Expert Systems With Applications 208 (2022) 118120

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Optimizing filter rule parameters with genetic algorithm and stock selection with artificial neural networks for an improved trading: The case of Borsa Istanbul

Mehmet Ozcalici [a,*,2], Mete Bumin [b,1,3]

[a] Kilis 7 Aralik University, Faculty of Economics and Administrative Sciences, Kilis, Turkey
[b] Banking Regulation and Supervision Agency of Turkey (BRSA), Istanbul, Turkey

ARTICLE INFO

Keywords:
Filter rule
Genetic algorithm
Artificial neural network

ABSTRACT

Filter rule along with other trading algorithms is used to identify potentially profitable trading points in stock markets. In this study, the scope of the filter rule has been expanded to include different moving average types. The filter rule parameters that will provide the highest return for each of the stocks listed in Borsa Istanbul have been optimized by using genetic algorithm. A number of 357 stocks traded in Borsa Istanbul is included in the dataset of the study between 06-07-2012 and 31-03-2020 period. To improve the poor performance in out-of-sample sets of optimal rules, the stock selection process was performed by means of artificial neural networks. The artificial neural network model predicts the performance of the stock in the test set by using the performance values in the training set. Results indicate that the returns of the selected stocks are significantly higher than the returns of the buy and hold strategy. Parameter optimization of filter rule with genetic algorithms and stock selection with the artificial neural networks can be used as a decision support system for investors, where they can make a profit above the market return. When only the genetic algorithm results are taken into account, it can be stated that Borsa Istanbul is a weak form efficient market. However, selecting the stocks with the assistance of artificial neural networks made it possible to obtain excess returns over the market.

1. Introduction

According to the efficient market hypothesis (EMH), current market prices of listed stocks reflect all available historical information such that investors should not be able to outperform on the market (buy-and-hold return) consistently by trading on past information (Almujamed, 2019). Since its introduction to the literature by Fama in the 1970s (Fama, 1970), many empirical studies have been conducted on EMH to determine whether it is valid in the markets of different countries.

Filter rule, which is developed by Alexander (Alexander, 1961) is used to identify potentially profitable trading points and it uses historical price data to generate trading signals. The filter rule is well known for testing the efficiency of stock markets such as Qatar Stock Exchange (Almujamed, 2019), European markets (Fifield et al., 2005), Taiwan Stock Exchange (Huang, 1995), Nigerian stock market (Olowe, 1999), Arab stocks (Dbouk et al., 2014), and Australian share market (Pereira, 2002). There are few studies examining the performance of technical trading rules in Turkish stock markets with the help of filter rules (Metghalchi et al., 2021). Researchers have applied various econometric models instead of filter rule in order to test the efficiency of Borsa Istanbul (Akgun & Sahin, 2017; Aliyev, 2019; Bektur & Aydın, 2019; Bulut, 2016; Coşkun & Seven, 2016; Gozbasi et al., 2014; Hailu & Vural, 2020; Karademir & Evci, 2020; Saymeh, 2013; Sevim et al., 1997; Yucel, 2016) and varied results have been reported.

This study investigates Borsa Istanbul in terms of whether filter trading strategies and the weak form efficiency of the EMH influence investment decisions in Borsa Istanbul-listed companies.

In the original filter rule developed by (Alexander, 1961), trading signals are determined by considering historical price data. In this study, the scope of the filter rule has been expanded by enabling the use of

* Corresponding author.
E-mail addresses: mozcalici@kilis.edu.tr (M. Ozcalici), mbumin@bddk.org.tr (M. Bumin).
[1] The views expressed in this paper are solely of the author, and do not necessarily reflect the views of the Banking Regulation and Supervision Agency of Turkey.
[2] ORCID: 0000-0003-0384-6872.
[3] ORCID: 0000-0002-4740-0007.

https://doi.org/10.1016/j.eswa.2022.118120
Received 30 October 2021; Received in revised form 11 May 2022; Accepted 7 July 2022
Available online 12 July 2022
0957-4174/

different moving average types in the generation of buy and sell signs, which is explained as follows. A buy signal is generated when the stock's *p* daily moving average (simple, linear, square-root, exponential, triangular moving average or closing price) appreciates *x*% value, and a buy signal when the stock's *q* daily moving average (simple, linear, square-root, exponential, triangular moving average or closing price) depreciates *y*% value. Each parameter that is used in the filter rule has been optimized for each stock and each period separately by genetic algorithm. The trading algorithm used in the fitness function of the genetic algorithm has been modified to prevent selling at a lower price than the purchase price, which will result in a net loss from the transaction.

The overall dataset is divided into three partitions. Each partition is further divided as training set and testing set. Parameters of the filter rules for each stock are optimized with genetic algorithm by using the training set. Out-of-sample performance of the optimized rules is measured by testing set. It was determined that optimized filter rules were insufficient to show higher performance than the buy-and-hold strategy. To improve the performance of the optimal filter rule against the buy-and-hold strategy, the stock selection process is performed with artificial neural networks. Artificial neural networks are used to increase the performance of process in various fields (Chiang et al., 2019; López-González et al., 2020; Mújica-Vargas, 2021; Rubio, 2021; J. de J. Rubio et al., 2021, 2022). The stock selection procedure with artificial neural networks is as follows. By using the training data of the previous partition, a model is created and is trained to forecast its performance in the test set. Inputs of the neural network model are the performance indicators of training set. The output of the neural network model is the trading performance in the testing set. To test whether the performance of the selected stocks is higher than the buy and hold strategy, the nonparametric Wilcoxon rank sign test was performed and it was determined that the performance of the stocks was higher than the buy and hold return.

The study is formed as follows. In Section 2, the studies that apply the filter rule and the studies that test the efficiency of Borsa Istanbul are summarized. The working principles of the filter rule, genetic algorithm and artificial neural networks are summarized in Section 3. In Section 4, the data set used in the study is introduced, and the results of the analysis are given. Finally, the results are interpreted and suggestions for further studies are presented in Section 5.

2. Literature review

Filter rule is used to test the efficiency of the markets and trading performance of the stocks in the financial systems of various countries by employing different numbers of datasets. The studies attempted to examine the stock markets by filter rule are summarized as follows.

Fama and Blume interpreted the results of (Alexander, 1961) by using different numbers of observations for stocks (Fama & Blume, 1966). Although the start dates of the observations vary from stock to stock, the ending date is September 26, 1962 and 24 different filters ranging from 0.5 % to 50 % were simulated. The authors report that trading with filter rule, in general, does not perform better than a simple buy-and-hold policy. Sweeney re-examined the fourteen stocks that were previously used in (Fama & Blume, 1966) and found that they show statistically significant profits for 1970–1982 (Sweeney, 1988). Corrado and Lee used a dataset that included 120 stocks from SP500 and Dow Jones Index, which had returns between January 1969 and December 1989 (Corrado & Lee, 1992). They examined the ability of filter rules to predict variation in expected daily returns. Their results indicated that the filter rule can predict significant variation in expected daily returns. Huang examined the trading performance of the filter rule relative to the buy-and-hold strategy on the Taiwan Stock Exchange (Huang, 1995). The dataset used in the study covers the period 1971–1993 and 24 filter sizes ranging from 0.005 to 0.50 were simulated daily. It is reported that medium size filters significantly outperform the passive buy-and-hold strategy even after considering the transaction costs.

Olowe tested the weak form of efficiency of the Nigerian stock market (Olowe, 1999) by using a dataset including end-of-the-month quoted stock prices of 59 randomly selected stocks listed throughout the period January 1981 to December 1992 on the Nigerian Stock Exchange. It is reported that technical analysis (filter rule) based on historical price information appears to be valueless in Nigeria. Pereira outlined how a genetic algorithm can be used to optimize technical trading rules in the Australian share market (Pereira, 2002). The dataset used in the study consists of the daily closing of all ordinary accumulation indexes and the daily 90-day Reserve Bank of Australia bill dealer rate over the period 1982 to 1997, consisting of 4065 observations. The study reported that generalized moving average rules were able to outperform the benchmark strategy over the out-of-sample test period.

Lin and colleagues applied genetic algorithm to optimize the filter size, which was tested on stocks of the Australia Stock Exchange (Lin et al., 2004). When computing subsequent highs/lows, the previous *p*-day price is considered. It is found that when the price over *p*-days is discorded before the time point, a signal may produce. Fifield and colleagues analyzed the predictive ability of technical trading rules for 11 European countries nearly ten years from January 1991 to December 2000 (Fifield et al., 2005) by applying 10 different filters ranging from 1 % to 30 %. It is reported that some of the European stock markets are informationally inefficient. Yen and Hsu tested the performance of the 1560 filter rule on ten futures markets (Yen & Hsu, 2010). Five of them are financial contracts while the remaining five are commodity contracts. Various training testing division is performed on the dataset. The study reported that trading rules are not generally able to beat the buy-and-hold strategy.

Dbouk and colleagues examined the profitability of technical analysis for a cross-section of individual Arab stocks (Dbouk et al., 2014), with a data set consist of closing prices of twenty most actively traded firms belonging to Qatar, Saudi, Dubai, and Kuwait stock markets between 2004 and 2012. Authors employed a 5 % filter rule in the study and they located that scant evidence of statistically buying returns are found for some technical trading rules and risk adjusted returns significantly weakens the evidence in favor of profitability. Almujamed examined the predictability of nine filter rule and tested the validity of the weak form of the efficient market hypothesis for the Qatar Stock Exchange (QSE) (Almujamed, 2019). The dataset used in the study covers the period of 2004–2017 for 44 companies. The author reported that it is possible to obtain excess returns relative to the buy-and-hold strategy by using the filter rule and QSE is not a weak form efficient market.

In the studies summarized above, the filter rules are examined in the stock markets of different countries from ready-made rules or by testing all values in a certain range. Ready-made rules may not provide high returns in every country's stock market. Moreover, different filter rules in different time periods can provide higher returns. For this reason, instead of ready-made rules, filter rules that provide high returns in different periods for each stock should be determined.

Various studies have been performed to investigate the efficiency of the markets in Turkish financial system by using datasets from the companies listed in Borsa Istanbul Stock Exchange. The studies attempted to examine the efficiency of the stocks in Borsa Istanbul are summarized as follows.

Sevim and colleagues tested the weak form of efficiency of Borsa Istanbul (Sevim et al., 1997). Monthly data for the period of 1988–2002 is examined with operational strategy in the overreaction hypothesis. They report that Borsa Istanbul is not weak form efficient market. Saymeh empirically tested the weak form efficiency of Borsa Istanbul (Saymeh, 2013). Monthly index for the period of 2000–2011 and various econometrical models/tools are used in the study. Mixed results were obtained from these tests and it is concluded that there was not enough evidence to consider Borsa Istanbul as weak form efficient market. Gozbasi and colleagues utilized various econometric models to re-

Procedure : Genetic Algorithms
begin
  $t \leftarrow 0$;
  initialize $P(t)$;
  evaluate $P(t)$;
  while (not termination condition) **do**
  begin
    recombine $P(t)$ to yield $C(t)$;
    evaluate $C(t)$;
    select $P(t+1)$ from $P(t)$ and $C(t)$;
    $t \leftarrow t+1$;
  end
end

Fig. 1. A general structure of the Genetic Algorithms (Gen & Cheng, 1999).

examine the Turkish stock market efficiency (Gozbasi et al., 2014). Daily data for the four indexes regarding the period 2002–2012 is used. It is concluded that Borsa Istanbul is a weak form efficient market.

Bulut utilized monthly data of BIST100 index between 2003 and 2015 (Bulut, 2016). According to the results obtained from various econometric models, it is concluded that Borsa Istanbul is a weak form efficient market. Coskun and Seven used monthly index data (namely BIST100) from 1993 to 2015 in order to test the weak form efficiency (Coşkun & Seven, 2016). The whole dataset is divided into various sub-periods and various econometric models have been used in the study. It is reported that they have not found enough evidence to support the weak form efficiency of Borsa Istanbul. Yucel employed daily closing prices of four indices from Borsa Istanbul for the period of 2010–2015 (Yucel, 2016). Various econometric models have been used to test the weak form efficiency of Borsa Istanbul. It is reported that Borsa Istanbul is a weak form efficient market.

