Eurasian Journal of Business and Economics 2018, 11(22), 83-103.

Generating Buy/Sell Signals for an Equity Share Using Machine Learning

Bugra ERKARTAL[*], **Linet OZDAMAR**[**]

Received: June 18, 2018 **Revised:** September 4, 2018 **Accepted:** September 15, 2018.

Abstract

This study proposes a novel model for predicting 5 days' ahead share price direction of GARAN (Garanti Bankasi A.Ş.), an equity share that is the top traded stock in BIST100, Istanbul Stock Exchange -Turkey. The first model includes global macroeconomic indicators as well as local inputs whereas the second model is focused more on local inputs. The performances of the two models are tested using Support Vector Machines (SVM), Neural Network with Back-Propagation (BPN), and Decision Tree (DT) algorithms. Though BPN and SVM have previously been used to predict BIST100 Index movement, DT has not been utilized before with this purpose. Forecasting is carried out tested for a time span of about 6 months on a rolling horizon basis, that is, algorithms are re-run weekly with updated data to generate daily buy/sell signals for the next week. A simple trading strategy is implemented based on buy/sell signals to calculate the rate of return on investment during the testing period. The results illustrate that DT having 80% prediction accuracy outperforms BPN and SVM that achieve 60% accuracy. Consequently, DT achieves a higher rate of return.

Keywords: *Machine learning algorithms, backward-propagation networks, support vector machines, decision tree learning, stock movement forecasting*

JEL Code Classification: *C53, C55, C82, C88, G12, G17*
UDC: 336.76
DOI: https://doi.org/10.17015/ejbe.2018.022.04.

[*] Yeditepe University, Istanbul, Turkey. E-mail: erkartalbugra@gmail.com
[**] Yeditepe University, Istanbul, Turkey. E-mail: ozdamar@yeditepe.edu.tr, ozdamar.linet@gmail.com

Copyright ©, 2018 Ala-Too International University.

1. Introduction

Stock market prediction is a challenging issue and draws the attention of many researchers and investors who rely on prediction systems to make important business decisions. Though substantial research exists in this field, a perfect solution is yet to be found. Predicting share prices or share price movement direction with 100% accuracy is impossible due to external factors that such as social, political, psychological, geopolitical and economic (White, 1988).

Estimating the direction of stock price movement requires processing a significant amount of historical data. Machine learning techniques (Artificial Neural Networks - ANN, Support Vector Machines –SVM, and other methods) are usually practical tools for predicting market states, because they do not rely on restrictive statistical assumptions. Furthermore, the factors affecting stock prices are usually unclear, and many anomalies exist between generally accepted inputs and the output.

This research proposes two novel models to predict the movement direction of the share price GARAN (Garanti Bankası A.Ş.) that is listed in the Istanbul Stock Exchange market BIST100. Garanti Bankası is the third largest bank in Turkey and GARAN has usually the highest trading volume in BIST100 (about 30% of the total transaction volume) and it is highly correlated with BIST100 index. The reason why we target GARAN share price direction in this study rather than the BIST100 index usually targeted in the previous literature is as follows. The average of GARAN share price's weekly rate of change is significantly higher than that of BIST100 contract price (traded on the VIOP platform, the Turkish derivatives market). Therefore, we expect to achieve a higher return on investment by targeting GARAN rather than BIST100.

The first model proposed here includes both global and local input variables whereas the second model includes only a subset of the first model's variables and it is more focused on local variables. Both proposed models are tested using three supervised learning methods: ANN with back-propagation (BPN), SVM and Decision Tree (DT) learning. In order to provide a good background of comparison, the performance of the two models are also compared against a recently proposed model (Oztekin et al., 2016) designed for generating buy/sell signals for the BIST 100 index. Öztekin et al.'s model is adapted and reproduced in this study to target GARAN share price. Other previously proposed models are not reproduced here, but their prediction accuracies are reported as cited.

Performance measures used in the comparison of results include the number of correct share price movement predictions, the number of correct positive and negative movement predictions, as well as the return on investment calculated by using a simple trading strategy. In this study, the parameters of all learning algorithms are re-optimized weekly on a rolling horizon basis in order to adapt to changing conditions. The re-optimization is carried out near the end of the trading session every Friday and the next 5 days' daily buy/sell signals are generated. All methods are tested during the first half of 2017 while validation phase covers the years 2010-2016.

This paper is organized as follows. The next section consists of a brief review of the three machine learning algorithms. Then, ANN, SVM and DT implementations in the literature are summarized. Next, the proposed models are explained. Finally, all models and methods are tested and numerical results are compared.

2. Methods

Here, we discuss the three supervised learning methods that are utilized for predicting GARAN’s price movement direction. Supervised learning is based on training a data sample from a data source with known labels.

2.1. ANN

Supervised learning techniques are utilized in feed forward hidden layer network models. ANN algorithm is inspired by the biological nervous system and it can be regarded as a whole organism consisting of a large number of computing units that interact with each other to solve a problem. Each neuron collects signals from neighboring neurons and transfers them to the next layer resulting in "excitatory" or "inhibitory" signals (Gorunescu, 2011). Hence, each neuron can be seen as a processor that makes a simple calculation, such as deciding whether to send signals to other neurons or not. Learning occurs when the effects of synapses are altered, i.e., the effect of a neuron on another neuron is changed (Tunçhan, 2008).

ANN models have three main characteristics:

i. One or more layers of hidden neurons that are not part of the network’s input or output layers enabling learning;

ii. Differentiable nonlinearity in neuronal activity;

iii. The interconnection model of the network exhibits a high degree of connectivity.

These characteristics along with learning through training help to solve difficult and diverse problems. Learning through training in a supervised ANN model is also called error back-propagation algorithm (BPN). The BPN algorithm trains the network based on the input-output samples and finds an error signal that is the difference of the calculated output and the desired output. The algorithm adjusts the synaptic weights of nodes in the network accordingly. Based on this principle, error back-propagation learning occurs in two passes:

Forward Pass: Here, an input matrix is presented to the network. Each input is connected to a node that creates a temporary output signal transferred to another node in the next layer by a transfer function. The input signal propagates forward through the network layers and emerges at the output end of the network as output signal. The output that is calculated at the output layer is compared with the desired response and the difference defines the error for that node. The synaptic weights of the network remain the same during the forward pass.

