

A Hybrid Artificial Intelligence Approach to Portfolio Management

Hamidreza Haddadian

Ph.D. Candidate, Department of Financial Management, Faculty of Management, Central Branch, Islamic Azad University, Tehran, Iran. (Email: hr.haddadian@gmail.com)

Mortaza Baky Haskuee*

*Corresponding Author, Visiting Professor and Research Fellow, York University, School of Liberal Art and Professional Studies, ON, Canada. (Email: mbaky@yorku.cam)

Gholamreza Zomordian

Assistant Prof., Department of Business Management, Central Branch, Islamic Azad University, Tehran, Iran. (Email: gh.zomorodian@gmail.com)

Document Type: Original Article

Received: 2021/05/23

Accepted: 2021/11/28

2022, Vol. 6, No. 1. 1-27.

Published: 2022/02/09

Abstract

The tremendous advances in artificial intelligence over the past decade have led to their increasing use in financial markets. In recent years a large number of investment companies and hedge funds have been implementing algorithmic and automated trading on their trading. The speed of decision-making and execution is the most important factor in the success of institutional and individual investors in capital markets. Algorithmic trading using machine learning methods has been able to improve the performance of investors by finding investment opportunities as well as time entry and exit of trading. The purpose of this study is to achieve a better portfolio performance by designing an intelligent and fully automated trading system that investors with the

support of this system, in addition to finding the best opportunities in the market, can allocate resources optimally. The present study consists of four separate steps. Respectively, tuning the parameters of technical indicators, detecting the current market regime (trending or non-trending), issuing a definite signal (buy, sell or hold) from the indicators' signals and finally portfolio rebalancing. These 4 steps respectively are performed using genetic algorithm, fuzzy logic, artificial neural network and conventional portfolio optimization model. The results show the complete superiority of the proposed model in achieving higher returns and less risk compared to the performance of the TEDPIX and other mutual funds in the same period.

Keywords: Portfolio Optimization, Artificial Intelligence, Algorithmic Trading, Trading Systems, Genetic Algorithm, Technical Analysis, Neural Network.

DOI: <https://doi.org/10.30699/IJF.2021.287131.1237>

Publisher: Iran Finance Association

Copyright: author(s)

Type of License: Creative Commons License (CC-BY 4.0)

Introduction

The stock market is always one of the most popular investment sectors due to its high returns. However, some factors have made the behaviour of this market unpredictable. So far, many models have been proposed using different tools and techniques to predict stock market behaviour. Since accurate forecasting of stock price changes can lead to significant benefits, it is important to provide a model for intelligently forecasting the market and allocating a portfolio based on it.

Success in stock trades extensively depends on proper decisions for on-time and correct enters and exits. This issue requires the investigation of information as well as a specialization for exploring investment opportunities. In recent decades, researchers have sought scientific and quantitative methods to obtain better results in financial markets. However, owing to many reasons, the majority of traders are incapable of using scientific analysis in trades. Hence, in recent years, there is a growing need for automated approaches to use financial data effectively to support investment decisions. . One of the most attractive fields for the use of artificial intelligence is stock trading and stock selection systems. The stock trade system, employed as an auxiliary tool in investment decisions, is reckoned as a novel research domain in the world. It

enjoys a thorough potential in increasing profitability besides its research potential .

This study aims to develop an intelligent trading system that exploits the conventional technical analysis tools and meta-heuristic algorithms to generate correct trading signals in the return points of trends. Therefore, backed by this system, investors can make timely and correct decisions in their trades and maximize the profitability of their investments .

This study comprises four separate phases. The indicators' parameters will be optimized by the genetic algorithm in the first phase. Each indicator has many input variables that are static for every stock and market. Thus, the computing complexity of this problem compels us to employ one of the meta-heuristic algorithms, such as the genetic algorithm to customize and optimize indicators for each stock. In the second phase, the market's current trend will be identified by the fuzzy inference system since a set of technical rules will be selected according to the market regime (neutral or trending market). In the third phase, the variables are aggregated, and with the help of a neural network, the final output signal will be presented as a buy, sell, or hold. Each indicator may generate a different signal. Therefore, a single answer can be generated by the neural network for decision making.

Finally, in the last phase, the portfolio is rebalanced by the candidate list of buying or selling. This optimal allocation is done by optimizing the risk and return along with using a few real constraints.

Literature Review

During the past recent decades, many researchers have employed soft-computing technics, including the genetic algorithm and artificial neural networks in mathematical problems (Goldberg, 1998). These methods were successfully applied to solving diverse problems. Soft-computing technics were employed in making many smart systems. In the meantime, one of their applicative domains is financial problems, such that many researchers are now investigating the applications of soft-computing technics (it is better to use "techniques" in formal settings) in the financial domain (Baba et al., 2001).

The conventional technical analysis, which is the base of many trading systems, usually emerges in the forms of different trading rules and concentrates on the peak and bottom points of prices in trends, and it is extensively prevalent in the world's markets. Unfortunately, the stock trading systems that are based on using merely the technical rules will be no more

suitable owing to their dependence on traders' experiences in selecting decision-making rules. This is the biggest drawback of using this type of analysis. The main problem is that the technical rules are based on some parameters. If technical parameters are truly and smartly determined, they will deliver appropriate buy and sell signals for the investor's profitability. Another problem is that different technical rules may simultaneously create contradictory trading signals and problematize making decisions in such conditions.

Refenes et al. (1994) modelled the behaviour of stock prices via neural networks and compared its performance to regression models. In this study, the researchers employed neural networks as an alternative for the classic statistical technics to predict the price of stocks. The results of this study showed that neural networks performed better than statistical methods and provided better models.

Tan et al. (1995) designed a system that predicted the considerable short-term changes in the stock price. At first, a pre-processing was conducted on the data, and then the neural network was modelled which estimated trading positions well.

Kuo et al. (2001) developed a prediction system regarding stocks hold, sell, and buy in the stock exchange markets. The characteristic of the developed system was that it allowed the quantification of qualitative variables interfering with the prediction of stock prices. These researchers have published a paper with the same title in 1998 without using genetic algorithms. In the mentioned paper, the researchers used a questionnaire with the Fuzzy Delphi Method (FDM) to make use of experts' opinions in predicting stock prices.

Yim (2002) conducted a study comparing the econometric methods using ARMA and GARCH Models. The results showed the superiority of neural networks to the classic ARMA and GARCH models. In a study, Nouri et al. (2014) compared artificial neural networks (ANN) and autoregressive integrated moving average model (ARIMA) in the modelling and short-term prediction of the exchange rate in Iran. Their results revealed that the employed neural network enjoyed a better prediction power compared to the ARIMA model. In another study, Foroughi et al (2014) predicted the profit of every stock according to the combination of artificial neural networks and the optimization algorithm of particle aggregative movement. Their results showed that the univariate and multivariate models predicted every stocks' profit with 78% and 92% precisions, respectively. Souto-Mairo (2011) used fuzzy logic to

predict the moving direction of the Brazil Stock Price Index, and their predictive results were finally evaluated to be suitable.

