considered as a binary classification problem for upward and downward directions (Ballings et al., 2015; Bisoi et al., 2019; Dash et al., 2019). In binary classifiers, two values representing the change in the upward or downward direction are considered class labels for classifiers. To the best of our knowledge, the use of more than two targets for stock index forecasting has been addressed in a very limited number of studies (Shynkevich et al., 2017; Weng et al., 2017; Zhang et al., 2018a; Kumar and Haider, 2019; Long et al., 2019). In addition, some researchers have used the threshold value to label stock price directions (Ballings et al., 2015; Zhang et al., 2018b; Kumar and Haider, 2019; Long et al., 2019). However, the threshold value determination process is generally not based on a theoretical basis, and the threshold value is determined by the researcher's experience in threshold value application.

This article presents a new approach to the stock index movement prediction. Unlike many previous studies, comparisons of three consecutive days were used to determine the price direction rather than two consecutive days. It was taken into consideration whether prices continue to move up or down, or if the price direction changes upward or downward. In addition, by working with example pairs instead of examples, an attempt was made to predict whether the examples that make up the pairs belonged to the same class or different classes. Thus, the multi-class problem is simplified by reducing it to the two-class problem. However, working with pairs of examples will allow a significant increase in the number of training examples and will increase the computational cost. Thus, the need has arisen to use an algorithm that promises faster training, less time, and fewer memory requirements. Therefore, the online LASVM algorithm proposed by Bordes et al. (2005) and adapted to the study with example pairs by Tas (2017) has been applied to the classification problem here. Additionally, Bayesian optimisation (BO) has been integrated to adjust the hyperparameters to optimise the experimental performance of the learning algorithm.

Considering the reviewed literature, five traditional techniques, multilayer perceptron, support vector machine, naive Bayes, k-nearest neighbor, and regularised multinomial regression were used for comparison purposes in this paper. Moreover, although various classification techniques were used for comparison, they performed worse than or similar to the existing ones, and the results were omitted.

To summarise, the main contributions of this study are as follows:

- A prediction framework based on pairwise classification is proposed for the multi-class classification task defined for price movement.

- It is aimed to predict continuation and reversal movements rather than upward and downward directions in price.

- The proposed approach has been evaluated on three data sets, namely, FTSE 100, KOSPI, and S&P 500 indices. The results show that the proposed approach is better than the other techniques and can achieve a classification accuracy of 84%, 89%, and 85%, respectively, for the indices.

2. Proposed framework for price movement prediction

This study proposes a classification approach using real-time financial data to predict the direction of stock market indices. In this context, four different classes are defined depending on the closing prices of the relevant trading day and the two preceding trading days. The values 1, 2, 3 or 4 representing the daily change in the stock market index prices are taken as class labels. The class label is determined based on past closing prices using the following rules:

If $CP_i<CP_{i-1}$ and $CP_{i-1}>CP_{i-2}$, then $T_i=1$

If $CP_i<CP_{i-1}$ and $CP_{i-1}<CP_{i-2}$, then $T_i=2$

If $CP_i>CP_{i-1}$ and $CP_{i-1}>CP_{i-2}$, then $T_i=3$

If $CP_i>CP_{i-1}$ and $CP_{i-1}<CP_{i-2}$, then $T_i=4$.

Here, CP_i is the index closing price at time i and T_i is the label for the index closing price movement at time i. Table 1 presents an example of how the class labels are assigned. For instance, if the closing price of the stock market index is lower than that of time t-1 at time t and higher than that of time t-2 at time t-1, it is class 1.

Table 1. An example showing how the class labels are assigned

Date	Closing price	Direction	Class label
. .	. .	. .	. .
2010-03-16	1159.459961	xxx	xxx
2010-03-17	1166.209961	↑	xxx
2010-03-18	1165.829956	↓	1
2010-03-19	1159.900024	↓	2
2010-03-22	1165.810059	↑	4
2010-03-23	1174.170044	↑	3

↑: Upward movement according to the previous closing price, ↓: Downward movement according to the previous closing price.

The proposed approach to predicting these four classes involves pairwise classification based on two input examples instead of one. It takes into account whether two input examples belong to the same class or different classes (Brunner

et al., 2012). Thus, our goal is to learn a function to predict whether pairs' examples are of the same class.