Backward Pass: The error signal originated at the output neuron of the last layer is propagated backwards through the network. This calculates the local gradient for each neuron in each layer and allows the synaptic weights of the network to undergo changes in accordance with the delta rule. The recursive computation is continued, with forward pass followed by the backward pass for each input pattern till the network converges (Quah, 2007, Majumder & Hussian, 2007, Mantri et al., 2014).

BPN identifies solutions to several linear and non-linear problems such as classification, plant control, forecasting, prediction and robotics (Mehrara et al., 2010).

2.2. SVM

SVM is first proposed by Cortes and Vapnik (1995) in the area of statistical learning theory and structural risk minimization. SVM is used in pattern recognition and regression estimation problems and are applied to the problems of dependency estimation, forecasting and constructing intelligent machines (Smola & Schölkopf, 2004).

In BPN classifiers, the weights are updated during the training phase for which the total sum of errors among the network outputs and the desired output is minimized. In contrast to this, SVM directly determines the class boundaries from training data for which the separating margins of the boundaries are maximized in feature space. That is, a maximum margin hyperplane splits the positive (labeled as 1) or negative (labeled as -1) training sample, so that the distance between the margin and the hyperplane is maximized. Support vectors are hyper planes that separate the training data by a maximal margin (An et al., 2005). If the data are linearly non-separable, a non-linear SVM classifier is applied. SVM transforms input vectors into a high dimensional feature space using a non-linear transformation Φ, and conducts a linear separation in feature space using a kernel function K(*x, y*), where {x_1 ... x_n} are input vectors and {y_1 … y_n} are their labels {-1, l}. A kernel is a function that returns the dot product $\langle \phi(x).\phi(y) \rangle$ of two vector images in feature space (Jung & Reggia, 2008). Various forms for K(*x*, *y*) exist in the literature (Nasrabadi, 2007). Some examples are provided below.

$$Linear\colon\ K(x,y) = x.y$$

$$Polynomial\colon K(x,y) = (x.y)^d\ or\ K(x,y) = (1 + x.y)^d$$

$$Gaussian\colon K(x,y) = exp\left[\frac{\left|\left|\mathrm{x} - \mathrm{y}\right|\right|^2}{2\delta^2}\right]$$

SVM is also applied to multi-class problems using “one against one” or “one-against-all” approaches (Mehrara et al., 2010).

2.3. DT

DT learning is a common data mining method where the goal is to create a model that predicts the value of a target variable based on several input variables. Each node in the tree corresponds to one of the input variables and produces two child nodes representing the class partition of that input variable. A leaf represents the class of the target variable resulting from the domains of the input variables that are fixed on the path from the root to the leaf (Figure 1).

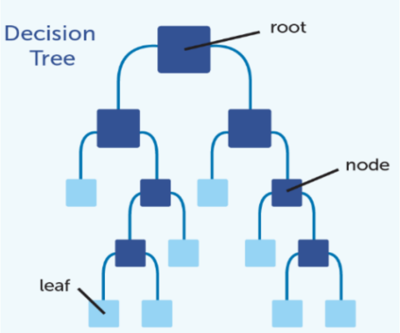

Figure 1. Illustration of a Decision Tree

In DT, different predictor functions such as majority class, or, naive Bayes classifier (Kim et al., 1998) are used to classify. Learning takes place by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset recursively as illustrated in Figure 2. The recursion ends when the subset at a node has the same value as the target variable, or when splitting no longer adds value to the predictions. This process involves top-down induction that is greedy in nature. Using this top down structure, DT partitions instances into separate classes leading to a generalized structure used to classify instances with unknown class values.

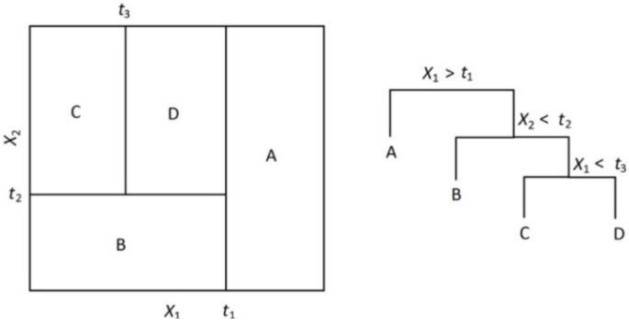

Figure 2. Recursive partitioning in a DT

Several learning algorithms are used in DT (e.g., CART, ID3, C4.5, MARS (Bao et al., 2005; Breiman et al., 1984; Quinlan, 1986/1987) that learn the classification rules from a set of training data in slightly different ways. CART, ID3, C4.5 are used to classify linguistic variables whereas MARS can deal with numerical data. The idea behind DT is to split instances into separate groups that are as homogeneous as possible. These splits are based upon different values for particular attributes. Typically, the most meaningful attribute of a dataset is selected to be the first branch of the tree. DT re-partitions the data recursively to achieve groups consisting of instances from a single class by branching from attributes that diminish in importance. The main difference between CART, ID3 and C4.5 is the way in which partitioning takes place. CART uses the Gini index to select the splitting attribute, whereas ID3 and C4.5 use a measurement of information gain and the information gain ratio. The Gini index measures the impurity of groups by calculating for each instance the probability of belonging to one class or another. CART selects the attribute that has provides the highest probability to partition the group. Information gain is a measurement developed by Shannon (2001) that quantifies the amount of information needed to classify a group of instances, and the gain ratio is an improvement upon this measurement. When information gain or the gain ratio are used to partition data, the attribute that requires the least information to classify the resulting groups is selected to split the data (Endou & Zhao, 2002).

MARS algorithm stands for Multivariate Adaptive Regression Spline. MARS uses piecewise linear basis functions consisting of reflected differences for each pair of inputs. These functions are assigned weights and transformed into a regression function. MARS decides on the number of pairwise terms in the regression function by conducting a forward pass and adding new terms consecutively to the regression function to reduce residual error. Then, a backward pass is conducted by removing one term at a time from the model in order to improve validation. The algorithm removes the term which increases the residual error the least. It continues removing terms until cross validation is satisfied.