Alejandro Rodríguez et al. (2011) exploited neural networks to improve technical analysis indicators. This Study used the genetic algorithm to improve the prediction parameters used in technical analysis and enhance the ENS networks via improved parameters for the purpose of predicting turnover points. The results of studies show that the employed approach performs better than the buy and hold policy (Lin et al., 2011).

Rahnamay et al. (2015) adopted a genetic algorithm approach to optimize a portfolio comprising the mutual funds stock of the Tehran Stock Exchange. The results indicated that the genetic algorithm could be more desirable for selecting the portfolios developed by the genetic algorithm. Tehrani and Abbasian (2008) used only a neural network tool to investigate the schedule of entering the stock trades with a technical approach. Their results revealed that the system's performance was only suitable for Bear markets. In Bull markets, there was not a significant difference between the proposed trading system and the hold and buy method.

Brogaard et al. (2018) investigate high-frequency trading around extreme price movements, classified as the top 0.1% of the individual stock price changes over 10-s trading intervals. The authors find that the net effect of high-frequency trading liquidity demand and supply is negatively associated with the direction of the price changes. Michell et al. (2020) used An embedded approach of STPG and a Fuzzy inference system to generate trading rules. They used a fuzzy inference system (FIS) and strongly typed genetic programming (STGP) to generate trading rules for the US stock market, a framework called FISTGP. The results show that the proposed model outperforms the Buy and Hold (B&H) strategy by 28.62% in the test period, considering excesses of return, with almost the same risk (1.28% higher). Manahov (2018) studied the effect of high-frequency data with an STGP approach, to see if it is beneficial or not to the market, finding that it is actually harmful to institutional traders, on a millisecond basis. Ha & Moon (2018) proposed GP to identify patterns in the cryptocurrency market. They focus the study on applying the signals in the market to increase investment return. Based on several experiments, they show that GP can create useful rules for trading, performing well in the tests.

Rastegar and Sedaghatipour (2019) also presented an algorithmic trading system for gold coin futures contracts based on overnight data. Due to the fact

that technical analysis is suitable for two-way markets (buying and selling position), Signal 8 technical tools have been used for the trading system. In order to create a trading system, the MOPSO algorithm has been used with the aim of optimizing the two functions of return and conditional value at risk (CVaR). Also, to complete the risk management system, the optimal profit and loss limit for the futures contract has been set. The results show that the designed trading system has a more return-to-risk ratio than other competing strategies such as buying and holding and selling and holding.

Baky et al. (2021) have tried to create an intelligent trading system based on technical analysis Using meta-heuristic algorithms and machine learning. Using a genetic algorithm, they optimized the parameters of technical indicators and using an artificial neural network converted the signals of indicators into buying or sell orders. The results show that there is a statistically significant difference between stocks buy and hold and the trading system presented by their research.

Research Methodology

This study comprises four separate phases. The variables will be optimized by the genetic algorithm in the first phase. There is a minimum m analytical rule, each of which includes n parameters for presenting signals. Thus, the computing complexity of this problem compels us to employ one of the meta-heuristic algorithms such as the genetic algorithm. In the second phase, the market's current trend will be identified by the fuzzy inference system since a set of technical rules will be selected according to the kind (Trending or non-trending). In the third phase, the variables are aggregated, and with the help of a neural network, the final output signal will be presented as a buy, sell, or hold signal.

In the last phase, the candidate list for buying and selling will be allocated and the portfolio will be rebalanced by one of the portfolio optimization models. The general structure of research is described in figure 1 including the construction and training model and figure 2 including the implementation of the trading system.