Assume a set of training examples represented by $X = \{x_i | i = 1, 2, \dots, m\}$ such that $x_i \in \mathbb{R}^d$. For each pair of training examples, $pair = (x_i, x_j) \in \mathbb{R}^d \times \mathbb{R}^d$, names should be given according to whether the examples of the pair belong to the same class or different classes. If the examples of the (x_i, x_j) pair belong to the same class, we name this pair as a positive pair and set $y_{pair} := +1$. Conversely, if the examples of the (x_i, x_j) pair do not belong to the same class, we name this pair as a negative pair and set $y_{pair} := -1$. The purpose of the pairwise classification task is to decide whether the examples of the (x_i, x_j) pair belong to the same class. In this context, the pairwise decision function, $f: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, is defined such that $f > 0$ if the examples of the pair belong to the same class and $f < 0$ if not.

One of the basic concepts of support vector learning is the kernel function. If for every $x, x' \in X$

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_H \tag{1}$$

there is a Hilbert space H and a feature map $\varphi: X \to H$, the function $k: X \times X \to \mathbb{R}$ is called a kernel. The idea is that the data which cannot be separated linearly in a lower dimensional space may be linearly separable in a higher dimensional space. A pairwise kernel $K : (X \times X) \times (X \times X) \to \mathbb{R}$, the extension of standard kernels to pairwise classification, proposed by Ben-Hur and Noble (2005), is used for the task of mapping to the higher dimensional space. The pairwise kernel used between (x_1, x_2) and (x_3, x_4) is defined as follows:

$$K_{TPPK}\big((x_1, x_2), (x_3, x_4)\big) = K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3) \tag{2}$$

where *TPPK* stands for tensor product pairwise kernel. The experiments conducted under this study take $K(.,.)$ as the radial basis function kernel and the parameters of the kernels are adjusted with the help of BO (Sec. 3.2.2) in the training process.

The pairwise decision function obtained for each class, derived from existing examples, is defined as follows:

$$f_k(pair) = \left(w^{k*}\right)^T \varphi(pair) + b^{k*} \quad k = 1, 2, \dots, l. \tag{3}$$

Here, $w^{k*} \in \mathbb{R}^d$ is the vector of parameters and $b^{k*} \in \mathbb{R}$ is the bias term. The pairwise data is mapped to a higher-dimensional space by φ function. This decision function takes a value greater than 0 if the examples in a pair belong to the class k, and less than 0 if they do not belong to the class k. After obtaining the decision function for each class, in the test phase, the new example x is matched with randomly selected examples from each class in the training set, and test pairs, one of the examples is x, are created. The voting method (Hsu and Lin, 2002) described below is used to make the final decision on the class of the new example:

$$V_k(x) = \sum s\big(f_k(testpair_i)\big) \tag{4}$$

$$s(t) = \begin{cases} 1, & t > 0 \\ -1, & otherwise \end{cases}. \quad (5)$$

$s(t)$ matches the decision function value, $f_k(testpair_i)$, with $\{-1,1\}$ for each test pair. If f_k decides that a pair's examples are not from the class ($f_k < 0$), it contributes −1 vote by voting for $V_k(x)$; otherwise, it contributes +1 vote. After collecting the votes from all classifiers and completing the voting process, the class label of the example x is decided as follows:

$$class\ of\ x = argmax_{k=1,2,\dots,l} V_k(x) \quad (6)$$

That is, x is assigned to the class with the highest number of votes.

To train the classifier, the following optimisation problem is solved (Schölkopf and Smola, 2001):

$$\min_{w^k,b^k,\xi^k} \left\|w^k\right\|^2 + C\sum_{i=1}^{m}\xi_i^k \quad with \quad \begin{cases} \forall i & y_i f_k(pair_i) \geq 1 - \xi_i^k \\ \forall i & \xi_i^k \geq 0 \end{cases} \quad (7)$$

Here $\xi^k = \xi_1^k, \xi_2^k, \dots, \xi_m^k$ is the slack variables and C the regularisation parameter. This restricted quadratic optimisation problem is solved by adding Lagrangian multipliers. With the use of pairwise kernels, the following dual optimisation problem is reached:

$$\max_{\alpha} W(\alpha) = \sum_i \alpha_i y_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j \mathrm{K}(pair_i, pair_j) \quad with \begin{cases} \sum_i \alpha_i = 0 \\ A_i \leq \alpha_i \leq B_i \\ A_i = min(0, Cy_i) \\ B_i = max(0, Cy_i) \end{cases} \quad (8)$$