3. Machine Learning Literature In Finance

A stock market is a public market for trading company stocks and derivatives at an agreed price that is determined by supply and demand. Companies can raise additional funds by public offerings of shares, allowing businesses to grow faster. Investors are keen on predicting share prices, especially those that are usually kept in pension fund portfolios.

There are different ways in which stock prices can be predicted. One way is to reduce the complexity by extracting best features or by feature selection (Mantri et al., 2014, Zhang et al., 2004). This approach helps in predicting stock prices with better accuracy as the complexity reduces. Investigating relationships between companies in the same sector can also help building machine learning models. Here, we provide a brief survey on ANN, SVM and DT applications in share price prediction.

We first discuss ANN and SVM applications. A review of ANN applications in stock price prediction, exchange rates and crisis prediction is presented by Li & Ma (2010).

Kimoto et al. (1990) predict the TOPIX (Tokyo Stock Exchange Price Index) by using a model with five inputs in a BPN: price vector curve, turnover, interest rate, foreign exchange rate, DOW index and a lag of the output variable whereas McCluskey (1993) targets the S&P 500 index using BPN and GA. Schierholt & Dagli (1996) predict Thailand's stock exchange index SET employing BPN while Kim et al. (1998) propose to mine reasonable trading rules by classifying the up/down fluctuant direction of the price for the Korean Stock Exchange index KOSPI 200 futures. Kutsurelis (1998) targets the monthly returns of the TOPIX using the dollar index, S&P 500 price/earnings (P/E) ratio, U.S. market performance index and liquidity as inputs as well as one period lagged TOPIX index level. Yamashita et al. (2005) utilize a multi-branch ANN targeting TOPIX. Chye & Suan (1999) target the share price of Singapore Airlines (SIA) listed in Singapore Stock Exchange. The model predicts one-week ahead closing price of the share based on inputs of historical high, low and closing prices, and traded volume. Dutt & Ghosh (1999) apply ANN to predict the 5 year US treasury; Burgess et al. (2000) describe ANN with 2 hidden layers for predicting Euro/Dollar futures using daily high, low, open and close prices. Majumder & Hussian (2007) predict the S&P CNX Nifty 50 index direction using lagged index attributes. Tilakaratne et al. (2007) predict one day ahead trading signals of the Australian All Ordinary (AORD) index using the current day's return of S&P 500 closing prices, FTSE 100 (UK) and CAC 40 (France) indices as inputs and lagged AORD index level. The authors find that feedforward ANN performs better than probabilistic networks. Charkha (2008) uses BPN and Radial Basis Function Neural Network (RBFNN) to predict the trend in a stock price with only past prices as inputs. Mandziuk & Jaruszewicz (2007) propose ANN/GA targeting the German DAX index with TOPIX and NYSE indices, EUR/USD and USD/JPY exchange rates as inputs. Here, the GA is used to find the best set of input variables for daily prediction and it is applied on a rolling horizon basis updating the inputs every 5 days. Mehrara et al. (2010) simulate BPN and ANN/GA to predict TEPIX (Iran Stock Exchange index) using technical analysis rules that are crossed over by the GA. Mantri et al. (2014) apply various Auto Regressive methods and ANN to calculate the volatility of the Indian stock market, BSE SENSEX, to discover that both approaches are equally efficient in predicting volatility. Trafalis & Ince (2000) compare SVM against BPN and RBFNN using the e-insensitive loss function and several different quadratic optimization algorithms and show that SVM is superior over BPN. Tay & Cao (2001) propose a modified SVM for financial series prediction called C-ascending SVMs. The goal of this approach is to increase the weights of most current e-insensitive errors and de-weigh more distant ones – analogous to the discounted least squares approach. Both linear and exponential weight functions are tested against several stock indices including the S&P 500. Van Gestel et al. (2001) propose SVM in a Bayesian evidence framework to predict US short term T-bill and the DAX30 index. Yang et al. (2002) propose a non-fixed and asymmetrical margin, along with the use of momentum, to improve SVM's ability to predict financial time series. The e-insensitive loss function is modified to have different upside and downside margins based on the standard deviation of the input data. The authors show that the mean absolute error of SVM's one step ahead predictions of the Hang Seng Index (HSI) and Dow Jones Industrial Average (DJIA) index are lower than those of Auto Regressive and RBFNN. Liang & Sun (2003) propose an adaptive method for modifying the RBF kernel function during the training of SVM. An optimal partitioning algorithm is used to modify the kernel, making the kernel data dependent. The authors target S&P 500 and Shanghai Stock Exchange (SSE) indices. Kim (2003) proposes a model for predicting the KOSPI index with twelve "technical indicators" as inputs. In Yang et al. (2004), a two phase SVM training method is proposed for detecting outliers in the data, thus reducing the mean absolute error in the predictions. Abraham et al. (2004) compare the one-step ahead time series prediction performances of ANN, the Levenberg-Marquardt algorithm, SVM, Takagi-Sugeno neuro-fuzzy model and the Difference Boosting Neural Network. The targets are the NASDAQ100 index and the NIFTY index. The authors show that SVM performs marginally better. Bao et al. (2004) propose SVM that uses the e- insensitive loss function and the RBF kernel function to predict five days ahead stock price direction of Haier, Inc. listed in SSE. Huang et al. (2005) propose SVM for predicting the direction of the NIKKEI 225 index based on several inputs including interest rates, consumer price index, and other market data. Bao et al. (2005) propose a Fuzzy Support Vector Machine Regression method (FSVMR) for predicting the SSE index. The FSVMR is trained during cross validation method to identify dynamic parameters. The latter approach is shown to be more effective than SVM. Jung & Reggia (2008) use SVM to predict stock price index as a time series problem and observe that SVM performs better than BPN and case based reasoning. They argue that SVM implements the structural risk minimization principle, which leads to better generalization than conventional techniques.