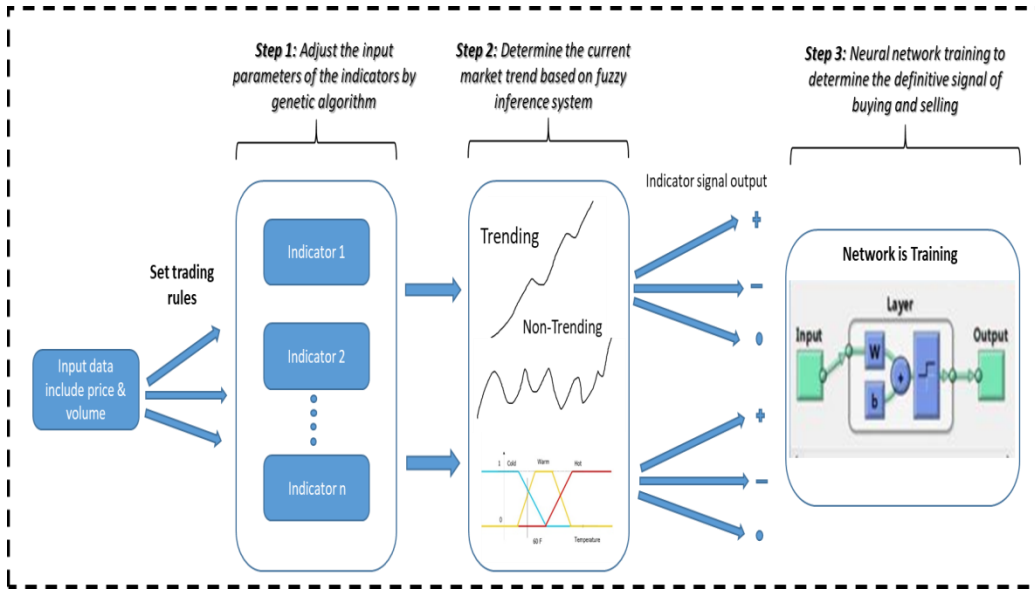


Figure 1. The first section includes construction & training of the model

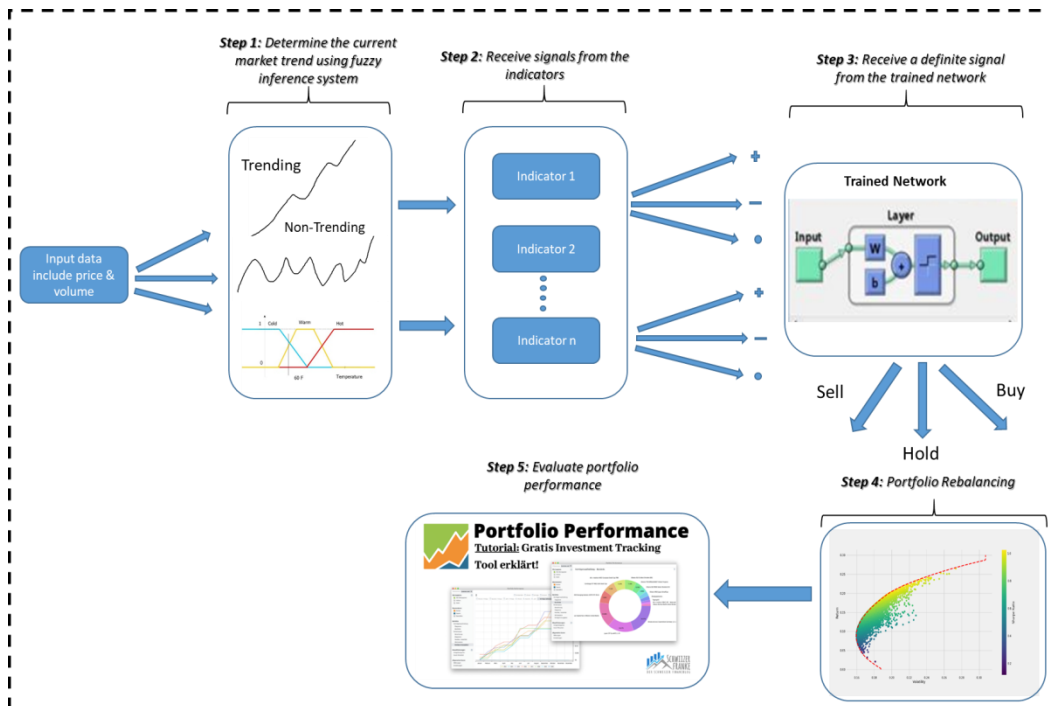


Figure 2. The second section includes implementation and evaluation

Parameters Optimization

The first step is to adjust the parameters used in the technical indicators. A change in the used parameters issues varying responses. Commonly, experts determine the parameters empirically and by trial and error based on their personal experiences. For example, a 10-day moving average may offer relatively correct and on-time signals for a certain stock, whereas a longer interval is needed for another stock. Accordingly, we must define an algorithm to properly adjust the parameters of the mentioned variables for every stock. The genetic pattern is one of the most suitable algorithms in terms of discovering correct and quick responses.

Suppose the sequence of the expected trading points in which asset buy and sell signals exist is like T ($T_1, T_2, T_3, \dots, T_n$). For every expected trading point (T_i), we will look for operational signals (S_j) proposed by the different rules of the technical analysis. Evidently, every signal is placed between the two expected trading points ($T_{i-1} < S_j < T_{i+1}$).

According to the type of the trading point; i.e. a buy trading point and sell trading point, we will have the following conditions for the fitness function:

If T_i is an anticipated trading point for asset buying, there will be three conditions for the fitness function as below:

If S_j (trading signal) is a buy proposal, the magnitude of the fitness function at the T_i point will be computed as follows:

$$Fitness(T_i) = Close(S_j) - Close(T_i) \quad (1)$$

In recent trades:

Close (T_i) indicates the closing price at T_i .

Close (S_j) indicates the closing price at S_j .

The more the closing price at S_j is closer to the closing price at T_i , the smaller the magnitude of the fitness function.

If S_j is a proposed sell signal, and the price at S_j is close to the price at T_i , the magnitude of the fitness function will be computed by the below Equation:

$$Fitness(T_i) = 2 \times (\max_{j \in [T_{i-1}, T_{i+1}]} \{ |Close(T_{i-1}) - Close(T_{i+1})| \}) - Close(T_i)$$

$$Fitness(T_i) = 2 \times (\max(close(T_{i-1} : T_{i+1})) - Close(T_i)) \quad (2)$$

The closeness term in the current equation is as below:

$$\frac{|close(S_j) - close(T_i)|}{close(T_i)} < 0.05 \quad (3)$$

In the above fitness function, $\max(close(T_{i-1} : T_{i+1}))$, meaning maximum closing price, is between T_{i-1} and T_{i+1} . Since asset selling in minimum price points of a trend is a wrong trading policy, such a decision will accompany a penalty.

If no operation between T_{i-1-1} and T_{i+1+1} is proposed, a penalty will be earmarked in the model for losing a trading opportunity. In this case, the fitness function will be as below:

$$Fitness(T_i) = \max(close(T_{i-1} + 1 : T_{i+1} - 1)) - close(T_i) \quad (3)$$

Similarly, if T_i is an expected sell point, the fitness function will be a penalty as below:

If S_j is a proposed sell signal, the fitness function at T_i will be as below:

$$Fitness(T_i) = close(T_i) - Close(S_j) \quad (5)$$

If the price at S_j is close to the price at T_i , and S_j is erroneously considered as a buy point, the fitness function will be as below:

$$Fitness(T_i) = 2 \times Close(T_i) - \min(close(T_{(i-1)+1} : T_{(i+1)-1})) \quad (6)$$

Such that $\min(close(T_{(i-1)+1} : T_{(i+1)-1}))$, which means minimum closing price, is between $T_{(i-1)+1}$ and $T_{(i+1)-1}$.

If the trading opportunity is lost in the between $T_{(i-1)+1}$ and $T_{(i+1)-1}$ interval, the fitness function will be as below:

$$Fitness(T_i) = Close(T_i) - \min(close(T_{(i-1)+1} : T_{(i+1)-1})) \quad (7)$$

Finally, the fitness function of the trading points sequence $S = \{S_1, S_2, \dots, S_n\}$ is defined as below:

$$Fitness(S) = \sum_{i=1}^n [fitness(T_i)] \quad (8)$$

It is anticipated that the genetic algorithm can deliver an optimal combination

of parameters such that the magnitude of the fitness function is minimized.

Below table 1 shows the formula of indicators used in the research as well as their parameters that need to be optimized. Table 1 shows the indicators that have been used as a trending market and table 2 shows indicators that have been used in the non-trending or range market.

Table 1. Indicators list have been used in trending market

Indicator	Formula	Signal Condition	Optimized Parameters
Moving Average	$SMA_t = \frac{\sum_{k=i-n}^i x_k}{n}$	Price above SMA shows upward market & conversely shows downside market	n
Parabolic SAR	In Bull Market: $SAR(i) = SAR(i-1) + ACCELERATION * (HIGH(i-1) - SAR(i-1))$ In Bear Market: $SAR(i) = SAR(i-1) + ACCELERATION * (LOW(i-1) - SAR(i-1))$	Price above SAR shows upward market & conversely shows downside market	Acceleration factor
MACD	$Z_t = MA(N) - MA(n)$ MA(N): longer moving average MA(n): shorter Moving average	The histogram above zero shows upside market & below zero shows downside market	N & n
ADX	$-DI = \frac{S - DM}{ATR} \quad +DI = \left(\frac{S + DM}{ATR} \right) \times 100$	The bulls have the edge when +DI is greater than -DI, while the bears have the edge when -DI is greater.	DI Length
Ichimoku	Conversion Line (kenkan sen) = $\frac{9-PH + 9-PL}{2}$ Base Line (kijun sen) = $\frac{26-PH + 26-PL}{2}$ Leading Span A (senkou span A) = $\frac{CL + Base Line}{2}$ Leading Span B (senkou span B) = $\frac{52-PH + 52-PL}{2}$ Lagging Span (chikou span) = Close plotted 26 periods in the past	As long as the price is higher than the cloud, the general trend is upward, and if the price is lower than the cloud, it is a downward trend, and if the price is inside the cloud, it is fluctuating.	Period of PH & PL in Conversion Line and Base Line