Akgun and Sahin utilized daily closing prices of four indices from Borsa Istanbul from 2010 to 2017 (Akgun & Sahin, 2017). Econometric test results indicate that there is not enough evidence to support the weak form efficiency of Borsa Istanbul. Aliyev utilized 10 years' weekly data to test the weak form efficiency of Borsa Istanbul (Aliyev, 2019). A smooth transition autoregressive type linear model is used in the study. The author reports that Borsa Istanbul returns are predictable at the given period since the proposed model outperformed the random walk model. Hailu and Vural assessed the weak form efficiency of BIST100, BISTBANK indices as well as banks listed in the BIST30 index by applying various econometric models (Hailu & Vural, 2020). Weekly adjusted closing prices of the stocks between 2010 and 2019 are used. The study concluded that it is difficult to give a general conclusion regarding the efficiency of the BIST banking sector as a weak form efficient market.

Bektur and Aydin tested the efficient market hypothesis for Borsa Istanbul (Bektur & Aydın, 2019). Various indexes with daily closing prices for the period of 2000–2017 were used in the study. Results from various econometric analyses indicate that Borsa Istanbul is a weak form efficient market. Karademir and Evci assessed Borsa Istanbul as weak form efficiency with unit root tests based on the monthly closing data of selected 27 indices in Borsa Istanbul from 2008 to 2018 (Karademir & Evci, 2020). It is reported that no strong evidence is found to support the weak form efficiency of the market.

Methgalci and colleagues investigated the relationship between trading rules and excess returns (Metghalchi et al., 2021). They applied trading rules to the FTSE Turkish all-cap and small-cap indexes from September 23, 2003 to August 9, 2019 to determine the rules that produce net excess returns over the buy-and-hold strategy. It is reported that some of the rules can be able to produce excess returns over the buy-and-hold strategy.

The common point of the studies performed for Borsa Istanbul, which are summarized above, is the application of various econometric analyzes to test the efficiency of Borsa Istanbul. Only in a few studies the performance of technical trading rules in Turkish stock markets as stated by Metghalci and colleagues (Metghalchi et al., 2021) is examined.

3. Methodology

In this study, a model is developed for the generation of buy-sell signals, which are expected to be profitable for stocks. The main model, which will generate trading points, is the expanded-filter rule model. The parameters of the filter rule that will provide the highest return are optimized with the help of genetic algorithm. The performance of the optimized expanded-filter rule is boosted with a stock selection technique based on artificial neural networks. Finally, artificial neural networks are used to select the stocks that are expected to show higher performance in the test set. In this section, these methodologies are summarized.

3.1. *Expanded-filter rule*

The filter rule is initially developed by Alexander (Alexander, 1961). The working principle of the filter rule is simple and can be explained as follows. Stock is bought when it appreciates by a certain percentage (x %) from its most recent trough (bottom or low value) and sold when it depreciates by a certain percentage (it can be the same x% or a different value as y%) from its most recent peak (that is the highest or top value) (Moosa, 2000).

In this study, the scope of the filter rule has been expanded. In the original filter rule, trading signals are determined by looking at historical price data. In this study, moving average values are also taken into consideration while determining the trading rules. In this developed model, a buy signal is generated when the stock's p daily moving average (simple, linear, square-root, exponential, triangular, or closing price) appreciates x% value, and a buy signal when the stock's q daily moving average (simple, linear, square-root, exponential, triangular or closing price) depreciates y%.

In expanded-filter rule model, the following parameters should be determined in verifying the buying and selling points for any stock:

- Which type of moving average will be used to generate a buy signal?
- What is the degree of the moving average to be used to generate the buy signal?
- What is the threshold value to be used to generate a buy signal?
- Which type of moving average will be used to generate a sell signal?
- What is the degree of the moving average to be used to generate the sell signal?

- What is the threshold value to be used to generate a sell signal?

It is possible to say that these values may vary from stock to stock and may differ for stocks in different periods. Therefore, instead of finding a valid rule for each stock in every period, these six values will be optimized with the help of genetic algorithm.

3.2. Genetic algorithms

The term metaheuristic describes higher level heuristics that are proposed for the solution of wide range of optimization problems (Dokeroglu et al., 2019). Metaheuristic seems to be a generic algorithm framework or a black box optimizer that can be applied to almost all optimization problems (Abdel-Basset et al., 2018). The appeal of using these algorithms for solving complex problems is that they obtain the best/optimal solutions even for very large problem sizes in small amounts of time (Dokeroglu et al., 2019).

Genetic algorithms (GA), is a population-based metaheuristic introduced by John Holland (Holland, 1992) and it is biologically inspired search approach that is suitable to a wide range of optimization problems (Kramer, 2017).

A genetic algorithm for a particular problem must have the following five components (Michalewicz, 1996): (1) a genetic representation for potential solutions to the problem, (2) a way to create an initial population of potential solutions, (3) an evaluation function that plays the role of the environment, rating solutions in terms of their fitness, (4) genetic operators that alter the composition of children and (5) values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc).

Genetic algorithm procedure (Fig. 1) can be summarized as follows (Gen & Cheng, 1999). An initial population *P(t)* is created in generation *t*. Each individual represents a potential solution to the problem. Each individual is evaluated to give some measure of its fitness with the help of fitness function. Some individuals undergo stochastic transformations using genetic operations to create new solution candidates (individuals). There are two types of transformation: (i) crossover, which creates new individuals by combining parts from two individuals, and (ii) mutation, which creates new individuals by making changes in a single individual. Created individuals, called offspring *C(t)*, are evaluated. A new population is formed by selecting the individuals having the highest fitness values from the parent population and the offspring population. After several iterations (generations), the algorithm converges to the best individual (solution), which hopefully represents an optimal or suboptimal solution to the problem.

The fitness function plays an important role in any successful GA implementation since the main task of GA is to minimize the fitness function. The fitness function is a function that returns a numerical value measuring the goodness of an individual (Kaya & Alhajj, 2005). The fitness function accepts a candidate solution and produces an objective value as a measure of the performance of the candidate solution.

Crossover operation in GA implements a mechanism that mixes the genetic material of the parents. In crossover operation, two solutions at n positions are split up and are alternately assembled to obtain new individuals. The motivation of such an operation is that both strings might represent successful parts of solutions when combined even outperform their parents (Kramer, 2017). Another genetic operation in GA is mutation, which changes a solution by disturbing random changes.

Users must determine when the GA will stop iterations. It is possible to allow GA to run for a predetermined number of generations and that is the most popular termination condition. It is also possible to allow run GA for a specific time or until no significant improvement in fitness value is observed.

There are some advantages and disadvantages of genetic algorithms (Sivanandam & Deepa, 2008). Some of the advantages are; parallelism, liability, using only function evaluations, being easily modified for different problems, easy handling of large or poorly understood search spaces. Limitations are; the problem occurred in identifying fitness function, the definition of representation of the problem, lack of usage of gradients, no effective terminator, and requiring a large number of response (fitness) function evaluations.

There are many variants of Genetic Algorithm. For example, a parallel genetic algorithm is employed in (López-González et al., 2020). However, proposed model requires special central processing unit hardware as well as software capabilities that can implement parallel computation. Sharapov reviewed the variants of the genetic algorithm (Sharapov, 2007). The author states that a normal genetic algorithm does not use any auxiliary information about the objective function value such as derivatives. Because of this reason, a normal genetic algorithm is employed in this study without considering its variations.

In order to increase the performance, hybrid metaheuristics are developed. Hybrid algorithms are two or more algorithms that run together and complement each other to produce a profitable synergy from their integration (Rodriguez et al., 2012; Ting et al., 2015). One of the disadvantages of hybrid algorithms is their complexity in implementation (Ting et al., 2015). Among the different evolutionary algorithms, Genetic Algorithms are considered the most developed method of EA that apply evolutionary principles (Sivanandam & Deepa, 2008). Moreover, the results of a recent survey reveal that the genetic algorithm is the most popular metaheuristic employed for solving complex real-world optimization problems among other metaheuristics (Dokeroglu et al., 2019). For these reasons, in this study, a genetic algorithm was preferred for the optimization of the parameters of the filter rule.

There are studies focused on automated hyperparameter optimization. Benecki and colleagues developed a model to detect anomalies in spacecraft telemetry using a genetic algorithm (Benecki et al., 2021). Genetic Algorithms are also employed to optimize the parameters of Support Vector Machine (Gauthama Raman et al., 2017; Tao et al., 2019; Wu et al., 2007; Zhao et al., 2011b), Support Vector Regression (Wu et al., 2009), artificial neural networks (Ding et al., 2011; Leung et al., 2003). In all of these studies, the genetic algorithm is successfully applied to optimize the hyperparameters of relevant methods. Genetic algorithms have a broader range of applications. However, parameter optimization has been a highly successful theme among other applications (Bäck & Schwefel, 1993).

3.3. Artificial neural networks (ANN)

Mathematically, an ANN is a universal approximator, proven to be highly effective for modeling non-linear problems, with application to a diversity of large-scale problems, including pattern recognition, classification and control (Samani et al., 2007). Multi-Layer Perceptron (MLP) is one of the most frequently used artificial neural network architecture and belongs to the supervised neural networks (Yan et al., 2006). MLP is a system of simply connected nodes arranged in layers. A typical MLP network consists of three types of layers; input layers, hidden layers (which may include more than one layer), and output layers. Input features are passed forward from an input layer to the next hidden layers. The neurons that perform the final computation, in other words, whose outputs are the outputs of the network, are called output neurons; the other neurons, which perform intermediate computations, are termed hidden neurons (Dreyfus, 2004). Each hidden node, j, performs a linear operation which can be expressed as follows (Chiang et al., 2019; Choi & Kim, 2021; López-González et al., 2020; Mújica-Vargas, 2021; Rubio, 2021; de J. Rubio et al., 2021, 2022; Yan et al., 2006);

$$y_j = f_j\left(\sum_{i=1}^{q} w_{ji}x_i + b_j\right) \quad (1)$$

where, y_i is the output at the node $f(\blacksquare)$ is the nonlinear activation function of the j th node, and w_{ji} is the connection weight between the input x_i and the node. The node, j, receives x_i values from the previous layer and performs a simple transfer by calculating the weighted sum.

```
Inputs: price, short term degree, long term degree
Output: transactions

STMA = moving average of price with short term degree
LTMA = moving average of price with long term degree
signal = STMA > LTMA
// 0 in signal indicates sell position
// 1 in signal indicates buy position


Begin
   for idx = 2 to length of price
      if signal(idx-1) = 0 AND signal(idx) = 1 then
         buy stock
         save buying price as buyPrice
      end



      if signal(idx-1) = 1 AND signal(idx) = 0 then
         sell stock
      end
   end
end
```

(a) Trading Algorithm

```
Inputs: price, short term degree, long term degree
Output: transactions

STMA = moving average of price with short term degree
LTMA = moving average of price with long term degree
signal = STMA > LTMA
// 0 in signal indicates sell position
// 1 in signal indicates buy position


Begin
   for idx = 2 to length of price
      if signal(idx-1) = 0 AND signal(idx) = 1 then
         buy stock
         save buying price as buyPrice
      end

      condition1 = signal(idx-1) = 1 AND signal(idx) = 0
      condition2 = currentPrice > buyPrice
      if condition1 AND condition2    then
         sell stock
      end
   end
end
```

(b) The Modified Trading Algorithm

Fig. 2. General Structures of Trading and the Modified Trading Algorithm.

Then the bias value, b_j, is added to the weighted sum. Finally, the output layer takes the values from the last hidden layer and transforms them into the final result value (Choi & Kim, 2021).

In the model utilized in this study, the tangent sigmoid transfer function is used in all of the hidden layers as following;

$$k = \frac{2}{1 + e^{-2x}} - 1 \tag{2}$$

The output layer has a pure linear transfer function which is f(x) = x.

The error (or cost) function of the network is defined as in Equation (3) (Ding et al., 2011; Rubio, 2021):

$$e = \frac{1}{m}\sum_{j=1}^{m} \left(t_j - y_j\right)^2 \tag{3}$$

where, e is the error of the current output, t_j is the target output, y_j is the predicted output and m is the total output number.

The initial values of the connection weights (w_{ji} in Eq. (1)) are assigned randomly. The artificial neural network model employed in this study uses the Levenberg-Marquardt (LM) training algorithm to adjust these weights. During training, the ANN is presented with patterns of input and corresponding output pairs, during which the learning algorithm (in this case LM back-propagation algorithm) iteratively adjusts the values of connection weights within the ANN structure (Samani et al., 2007) to minimize the overall network error (Equation). The details of the LM algorithm can be followed in (Rubio, 2021).

3.4. The modified trading algorithm

According to the traditional filter rule, it is proposed to buy when the stock's price rises x% above its post local low and sell when it falls, y% below its local high (Sweeney, 1998). However, the rule can generate a sell signal that will result in a loss. For example, the rule might return a decision that we should sell the stock for less than the price at which we bought it. Implementation of the decision will result in a net loss from the transaction. This will lead to a decrease in profit that the investor can obtain at the end of the period.