Here, we summarize the SVM and BPN approaches targeting the Istanbul Stock Exchange index (ISE) recently re-named as Borsa Istanbul Stock Exchange (BIST). Diler (2003) present a model that uses technical indicators targeting the ISE index and implement BPN. Altay & Satman (2005) also propose a similar BPN model. Both Diler (2003) and Altay & Satman (2005) achieve nearly 60% prediction accuracy with BPN. Yümlü et al. (2005) target the ISE index using the USD\TRY exchange rate and two bond interest rates in a modular ANN model. Kara et al. (2011) target the ISE index movement direction using another set of technical analysis indicators as inputs and compare BPN, OLS (Ordinary Least Squares) and SVM to discover that BPN outperforms SVM and OLS with 75.74% accuracy in predicting ISE's direction. OLS is the worst performer with 55% accuracy and SVM comes in second with 71.52% accuracy. The more recent model by Oztekin et al. (2016) is also reported to have a good performance with a prediction accuracy of 72% for SVM during cross-validation and 60% accuracy for BPN. This model includes daily gold price in

USD, USD/TRY and Euro/TRY exchange rates, Trend calculated for BIST100, NASDAQ Composite index, and previous BIST100 index closing price as inputs.

Finally, we discuss DT applications. Wu et al. (2006) present a stock trading method by combining a filter rule and DT where the filter rule is used to generate candidate trading points that are subsequently clustered and screened by DT. This method is used to identify the right stocks in the Taiwan and NASDAQ stock exchange markets and the right purchase timing. Wang & Chan (2006) propose a two-layer bias DT using technical indicators as inputs to create a decision rule that makes 'buy' or 'don't buy' recommendations in the stock exchange market. Sun & Li (2008) present a data mining method combining attribute-oriented induction, information gain, and DT for financial distress prediction using financial ratio attributes and an entropy-based discretization method. An empirical experiment with 35 financial ratios and 135 pairs of listed companies results in a satisfactory performance. Lu & Chen (2009) employ DT based multi-learner mining techniques to explore classification rules of information transparency levels of listed firms in Taiwan's stock market. The approach is able to provide explicit classification rules that are able to discriminate good information disclosure data from poor information disclosure data.

In this study, we borrow a couple of variables from Oztekin et al.'s model (2016) and add new variables to create an enhanced forecasting model. Unlike previous works developed for forecasting BIST100 index, we also utilize DT approach to predict GARAN stock price direction.

4. The Proposed Models

In this section, we elaborate on two models (Model A and B) proposed for predicting GARAN share price movement direction.

4.1. Model A

In the proposed Model A, we concentrate on input variables that reflect macroeconomic outputs of leader countries and their stock exchange index directions. We include the Purchasing Managers Indices (PMI) and Non-Manufacturing Indices (NMI) of countries leading the global economy, i.e., USA, the EU, China, Japan and Germany. Prominent stock exchange indices included in Model A are the DOW, DAX, SSE, NIKKEI, and S&P500. The reason behind our motivation is that more than 65% of the transactions in the BIST100 stock exchange are conducted by foreign investors, and the state of the global financial system certainly impacts GARAN share price. In Table 1, we present the correlation coefficients between Model A's input variables and GARAN closing price during the years 2010-2016 where all correlations are statistically significant. In Model A, we also include as an input the 5-year ahead expected USA inflation rate denoted here as FRED, because it is an indicator of the Federal Reserve Bank's potential tightening of monetary policy that affects liquidity and investment on stocks. In a similar fashion, the decisions made in the European Central Bank's (ECB) monetary policy meetings also impact European economies as well as the peripheral economies like Turkey. We note that Turkish stocks are affected by approaching ECB meetings even when ECB is not expected to make a change in its monetary policy. Therefore, we include an input variable that counts the number of days till the next ECB meeting, and denote it as ECBM.

Table 1. Correlation analysis between input variables in Model A and GARAN share price during 2010-2016

	Pearson Correlation Coefficient								
	German PMI	EU PMI	China PMI	German NMI	USA NMI	NIKKEI	EU NMI	Japan NMI	USA PMI
Correlation with GARAN	0.180	0.116	0.298	0.280	0.097	0.380	0.405	0.149	0.328
	SSE	DAX	DOW	S&P 500	BIST100	BOND	FRED	ECBM	
Correlation with GARAN	0.155	0.530	0.525	0.512	0.882	0.597	0.321	0.107	

As for local variables, we include the previous BIST100 index level, the trend in GARAN share price movement and the Turkish bond index. All data used in the proposed models are acquired from Bloomberg server whose address is "https://www.bloomberg.com/professional/solution/bloomberg-terminal/".

Below, we provide brief definitions of the input variables and the target.

Purchasing Managers Index (PMI): PMI is a monthly indicator of the economic health of the manufacturing sector that is calculated using new orders, inventory and production levels, supplier deliveries and the employment environment. The PMIs for USA, Germany, EU and China are included in Model A. Japan's PMI is excluded due to lack of statistically significant correlation with GARAN's price.

Non-Manufacturing Index (NMI): NMI is an index based on the surveys of a large number of non-manufacturing firms' purchasing and supply executives. NMI tracks economic data and it is a composite diffusion index based on national survey. The NMIs for Germany, USA, EU, and Japan are included in Model A.

Stock Exchange Indices: Stock exchange index levels considered as inputs in Model A are Japan's NIKKEI, China's SSE, Germany's DAX, USA's DOW and S&P500 and Turkey's BIST100.

Trend: Trend variable is used for capturing the recent history of GARAN share price movements, tracking its past up and down motions. For instance, a value of -2 for trend means there are two consecutive negative patterns in the price, and, value of 0 means the pattern has changed. A small numerical example is given in Table 2, where -1 implies the price went down that day and a value of 1 implies it went up.

Turkish Government Bond Index: The Bond index was first announced by FTSE (UK) in 2010 as a new fixed income index for Turkish Lira Government Bonds. The index contains short term Turkish government bonds, and offers domestic and international investors a tool to access the returns. The index may be followed on Bloomberg by the following abbreviation, TRGV01.

Table 2. Calculating Trend variable value for GARAN share price starting on 05.04.2010.

Date	Share Price Direction	Trend Value
05.04.2010	1	-
06.04.2010	1	1
07.04.2010	1	2
08.04.2010	-1	3
11.04.2010	-1	0
12.04.2010	1	-1
13.04.2010	1	0

St Louis Federal Reserve Bank 5 year Inflation Anticipation (FRED): The Federal Reserve Bank of St. Louis maintains a database of more than 421,000 economic time series from 81 sources. These include consumer price indices, employment rates, exchange rates, interest rates, GDP data, etc. The time series that Model A acquires as input is the 5-year ahead expected inflation rate that utilizes the 10- and 5-year fixed and inflation based treasury yields (https://fred.stlouisfed.org/series/T5YIFR). FRED changes daily based on macroeconomic conditions and market pricing.