Table 2. Indicators lists have been used in non-trending market

Indicator	Formula	Signal Condition	Optimized Parameters
RSI	$RSI_{\text{step one}} = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}} \right]$	RSI above 80 markets is overbought & RSI below 20 markets is oversold.	period
Stochastic	$\%K = \frac{\text{Price} - \text{Low}_N}{\text{High}_N - \text{Low}_N} \times 100$ $\%D = \frac{\%K_1 + \%K_2 + \%K_3}{3}$	Stochastic above 80 market is overbought & Stochastic below 20 markets is oversold.	Smooth D
			Smooth K
			Length
MFI	$MFI = 100 - \frac{100}{1 + \text{money ratio}}$	MFI above 80 markets is overbought & MFI below 20 markets is oversold	n-period
Williams R%	$\%R_i = \frac{\text{close}_i - \max_{k \in [i-n, i]} \text{high}_k}{\max_{k \in [i-n, i]} \text{high}_k - \min_{k \in [i-n, i]} \text{low}_k} * 100$	R% above 80 markets is overbought & R% below 20 markets is oversold	n
CCI	$CCI_i = \frac{TP_i - SMA_{\text{period}}(TP_i)}{0.015 * \frac{\sum_{j=i-p+1}^i TP_j - SMA_{\text{period}}(TP_j) }{\text{period}}}$	CCI above 80 markets is overbought & CCI below 20 markets is oversold	period

Identifying the Market Regime

Every variable used in this study issues a proper signal according to a trending or non-trending market. For example, the stochastic oscillator will not enjoy the necessary efficiency in an up-trending or down-trending market. However, this oscillator can be more properly applied to a non-trending market. Accordingly, it is necessary to determine the kind and current situation of the market before using these kinds of variables.

However, determining a market's situation is reckoned as the main challenge of the capital market activists. It is because there is no firm criterion for it, such that an expert may perceive a market situation as non-trending, and another one may perceive it as a trend. This problem is an intuitive issue, and we can define no definiteness for it. One of the strong mathematical tools

extensively used as an uncertainty explanation and description is the fuzzy logic theory. It is a novel tool for solving problems that the probability theory does not offer a solution.

Fuzzification

In this phase, we consider membership functions for every input variable to transform crisp inputs to fuzzy ones and place them in the fuzzy inference system. The membership functions are of different kinds, such as triangular, trapezoidal, arch, etc. In this study, we use a trapezoidal kind. Here, the three variables, including the distance from moving average (d), the slope of moving average (s), and the Index of average directional movement (that is a kind of Technical indicator) are proposed as input linguistic variables.

All these variables are thermalized in a numerical interval between -1 and 1 according to the below Equation. For normalizing or fuzzification of numbers in fuzzy theory we should use the below equation:

$$v^{*} = (2v - (\max_{i \in I}(v) + \min_{i \in I}(v))) / (\max_{i \in I}(v) - \min_{i \in I}(v)) \tag{9}$$

Where $\max(v)$ is the maximum value in the training data, and $\min(v)$ is the minimum value in the total training data. The below Figure illustrates the membership function applied in this study, and var_a , var_b , var_c , and var_d indicate the linguistic value of the linguistic variables.

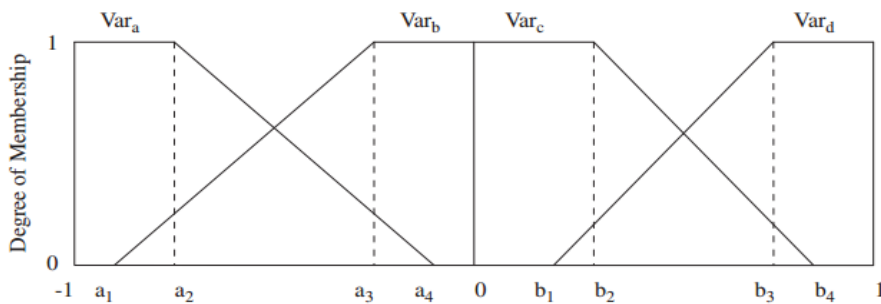


Figure 3. Trapezoidal Membership Function for the Input Linguistic Variables

a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4 are the parameters that specify the shape of the membership function. Table 3 outlines the selected interval for the above-mentioned parameters.

Table 3. Selected Intervals for the Parameters of the Membership Function

Parameter	Lower bound	Upper bound
a1	-1	-0.5
a2	-1	-0.5
a3	-0.5	0
a4	-0.5	0
b1	0	0.5
b2	0	0.5
b3	0.5	1
b4	0.5	1

In the following, we present the membership functions employed in the study to determine the rate of membership degree:

$$\mu_{var_a}(x) = \begin{cases} 1 & \text{if } x \leq a_2 \\ 0 & \text{if } x \geq a_4 \\ 1 + (x - a_2) \left(\frac{-1}{a_4 - a_2} \right) & \text{if } a_2 < x < a_4 \end{cases} \quad (10)$$

$$\mu_{var_b}(x) = \begin{cases} 1 & \text{if } x \leq a_1 \\ 0 & \text{if } x \geq a_3 \\ \left(\frac{x - a_1}{a_3 - a_1} \right) & \text{if } a_1 < x < a_3 \end{cases} \quad (11)$$

$$\mu_{var_c}(x) = \begin{cases} 1 & \text{if } x \leq b_2 \\ 0 & \text{if } x \geq b_4 \\ 1 + (x - b_2) \left(\frac{-1}{b_4 - b_2} \right) & \text{if } b_2 < x < b_4 \end{cases} \quad (12)$$

$$\mu_{var_d}(x) = \begin{cases} 1 & \text{if } x \leq b_1 \\ 0 & \text{if } x \geq b_3 \\ \left(\frac{x - b_1}{b_3 - b_1} \right) & \text{if } b_1 < x < b_3 \end{cases} \quad (13)$$

Table 4 illustrates the characteristics of all linguistic variables that are employed in this system.

Table 4. A Summary of the Conditions of Linguistic Variables

Linguistic Value	Interval	Symbol	Linguistic Value
(vara) large (varb) small (varc) small (vard) large	[-1, 0] [0, 1]	-d +d	D (Mat – Pt)
(vara) high slope (varb) low slope (varc) high slope (vard) low slope	[-1, 0] [0, 1]	-s +s	SMA(Mat– Mat-10)
(vara) strong (varb) weak (varc) weak (vard) strong	[-1, 0] [0, 1]	-adx +adx	Adx
(vara) directional (varb) neutral (varc) neutral (vard) directional	[-1, 0] [0, 1]	neutral directional	Situation

Rule Base

The rule base is defined as a fuzzy "if-then" set, which comprises the heart of the fuzzy inference system. There are two main methods for determining fuzzy rules: one is to use expert knowledge, and the other is to use self-organized knowledge such as novel algorithms and neural networks. This study uses the first method to determine fuzzy rules.