For cases where the dual objective function $W(\alpha)$ works with examples instead of example pairs, Bordes et al. (2005) proposed the LASVM algorithm that increases the $W(\alpha)$ incrementally. The proposed online LASVM algorithm to overcome the time and memory limitations of SVMs for large data sets is associated with SMO and converges to the solution of the SVM QP problem. Tas (2017) modified all the procedures performed to process an example in the online LASVM algorithm to handle pairs of examples. This adaptation includes a number of fundamental changes. The set P is defined to keep the indices of the support pairs and S to keep the indices of the corresponding examples of the pairs. That is, when a pair is added to the existing kernel expansion, the index of the pair is added to the set P, and the corresponding indices of the two examples are added to the set S. Since the P set does not need a kernel cache, it does not have one, while the S cluster has a kernel cache that keeps the kernel values between examples. Procedures adapted to process pairs of examples involve calculating the gradient of a pair, description of the τ-violating quadruple with the maximal gradient and the direction searches. The reprocessing means removing pairs from the P set and examples of pairs from the S set. Finally, two useful quantities are computed: the bias term of the decision function and the gradient δ of the most τ-violating quadruple in P. In addition, each

LASVM rework takes a new example pair and updates the existing coefficient vector by performing two types of directional searches, called process and reprocess. The processing step aims to add a new support pair to the set of potential support pairs, while the reprocessing step can remove the support pairs already existing in the existing kernel expansion.

3. Experiments

3.1. Data acquisition and preprocessing

All techniques in this study are evaluated based on data from the FTSE 100, KOSPI, and S&P 500 indices between January 4, 2010 and December 31, 2020. All three data sets were downloaded from the Yahoo! Finance website via the tidyquant R package. A single data point corresponding to a trading day contains the daily opening and closing, as well as the highest and lowest prices. The data set for the FTSE 100 index includes 2741 trading days, 2673 for the KOSPI index, and 2740 for the S&P 500 index.

Technical indicators are used as input variables to predict the direction of daily price changes in stock market indices. Following the review of many studies (Kara et al., 2011; Patel et al., 2015; Bisoi et al., 2019; Ismail et al., 2020), nine technical indicators were selected as input. The calculation formulas for these technical indicators are given in Table 2.

Table 2. Selected technical indicators and their formulas

Indicator	**Formula**
SMA	$\frac{1}{n}\sum_{i=0}^{n-1} C_{t-i}$
EMA	$(C_t - EMA_{yesterday})\frac{2}{n+1} + EMA_{yesterday}$
Momentum	$C_t - C_{t-n}$
RSI	$100 - \frac{100}{1 + \frac{\frac{\sum_{i=0}^{n-1} Up_{t-i}}{n}}{\frac{\sum_{i=0}^{n-1} Down_{t-i}}{n}}}$
Stochastic K%	$\frac{C_t - Low_n}{High_n - Low_n} \times 100$
MACD	$EMA(12) - EMA(26)$
n-day EMA of the MACD	$MACD(n)_{t-1} + \frac{2}{n+1}(DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{High_n - C_t}{High_n - Low_n} \times 100$
ROC	$\left(\frac{C_t}{C_{t-n}} - 1\right) \times 100$

SMA is simple moving average, EMA is exponential moving average, RSI is relative strength index, MACD is moving average convergence and divergence, ROC is rate of change, C_t is the current closing price, Low_n is the lowest price of last n days, $High_n$ is the highest price of last n days, Up_t means upward price change and $Down_t$ means downward price change, $DIFF_t = EMA(12)_t - EMA(26)_t$.

3.2. Experimental setup

3.2.1. Construction of comparison classifiers

The performance of the proposed method has been compared with five classifiers. Classifiers were implemented using Cortez's rminer R package (Cortez, 2020). To ensure fair use of classifiers, parameter configurations employed in Fernández-Delgado et al.'s (2014) extensive study, in which they compared 179 classifiers in 121 different data sets, were used. Data sets for experiments on benchmarking models were randomly divided into three independent parts, namely training, validation, and testing at the rates of 3/5, 1/5, and 1/5, respectively. During the training and validation stages, the parameters of the classifiers are adjusted, and each parameter combination is applied to the training and validation data sets. The combination of parameters that provides the best accuracy value is kept for the independent test phase. This procedure was repeated ten times for performance evaluations, and the averages of the classification metrics were obtained and given in Tables 3-5.

Multilayer Perceptron (MLP): The multilayer perceptron is the most popular neural network architecture in use today. MLP is a feed-forward network consisting of at least three node layers, including an input layer, a hidden layer, and an output layer. MLPs are fully connected, and each link has a numerical weight value that indicates the strength of the link. While the input data is brought to the network through the input layer, the values of all other nodes are calculated by a mathematical function that includes the node values preceding it and the weights of the connected links. Using the learning algorithm, the connection weights are adjusted by comparing the network output with the desired output.