In order to avoid such net losses, the trading algorithm has been modified. In this modification, the sale transaction is subject to two conditions; (i) being in-the-market position, and (ii) the selling price to be higher than the buying price. In other words, if the investor is in the in-market position, he/she has to wait for the filter rule to generate a signal where the selling price will be higher than the buying price. This modification prevents transactions in which a clear loss would occur.

The trading signals generated by the rule are implemented with the modified algorithm. In out-of-market positions, the return to be provided is calculated assuming that the interbank rate is compounded daily. It is assumed that 0.3 % commission expense is paid for each trading transaction executed on the exchange. Adjusted return is calculated by using the following equation:

$$r_a = \frac{(1 + r_f)(1 + r_o)}{(1 + r_c)} - 1 \tag{4}$$

where, r_a is stands for the adjusted return that includes the total effect of the filter rule, the interest earned when the market is not invested, and the commission expense paid on trading transactions. r_f stands for the rate of return generated by the signals produced by the filter rule. r_o and r_c represents, the return from interest and commission expenses, respectively.

The following performance measures are calculated with the modified trading algorithm;

- Adjusted return (r_a in Eq. (4))
- Total number of transactions
- Number of transactions with a profit
- Number of transactions with a loss
- Buy-and-hold return
- The number of days in out-of-market positions
- The number of days in in-the-market positions
- The number of sell signals where an obvious loss is prevented with a modified trading algorithm
- The rate of return generated by the signals produced by the filter rule (r_f in Eq. (4)).
- The rate of return from interest earned (r_o in Eq. (4)).

If there are unsold stocks at the end of the period (in other words, if the investor is inside the in-the-market position at the end of the period), the stocks are sold at the closing price of the last day. That price may be lower than the stock's purchase price (indicating a clear loss). Still, such

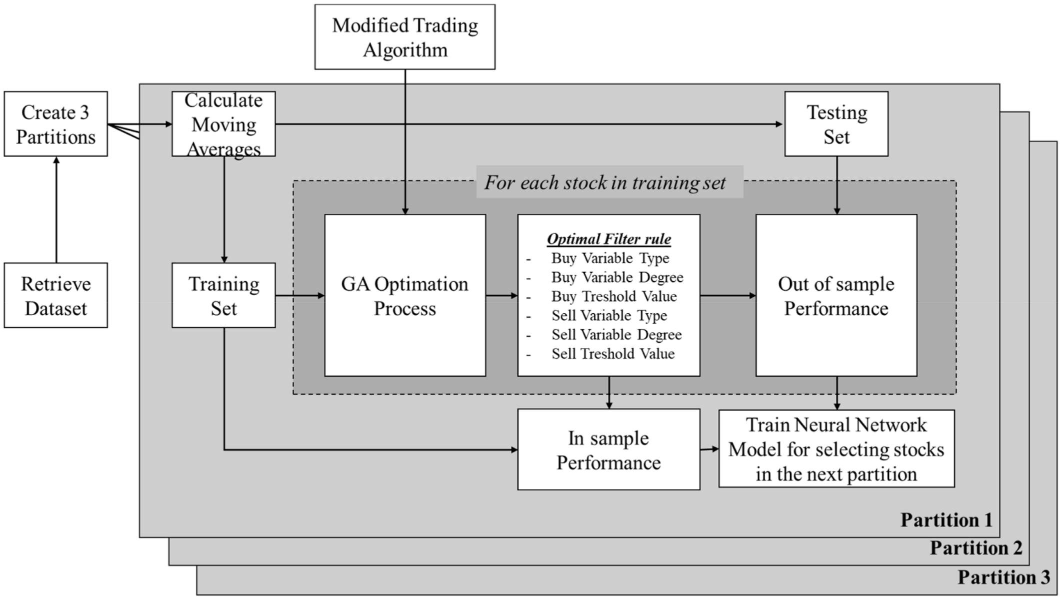

Fig. 3. Conceptual framework of the study.

Table 1
Initial Feature Pool.

Feature Index	Feature Name	Degree
1 ~ 50	Closing (Raw) Price	1 ~ 50
51 ~ 99	Simple Moving Average	2 ~ 50
100 ~ 148	Linear Moving Average	2 ~ 50
149 ~ 197	Square-root Moving Average	2 ~ 50
198 ~ 246	Exponential Moving Average	2 ~ 50
247 ~ 295	Triangular Moving Average	2 ~ 50

Table 2
Periods used in the study.

	Partition 1	Partition 2	Partition 3
Training Set Start	July 6, 2012	July 6, 2012	July 6, 2012
Training Set End	December 31, 2017	December 31, 2018	December 31, 2019
Testing Set Start	January 1, 2018	January 1, 2019	January 1, 2020
Testing Set End	March 31, 2018	March 31, 2019	March 31, 2020

a procedure is needed to make comparisons between periods.

General structures of the trading algorithm and the modified trading algorithm is compared in Fig. 2. The code of the trading algorithm developed on the MATLAB platform is shared as supplementary material. In this code, users can create a filter rule and see the buy and sell points on the time-series graph as plotted in the article. In this way, users will be able to see the trading points on the chart with different data sets and different filter rule parameters.

3.5. Moving average types

The average types used in the study are as follows.

- Simple moving average.

$$SMA_p = \frac{1}{p}\sum\nolimits_{i=n-p+1}^{n} P_i \tag{5}$$

where, SMA_p represents the simple moving average value of series (average over the last p days), n represents the window size of moving average, P_i represents the raw price data point on day i.

- Linear moving average

$$LMA_p = \frac{1}{\sum W}\sum\nolimits_{i=n-p+1}^{n} P_i W_i \tag{6}$$

where, W represents the weights assigned to each period. The highest weight is assigned to the first period. If the window size is equal to 3, then weights will be 3, 2, 1.

- Square-root moving average

$$SRMA_p = \frac{1}{\sum W}\sum\nolimits_{i=n-p+1}^{n} P_i W_i \tag{7}$$

where, W represents the weights assigned to each period. The highest weight is assigned to the first period. If the window size is equal to 5, then weights will be $\sqrt{5}, \sqrt{4}, \sqrt{3}, \sqrt{2}, \sqrt{1}$.

- Exponential moving average

$$EMA_p = \begin{cases} P_1, p = 1 \\ \alpha P_t + (1-\alpha)EMA_{t-1}, p > 1 \end{cases} \tag{8}$$

where, EMA_p represents the exponential moving average for the data point p. α is the smoothing factor in exponential moving average, and $0 < \alpha < 1$. In this study $\alpha = \frac{2}{n+1}$.

- Triangular moving average

Buy Variable	Buy Rate	Sell Variable	Sell Rate
Integer between (1~259)	Decimal value between [0.0, 5.0]	Integer between (1~259)	Decimal value between [0.0, 5.0]

Fig. 4. Gene design used in the genetic algorithm.

Output : Error Performance
Inputs : training data, candidate solution, interbank rates
Candidate Solution (1) : buy variable
Candidate Solution (2) : buy rate
Candidate Solution (3) : sell variable
Candidate Solution (4) : sell rate

adjustedReturn = ModifiedTradingAlgorithm (training data, candidate solution, intervank rates)

Error Performance = (-1)**adjustedReturn*

Fig. 5. The Design of the Fitness Function.

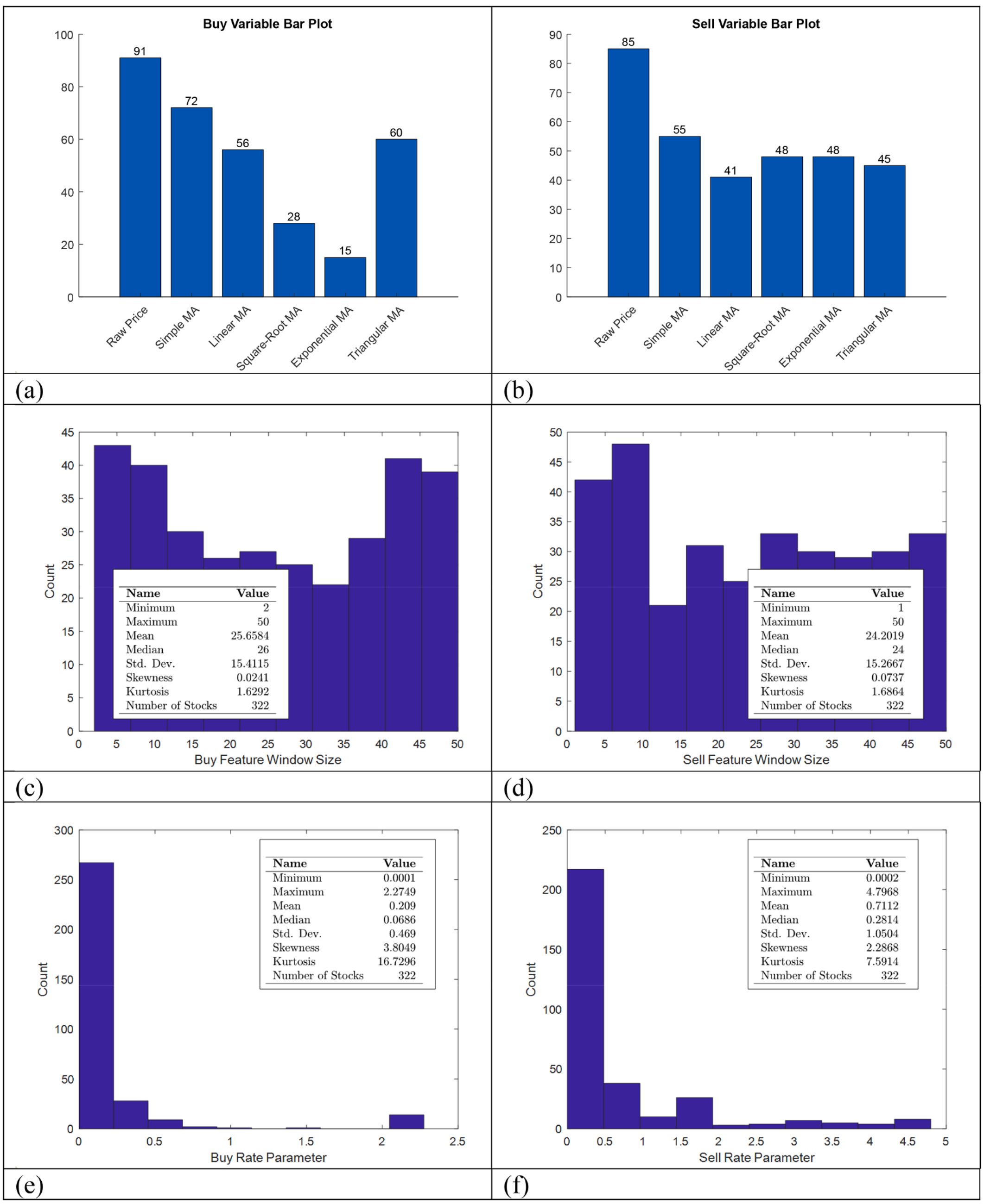

Fig. 6. Frequency Distribution of buy variables (a) and sell variables (b). Histogram of buy window size (c) and sell window size (d). Histogram of buy (e) and sell (f) thresholds (Partition 1).

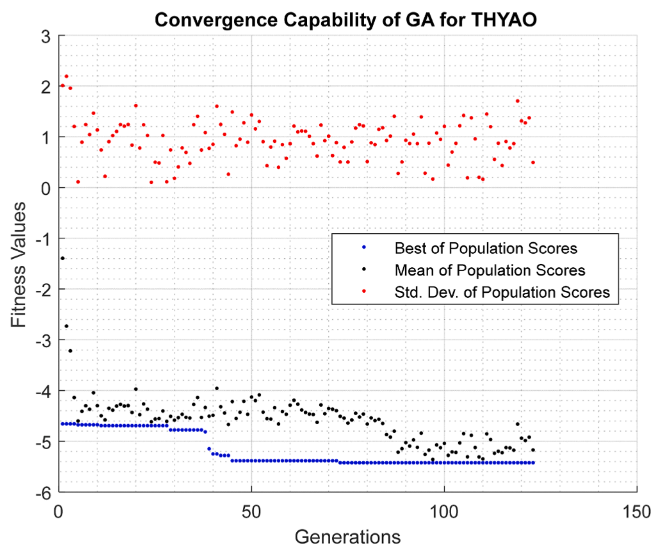

Fig. 7. Convergence Capability of the Genetic Algorithm for the stock THYAO.

$$TMA_p = \frac{1}{p}\sum\nolimits_{i=n-p+1}^{n} SMA_i \quad (9)$$

Where, *SMA* represents the simple moving average. *n* is the degree of the triangular moving average degree.