The knowledge base for fuzzy inference in this system is defined as below:

Rule 1: If +d is large and s- is small; then, the trend will be neutral.

Rule 2: If d- is small and –s is large; then, the trend will be directional.

Rule 3: If d- is small, –s is small, and adx is strong; then, the trend will be directional.

Rule 4: If –d is small, s- is small, and adx is weak; then, the trend will be neutral.

Rule 5: If d+ is small, –s is large, and adx is strong; then, the trend will be directional.

Rule 6: If d+ is small, –s is small, and adx is strong, then, the trend will be directional.

Rule 7: If d+ is small, –s is small, and adx is weak, then, the trend will be neutral.

Rule 8: If d- is large, -s is small, and adx is weak, then, the trend will be neutral.

Rule 9: If d+ is small and s+ is large, then, the trend will be directional.

Rule 10: If d+ is small, s+ is small, and adx is strong, then, the trend will be directional.

Rule 11: If d+ is small, s+ is small, and adx is weak, then, the trend will be neutral.

Rule 12: If -d is small and +s is large, then, the trend will be directional.

Rule 13: If -d is small, +s is small, and adx is strong, then, the trend will be directional.

Rule 14: If -d is small, +s is small, and adx is weak, then, the trend will be neutral.

Defuzzification

Defuzzification is a process that converts a fuzzy set into a crisp number. Thus, the input to the defuzzification process is a fuzzy set (the sum of output fuzzy sets), and its output is a number. Many methods are used for this purpose; however, in this study, we use the center of gravity (COG) method to create the crisp output.

$$\text{COG} = \frac{\int_a^b \mu_A(x) \cdot x \, dx}{\int_a^b \mu_A(x) \, dx} \quad (14)$$

The created crisp output is compared by the confidence interval that we have defined. Accordingly, if COG is larger than 0.6, the market will be trending; otherwise, we will face a non-trending market.

Creating the Final Signal

In the previous two phases, we dealt with optimizing and determining the market situation. In the second phase, after determining the market situation, we will take the output signals from the variables with respect to the separation developed among them. The variable kinds can be separated as below:

The variables used in a trending market:

Double moving average, directional average indicator, parabolic SAR, moving average convergence/divergence (MACD), and OBV indicator.

The variables used in a non-trending market:

Momentum index, relative strength indicator (RSI), stochastic index, MACD indicator, and money flow index (MFI).

Each of the above-mentioned indices will issue a signal that is not necessarily equal to another one. Hence, 5 issued signals should be integrated, and the tool used for this purpose is a neural network. Different indicators may produce different signals. Here, by application of the neural network as a classification problem, different signals are taken from Technical indicators and determine if a correct signal is a buy, sell or hold signal. The input to the network is a buy, sell or no decision signal, which we indicate with -1, 1 and 0 respectively. In fact, the signal of each Technical rule is a member of the network input array. Respectively output neurons are bought (+1), sell (-1) or hold (0).

We are using the Elman network in this Research. An Elman network is a three-layer network (arranged horizontally as x , y , and z in figure .4) with the addition of a set of context units (u in the illustration). The middle (hidden) layer is connected to these context units fixed with a weight of one. At each time step, the input is fed forward and a learning rule is applied. The fixed back-connections save a copy of the previous values of the hidden units in the context units (since they propagate over the connections before the learning rule is applied). Thus the network can maintain a sort of state, allowing it to perform such tasks as sequence-prediction that are beyond the power of a standard multilayer perceptron (15).

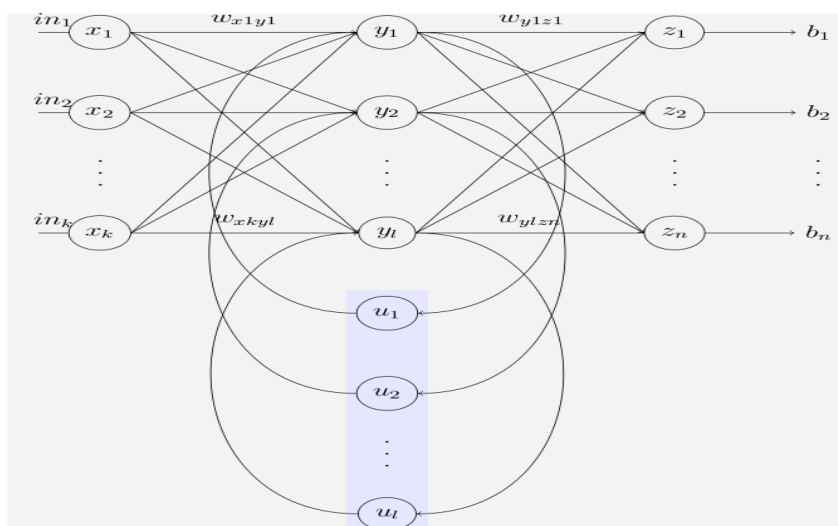


Figure 4. A View from Elman Network

Assume that x_t is an input vector, so the activation of hidden state h_t is updated according to:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad (15)$$

Where σ_h is the internal activation function (typically sigmoid functions). Also, w , U , b_h are parameter metrics and vectors. The network output is obtained through the following equation:

$$y_t = \sigma_y(W_y h_t + b_y) \quad (16)$$

Where σ_y is the output activation function,

The main difference between the two-layered Elman network and the other two-layered networks is that the connection in the first layer of the network is recurrent. The delay that occurs in this connection type triggers the storage of the past values as well as their uses in later steps. The feedback mechanism in this network is such that even two Elman networks with similar weights, biases, and inputs can have varying outputs.

Portfolio Rebalancing

Candidate stocks in the previous step must be allocated to enter the portfolio. Therefore, the portfolio should be re-weighted or so-called Rebalanced based on the amounts of money obtained from the sale of outgoing stocks and candidate stock for entry. For this purpose, the traditional Markovitz model is used.

In this research, to make the model more practical, the model has been improved by considering several real constraints. On the other hand with considering coefficient λ in the objective function we tried to include both risk and return criteria in the objective function to Maximize returns while minimizing risk. In fact, λ is only one weighting parameter whose value varies in the range (0,1) and varies by investor risk preferences. The constraints used in the model are described in Table 5.

Table 5. Constraints used in the model

Constraint	Equation	Parameter definition
Limit the number of equity in the portfolio	$k_{\min} \leq \sum_{i=1}^n z_i \leq k_{\max}$	$Z_i : 0, 1$ parameter
		K_{\min} : the minimum amount of assets in the portfolio
		K_{\max} : maximum amount of assets in the portfolio
Above and below allowable limit per share	$W_i \leq \delta_i Z_i \leq Z_i \square_i$	\square_i :the minimum amount allowed per share
		δ_i :the maximum amount allowed per share
Liquidity limitation for controlling the market impact	$W_i \leq Z_i . a . ADV$	ADV: average daily trading volume in last month
		a: percent of daily volume

Finally proposed model is rewritten as Equation (17):

$$\text{Max} Z = \lambda \sum_{i=1}^n w_i \mu_j - (1-\lambda) \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij}$$

Subject to :

$$\sum_{i=1}^n w_i = 1 \tag{17}$$

$$k_{\min} \leq \sum_{i=1}^n z_i \leq k_{\max}$$

$$W_i \leq \delta_i Z_i \leq Z_i \square_i$$

$$W_i \leq Z_i . a . ADV$$

Research Findings

In this section, the results of the implementation and execution of the proposed system are presented and compared. In addition to comparing with TEDPIX, the performance of the model is also compared with the performance of 3 of the best mutual funds that had the highest returns during the period under review.