In this study, the RSNNS R package was used to improve the neural network applied to perform the classification. A hidden layer feed-forward ANN structure was used, and the network was trained using the back-propagation algorithm. The number of neurons in the hidden layer was set with {1, 3, 5, …, 19} values. The logistic function was chosen as the activation function for the hidden and the output layer. Since nine input variables are used in the study, there are nine input nodes, and four output nodes representing four classes because the directions of price movements are classified into four classes.

Support Vector Machine (SVM): The support vector classification is based on the simple idea that if categories in training data can be separated by a linear boundary, the data can be classified depending on which side of that decision boundary it is on. In practice, since the data is generally not linearly separable, it can be ensured that

categories can be separated as linearly as possible through transformations that map the data to a higher dimensional space. Thus, the decision boundary is now a hyperplane in this high-dimensional space. For this type of conversion, the computationally cheaper kernel function is used in SVM. In this study, five different kernel functions were adopted for SVM: Gaussian kernel, linear kernel, and polynomial kernel of degree 1, 2, 3. For linear kernel SVM, parameter consists of C={2^{-2}, …, 2^{7}} penalty; parameters for Gaussian kernel SVM consist of C={2^{-5}, …, 2^{14}} penalty and γ={2^{-16}, …, 2^{8}} kernel parameter; and parameters for polynomial kernel SVMs consist of C={0.25, 0.5, 1} penalty and s={0.001, 0.01, 0.1} scale. The kernlab R package was used to carry out the experiments.

k-Nearest Neighbor (kNN): The k-nearest neighbor algorithm is an instance-based machine learning algorithm. To perform the prediction through kNN, the value of k, which indicates the number of nearest neighbors to be considered, must be determined. A distance measure is used to determine which of the k examples in the training dataset is most similar to the test example. The class of the test example is predicted to be the majority class among the nearest k examples. In this study, kNN was designed with 19 different k values, such as 1, 3, 5, …, 37, and the Euclidean distance using the class R package.

Naive Bayes: Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. First, a frequency table is created for each feature corresponding to the target, and then a likelihood table is created, and the posterior probability is calculated for each class using the naive Bayesian equation. The class with the highest posterior probability is determined as the output class. Within the scope of this study, the Gaussian Naive Bayes classifier, which is assumed to be distributed according to a Gaussian distribution of continuous values for each feature, was implemented via the e1071 R package.

Regularized Multinomial Regression (RMR): Linear logistic regression is used when the response variable has two levels, and when the variable has more than two levels, multinomial logistic regression, a generalisation of the linear logistic regression model, is commonly used. The multinomial logistic regression model is included in the study as a generalised linear model type. For the current multi-class classification problem, the lasso-penalised multinomial regression was applied within the scope of the glmnet R package.

3.2.2. Construction of the proposed framework

For the proposed method in which pairwise learning is used, the data is firstly divided randomly in ratios of 3/5, 1/5, and 1/5 as training, validation, and test. Then the following path is followed. Pairwise training data is formed from the binary selections of the examples in the training data in a ratio of 3/5. Although the number of pairs to be used for training is created and implemented in varying numbers between 10^3 and 10^5 with increments of 10^3, the number of pairs used for training is limited to 10^3, as the acceptable classification performance is achieved with the

pairwise training data consisting of 10^3 pairs. Also, the number of positive and negative pairs is balanced, i.e., 5000 positive, 5000 negative pairs are created for the pairwise training set. Pairwise validation data is produced by matching the examples in the 1/5 validation data with 201 of the training examples whose classes are known and determined to be sufficient in number.

In order to create pairwise test data, the examples in the test data are matched with 201 of the training examples whose classes are known. While creating pairwise validation and pairwise test data, we take into account the distribution of classes in training data by matching examples in 3/5 training data with examples in validation and test data. Pairwise SVM models built with pairwise training and validation data are implemented on pairwise test data. The classes of the test examples are predicted through the voting process.

Bayesian Optimization: One of the two important decisions to be made for pairwise SVM is the choice of the pairwise kernel function to be used, and the other is the choice of parameter C. The explanations for the pairwise kernel function used are already given in Section 2. In fact, hyperparameters that need to be adjusted for the pairwise SVM; the regularisation parameter C in the range of $[1, 1{,}000]$ and the parameter γ in the range of $[1e-5, 1]$ determining the spreading of the RBF kernel used in the pairwise kernel.