These averages are calculated based on the historical price of the stocks. There are some available filter rules such as buy when 200-day simple moving average is greater than stock price (Metghalchi et al., 2021). It is possible to create a large number of such rules. However, in this study, the optimal rules for each stock are determined by the genetic algorithm. The most frequently occurring moving average type and degree in optimal rules are examined with the help of figures (Fig. 6, Fig. 10, Fig. 14).

The average type and average degrees are optimized with genetic algorithms. The fitness function of the genetic algorithm runs the neural network model with relevant moving average type and degree (Equations (5)-(9)). In other words, the results of the equations are used as input in the neural network model used in the study.

4. Analysis

4.1. Proposed model and dataset

The conceptual framework of the study is depicted in Fig. 3. Firstly, the retrieved dataset is divided into three sequential parts. A three-dimensional data cube is prepared for each stock, including all moving average types. Each partition is also divided into two sets as training set and testing set. For each stock, parameters of the expanded-filter rule are optimized with genetic algorithms using the training set data. The performance of the optimized filter rule is determined with the testing set. A neural network model is developed to forecast the performance of the testing set by using the training set performance indicators as inputs. This trained model is used to forecast the testing performance in the next partition.

It is accepted that there are two positions in the study. These are in-the-market position and out-of-the-market position. While in-the-market position indicates that the stock has been bought, out-of-the-market position indicates that the stock was sold. In this latter case, it is assumed that money is valued with the help of the interbank rate. The interbank rate differs from year to year and it is obtained from the Central Bank of Turkey (https://www.tcmb.gov.tr).

The dataset initially contains closing price information for 357 stocks traded in Borsa Istanbul. The dataset was downloaded from https://www.finance.yahoo.com.The overall period considered in the study is between 06-07-2012 and 31–03-2020.

For each stock in the dataset, a feature pool is calculated. The initial

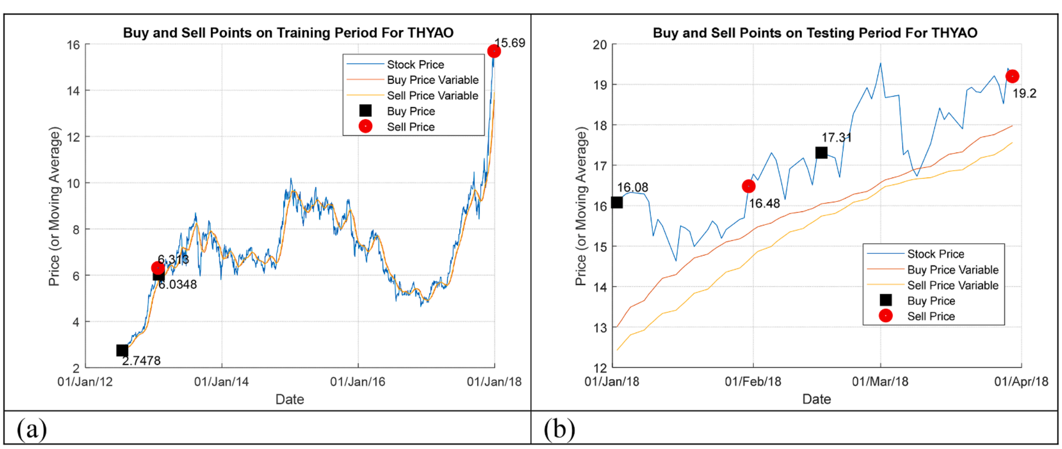

Fig. 8. Trading points for THYAO (in Partition 1) for training set (a) and testing set (b).

Table 3
Trading Algorithm Results for Training Set (Partition 1).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	0.4153	1.0000	1.0000	0.0000	0.0152	6.0000	92.0000	0.0000	−1.4367	4.4546
Max	50.4866	94.0000	94.0000	1.0000	23.6849	1283.0000	1423.0000	239.0000	7.1749	118.8390
Mean	5.4810	13.9159	13.5752	0.3230	1.7580	397.7389	965.1726	29.1991	0.8491	15.1880
Median	3.6940	9.0000	8.0000	0.0000	0.8789	335.5000	1015.5000	0.0000	0.8132	12.9892
Std	7.0433	15.8136	15.7583	0.4687	3.0322	312.4849	355.9668	52.7376	1.0370	13.5069
Skewness	3.7352	2.2995	2.3328	0.7570	4.7500	0.8806	−0.7721	2.2892	1.8401	5.6337
Kurtosis	18.5187	8.8296	9.0211	1.5730	29.6472	3.1394	2.8067	7.9263	11.7665	40.4035

Table 4
Trading Algorithm Results for Testing Set (Partition 1).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	−0.5105	1.0000	0.0000	0.0000	−0.5090	0.0000	2.0000	0.0000	−4.2753	2.2578
Max	0.5710	10.0000	10.0000	1.0000	5.0943	62.0000	64.0000	51.0000	4.0933	28.1993
Mean	0.0374	1.5487	1.0088	0.5044	0.0884	19.3363	44.6637	0.4558	0.5441	6.7977
Median	0.0100	1.0000	1.0000	1.0000	0.0140	12.0000	52.0000	0.0000	0.4073	4.9172
Std	0.1782	1.2256	1.3792	0.5011	0.4317	19.8365	19.8365	3.9610	1.1944	5.0667
Skewness	0.4954	3.8868	3.1932	−0.0177	7.9029	0.5780	−0.5780	10.9796	0.0845	2.1099
Kurtosis	4.0569	23.1136	18.4942	1.0003	86.0871	1.8715	1.8715	130.5221	5.5377	7.4056

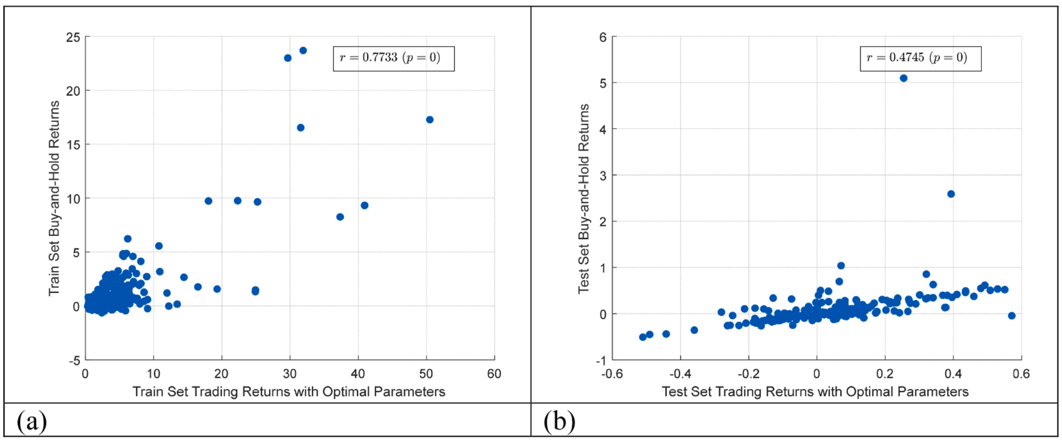

Fig. 9. Correlations between filter rules and buy-and-hold returns. (a) training set, (b) testing set (Partition 1).

feature pool is presented in Table 1. There are 295 features in the pool. This means that genetic algorithm will search among these features to determine the best ones, along with the percentage rates that will be used to generate buy and sell signals.

After the variables are prepared and the data set is arranged for analysis, the data set is divided into three different partitions to determine whether the proposed system will produce successful results in different periods. Each partition is also divided into two parts as training set and test set. Periods for the partitions are listed in Table 2.

There are four parameters optimized with the help of genetic algorithms (Fig. 4). These variables are;

- Buy variable: This variable is used to generate the buy signals and it is selected from the initial feature pool. This variable also includes the number of degrees of moving averages.
- Buy rate: This variable is also used to generate the buy signals and it is optimized in the range of [0.0, 5.0].
- Sell variable: This variable is used to generate the sell signals and it is selected from the initial feature pool. This variable also includes the number of degrees of moving averages.
- Sell rate: This variable is also used to generate the sell signals and it is optimized in the range of [0.0, 5.0].

The fitness function used in the study has a simple structure (Fig. 5). It uses training data, candidate solution, and interbank rates data as inputs. The training data contains the raw price and average data in the training set. The candidate solution, on the other hand, is the solution currently under evaluation; it includes the variable and threshold value (percentage) to be used to create the buy signal, the variable to be used to create the sell signal, and the threshold value (percentage), respectively. Interbank rates include the rates used in the interbank lending for the years 2012–2020.

The adjusted profit amount in the training set of the candidate solution is calculated. Since this value is intended to be maximum, the error value returned by the function is to multiply the adjusted return value by −1.

(Jabari Lotf et al., 2022) determined the parameters of the genetic algorithm by testing different values. In some studies, the parameters of the genetic algorithm are listed without mentioning how they are determined (Ding et al., 2011; Gauthama Raman et al., 2017; Tao et al., 2019; Zhao et al., 2011). In several studies, the parameters of GA are determined with trial and error through experiments (Leung et al., 2003; Wu et al., 2009). There are no clear procedures to select the parameters of GA, parameter values are mostly selected by conventions (Eiben & Smit, 2011; López-González et al., 2020). In this study, the population size, crossover rate, mutation rate, and elite count are determined by a trial and error approach with considering the hardware specifications. Parameters of the genetic algorithm utilized in the study are as follows: crossover function is crossover scattered; crossover rate is 0.8; population size is 50; elite count is 3; fitness limit is negative infinity; the maximum number of generations is 400; no time limit is defined.

4.2. Results

4.2.1. Genetic algorithm Results for the partition 1

Genetic algorithm was run with the first partition and optimal values were found for 322 stocks out of 357 stocks. Optimal values could not be found for the remaining 35 stocks. The reasons for the failure to find optimal values for the remaining stocks are as follows;

- Insufficient number of observations,
- The stock is not traded in the market during that period,
- Not even a single buy signal can be produced in the period.

Bar and histogram graphs of four different variables optimized by genetic algorithm for 322 stocks are shown in Fig. 6.

Part (a) of Fig. 6 shows the bar graph of the frequencies of the variable types optimized for buy signals. The most frequently chosen

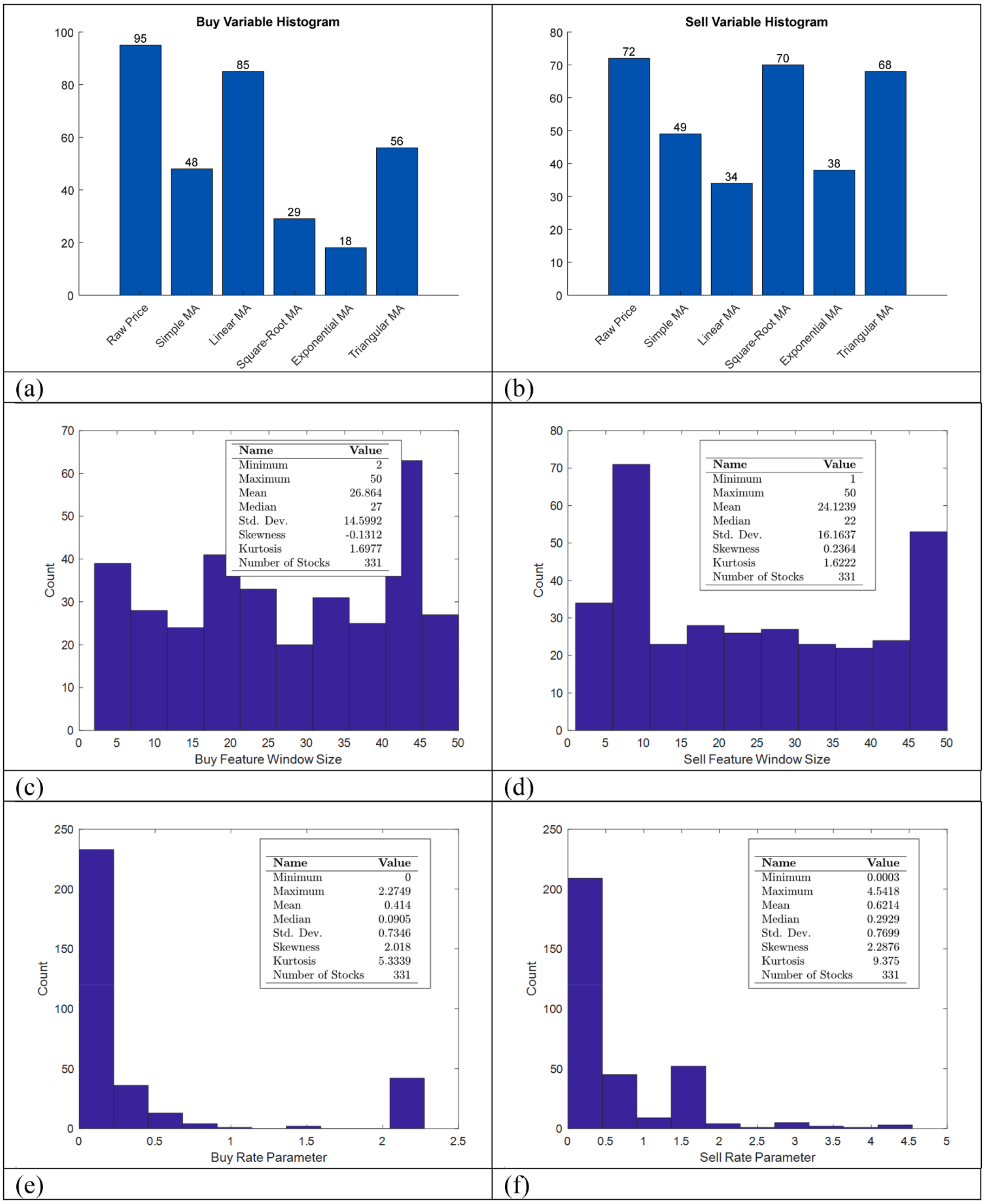

Fig. 10. Frequency Distribution of buy variables (a) and sell variables (b). Histogram of buy window size (c) and sell window size (d). Histogram of buy (e) and sell (f) thresholds (Partition 2).

variable is the closing price variable. In 91 of the 322 stocks (for 91 stocks), the optimal buying variable was determined as the closing price.