For this comparison, in addition to the return criterion during the period, 4 well-known portfolio performance evaluation criteria have been used. ‘Sharpe ratio’, ‘Treyner ratio’, ‘Sortino’ as well as ‘maximum Drawdown’ have been

used for this purpose. Also, to evaluate the performance and efficiency of the second and fourth phases of the research, including the use of fuzzy logic to determine the current market trend and also portfolio rebalancing,

The model was performed once without the second phase and once without the fourth phase and the results were compared with the complete algorithmic model.

1. Assumptions

The assumptions used in the research include two parts. Assumptions during construction and training the model, and assumptions of testing and implementing the model.

The assumptions of the Training section are as follows:

1) To implement the model, data from the beginning of 1390(Solar Hijri) to the end of 1396 have been used as training data. In this period, we have witnessed a strong upward, downward, volatile and neutral market. The same period was divided into two parts of the Trending market and non-Trending market so that the indicators can be optimized correctly. From 1391/01/01 to 1391/06/31 as well as 1395/10/01 to 1396/06/31 were considered as a market without trend and 1391/07/01 to 1392/09/30 as well as 1394/12/29 as a market with a strong trend. This classification is more intuitive based on the Tehran Stock Index chart (figure 5). Accordingly, when the market was trending, we only optimized the indicators that followed the trend, and in the non-trending periods, we only optimized the oscillators.

2) In this research, the amount of 500 billion Rials is considered as the initial amount of the portfolio. This amount has been considered due to its comparability with other mutual funds.

3) To make the model more realistic, like mutual funds, when stocks are sold and there are unallocated resources to stocks in a bear market, we allocate them in fixed income securities in order to make the best use of resources. We set the bond rate at 20 percent. The minimum and maximum government bonds rates in this period were 19 to 25 percent and most of the time around 20. So we have considered 20 percent.

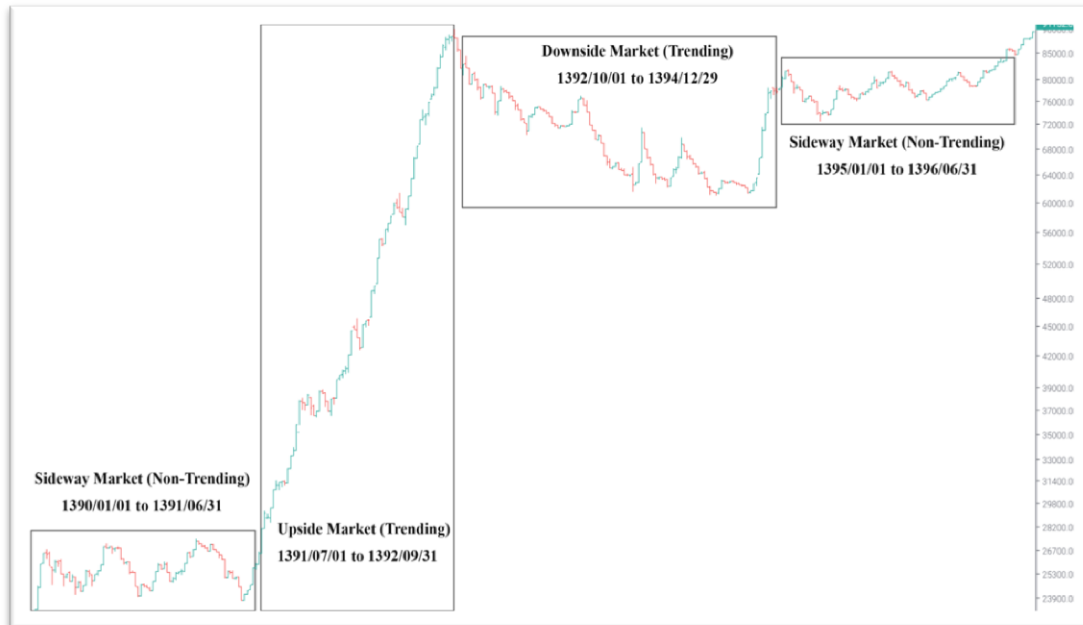


Figure 5. Classification of TEDPIX Trending

The assumptions of the second running model are as follows:

- 1) There will be weekly portfolio review and Rebalancing
- 2) The evaluation period of portfolio performance is from the beginning of 1397 to mid-1399.
- 3) The minimum number of portfolio assets is 10 and the maximum is 20.
- 4) The minimum allowable ratio for an asset is 3% of the total capital and the maximum is 10% (according to the investment fund management instructions by Securities & Exchange Organization).
- 5) 10% of the average daily trading volume of one symbol per month (ADV) is used as the liquidity constraint in the model.
- 6) The transaction fee is 0.5% in purchase transactions and 1% in sales transactions.

2. Performance evaluation

In this research as mentioned, five criteria of return, ‘Sharpe ratio’, ‘Treyner ratio’, ‘Sortino’ and ‘maximum drawdown’ have been used.

The Sharpe ratio is known for its reward to risk ratio. This ratio is based on capital market theory.

$$RVAR = \frac{TR_p - R_f}{SD_p} = \frac{\text{Excess Return}}{\text{Risk}} \tag{18}$$

Where TR_p is the average return on the total portfolio over a period of time, R_f is the average risk-free rate of return over the period under review, SD_p is the standard deviation of the daily returns on the portfolio over the period under review, and Excess Return is the additional return. Sharp ratio measures stock portfolio surplus returns per unit of risk, and the higher the ratio, the better portfolio performance. Treynor introduced a similar criterion called the reward-to-volatility ratio as below:

$$RVOR = \frac{TR_p - R_f}{\beta_p} \tag{19}$$

Where TR_p is the average return on the entire stock portfolio over a period of time, β_p is the systematic risk and indicates an investor's return on a unit of systematic risk. If the adverse risk criterion is used in the performance evaluation instead of the standard deviation, the Sortino index is achieved. In fact, if x is a portfolio return variable and the function $f(x)$ is a function of the probability density of this variable, and μ the mean and r the minimum acceptable rate of return, the Sortino index can be shown as a (20) ratio:

$$SOR = \frac{(\mu - r)}{\sigma} \tag{20}$$

In this study, the following equation has been used to calculate the adverse risk:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (\text{Max}\{\cdot, R_f - R_{ij}\})^2} \tag{21}$$

The maximum drawdown is the maximum amount that the portfolio loses its value from a peak value. This changing value in the portfolio can be expressed as a percentage according to Equation:

$$MDD = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}} \quad (22)$$

The weekly Time frame is used to calculate return and risk. The basis of the comparison period is from 1397/01/01 to 1398/06/30. The results according to the table below show the superiority of the algorithmic portfolio over other funds as well as the TEDPIX index. Also, the efficiency of the algorithm when using both the portfolio optimization model and fuzzy logic shows its complete superiority over the non-use mode. Figure 6.1 and 6.2 And Table 6 shows a comparative graph of the cumulative returns of the proposed algorithmic portfolio (in three different modes) as well as the TEDPIX index, the index of 50 top companies and the top three funds.