Bayesian optimisation (Archetti and Candelieri, 2019) used for the optimisation of these hyperparameters is explained as follows. The hyperparameter optimisation problem can be thought of as a black-box optimisation problem:

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega}\in\Omega}{argmax}\, f(\boldsymbol{\omega}). \quad (9)$$

Here $\boldsymbol{\omega} = (C, \gamma)$, Ω specifies the domain of hyperparameters and f is a black-box function that corresponds to validation accuracy. f has no simple closed-form, but the noisy value $y_i = f(\omega_i) + \varepsilon$ based on f can be observed. Here, ε indicates the observation noise, assumed to be $\varepsilon \sim N(0, \sigma^2)$. BO is a sequential model-based approach to solve the problem in (9), that is, to find $\boldsymbol{\omega}^*$, or it is a strategy for optimisation of the noisy black-box function. At a number of points in the domain, evaluations of the function are iteratively requested, with each new evaluation the approach tries to pick the next best point, so that the determined values of $\boldsymbol{\omega}$ gradually come closer to the value $\boldsymbol{\omega}^*$. The BO consists of two basic steps in which a surrogate model is created for f and new parameter points are proposed for f evaluations based on this surrogate model.

We can list the repeating steps in BO as follows:

1. Finding the most promising point based on the acquisition function.
2. Evaluating the objective function and adding the resulting new data point to the set of observations.
3. Updating the surrogate model.

This approach starts with creating a simpler surrogate model of the objective function that will be used to solve the optimisation problem. Gaussian processes are widely preferred for surrogate models. The Gaussian process model characterises the black-box function at each point as associated Gaussian random variables. A Gaussian process is fully defined by its mean function $\mu(\omega) = E[f(\omega)]$ and covariance function $k(\omega, \omega') = E\big[\big(f(\omega) - \mu(\omega)\big)\big(f(\omega') - \mu(\omega')\big)\big]$.

The covariance function is also called the kernel. The Gaussian process is initially fitted to $D_{1:n} = \{(\omega_i, y_i)\}_{i=1,2,\dots,n}$, a set of existing observations. For a predictive distribution of the function value in the new ω, the predictive mean and variance can be written as follows:

$$\mu_n(\omega) = E[f(\omega)|D_{1:n}, \omega] = k(\omega, \omega_{1:n})[K(\omega_{1:n}, \omega_{1:n}) + \sigma^2 I]^{-1} y \tag{10}$$

$$\sigma_n^2(\omega) = k(\omega, \omega) - k(\omega, \omega_{1:n})[K(\omega_{1:n}, \omega_{1:n}) + \sigma^2 I]^{-1} k(\omega_{1:n}, \omega). \tag{11}$$

Here the vector y yields the value of the function at the previous points and the covariance matrix, K, has $K_{ij} = k\big(\omega_i, \omega_j\big)$ entries where $i, j = 1, \dots, n$. $k(\omega, \omega_{1:n})$ is a n dimensional vector with $k_i = k(\omega, \omega_i)$ components. For this study, Matérn covariance, also known as the Matérn kernel (Archetti and Candelieri, 2019), has been chosen to describe the relationship between ω and ω'

$$k_{v=3/2}(\omega, \omega') = \left(1 + \frac{|\omega - \omega'|\sqrt{3}}{l}\right) e^{-\frac{|\omega - \omega'|\sqrt{3}}{l}}. \tag{12}$$

Also, l representing the characteristic length scale and σ^2 representing the noise variance were taken as 1 and 10^{-6}, respectively. The other basic component of the BO is the acquisition function calculated on the basis of the surrogate model and used to guide the selection of the next evaluation point. Integrated version of expected improvement was used for this study. This acquisition function is approached with the Markov Chain Monte Carlo approach using slice sampling. For more information on hyperparameter optimisation, we refer the reader to Jiménez and Ginebra (2017). The procedure of the proposed approach is presented in Figure 1.