European Central Bank Meeting Anticipation (ECBM): This input counts the remaining number of days to the next ECB monetary policy meeting. Once a meeting takes place, the counter is reset to zero. The motivation behind including this variable is the fact that 50% of Turkey's foreign trade is with the EU countries and that the majority of her foreign direct and indirect investments come from the EU. Therefore, GARAN share price shows sensitivity to the coming ECB announcements.

All inputs that take place in the model are inserted in terms of their daily rate of change except for Trend variable and ECBM. The rate of change of the local variable Bond index on day t is defined as $100*(Bond_{t-1}-Bond_{t-2}/Bond_{t-2})$ where t is the index for days. Due to time zone differences, Asia oriented rates of change are calculated by the difference between days t and t-1, whereas, the USA related rates of change are calculated by the difference between days t-2 and t-3.

As for the target GARAN's closing price, the rate of change is converted into a binary value: If rate of change is negative, the target value is -1, otherwise, the target value is set to 1.

4.2 Model B

The proposed Model B involves only a subset of the variables that take place in Model A. These are the Bond Index, Trend, BIST100, FRED and ECBM. These variables have been selected after preliminary experiments regarding their impacts on forecasting performance.

4.3. Oztekin et al. (2016) Model Implemented for GARAN

In order to compare the performance of the proposed Models A and B with the literature, we reproduce the recent model presented by Oztekin et al. (2016) that predicts the BIST100 index movement direction using BPN and SVM. Here, we denote this model the OZ_BIST model. We also adapt the model to target GARAN share price direction by calculating Trend variable for GARAN share price rather than the BIST100 index and by replacing the NASDAQ index with the S&P500 index. The latter change is made, because NASDAQ index has no correlation with GARAN, but S&P500 index has a statistically significant correlation. We denote this model OZ_GARAN. Similar to Model A and B, all model inputs except Trend are represented in the model in terms of rate of change.

In Table 3, we summarize the inputs of Models A and B proposed here as well as those of Öztekin et al.'s (2016) adapted for GARAN (OZ_GARAN) and the original Oztekin et al. model targeting BIST100 (OZ_BIST).

Table 3. Summary of the model inputs

MODEL A		**MODEL B**		**Oztekin et al. (2016) OZ_GARAN**		**Oztekin et al. (2016) OZ_BIST**	
INPUTS	**TARGET**	**INPUTS**	**TARGET**	**INPUTS**	**TARGET**	**INPUTS**	**TARGET**
USA PMI GERMANY PMI EU PMI CHINA PMI USA NMI GERMANY NMI EU NMI JAPAN NMI SSE DAX DOW S&P500 BOND INDEX TREND BIST100 FRED ECBM	GARAN	BOND INDEX TREND BIST100 FRED ECBM	GARAN	S&P500 (new) USD/XAU USD/TRY EURO/TRY TREND (GARAN) BIST100 BOND INDEX	GARAN	NASDAQ USD/XAU USD/TRY EURO/TRY TREND (BIST100) BIST100 BOND INDEX	BIST100

4.4. Trading Strategy

All methods implemented in this study use the following trading strategy in order to monetize the buy/sell signals that are generated by the models. The trading strategy requires that at the end of each trading day, we decide to buy stocks if the signal is up for the next day and there is money in the account, or, sell stocks at hand if the signal is down for the next day. Otherwise, there is no action. Here, unlike the strategy provided by Oztekin et al. (2016), short selling is not allowed, because the rules of BIST were changed by the regulating body Capital Markets Board of Turkey (SPK) such that short selling of stocks now requires special permission.

5. Numerical Experiments

5.1. Implementation Details

All methods use input and target data during the years 2007-2016. Training data covers the years 2010-2016 (first half) and training sample size is 1509. Predictions are made during the first half of 2017 between the dates July 1st – November 30th 2016. All data are retrieved from the Bloomberg server.

All methods are implemented using Matlab 2016a® software by MathWorks, Inc. on an Intel i7 dual core 6700K processor desktop with 16 GB DDR4 RAM.

The main parameters of BPN implementation are set as follows. The number of hidden layers is 50, the number of epochs is 1000; a single feed-forward pass and two feedback passes are executed. Target weights are computed using "logsig" function in order to force the system to have a binary decision on the next day's direction of price with threshold value of 0.5 used to determine the trigger signal (buy/sell). Five-fold cross-validation is executed for fine tuning the parameters using training data (2010- 2016). In each iteration, the training data is divided into five equal subsets in which one subset represents the validation set and the other four subsets represent the training data for the model. The validation data set changes consecutively during the 5 repetitions of the procedure. After all iterations are completed, the average of the weight values resulting from 5 iterations is stored.

In SVM, Kernel scale is set to sqrt(P) where, P is the number of predictors. The Kernel scale mode is selected automatically by Matlab 2016a®. The kernel functions tested by Matlab are Gaussian, linear and cubic polynomial functions. The best kernel that minimizes prediction errors is the medium Gaussian function that achieved 81% prediction accuracy during validation.

In DT, we used the MARS algorithm that is appropriate for non-binary data and the following parameters are used. The maximum number of splits is n-1, where n is the number of training data, the minimum leaf size is 1 and the minimum parent size is 10. In DT algorithm, the minimum parent size controls the depth of the tree, i.e., the minimum number of nodes on a branch. The minimum leaf size is the least number of children arising from a partitioned node.

The following assumptions are made for calculating the rate of return on investment during the testing period. At the beginning of the testing period, 1600 TRY is deposited in the bank account. Each transaction consists of 100 lots and transaction commission is 0.4%. Shares are sold and bought at closing price. Trading strategy excludes short selling.

5.2. Performance Measures

The performance of each model is measured in terms of the percentage of correct price movement direction predictions (denoted as the Hit Rate - HR), the percentage of correct downward movement predictions (denoted as True Negatives -TN) and the percentage of correct upward movement predictions (denoted as True Positives- TP).