Part (b) of Fig. 6 contains the bar chart for the variable types optimized for sell signals. The most preferred variable as a sales variable is the closing price. The optimal selling rule variable for 85 stocks is the closing price.

In the (c) and (d) parts of Fig. 6, there is the histogram distribution of the window size variable, which is optimized for buy (sell) signals. This variable represents how many historical values of the optimal feature will be used in trading rules.

In the (e) and (f) parts of Fig. 6, there is the histogram graph of the optimized rate for buy (sell) signals. A buy (sell) signal is generated when there is an increase (decrease) of at least this rate from a price in the past. The search range is [0.0, 5.0]. For the vast majority of stocks, the optimal value is set below 50 %.

Convergence capability of genetic algorithm for one of the randomly selected stocks (THYAO: Turkish Airlines) is presented in Fig. 7. Genetic algorithm stopped iterations due to no improvements are made on the score of the best individual in the populations during the last 50 generations. The best (minimum) score in the initial population (which is randomly created) is calculated as −4.6571. The best score of the last iteration is −5.4230 indicating an increase of 16.45 % $\left(=\frac{-5.4230-(-4.6571)}{-4.6571}\times 100\right)$ on the best individual score. It can be observed that the standard deviation values of population scores varies between 0 and 2.

The trading points on the training set (a) and test set (b) for THYAO, in the first partition are shown in Fig. 8. The optimal rule determined for THYAO stock by genetic algorithm is as follows; "buy when the stock's simple moving average (window size = 38) rises 9.7 % above its

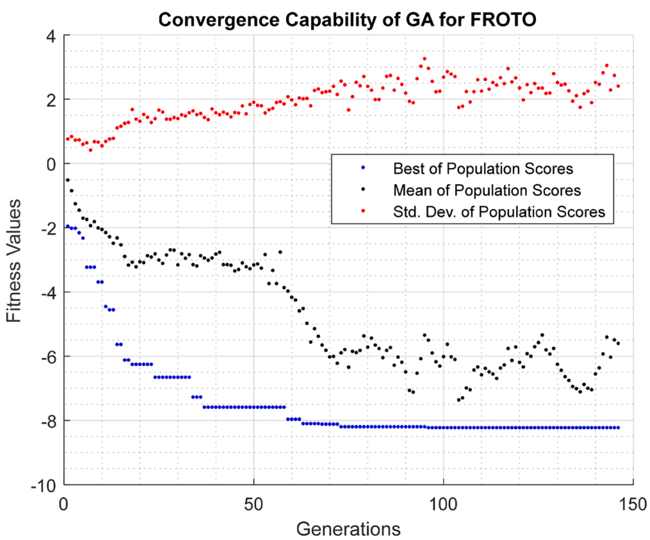

Fig. 11. Convergence Capability of the Genetic Algorithm for the stock FROTO.

previous value and sell when linear moving average (window size = 48) fall 49.08 % below its previous value. When the optimal rule is followed, only two trades take place during 1430 day and profit is made from both trades. When applying the trading algorithm described earlier during the training time frame, a profit of 546.37 % is obtained, which is higher than the return of the buy and hold strategy, and gives a return of 467.41 %. While the number of in-the-asset days is 1413, the number of out-of-the-asset days is 17. During the training period, the modified trading algorithm did not prevent any trades that would incur a clear loss when implemented.

In panel (b) of Fig. 8, there are trading points that will occur in the test set because of the pursuing of the optimal rule. When the modified trading algorithm is applied during the 63-day test period, a return of 13.34 % is realized. In the same period, the profit of an investor who applies the passive buy and hold strategy will be 19.40 %. For THYAO stock, although the rule was successful in the training set, it could not provide a higher return on the test set than the buy and hold strategy. While the number of in-the-asset days is 52, the number of out-of-the-asset days is 12. When the optimal rule is followed, two trades are made in the test set and profit is made from both trades. In the test set, the number of uneconomic transactions blocked by the modified algorithm is 2.

Table 3 contains the descriptive statistics of trading performance in the training set when the optimal rules for 322 stocks (determined by genetic algorithm for each stock separately) are followed. The minimum profit rate (0.4153 %) is much higher than the minimum buy-and-hold return (0.0152 %). “The median of the profit rate is greater than the median of the buy-and-hold return” alternative hypothesis is tested with the Wilcoxon rank sum test ($z = 14.1378$, ranksum = 91951 and $p = 0$). Results indicate that optimized filter rules performed better than buy-and-hold return in the training set.

Table 4 shows the performance of the optimal rules of the same 322 stocks on the test set (on a set that the models have never encountered before). “The median of the profit rate is greater than the median of the buy-and-hold return alternative hypothesis” is tested with the Wilcoxon rank sum test ($z = -0.2741$, ranksum = 672261 and $p = 0.6080$). Results indicate that not enough evidence was found on the superiority of the optimal filter rules over buy-and-hold return. In other words, the high performance in the training set could not be sustained in the test set.

The relationship between the return values of the optimal rules and the returns of the buy and hold strategy is shown in Fig. 9. On the horizontal axis of the figures, the return rates of the optimal filter rules are located, on the vertical axis are the return rates of the buy and hold strategy. In both sets, a positive and statistically significant relationship was found between the return rates of the filter rules and the return rates

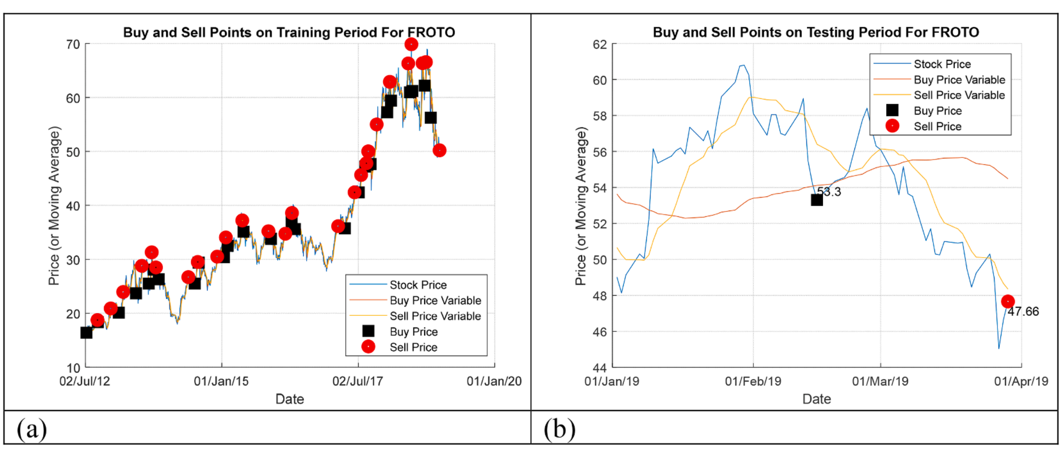

Fig. 12. Trading points for FROTO (in Partition 2) for training set (a) and testing set (b).

Table 5
Trading Algorithm Results for Training Set (Partition 2).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	0.1335	1.0000	1.0000	0.0000	−0.7631	6.0000	58.0000	0.0000	−6.5657	3.4483
Max	73.9111	92.0000	92.0000	1.0000	11.8433	1632.0000	1684.0000	523.0000	33.4823	1283.4830
Mean	5.2975	15.4493	15.0220	0.4185	0.8518	575.4978	1026.2030	39.7093	1.0425	26.6167
Median	3.9183	9.0000	9.0000	0.0000	0.4033	517.0000	1107.0000	3.0000	0.7690	13.227
Std	6.3885	16.4758	16.4428	0.4944	1.6313	415.6591	448.9505	75.8081	3.2623	119.7357
Skewness	6.3459	1.8811	1.9055	0.3304	3.5458	0.6630	−0.4725	3.2619	8.6089	10.2661
Kurtosis	62.0339	6.4134	6.5247	1.1092	19.6445	2.5639	2.2376	16.1750	87.1936	107.9291

Table 6
Trading Algorithm Results for Testing Set (Partition 2).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	−0.4933	1.0000	0.0000	0.0000	−0.5974	0.0000	1.0000	0.0000	−4.2201	2.5496
Max	0.9496	8.0000	8.0000	1.0000	1.6452	62.0000	63.0000	15.0000	2.9733	31.4780
Mean	0.0312	1.4185	0.8767	0.5242	0.0971	28.2996	34.7004	0.3524	0.6541	6.4567
Median	0.0068	1.0000	1.0000	1.0000	0.0396	29.0000	34.0000	0.0000	0.5016	4.4569
Std	0.1907	0.9849	1.0696	0.5005	0.2637	19.9064	19.9064	1.8694	1.1561	5.4568
Skewness	1.1462	3.3746	2.7257	−0.0970	2.2731	−0.0277	0.0277	5.6590	−0.2774	1.7207
Kurtosis	6.1009	16.4556	14.5391	1.0094	12.3171	1.7011	1.7011	35.6964	3.4617	7.1669

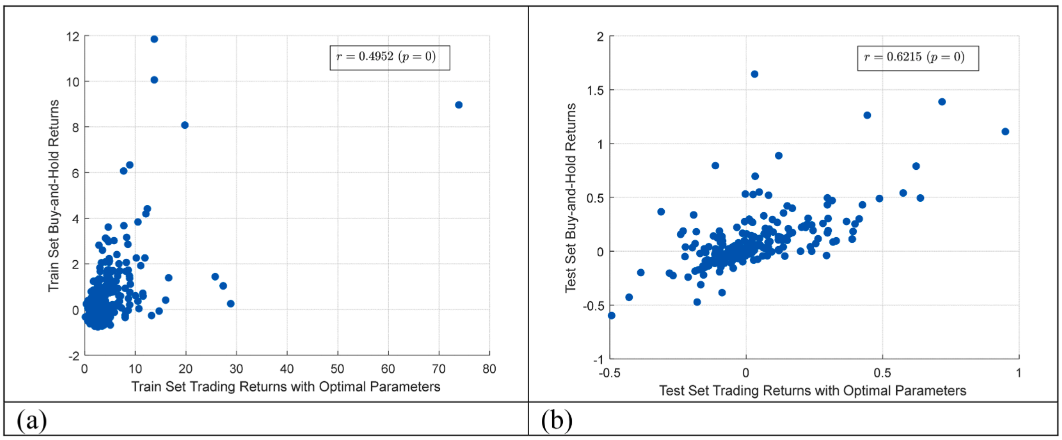

Fig. 13. Correlations between filter rules and buy-and-hold returns. (a) training set, (b) testing set (Partition 2).

of the buy and hold strategies.

4.2.2. Genetic algorithm Results for the partition 2

In the second partition, the optimal rule could be determined for 331 out of 357 stocks. Bar and histogram graphs of four different variables optimized by genetic algorithm for 322 stocks are shown in Fig. 10.

Part (a) of Fig. 10 shows the bar graph of the frequencies of the variable types optimized for buy signals. The most frequently chosen variable is the closing price variable. In 95 of the 331 stocks, the optimal buying variable was determined as the closing price.