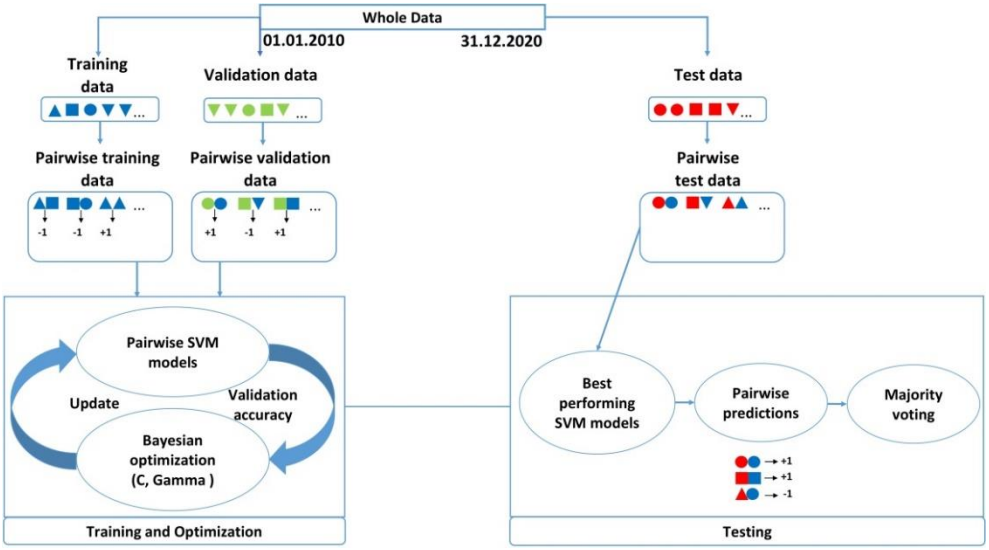

Figure 1. An overview of the proposed method (Each of the geometric shapes represents a different category.)

3.3. Experimental results

The purpose of the comparison experiments in this section is to evaluate the predictive performance of all models for the best parameter combinations. For this purpose, the evaluation criteria averages obtained from ten repetitions of the experiments of each model for each index are used. Tables 3-5 show the classification metrics obtained for the FTSE 100, KOSPI, and S&P 500 indices, respectively. The accuracy of the proposed approach for the FTSE 100 and S&P 500 indices reaches over 84% and above 89% for KOSPI outperforming other models. An evaluation of the performances of the comparison models has revealed that for all indices, SVMs are generally behind the multilayer perceptron, although they are competitive in some metrics, and that the naive Bayes and kNN models are worse-performing models. In addition, the RMR model performs similarly to the MLP model for the FTSE 100 and S&P 500 indices, while it outperforms other models for the KOSPI index.

Table 3. Classification results of FTSE 100 dataset

Method	Recall	Precision	F1-Score	Accuracy
Proposed classifier	0.8434	0.8526	0.8423	0.8440
MLP	0.5577	0.5477	0.5479	0.5606
Naive Bayes	0.3725	0.3284	0.3059	0.3770
kNN	0.4532	0.4324	0.4362	0.4575

Method	Recall	Precision	F1-Score	Accuracy
SVM_gaussian kernel	0.5394	0.5387	0.5368	0.5394
SVM_linear kernel	0.5347	0.5325	0.5322	0.5350
SVM_polynomial kernel	0.5457	0.5452	0.5440	0.5464
RMR	0.5472	0.5413	0.5404	0.5507

Table 4. Classification results of KOSPI dataset

Method	Recall	Precision	F1-Score	Accuracy
Proposed classifier	0.8870	0.9040	0.8910	0.8933
MLP	0.5365	0.5227	0.5221	0.5384
Naive Bayes	0.3817	0.3428	0.3146	0.3831
kNN	0.4385	0.4145	0.4185	0.4403
SVM_gaussian kernel	0.5261	0.5221	0.5224	0.5253
SVM_linear kernel	0.5142	0.5141	0.5107	0.5114
SVM_polynomial kernel	0.5142	0.5094	0.5108	0.5142
RMR	0.5367	0.5295	0.5274	0.5401

Table 5. Classification results of S&P 500 dataset

Method	Recall	Precision	F1-Score	Accuracy
Proposed classifier	0.8463	0.8666	0.8409	0.8485
MLP	0.5575	0.5465	0.5450	0.5580
Naive Bayes	0.3800	0.3450	0.3080	0.3810
kNN	0.4665	0.4505	0.4503	0.4687
SVM_gaussian kernel	0.5454	0.5439	0.5439	0.5446
SVM_linear kernel	0.5147	0.5238	0.5167	0.5126
SVM_polynomial kernel	0.5019	0.5043	0.5020	0.5018
RMR	0.5489	0.5493	0.5429	0.5532

A visual summary of the mean correct positives for four targets (categories) from ten replicates of experiments for all models is illustrated in Figure 2. Based on correct positives for targets 1 and 4, the approach proposed in all indices outperforms other models, and naive Bayes lags behind all models. The proposed approach also outperforms other models for targets 2 and 3. Of all the other models, the best

performing model for target 2 is MLP, and the worst performing models for target 3 are DVMs.