The following equation shows how the hit rate, HR, is calculated.

$$\mathrm{HR} = \frac{\text{correct number of predicted labels}}{\text{total number of testing labels}} \mathrm{x}100 \qquad (1)$$

5.3. Results

Each model is tested using SVM, BPN and DT methods. The results of Models A and B are presented in Table 4. We provide the percentage of correctly predicted labels, i.e., the hit rate, HR, the percentage of correct upward movement predictions, TP, and correct downward movement predictions, TN, as well as the percentile rate of return on investment and computation times. We also report the rate of return on investment for the Buy and Hold and fixed income investment options.

Table 4. Performance of Model A and Model B

	MODEL A						MODEL B					
	HR %	TP %	TN %	Validation HR	Return Rate %	Comput. Time (CPUsec.)	HR %	TP %	TN %	Validation HR	Return Rate %	Comput. Time (CPUsec.)
SVM	59	28	31	65	10	9	54	26	28	69	2.2	7
BPN	56	25	31	59	13	1011	54	24	30	61	2.3	354
DT	77	33	40	72	38	8	79	36	43	82	45	4
Buy & Hold					25.5						25.5	
Fixed Income					4.23						4.23	

In Table 4, we observe that the best prediction accuracy is obtained by DT in both Models A and B. Model B validation results are slightly better than Model A's for all three methods. However**,** SVM and BPN's predictions are slightly better in Model A than in Model B affecting the return on investment. In DT's predictions and earnings, the reverse is true, Model B's results including validation, prediction and earnings are all better than those of Model A. Since DT is a tree based method and since Model B has a lower number of variables than Model A, DT performs better in the simpler model. DT's prediction accuracy is significantly superior to those of BPN and SVM in both models during both validation and prediction phases.

In Table 4, we observe that in both proposed models and in all methods, TN accuracy is higher than TP accuracy. This is important, because when TN accuracy is low, the investor actually loses money. On the other hand, when TP accuracy is lower, the investor loses a profit opportunity. Since DT has the best prediction accuracy, its rate of return on investment is also significantly higher than that of BPN and SVM. If the investor buys and holds GARAN share during the testing period, he/she makes a profit of 25.5%, however, DT's return on investment in both Model A and Model B outperforms the Buy & Hold option by about 13% and 20%, respectively. BPN's and SVM's return on investment rates are very poor due to their lower prediction accuracies despite the fact that their performance were not too inferior in validation phase. Even the fixed income return during the testing period provides better profit than BPN and SVM in Model B, but not in Model A. Using Model A, SVM and BPN can return almost triple the fixed income profit.

An important point to note is that due to the extensive training phase, BPN requires a much higher amount of computation time (about 16 CPU minutes for Model A) than SVM and DT (less than 10 CPU seconds). These programs are re-run every Friday near the end of the session to make an investment decision for Monday. Therefore, BPN's longer computation time may become a liability.

In Table 5, we present the results of Öztekin et al.'s OZ_BIST and OZ_GARAN models. Unfortunately, OZ_GARAN model performs poorly despite the fact that the NASDAQ index that is not correlated with GARAN has been replaced by the S&P500 index. The prediction accuracy of OZ_GARAN model is about 50%. Unlike Models A and B, the OZ_GARAN model is better in predicting positive movement (TP) than negative movement (TN). The same observation holds for the OZ_BIST model. This feature is also mentioned by Oztekin et al. (2016). Another surprising outcome observed in Table 5 is that BPN is not able to classify output labels when OZ_BIST model is utilized (all labels are decided as an upward movement). We could not find a good reason for this outcome. DT performs very poorly when used with OZ_BIST model. However, SVM's performance with OZ_BIST model is almost as good as our best result in Table 4, DT/Model B. Yet, the rate of return on investment achieved by SVM/OZ_BIST is lower than that of DT/Model B and comparable with our second best result, DT/Model A. The reason might be that GARAN share price's up/down movement range is higher than that of BIST 100 index. As mentioned previously, having better negative movement prediction accuracy, TN, also reduces losses.

In order to check if the differences of results between the three best performing combinations DT/Model A, DT/Model B and SVM/OZ_BIST are statistically significant, we ran McNemar’s difference test (McNemar, 1947) on binary validation and prediction outputs and paired t-test for the rate of return on investment outputs. In Table 6, first we compare Model A and Model B. We find that there is no statistically significant difference between the predictions obtained by DT using Models A and B, but the differences between validation and earnings results are statistically significant. Next, we check for the differences between the best two performers, DT/Model B and SVM/OZ_BIST. Though no statistically significant difference exists between both validation and prediction results, the difference in the return on investments is statistically significant.

Table 5. Performance of OZ_BIST and OZ_GARAN models

	OZ_GARAN Model						OZ_BIST Model					
	HR %	TP %	TN %	Validation HR %	Return Rate %	Comput. Time (CPUsec.)	HR %	TP %	TN %	Validation HR%	Return Rate %	Comput. Time (CPUsec.)
SVM	50	31	19	65	4.2	7	78	44	34	81	38.9	8
BPN	44	34	10	53	2.1	667	FAILED TO CLASSIFY					1155
DT	53	35	18	70	1.7	6	36	26	10	62	-6.2	7
Buy and Hold					25.5						25.5	
Fixed Income Return					4.23						4.23	

Table 6. Paired difference tests on validation, prediction and earnings data resulting from DT/Model A and DT/Model B; and; DT/Model B and SVM/OZ_BIST

Validation Data		Prediction Data		Earnings Data	
McNemar Test on DT Results Model A vs. Model B	Exact Significance (2-sided)	McNemar Test on DT Results Model A vs. Model B	Exact Significance (2-sided)	Paired t-test on DT Results Model A vs. Model B	Exact Significance (2-sided)
	0.028*		0.720		0.028*
Validation Data		**Prediction Data**		**Earnings Data**	
McNemar Test on DT/Model B vs. SVM/ OZ_BIST	Exact Significance (2-sided)	McNemar Test on DT/Model B vs. SVM/ OZ_BIST	Exact Significance (2-sided)	Paired t-test on DT/Model B vs. SVM/ OZ_BIST	Exact Significance (2-sided)
	0.719		0.706		0.029*

In Table 7, we present previously reported prediction accuracy results targeting BIST100 index direction by several authors mentioned in the literature survey: Kara et al. (2011), Diler (2003), Altay & Satman (2005) and Oztekin et al. (2016). For comparison purposes, in Table 7, we also summarize the prediction accuracies illustrated in Tables 4 and 5 (Models A, B, OZ_BIST and OZ_GARAN). Among Öztekin et al.'s cited results (2016) ANFIS (an adaptive neuro-fuzzy inference method that uses fuzzy if-then rules) is their least successful method. Kara et al.'s model (2011) performs worst under OLS. Among these previously cited results, the best prediction accuracy is obtained by Kara et al. (2011) using BPN with almost 76% hit rate.