Part (b) of Fig. 10 contains the bar chart for the variable types optimized for sell signals. The most preferred variable as a sales variable is the closing price. The optimal selling rule variable for 72 stocks is the closing price.

In the (c) and (d) parts of Fig. 10, there is the histogram distribution of the window size variable, which is optimized for buy (sell) signals.

In the (e) and (f) parts of Fig. 10, there is the histogram graph of the optimized rate for buy (sell) signals. The search range is [0.0, 5.0]. For the vast majority of stocks, the optimal value is set below 50 %.

Convergence capability of genetic algorithm for one of the randomly selected stock (FROTO: Ford Otomotiv Sanayi) is presented in Fig. 11. Genetic algorithm stopped iterations due to no improvements are made on the score of the best individual in the populations during the last 50 generations. The best (minimum) score in the initial population (which is randomly created) is calculated as −1.9552. The best score of the last iteration is −8.2231 indicating an increase of 320.58 % $\left(= \frac{-8.2231-(-1.9552)}{-1.9552} \times 100\right)$ on the best individual score. An increase in the standard deviations of the population scores can be observed from the Fig. 11.

The trading points on the training set (a) and test set (b) for the stock (FROTO) in the second partition are shown in Fig. 12. The optimal rule determined for FROTO stock by genetic algorithm is as follows; "buy when the stock's simple moving average (window size = 50) rises 0.51 % above its previous value and sell when square-root moving average (window size = 10) falls 13.25 % below its previous value. When the optimal rule is followed, 25 trades take place during the 1690 days, and profit is made except for the last transaction. According to the trading algorithm, if the position on the last trading day is in-the-asset position, the stocks are converted into money with the closing price of the last day. This rule has been applied for this stock and the shares have been converted into cash despite the loss. When applying the trading algorithm during the training time frame, a profit of 822.31 % is obtained, which is higher than the return of the buy and hold strategy, giving a return of 206.10 %. While the number of in-the-asset days is 1191, the number of out-of-the-asset days is 499. During the training period, the modified trading algorithm prevented 66 transactions that would incur a clear loss when implemented.

In panel (b) of Fig. 12, there are trading points that will occur in the test set because of the application of the optimal rule. When the modified trading algorithm is applied during the 63-day test period, a loss of −10.85 % occurs. In the same period, the loss of an investor who applies the passive buy and hold strategy will be −2.77 %. For FROTO stock, although the rule was successful in the training set, it could not provide a higher return on the test set than the buy and hold strategy. While the number of in-the-asset days is 31, the number of out-of-the-asset days is 32 for this stock. When the optimal rule is followed, two trades are made in the test set and profit is made from both trades. In the test set, the number of uneconomic transactions blocked by the modified algorithm is 0.

Performance of the trading algorithm in the training set of partition 2 presented in Table 5. Results indicate that the trading algorithm performed better than buy and hold return. In the training set, the trading algorithm outperformed the buy and hold strategy, since the genetic algorithm performed the optimization using the training set. "The median of the profit rate is greater than the median of the buy-and-hold return" alternative hypothesis is tested with Wilcoxon rank sum test (z = 15.3944, rank sum = 73161 and p = 0). Results indicate that

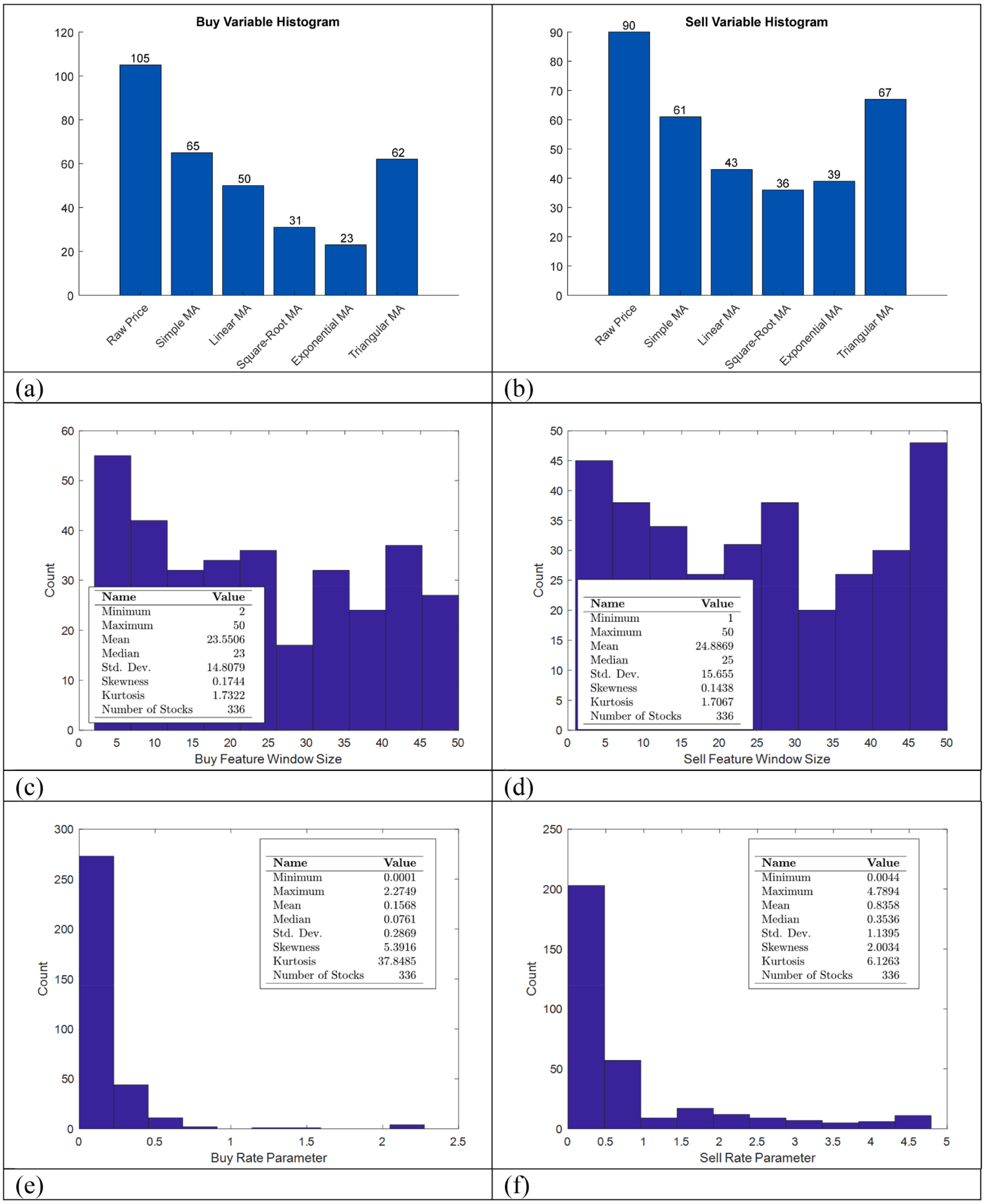

Fig. 14. Frequency Distribution of buy variables (a) and sell variables (b). Histogram of buy window size (c) and sell window size (d). Histogram of buy (e) and sell (f) thresholds (Partition 3).

optimized filter rules performed better than buy-and-hold return in the training set.

Table 6 shows the performance of optimized rules on the testing set for partition 2. "The median of the profit rate is greater than the median of the buy-and-hold return" alternative hypothesis is tested with Wilcoxon rank sum test ($z = -3.2888$, rank sum $= 47046$ and $p = 0.9995$). Results indicate that enough evidence was not found on the superiority of the optimal filter rules over buy-and-hold return. In other words, the high performance in the training set could not be sustained in the test set.

The scatter diagram between the return values of the optimal rules and the returns of the buy and hold strategy is shown in Fig. 13. In both sets, a positive and statistically significant relationship was found between the return rates of the filter rules and the return rates of the buy and hold strategies.

4.2.3. Genetic algorithm Results for the partition 3

The genetic algorithm was applied for partition 3 and it was able to perform the optimization process for 336 stocks out of 357 stocks. Descriptive statistics of optimal genes are presented in Fig. 14.

Part (a) of Fig. 14 shows the bar graph of the frequencies of the variable types optimized for buy signals. The most frequently chosen variable is the closing price. In 105 of the 336 stocks, the optimal buying variable was determined as the closing price. Part (b) of the figure contains the bar chart for the variable types optimized for sell signals. The most preferred variable as a sales variable is the closing price. The optimal selling rule variable for 90 stocks is the closing price.

In the (c) and (d) parts of Fig. 14, there is a histogram distribution of the window size variable, which is optimized for buy (sell) signals. Figures also include descriptive statistics as tables on the legends.

In the (e) and (f) parts of Fig. 14, there is a histogram graph of the

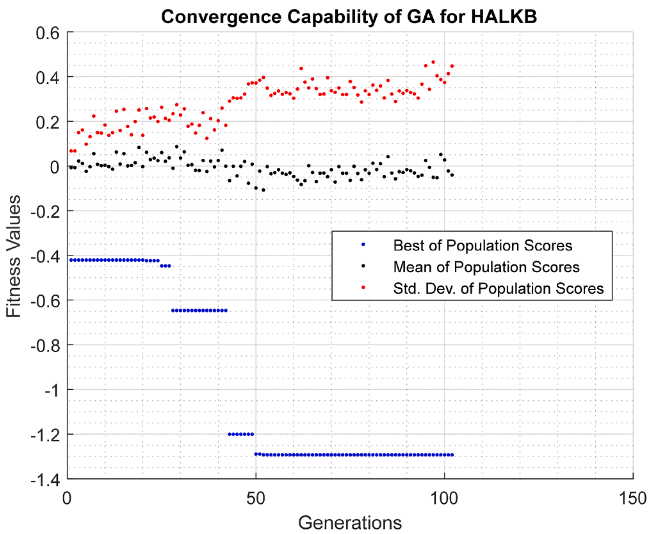

Fig. 15. Convergence Capability of the Genetic Algorithm for the stock HALKB.

optimized rate for buy (sell) signals. The search range is [0.0, 5.0]. For the vast majority of stocks, the optimal value is set below 50 %.

Convergence capability of genetic algorithm for one of the stock (HALKB: Turkiye Halk Bankasi) is presented in Fig. 15. Genetic algorithm stopped iterations due to no improvements are made on the score of the best individual in the populations during the last 50 generations. The best (minimum) score in the initial population (which is randomly created) is calculated as −0.4209. The best score of the last iteration is −1.2926 indicating an increase of 207.10 % $\left(=\frac{-1.2926-(-0.4209)}{-0.4209}\times 100\right)$ on the best individual score. An increase in the standard deviations of the scores can be observed from the figure.

The trading points on the training set (a) and test set (b) for the selected stock (HALKB) in the third partition are shown in Fig. 16. The reason for choosing this stock is that the price of the stock decreases in both the training set and the test set. In this way, the performance of the trading algorithm on a stock where the price decreased has been visualized in detail. The optimal rule determined for HALKB stock by genetic algorithm is as follows "buy when the stock's simple moving average (window size = 2) rises 1.13 % above its previous value and sell when simple moving average (window size = 7) falls 0.44 % below its previous value. When the optimal rule is followed, 32 trades take place within 1950 days, and profit is made from each one. When applying the trading algorithm during the training time frame, a profit of 1.366 % is obtained, which is higher than the return of the buy and hold strategy, giving a loss of 0.5853 %. While the number of in-the-asset days is 334, the number of out-of-the-asset days is 1616. During the training period, the modified trading algorithm prevented 126 transactions that would incur a clear loss when implemented.

In panel (b) of Fig. 16, there are trading points that will occur in the test set because of the application of the optimal rule. When the modified trading algorithm is applied during the 63-day test period, a profit of 0.0011 % occurs. In the same period, the loss of an investor who pursues the passive buy and hold strategy will be −0.1925 %. For HALKB stock, the rule was outperformed the buy-and-hold strategy in both the training and test set. While the number of in-the-asset days is 4, the number of out-of-the-asset days is 59. When the optimal rule is followed, a single trade is made in the test set, and profit occurs. In the test set, the number of uneconomic transactions blocked by the modified algorithm is 0.