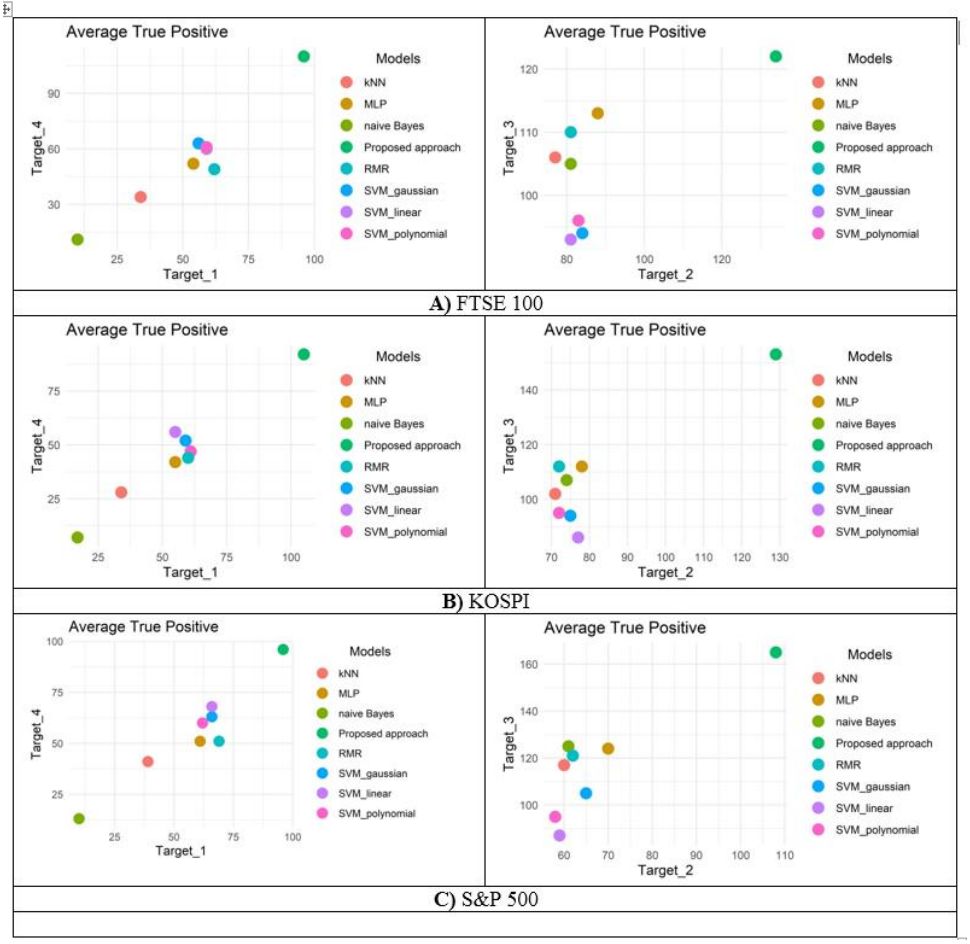

Figure 2. A visual summary of the performance of the models for four targets

4. Conclusions

The prediction of changes in stock market prices plays an important role in financial decision making and investment management. Studies on price prediction with machine learning have attracted great interest in recent years owing to the convenience of storing big data, accelerated flow of information, and increased variety in databases. This article presents the application of an online pairwise LASVM solver based on the Bayesian optimisation proposed to predict four kinds of price changes. The main advantages of the framework introduced in the study can be summarised as follows: Unlike previous studies, it is taken into consideration whether prices continue to go up or down, or fluctuate. To our knowledge, it also

contributes to the literature as it is the first study to predict the direction of price movements based on such pairwise classification.

The predictive performance of the proposed model with five models, namely MLP, SVM, naive Bayes, kNN, and RMR, was compared based on 11-year historical data retrieved from the FTSE 100, KOSPI, and S&P 500 indices. Experimental results from three important indices from America, Europe and Asia, three different continents, prove the effectiveness and applicability of the model. In predicting index movement, the proposed model reached an accuracy level of 84% or higher. The following focal points can be considered for future work:

- The proposed framework can be redesigned to predict reverse and continuation patterns or buy/sell points.
- Exact stock price prediction can be performed based on pairwise learning.
- To achieve a better performance in stock market trend prediction, using different data types such as financial texts and financial news besides technical indicators for the proposed classifier can be considered.
- The application of the classifier to multivariate time series may be considered in other areas.