Table 6. Comparison of algorithmic portfolios with other funds in all criteria

cumulative Return(percent)	Name
۲۹۸۷	Complete Algorithmic portfolio
2760	Arzesh Kavan fund
2620	noRebalance_algorithmic
2080	Boursiran fund
۲۰۲۶	noTrend_algorithmic
2065	Khavaremiyane fund
1560	TEDPIX
1485	Index of 50 companies

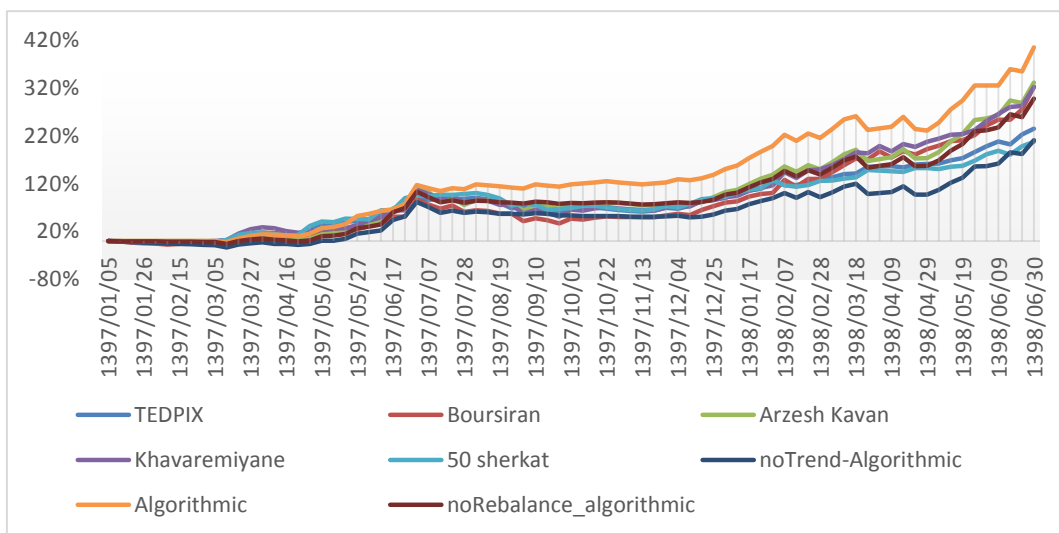


Figure 6.2 Comparison of return on net asset value from 1397/01/01 to 1398/06/30

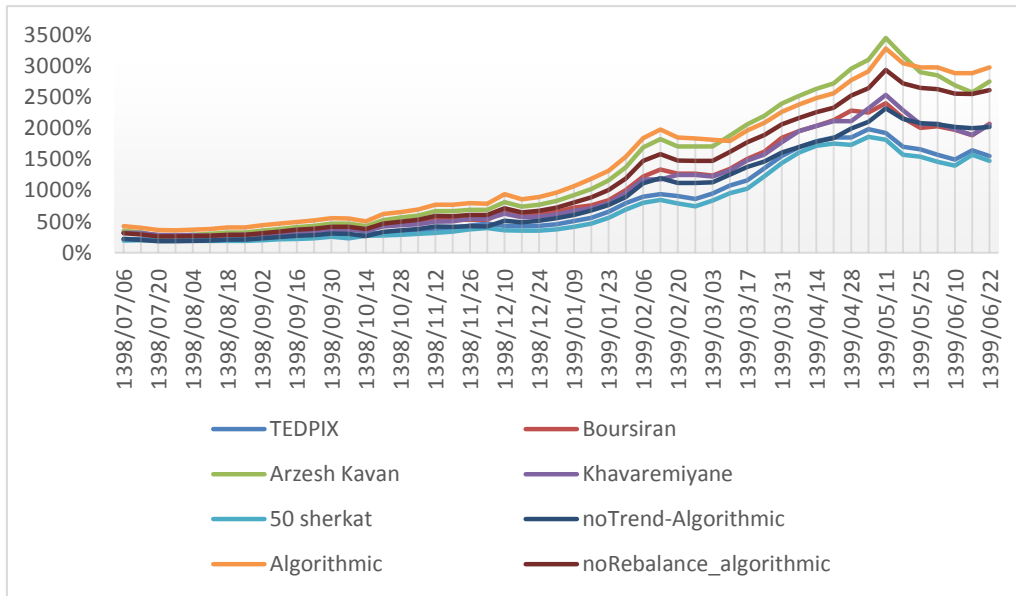


Figure 6.2 Comparison of return on net asset value from 1398/06/30 to 1399/06/30

Table 7 compares the portfolio performance of the proposed model and the top 3 funds, as well as the total index and the index of 50 companies in all 5 criteria.

Table 7. Comparison of algorithmic portfolios with other funds in all criteria

Maximum Drawdown	Sortino	Treynor Ratio	Sharp ratio	Average weekly Return	Name
-18.40%	۳,۲۳	۰,۰۱۸	۰,۳۶	%۲,۱۹	TEDPIX Index
-17.60%	۲,۷۸	۰,۰۲	۰,۳۲	%۲,۱۰	50 Company index
-13.10%	۳,۱۳	۰,۰۷۱	۰,۳۰	%۲,۴۰	Boursiran Fund
-19.60%	۲,۰۶	۰,۰۸۳	۰,۳۷	%۲,۶۱	Arzesh Kavan Fund
-18.10%	۲,۹۳	۰,۰۶	۰,۳۰	%۲,۳۹	Khavaremiyane fund
-8.90%	۳,۸	۰,۰۶۹	۰,۳۹	%۲,۰۷	Algorithmic Portfolio
-10.60%	3.76	0.068	0.38	%۲,۴۸	noRebalance_algorithmic
-12.50%	3.47	0.066	0.35	%۲,۳۷	noTrend_algorithmic

Conclusion

In this study, we tried to improve the decision parameters in the technical rules by using the meta-heuristic algorithm in the first place, and then these rules become the basis for predicting and identifying signals. Also, using the power of fuzzy logic, we tried to determine the correct direction of the market situation (trend market or neutral). Finally, using an artificial neural network, the trading signals generated by each of the technical rules are converted into a signal to decide whether to buy or sell an asset. Finally, the candidate stock list was optimized through one of the traditional portfolio optimization models.