Performance of the trading algorithm in the training set for partition 3 is presented in Table 7. Results indicate that the trading algorithm performed better than buy and hold return. In the training set, the trading algorithm outperformed the buy and hold strategy, since the genetic algorithm performed the optimization using the training set. "The median of the profit rate is greater than the median of the buy-and-

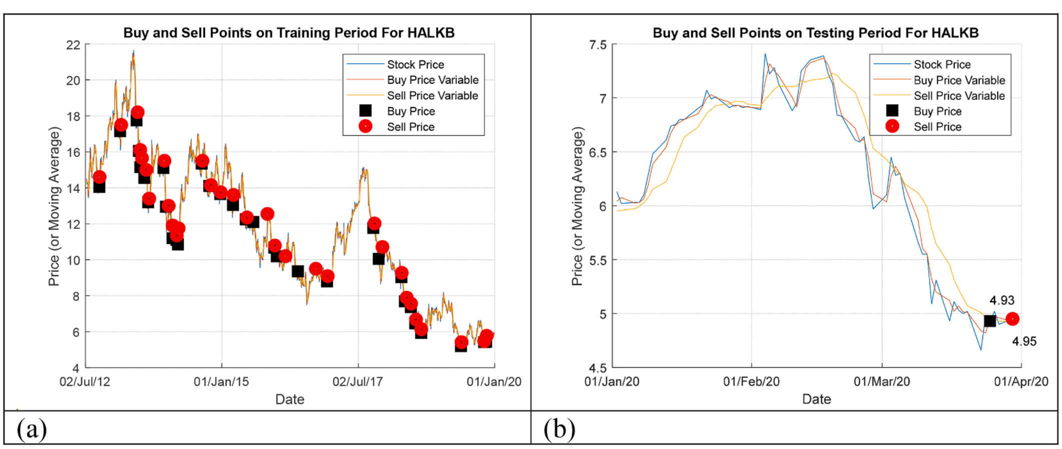

Fig. 16. Trading points for HALKB (in Partition 3) for training set (a) and testing set (b).

Table 7
Trading Algorithm Results for Training Set (Partition 3).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	−0.6227	1.0000	0.0000	0.0000	−0.5654	1.0000	35.0000	0.0000	−5.5412	4.1704
Max	86.5034	81.0000	80.0000	1.0000	38.5302	2050.0000	2084.0000	730.0000	34.6298	1438.5200
Mean	7.0347	11.5855	11.3783	0.2039	2.3796	452.2467	1468.7200	52.2401	1.1053	28.8581
Median	4.6614	6.0000	6.0000	0.0000	1.1059	279.5000	1655.0000	0.0000	0.7765	13.3574
Std	8.3874	14.2539	14.2474	0.4036	3.8160	476.1481	591.6029	102.7846	3.2484	125.2224
Skewness	4.1206	2.3898	2.3833	1.4695	4.4881	1.3328	−0.8519	2.8247	8.2088	10.0921
Kurtosis	31.7910	9.1944	9.1297	3.1594	33.6257	4.2008	2.6191	12.5683	80.700	109.4885

Table 8
Trading Algorithm Results for Testing Set (Partition 3).

	Profit (%)	Number of Transactions	Number of transactions with a profit	Number of transactions with a loss	Buy and hold return (%)	Number of out-of-the asset days	Number of in-the-asset days	Prevented loss	Asset return (%)	Interest return (%)
Min	−0.8177	1.0000	0.0000	0.0000	−0.8172	0.0000	1.0000	0.0000	−5.9621	1.9734
Max	1.1487	5.0000	4.0000	1.0000	1.3159	62.0000	63.0000	10.0000	4.5159	43.1971
Mean	−0.1309	1.3553	0.6086	0.7336	−0.1873	22.7993	40.2007	0.4243	−0.5227	6.2020
Median	−0.1724	1.0000	0.0000	1.0000	−0.2372	17.5000	45.5000	0.0000	−0.4183	5.0787
Std	0.2684	0.6691	0.7587	0.4428	0.2805	22.3523	22.3523	1.3646	1.0043	4.5585
Skewness	0.9054	2.4263	1.4689	−1.0566	2.2011	0.3726	−0.3726	3.933	−0.9804	4.3012
Kurtosis	5.3745	10.4123	6.0893	2.1163	11.6765	1.5685	1.5685	20.0544	10.1792	26.8477

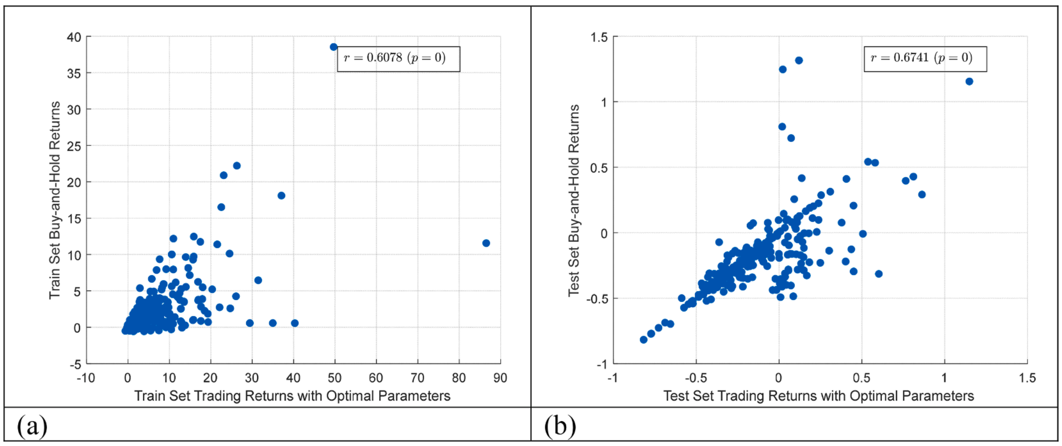

Fig. 17. Correlations between filter rules and buy-and-hold returns. (a) training, (b) testing set (Partition 3).

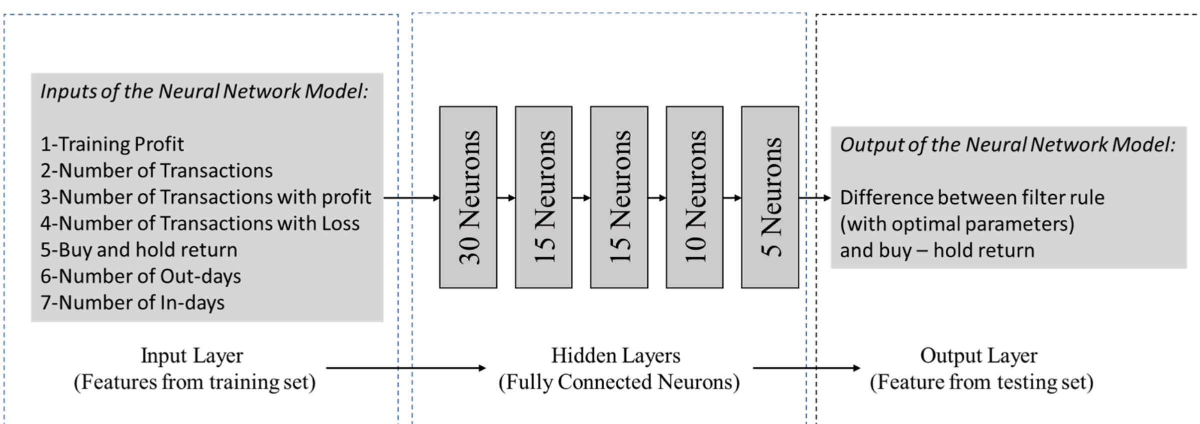

Fig. 18. Artificial Neural Network Model utilized in the study.

Table 9
Wilcoxon Ranked Sign Test Results for Different Number of Stocks for Partition 2.

Number of Stocks	Z val	Ranksum	p
10	1.8527	130	0.0320
15	1.7421	275	0.0407
20	1.9344	482	0.0265
25	1.4360	712	0.0755

Table 10
Wilcoxon Ranked Sign Test Results for Different Number of Stocks for Partition 3.

Number of Stocks	Z val	Rank sum	p
10	2.3056	136	0.0106
15	2.8633	302	0.0021
20	3.0172	522	0.0013
25	2.8916	787	0.0019

hold return" alternative hypothesis is tested with the Wilcoxon rank sum test (z = 11.6281, rank sum = 117751 and p = 0). Results indicate that optimized filter rules performed better than buy-and-hold return in the training set.

Table 8 shows the performance of optimized rules on the testing set for partition 3. The median of the profit rate is greater than the median of the buy-and-hold return alternative hypothesis is tested with Wilcoxon rank sum test (z = -3.4745, rank sum = 100093.5 and p = 0.). Results indicate that enough evidence was found on the superiority of the optimal filter rules over buy-and-hold return. In other words, the high performance in the training set is sustained in the test set.

The scatter diagram between the return values of the optimal rules and the returns of the buy and hold strategy is shown in Fig. 17. In both sets, a positive and statistically significant relationship was found between the return rates of the filter rules and the return rates of the buy and hold strategies.

4.3. Neural network model for stock selection

In this section, an artificial neural network model is developed which will enable us to predict the stocks that can be successful in the test set with the assistance of the data in the training set. While the performance indicators in the training set are included as inputs in the artificial neural network model, the output variable is the difference between the rate of return of the trading algorithm and the rate of return of the buy and hold strategy (Fig. 18).

Five intermediate layers are used in the artificial neural network model. The number of neurons in these layers is 30, 15, 15, 10, and 5, respectively. The number of layers and neurons were determined by considering the computational power of the computer on which the calculations were made. Neural network is trained with the Levenberg-Marquardt training algorithm. While 70 % of the data set presented as input was used in training the model, 15 % in validating the model, and the remaining 15 % in testing the model. All of the neurons in the hidden layers have a hyperbolic tangent sigmoid transfer function, while the transfer function in the output layer is a pure linear transfer function.

The model is trained by using the values in the training set in the first partition as the input variable and the values in the test set (the difference between the return of the algorithm and the return of the buy and hold strategy) as the output variable. The trained model is used to predict the performance of the trading rules in the testing set. The values in the training set in the second partition were presented as input in the trained model to predict the performance in the test set. As a result, each stock is assigned a numerical value that includes the performance estimation on the test set before it even happens. The higher this value is assumed, the better the stock will perform on the test set. It is possible to select different numbers of stocks by using the outputs of artificial neural networks. Table 9 shows the results of the one-sided hypothesis test "The median of the profit rate is greater than the median of the buy-and-hold return". When 25 stocks were selected, the results were significant at the 10 % level, while a significant difference was achieved at the 5 % level in the number of other stocks.

Another artificial neural network model is trained by using the values in the training set in the second partition as the input variable and the values in the test set (the difference between the return of the algorithm and the return of the buy and hold strategy) as the output variable. The values in the training set in the third partition were presented as input to this trained model to predict the performance in the test set. As a result, each stock is assigned a numerical value that includes the performance estimation on the test set before it even happens. Table 10 shows the results of the one-sided hypothesis test "The median of the profit rate is greater than the median of the buy-and-hold return" in cases where the different numbers of stocks are selected. All results are significant at the 5 % level.

5. Conclusion

In this study, the scope of the traditional filter rule has been extended to include moving average types. Genetic algorithm is used to optimize the parameters of the extended filter rule for each stock. The genetic algorithm optimizes the type of variable, degree of variable, and threshold value to be used both in the buy signal, and in the sell signal.

In the experiments carried out with the data set covering 357 stocks traded in Borsa Istanbul and covering nearly-eight years period, it was determined that the rules optimized with the genetic algorithm provided a higher return on the training set than the buy-and-hold strategy. However, it was determined that the performance of the optimal rules on the test set was not sufficient to exceed the buy-and-hold rate of return. In this respect, it can be stated that Borsa Istanbul is an efficient market in a weak form.

An artificial neural network model was developed to predict the performance of stocks in the test set by using the performance in the training set. The stock selection process was carried out by using the results of the artificial neural network model. Thus, only stocks that are predicted to have high performance were considered. It was determined that the performance of the selected stocks outperformed the buy and hold strategy at a statistically significant level.

The scatter plots drawn between the performance in the training set and the performance values in the test set show that the performance of these two sets is positively related to each other, although the strength of this relationship varies from partition to partition.

Assumptions of the study are as follows; any number of purchases and sales can be made at the closing price of the day, tax calculations are ignored, dividend payments are ignored, the possibility of short selling is ignored.

In this study, only stocks in Borsa Istanbul are considered. The time frame used in the study is another limitation. Although research on optimizing the structure of artificial neural networks continues, in this study, the structure of the network was designed to consider the computational power of the computer used in the analysis by testing different architectures.

In the future studies; ratios or items from financial statements can be used as inputs to improve the predicting capability of the neural network. Proposed system may be run for the stock data of exchange markets of other countries. Moreover, it may be applied to different financial time series that are suitable for trading (such as exchange rates or commodity prices). This study considers one-indicator trading rules, however, the performance of two or more indicators for trading rules can be researched in future studies.