Table 7. Summary of prediction accuracies (HR) of previous and proposed models. (FTC: Failed to Classify)

TARGET: BIST100 Direction										
Kara et al. (2011)			Oztekin et al. (2016)			OZ_BIST			Diler (2003)	Altay & Satman (2005)
OLS	BPN	SVM	ANN	ANFIS	SVM	SVM	BPN	DT	BPN	BPN
55%	76%	72%	60%	52%	72%	78%	FTC	36%	61%	58%
TARGET: GARAN Share Price Direction										
MODEL A			MODEL B			OZ_GARAN				
SVM	BPN	DT	SVM	BPN	DT	SVM	BPN	DT		
59%	56%	77%	54%	54%	79%	50%	44%	53%		

SVM results of both Kara et al. (2011) and Oztekin et al. (2016) models are cited to be second best with 72% accuracy. As for the models OZ_BIST and OZ_GARAN that we reproduced here, SVM/OZ_BIST provides a better than cited prediction accuracy of 78%, however, the return on investment is about 38% and comparable to that of proposed DT/Model A, but inferior to that of DT/Model B. BPN/OZ_BIST did not return any result and it failed to classify the direction of movement of BIST100. DT/OZ_BIST resulted in a very inferior prediction accuracy of 36%. As illustrated in Table 7, OZ_GARAN model also results in an inferior prediction accuracy. When we compare the proposed DT/Model B results with SVM/OZ_BIST results (the two best achievements among all cited/proposed methods and models), we observe that though their prediction accuracies are similar, the return on investment is higher for Model B/DT and the difference is statistically significant.

6. Conclusion

The motivation of this study is to achieve more profitable rate of return on stock exchange investments. As mentioned previously, Turkish researchers have focused on the BIST100 index of the Istanbul Stock Exchange and the learning algorithms they used are mainly BPN and SVM. We noted that the stock GARAN has a higher weekly rate of change as compared to BIST100 index while being a very liquid stock with top trading volume in BIST100. Therefore, in this study we targeted GARAN stock rather than the BIST100.

We also included the learning algorithm DT/MARS in this study and discovered that its prediction rate is higher than those of SVM and BPN.

The two novel models developed here generate 5 days ahead buy/sell signals for GARAN equity share listed in Borsa Istanbul Stock Exchange, BIST100. GARAN is the top traded BIST100 share in terms of volume. The first model (Model A) focuses on global macroeconomic indicators and leading stock exchange indices as well as local market attributes and USA and EU central banks' potential monetary policy actions. The second model (Model B) includes only a subset of the first model's input variables and focuses on central bank influence and local market attributes, thereby reducing the number of variables to five. Two novel variables take place in Models A and B. These are the "*St Louis Federal Reserve Bank 5 year Inflation Anticipation (FRED)*" and the "*the days remaining to the ECB monetary policy meeting dates*". These two variables have a significant impact on bank stocks, because they reflect future foreign loan availability. Turkey's economy thrives when foreign loans are abundant. To our knowledge, these variables have not been included in previous models and they are the main contributions of Model A and B.

Both models are tested using three machine learning methods, namely, BPN, SVM and DT. Among them, the best prediction accuracy and return on investment is obtained by DT algorithm. DT works more efficiently on Model B, but there seems to be no statistically significant difference between the two models' prediction accuracies. However, their returns on investment differ significantly.

We also tested and compared the previous application of Oztekin et al. both on the BIST100 index and GARAN. We show that Oztekin et al.'s model does not work well for GARAN stock, its rate of return is very low. It works for BIST100 index, but its rate of return is lower than our proposed Model B/DT algorithm. In fact, the proposed Model B/DT algorithm outperforms all previously proposed models that target the BIST100 index.

References

Abraham, A., Philip, N. S., & Saratchandran, P. (2004). Modeling chaotic behavior of stock indices using intelligent paradigms.Neurali Parallel & Scientific Computations 11(1&2), 143-160.

Altay, E., & Satman, M. H. (2005). Stock market forecasting: artificial neural network and linear regression comparison in an emerging market. *Journal of Financial Management & Analysis*, *18*(2), 18.

An, J., Wang, Z.-O., Yang, Q., & Ma, Z. (2005). A SVM function approximation approach with good performances in interpolation and extrapolation. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* 3, 1648–1653).

Bao, Y.-K., Liu, Z.-T., Guo, L., & Wang, W. (2005). Forecasting stock composite index by fuzzy support vector machines regression. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* (Vol. 6, pp. 3535–3540).

Bao, Y., Lu, Y., & Zhang, J. (2004). Forecasting stock price by SVMs regression. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications* 295–303 https://doi.org/10.1007/978-3-540-30106-6_30

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression* *trees*. Boca Raton: Chapman & Hall/CRC press.

Burgess, A. N., & others. (2000). *A computational methodology for modelling the dynamics of statistical arbitrage*. Phd Thesis: University of London.

Charkha, P. R. (2008). Stock price prediction and trend prediction using neural networks. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on* 592–594. https://doi.org/10.1109/ICETET.2008.223

Koh, H. C., & Tan, S. S. (1999). A neural network approach to the prediction of going concern status. *Accounting and Business Research*, *29*(3), 211-216 https://doi.org/10.1080/00014788.1999.9729581

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Diler, A. I. (2003). Predicting direction of ISE national-100 index with back propagation trained neural network. *Journal of Istanbul Stock Exchange*, *7*(25–26), 65–81.