As obtained from the results, the algorithmic portfolio proposed in the research had a much higher performance in both efficiency and risk factors. The return of the algorithmic portfolio during the test period was 2987%, which had the best performance among all mutual funds and with a much better distance from the TEDPIX index and the index of the top 50 companies. During the same period, the total index has experienced 1560% growth. On the other hand, algorithmic portfolio status has been completely superior in all portfolio performance measurement criteria except Treynor. The maximum drawdown in the algorithmic portfolio during this period was about 8%, while in the top funds it was on average 17 and in the overall index 18.5%.

The subject of algorithmic trading and trading systems is very wide and diverse. The most significant suggestive topics for future studies can be described as below:

1. The proposed mathematical model and solution method have many capabilities for development. Therefore, in the first stage, using other meta-heuristic models is recommended to replace the genetic algorithm for finding a faster and more efficient answer and compare it with the model in our research.
2. The power of fuzzy logic was used to solve the problem of finding trends in the market in the existing research. However, other mathematical methods can be suggested.
3. An artificial neural network was used to weigh the final stock selection criteria. It is possible to achieve better performance by changing the type and structure of the neural network.
4. The biggest limitation in solving similar problems in algorithmic trading is the impossibility of covering the whole market stocks. Providing a model to shorten the time to reach the answer will be helpful.

Provide a model to solve the problem of market impact at the time of ordering.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest concerning the research, authorship and, or publication of this article.

Funding

The authors received no financial support for the research, authorship and, or publication of this article.

References

- Alejandro Rodríguez et al. (2011). Using neural networks to improve trading systems based on Technical analysis by means of the RSI financial indicator, *Expert Systems with Applications*, 38(9):11489-11500. DOI: 10.1016/j.eswa.2011.03.023
- Baba, N., et al. (2001). Knowledge-Based Decision Support Systems for Dealing Nikkei-225 by Soft Computing Techniques. in Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies *KES. 2001: IOS Press*. The Netherlands. DOI:https://doi.org/10.1007/11552413_2
- Baky, M. Haddadian, H. Zomorodian, G. (2021). An Algorithmic Trading system Based on Machine Learning in Tehran Stock Exchange. *Advances in Mathematical Finance & Applications*, 6(3), 653-669. DOI: 10.22034/AMFA.2020.1894049.1380
- Dariush Forougi et al. (2014). Earnings Per Share Forecast: the Combination of Artificial Neural Networks and Particle Swarm Optimization Algorithm, *Investment Science*, 6, 63-82. DOI: http://jik.srbiau.ac.ir/article_7480.html
- Goldberg, D. (1998). Approximation by superposition of a sigmoidal function. *Journal of Mathematics of Control Signals & Systems*, 2, 303-314. DOI:<https://doi.org/10.1007/BF02551274>
- Kuo, R. J., Chen, C., & Hwang, Y. (2001). An intelligent stock trading decision

- support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems*. 118, 21–45. DOI: [https://doi.org/10.1016/S0165-0114\(98\)00399-6](https://doi.org/10.1016/S0165-0114(98)00399-6)
- Juliana Yim. (2002). A Comparison of Neural Networks with Time Series Models for Forecasting Returns on a Stock Market Index. *Applications of Artificial Intelligence and Expert Systems*. IEA/AIE, Cairns, Australia, June 17-20, 2002, Proceedings (pp.25-35). DOI: https://doi.org/10.1007/3-540-48035-8_4
- Nouri, A, et al. (2014). Comparing the performance Of Artificial Neural Networks (ANN) and Auto-Regressive Moving Average (ARIMA) Model in Modeling and Forecasting Short-term Exchange Rate Trend in Iran. *Investment Science Journal*. 10, 85-100. DOI: http://jik.srbiau.ac.ir/article_7607.html
- Souto-Mairo, (2011). S&P 500 Index Direction Forecasting from 1976 to 2010: A Fuzzy System Approach, *The International Journal of Digital Accounting Research*. 111-134. DOI: 10.4192/1577-8517-v11_6
- N. Baba, T. Kawachi, T. Nomura, Y. Sakatani, (2004), Utilization of NNs & Gas for improving the traditional technical analysis in the financial market. *SICE Annual Conference*, 2(2), 1409-1412.
- Lin, X. Yang, Z. Song, Y. (2011). Intelligent stock trading system based on improved technical analysis and Echo State Network. *Expert Systems with Applications*. 34, 620-627. DOI: <https://doi.org/10.1016/j.eswa.2011.03.001>
- Rahnamay Roodposhty et al. (2015). Optimization of portfolio Constituted from mutual funds of Tehran stock exchange using genetic algorithm. *Investment Science*. 12, 217-232. DOI: http://jik.srbiau.ac.ir/article_7672.html
- Refenes A. N., Zaprakis A., & Francis G. (1994). Stock Ranking: Neural Networks vs multiple linear regression, *IEEE*. DOI: <https://doi.org/10.1109/ICNN.1993.298765>
- Tehrani, R. Abbasion, V. (2008). Application of artificial neural networks in Stock Market Timing: A Technical Analysis Approach. *Journal of Economic Research*. 8(1), 151-178. URL: <http://ecor.modares.ac.ir/article-18-10926-fa.html>
- J. Brogaard, et al. (2018). High-frequency trading and extreme price movements. *Finance. Econ*. 128 (2), pp. 253-265 DOI:10.1016/j.jfineco.2018.02.002
- Michel, K. Kristjanpoller, W. (2020). Strongly-typed genetic programming and fuzzy inference system: An embedded approach to model and generate trading rules. *Journal of Applied soft computing*. 90, 6-25. DOI: <https://doi.org/10.1016/j.asoc.2020.106169>
- Manahov, A (2018). The rise of the machines in commodities markets: new evidence obtained using strongly typed genetic programming. *Annals of Operations*

Research. vol. 260, no. 1-2, pp. 321–352, 2018.

Ha, S. Moon, B. (2018). Finding attractive technical patterns in Crypto-currency markets. *Memetic Computing*. vol. 10, no. 3, pp. 301– 306.

Rastegar, MA. Sedaghatipour, A. (201⁹). Algorithmic Trading System for the future contract of gold coin based on intra-day data. *Science of investment*. Vol 7, no 28, pp. 49-67.

Tan, H. Prokhorov, D. Wunsch, D. (1995). Probabilistic and time-delay neural network techniques for conservative short-term stock trend prediction. in Proc. *World Congr. Neural Networks*, Washington, D.C. DOI: <https://doi.org/10.1109/72.728395>

Bibliographic information of this paper for citing:

Haddadian, Hamidreza; Baky Haskuee, Mortaza & Zomordian, Gholamreza (2022). A Hybrid Artificial Intelligence Approach to Portfolio Management. *Iranian Journal of Finance*, 6(1), 1-27.

Copyright © 2022, Hamidreza Haddadian, Mortaza Baky Haskuee and Gholamreza Zomordian