CRediT authorship contribution statement

Mehmet Ozcalici: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Mete Bumin:** Validation, Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to thank to the editors and the anonymous reviewers for all of their careful, constructive and insightful comments and suggestions towards improving this study.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2022.118120.

References

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic Algorithms: A Comprehensive Review. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, 185–231*. https://doi.org/10.1016/B978-0-12-813314-9.00010-4

Akgun, A., & Sahin, E. (2017). AGLC 4th ed. Ali Akgun and I Erem Sahin, 'The Testing of Efficient Market Hypothesis in Borsa Istanbul. In *Annals of the Constantin Brancusi University of Targu Jiu, Letters and Social Sciences Series* (Vol. 35, Issue 2).

Alexander, S. S. (1961). Price movements in speculative markets: Trends or random walk? *Industrial Management Review, 2*(2), 7–26.

Aliyev, F. (2019). Testing market efficiency with nonlinear methods: Evidence from borsa istanbul. *International Journal of Financial Studies, 7*(2). https://doi.org/10.3390/ijfs7020027

Almujamed, H. I. (2019). Filter rule performance in an emerging market: Evidence from Qatari listed companies. *International Journal of Productivity and Performance Management, 68*(1), 231–247. https://doi.org/10.1108/IJPPM-04-2018-0129

Bäck, T., & Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation, 1*(1), 1–23. https://doi.org/10.1162/EVCO.1993.1.1.1

Bektur, Ç., & Aydın, M. (2019). Borsa İstanbul ve alt endekslerinde zayıf formda piyasa etkinliğinin analizi: Fourier yaklaşımı. *Akademik İncelemeler Dergisi, 14*(2), 59–76. https://doi.org/10.17550/akademikincelemeler

Benecki, P., Piechaczek, S., Kostrzewa, D., & Nalepa, J. (2021). Detecting anomalies in spacecraft telemetry using evolutionary thresholding and LSTMs. *GECCO 2021 Companion - Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion*, 143–144. https://doi.org/10.1145/3449726.3459411.

Bulut, U. (2016). Testing the weak-form of the efficient market hypothesis: The case of Turkey. *EconWorld2016.*

Chiang, H. Sen, Chen, M. Y., & Huang, Y. J. (2019). Wavelet-based EEG processing for epilepsy detection using fuzzy entropy and associative petri net. *IEEE Access*, *7*, 103255–103262. https://doi.org/10.1109/ACCESS.2019.2929266.

Choi, S., & Kim, Y. J. (2021). Artificial neural network models for airport capacity prediction. *Journal of Air Transport Management*, *97*(June 2019), 102146. https://doi.org/10.1016/j.jairtraman.2021.102146.

Corrado, C. J., & Lee, S.-H. (1992). Filter rule tests of the economic significance of serial dependencies in daily stock returns. *The Journal of Financial Research, XV*(4).

Coşkun, Y., & Seven, Ü. (2016). Efficiency of Financial Markets. In *Financial Markets and Institutions: Theory and Practice in Turkey* (pp. 287–319).

Dbouk, W., Jamali, I., & Soufani, K. (2014). The effectiveness of technical trading for Arab stocks. *Emerging Markets Finance and Trade, 50*(4), 5–25. https://doi.org/10.2753/REE1540-496X500401

Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic algorithm. *Artif Intell Rev, 36*, 153–162. https://doi.org/10.1007/s10462-011-9208-z

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering, 137*, Article 106040. https://doi.org/10.1016/J.CIE.2019.106040

Dreyfus, G. (2004). Neural Networks, Methodology and Applications. In *Neural Networks*. https://doi.org/10.1007/3-540-28847-3.

Eiben, A. E., & Smit, S. K. (2011). Evolutionary Algorithm Parameters and Methods to Tune Them. In *Autonomous Search* (Vol. 9783642214, pp. 15–36). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21434-9_2.

Fama, E. F. (1970). American Finance Association Efficient Capital Markets : A Review of Theory and Empirical Work Author (s): Eugene F . Fama Source : The Journal of Finance , Vol . 25 , No . 2 , Papers and Proceedings of the Twenty- Eighth Annual Meeting of the American. *The Journal of Finance*, *25*(2), 383–417.

Fama, E. F., & Blume, M. E. (1966). Filter rules and stock-market trading. *Source: The Journal of Business, 39*(1).

Fifield, S. G. M., Power, D. M., & Donald Sinclair, C. (2005). An analysis of trading strategies in eleven European stock markets. *European Journal of Finance, 11*(6), 531–548. https://doi.org/10.1080/1351847042000304099

Gauthama Raman, M. R., Somu, N., Kirthivasan, K., Liscano, R., & Shankar Sriram, V. S. (2017). An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems, 134*, 1–12. https://doi.org/10.1016/J.KNOSYS.2017.07.005

Gen, M., & Cheng, R. (1999). Genetic Algorithms and Engineering Optimization. *In Genetic Algorithms and Engineering Optimization*. https://doi.org/10.1002/9780470172261

Gozbasi, O., Kucukkaplan, I., & Nazlioglu, S. (2014). Re-examining the Turkish stock market efficiency: Evidence from nonlinear unit root tests. *Economic Modelling, 38*, 381–384. https://doi.org/10.1016/j.econmod.2014.01.021

Hailu, S. M., & Vural, G. (2020). Testing the weak form market efficiency of Borsa Istanbul: An empirical evidence from Turkish banking sector stocks. *Pressacademia, 7* (3), 236–249. https://doi.org/10.17261/pressacademia.2020.1291

Holland, J. H. (1992). Adaptation in natural and artificial systems. *Adaptation in Natural and Artificial Systems*. https://doi.org/10.7551/mitpress/1090.001.0001

Huang, Y.-S. (1995). The trading performance of filter rules on the Taiwan Stock Exchange. *Applied Financial Economics, 5*.

Jabari Lotf, J., Abdollahi Azgomi, M., & Ebrahimi Dishabi, M. R. (2022). An improved influence maximization method for social networks based on genetic algorithm. *Physica A: Statistical Mechanics and Its Applications, 586*, Article 126480. https://doi.org/10.1016/J.PHYSA.2021.126480

Karademir, F., & Evci, S. (2020). Borsa İstanbul'da zayıf formda piyasa etkinliğinin test edilmesi: Sektörel çerçevede bir analiz. *Business & Management Studies: An International Journal, 8*(1), 82–100.

Kaya, M., & Alhajj, R. (2005). Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets and Systems*. https://doi.org/10.1016/j.fss.2004.09.014

Kramer, O. (2017). Genetic algorithm essentials. *Springer International Publishing AG*. https://doi.org/10.1007/978-3-319-52156-5

Leung, F. H. F., Lam, H. K., Ling, S. H., & Tam, P. K. S. (2003). Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks, 14*(1), 79–88. https://doi.org/10.1109/TNN.2002.804317

Lin, L., Cao, L., Wang, J., & Zhang, C. (2004). The Applications of Genetic Algorithms in Stock Market Data Mining Optimisation. *WIT Transactions on Information and Communication Technologies*. https://doi.org/10.2495/DATA040261

López-González, A., Meda Campaña, J. A., Hernández Martínez, E. G., & Contro, P. P. (2020). Multi robot distance based formation using Parallel Genetic Algorithm. *Applied Soft Computing Journal, 86*. https://doi.org/10.1016/J.ASOC.2019.105929

Metghalchi, M., Durmaz, N., Cloninger, P., & Farahbod, K. (2021). Trading rules and excess returns: Evidence from Turkey. *International Journal of Islamic and Middle Eastern Finance and Management*. https://doi.org/10.1108/IMEFM-01-2020-0043

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer.

Moosa, I. A. (2000). *Exchange Rate Forecasting: Techniques and Applications*. MacMillian Press Ltd.

Mújica-Vargas, D. (2021). Superpixels extraction by an Intuitionistic fuzzy clustering algorithm. *Journal of Applied Research and Technology, 19*(2), 140–152. https://doi.org/10.22201/ICAT.24486736E.2021.19.2.1581

Olowe, R. A. (1999). *Weak Form Efficiency of the Nigerian Stock Market: Further Evidence*.

Pereira, R. (2002). Forecasting Ability But No Profitability: An Empirical Evaluation of Genetic Algorithm-Optimised Trading Rules. In *Evolutionary Computation in Economics and Finance* (pp. 287–311). Springer. https://doi.org/10.2139/ssrn.2735939.

Rodriguez, F. J., García-Martinez, C., & Lozano, M. (2012). Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: Taxonomy, comparison, and synergy test. *IEEE Transactions on Evolutionary Computation, 16*(6), 787–800. https://doi.org/10.1109/TEVC.2012.2182773

Rubio, J. D. J. (2021). Stability Analysis of the Modified Levenberg-Marquardt Algorithm for the Artificial Neural Network Training. *IEEE Transactions on Neural Networks and Learning Systems, 32*(8), 3510–3524. https://doi.org/10.1109/TNNLS.2020.3015200

Rubio, J. de J., Islas, M. A., Ochoa, G., Cruz, D. R., Garcia, E., & Pacheco, J. (2022). Convergent newton method and neural network for the electric energy usage prediction. *Information Sciences*, *585*, 89–112. https://doi.org/10.1016/J.INS.2021.11.038.

Rubio, J. de J., Lughofer, E., Pieper, J., Cruz, P., Martinez, D. I., Ochoa, G., Islas, M. A., & Garcia, E. (2021). Adapting H-infinity controller for the desired reference tracking of the sphere position in the maglev process. *Information Sciences*, *569*, 669–686. https://doi.org/10.1016/J.INS.2021.05.018.

Samani, N., Gohari-Moghadam, M., & Safavi, A. A. (2007). A simple neural network model for the determination of aquifer parameters. *Journal of Hydrology, 340*(1–2), 1–11. https://doi.org/10.1016/J.JHYDROL.2007.03.017

Saymeh, A. A. F. (2013). Empirical Testing for Weak Form Hypothesis of Emerging Capital Markets : A Comparative study of Jordan ' s ASE and Turkey ' s BORSA IST. *International Journal of Empirical Finance, 1*(2), 20–26.

Sevim, Ş., Yıldız, B., & Akkoç, S. (1997). Overreaction Hypothesis and an Emprical Work on the Istanbul Stock Exchange. *The Istanbul Stock Exchange Review, 9*(35), 21–36.

Sharapov, R. R. (2007). Genetic Algorithms: Basic Ideas, Variants and Analysis. In *Vision Systems: Segmentation and Pattern Recognition*. https://doi.org/10.5772/4971.

Sivanandam, S. N., & Deepa, S. N. (2008). Introduction to genetic algorithms. *Introduction to Genetic Algorithms*. https://doi.org/10.1007/978-3-540-73190-0

Sweeney, R. J. (1988). Some new filter rule tests: methods and results. *The Journal of Financial and Quantitative Analysis, 23*(3), 285. https://doi.org/10.2307/2331068

Tao, Z., Huiling, L., Wenwen, W., & Xia, Y. (2019). GA-SVM based feature selection and parameter optimization in hospitalization expense modeling. *Applied Soft Computing, 75*, 323–332. https://doi.org/10.1016/J.ASOC.2018.11.001

Ting, T. O., Yang, X. S., Cheng, S., & Huang, K. (2015). Hybrid metaheuristic algorithms: Past, present, and future. *Studies in Computational Intelligence, 585*, 71–83. https://doi.org/10.1007/978-3-319-13826-8_4/TABLES/2

Wu, C. H., Tzeng, G. H., Goo, Y. J., & Fang, W. C. (2007). A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications, 32*(2), 397–408. https://doi.org/10.1016/J.ESWA.2005.12.008

Wu, C. H., Tzeng, G. H., & Lin, R. H. (2009). A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications, 36*(3), 4725–4735. https://doi.org/10.1016/J.ESWA.2008.06.046

Yan, H., Jiang, Y., Zheng, J., Peng, C., & Li, Q. (2006). A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications, 30*(2), 272–281. https://doi.org/10.1016/j.eswa.2005.07.022

Yen, S. M. F., & Hsu, Y. L. (2010). Profitability of technical analysis in financial and commodity futures markets - A reality check. *Decision Support Systems, 50*(1), 128–139. https://doi.org/10.1016/j.dss.2010.07.008

Yucel, Ö. (2016). Finansal Piyasa Etkinliği: Borsa İstanbul Üzerine Bir Uygulama. *International Review of Economics and Management, 4*(3). https://doi.org/10.18825/irem.16916

Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications, 38*(5), 5197–5204. https://doi.org/10.1016/j.eswa.2010.10.041