Dutt, S. D., & Ghosh, D. (1999). A note on the foreign exchange market efficiency hypothesis. *Journal of Economics and Finance*, *23*(2), 157–161. https://doi.org/10.1007/BF02745949

Endou, T., & Zhao, Q. (2002). Generation of comprehensible decision trees through evolution of training data. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* 2, 1221–1225).

Gorunescu, F. (2011). *Data Mining: Concepts, models and techniques* (Vol. 12). Berlin: Springer Science & Business Media. https://doi.org/10.1007/978-3-642-19721-5

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

Jung, J.-Y., & Reggia, J. A. (2008). The automated design of artificial neural networks using evolutionary computation. In *Success in Evolutionary Computation* (pp. 19–41). Springer. https://doi.org/10.1007/978-3-540-76286-7_2

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.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2

Kim, K., Han, I., & Chandler, J. S. (1998). Extracting trading rules from the multiple classifiers and technical indicators in stock market. In *Proceedings of KMIS'98 International Conference*.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 1–6). https://doi.org/10.1109/IJCNN.1990.137535

Kutsurelis, J. E. (1998). *Forecasting financial markets using neural networks: An analysis of methods and accuracy*. (Doctoral dissertation, Monterey, California. Naval Postgraduate School).

Li, Y., & Ma, W. (2010). Applications of artificial neural networks in financial economics: a survey. In *Computational Intelligence and Design (ISCID), 2010 International Symposium on* (Vol. 1, pp. 211–214).

Liang, Y., & Sun, Y. (2003). An improved method of support vector machine and its applications to financial time series forecasting. *Progress in Natural Science*, *13*(9), 696–700.

https://doi.org/10.1080/10020070312331344260

Lu, C.-L., & Chen, T.-C. (2009). A study of applying data mining approach to the information disclosure for Taiwan's stock market investors. *Expert Systems with Applications*, *36*(2), 3536–3542. https://doi.org/10.1016/j.eswa.2008.02.007

Majumder, M., & Hussian, M. D. A. (2007). Forecasting of Indian stock market index using artificial neural network. *Information Science*, 98–105.

Mandziuk, J., & Jaruszewicz, M. (2007). Neuro-evolutionary approach to stock market prediction. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on* (pp. 2515–2520).

Mantri, J. K., Gahan, P., & Nayak, B. B. (2014). Artificial neural networks--an application to stock market volatility. *Soft-Computing in Capital Market: Research and Methods of Computational Finance for Measuring Risk of Financial Instruments*, *179*.

McCluskey, P. C. (1993). Feedforward and recurrent neural networks and genetic programs for stock market and time series forecasting. *Master's Thesis, Brown University*.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12 (2), 153–157. https://doi.org/10.1007/BF02295996

Mehrara, M., Moeini, A., Ahrari, M., & Ghafari, A. (2010). Using technical analysis with neural network for forecasting stock price index in Tehran stock exchange. *Middle Eastern Finance and Economics*, *6*(6), 50–61.

Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of Electronic Imaging*, *16*(4), 49901. https://doi.org/10.1117/1.2819119

Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, *253*(3), 697–710. https://doi.org/10.1016/j.ejor.2016.02.056

Quah, T.-S. (2007). Using Neural Network for DJIA Stock Selection. *Engineering Letters*, *15*(1).

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/BF00116251

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, *27*(3), 221–234. https://doi.org/10.1016/S0020-7373(87)80053-6

Schierholt, K., & Dagli, C. H. (1996). Stock market prediction using different neural network classification architectures. In *Computational Intelligence for Financial Engineering, 1996., Proceedings of the IEEE/IAFE 1996 Conference,* 72–78. https://doi.org/10.1109/CIFER.1996.501826

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55. https://doi.org/10.1145/584091.584093

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Sun, J., & Li, H. (2008). Data mining method for listed companies' financial distress prediction. *Knowledge-Based Systems*, *21*(1), 1–5. https://doi.org/10.1016/j.knosys.2006.11.003

Tay, F. E. H., & Cao, L. J. (2001). Improved financial time series forecasting by combining support vector machines with self-organizing feature map. *Intelligent Data Analysis*, *5*(4), 339–354.

Tilakaratne, C. D., Morris, S. A., Mammadov, M. A., & Hurst, C. P. (2007). Predicting stock market index trading signals using neural networks. In *Proceedings of the 14th Annual Global Finance Conference (GFC'07)*, 171–179.

Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on* Vol. 6, 348–353).

Tunçhan, C. (2008). *Modern sezgisel teknikler ve uygulamaları*. Istanbul: Papatya Yayınevi.

Van Gestel, T., Suykens, J. A. K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., Vandewalle, J. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks*, *12*(4), 809–821. https://doi.org/10.1109/72.935093

Wang, J.-L., & Chan, S.-H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, *30*(4), 605–611. https://doi.org/10.1016/j.eswa.2005.07.006

White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. Proceedings of the *IEEE International conference on Neural Networks*. 2, 451-458. https://doi.org/10.1109/ICNN.1988.23959

Wu, M.-C., Lin, S.-Y., & Lin, C.-H. (2006). An effective application of decision tree to stock trading. *Expert Systems with Applications*, *31*(2), 270–274. https://doi.org/10.1016/j.eswa.2005.09.026

Yamashita, T., Hirasawa, K., & Hu, J. (2005). Application of multi-branch neural networks to stock market prediction. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on* 4, 2544–2548 https://doi.org/10.1109/IJCNN.2005.1556303

Yang, H., Huang, K., Chan, L., King, I., & Lyu, M. R. (2004). Outliers treatment in support vector regression for financial time series prediction. In *International Conference on Neural Information Processing*, 1260–1265. https://doi.org/10.1007/978-3-540-30499-9_196

Yang, H., King, I., & Chan, L. (2002). Non-fixed and asymmetrical margin approach to stock market prediction using support vector regression. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on* 3, 1398–1402). https://doi.org/10.1109/ICONIP.2002.1202850

Yümlü, S., Gürgen, F. S., & Okay, N. (2005). A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Pattern Recognition Letters*, *26*(13), 2093–2103.

Zhang, D., Jiang, Q., & Li, X. (2004). Application of neural networks in financial data mining. *International Journal of Computational Intelligence*, *1*(2), 116–119.