REFERENCES

[1] **Archetti, F., Candelieri, A. (2019),** ***Bayesian Optimization and Data Science*****.** Springer;

[2] **Ballings, M., Van den Poel, D., Hespeels, N., Gryp, R. (2015),** ***Evaluating multiple classifiers for stock price direction prediction.*** *Expert systems with Applications*, 42(20), 7046-7056;

[3] **Ben-Hur, A., Noble, W.S. (2005),** ***Kernel methods for predicting protein–protein interactions.*** *Bioinformatics*, 21(suppl. 1), i38-i46;

[4] **Bisoi, R., Dash, P.K., Parida, A.K. (2019),** ***Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis.*** *Applied Soft Computing*, 74, 652-678;

[5] **Bordes, A., Ertekin, S., Weston, J., Bottou, L. (2005),** ***Fast kernel classifiers with online and active learning.*** *Journal of Machine Learning Research*, 6, 1579-1619;

[6] **Brunner, C., Fischer, A., Luig, K., Thies, T. (2012),** ***Pairwise support vector machines and their application to large scale problems.*** *Journal of Machine Learning Research*, 13(1), 2279-2292;

[7] **Chai, L., Xu, H., Luo, Z., Li, S. (2020),** ***A multi-source heterogeneous data analytic method for future price fluctuation prediction.*** *Neurocomputing*, 418, 11-20;

[8] **Cortez, P. (2020),** ***R package: rminer: Data Mining Classification and Regression Methods.*** Available at: https://cran.r-project.org/web/packages/rminer/index.html (accessed 16 December 2022);

[9] **Dash, R., Samal, S., Dash, R., Rautray, R. (2019),** ***An integrated TOPSIS crow search based classifier ensemble: In application to stock index price movement prediction.*** *Applied Soft Computing*, 85, 105784;

[10] **Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D. (2014),** ***Do we need hundreds of classifiers to solve real world classification problems?*** *Journal of Machine Learning Research*, 15(1), 3133-3181;

[11] **García, S., Luengo, J., Herrera, F. (2015),** ***Data Preprocessing in Data Mining*****.** Springer;

[12] **Hsu, C.W., Lin, C.J. (2002),** ***A comparison of methods for multiclass support vector machines.*** *IEEE Transactions on Neural Networks*, 13(2), 415-425;

[13] **Ismail, M.S., Noorani, M.S.M., Ismail, M., Razak, F.A., Alias, M.A. (2020),** ***Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur Stock Exchange.*** *Applied Soft Computing*, 93, 106422;

[14] **Jiménez, J., Ginebra, J. (2017*), pygpgo: Bayesian optimization for python.*** *The Journal of Open Source Software*, 2(19), 431;

[15] **Kara, Y., Boyacioglu, M.A., Baykan, Ö.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(5), 5311-5319;

[16] **Kumar, K., Haider, M.T.U. (2019),** ***Blended computation of machine learning with the recurrent neural network for intra-day stock market movement prediction using a multi-level classifier.*** *International Journal of Computers and Applications*, 43(8), 733-749;

[17] **Long, J., Chen, Z., He, W., Wu, T., Ren, J. (2020),** ***An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market.*** *Applied Soft Computing*, 91, 106205;

[18] **Long, W., Lu, Z., Cui, L. (2019),** ***Deep learning-based feature engineering for stock price movement prediction.*** *Knowledge-Based Systems*, 164, 163-173;

[19] **Patel, J., Shah, S., Thakkar, P., Kotecha, K. (2015),** ***Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques.*** *Expert Systems with Applications*, 42(1), 259-268;

[20] **Schölkopf, B., Smola, A.J. (2001),** ***Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*****.** MIT Press;

[21] **Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A., Li, Y. (2017),** ***Forecasting price movements using technical indicators: Investigating the impact of varying input window length.*** *Neurocomputing*, 264, 71-88;

[22] **Tas, E. (2017),** ***A single pairwise model for classification using online learning with kernels.*** *Hacettepe Journal of Mathematics and Statistics*, 46(3), 547-557;

[23] **Weng, B., Ahmed, M.A., Megahed, F.M. (2017),** ***Stock market one-day ahead movement prediction using disparate data sources.*** *Expert Systems with Applications*, 79, 153-163;

[24] **Zhang, J., Cui, S., Xu, Y., Li, Q., Li, T. (2018a),** ***A novel data-driven stock price trend prediction system.*** *Expert Systems with Applications*, 97: 60-69;

[25] **Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., Philip, S.Y. (2018b),** ***Improving stock market prediction via heterogeneous information fusion.*** *Knowledge-Based Systems*, 143: 236-247.