ISSN Online: 2821-1936

Transactions on Data Analysis in Social Science

Journal Homepage: https://transoscience.ir

A Novel Artificial Intelligence-Based Model for Stock Price Prediction

S. A. Mirzaei[1], M. Doaei[2,*]

[1] M.Sc. Student in Financial Management, Esfarayen Branch, Islamic Azad University, Esfarayen, Iran
[2] Assistant Professor, Department of Financial Management, Esfarayen Branch, Islamic Azad University, Esfarayen, Iran

ARTICLE INFO

Article History:
Received 4 August 2020
Received in revised form 12 October 2020
Accepted 23 December 2020
Available online 25 December 2020

Keywords:
Stock Price Prediction, Artificial Neural Networks, Metaheuristic Algorithms, Feature Selection, Ensemble Learning

ABSTRACT

Predicting stock price movements is a crucial, intriguing, and highly challenging task for researchers, traders, and market analysts. Daily stock price prediction is particularly difficult due to the nonlinear and chaotic nature of stock price fluctuations. Artificial intelligence (AI) techniques have been widely applied to predict data with nonlinear and chaotic structures. Most previous studies have utilized single artificial neural networks (ANNs) along with a limited number of technical indicators for stock price and index prediction. However, relying on a single ANN with fixed input variables often leads to increased prediction errors. Therefore, a more robust model is required to achieve higher prediction accuracy. In this study, a two-stage hybrid framework is proposed. In the first stage (feature selection), the genetic algorithm (GA) and multilayer perceptron (MLP) neural network are employed to identify the most suitable technical indicators. In the second stage (ensemble learning), three predictors namely, the multilayer perceptron (MLP), adaptive neuro-fuzzy inference system (ANFIS), and radial basis function (RBF) neural network are combined to forecast stock prices. The final output is obtained by averaging the results of these predictors. Experimental results demonstrate that the proposed model significantly reduces prediction errors compared to individual methods.

1. INTRODUCTION

The process of decision-making has always been one of the fundamental challenges faced by human beings, as it inherently requires access to information. Among the most critical decisions in the field of finance is investment decision-making. One of the major investment options is the capital market. Investment in the capital market necessarily involves making informed decisions, which, in turn, depends on acquiring information about the future behavior of stock prices. If future stock price trends can be predicted using reliable methods, investors can maximize their returns on investment.

Predicting stock price movements is a complex, appealing, and highly challenging task for researchers, traders, and market analysts. In recent years, researchers and market analysts have strived to develop models capable of predicting stock prices with higher accuracy and lower error rates. Daily stock price prediction is particularly difficult

* Corresponding Author: doaei@iauesf.ac.ir
Assistant Professor, Department of Financial Management, Esfarayen Branch, Islamic Azad University, Esfarayen, Iran

http://dx.doi.org/10.47176/TDASS.2020.225

due to the nonlinear and chaotic nature of price movements [1]. The Efficient Market Hypothesis (EMH) posits that current market prices fully reflect all recently available public information. Various factors, including government and economic policies, can influence the capital market, leading to fluctuations in stock prices [2]. Therefore, professional investors employ fundamental analysis, technical analysis, and artificial intelligence (AI)-based methods to make more informed decisions.

AI techniques have been widely utilized to predict data with nonlinear and chaotic structures. Researchers have employed various approaches for stock price and index prediction, among which artificial neural networks (ANNs) are particularly noteworthy. Neural networks are well-established techniques used in learning, approximation, and prediction problems [3]. Most previous studies have relied on single neural networks combined with a limited number of technical indicators for stock price and index forecasting. However, using a single ANN with fixed input variables tends to increase prediction error. Consequently, there is a need for a more robust model to enhance prediction accuracy.

The proposed method in this study consists of two stages. In the first stage (feature selection), the Genetic Algorithm (GA) and Multilayer Perceptron (MLP) neural network are employed to select the most suitable input variables. In the second stage (ensemble learning), three different neural networks the Multilayer Perceptron (MLP), the Adaptive Neuro-Fuzzy Inference System (ANFIS), and the Radial Basis Function (RBF) neural network are used to predict stock prices. The final output is obtained by averaging the results of these predictors.

2. RESEARCH BACKGROUND

2.1. Domestic Studies

Sohail Ahmadkhan Beigi and Neda Abdolvand [4], in their study entitled “Stock Price Prediction Using a Hybrid Approach Based on Artificial Neural Networks and the Imperialist Competitive Algorithm with Chaotic Theory”, investigated stock price forecasting. In their research, the authors used the Imperialist Competitive Algorithm (ICA) with chaotic initialization to train a neural network, utilizing daily price data from Iran Khodro Company between 2010 and 2016 to predict stock prices. The input variables to the neural network included the first price, highest price, lowest price, stock value, and trading volume as independent variables, while the opening price of the following day served as the dependent variable. The proposed model employed a three-layer perceptron network with five neurons in the input layer, three neurons in the hidden layer, and one neuron in the output layer. The study compared the proposed approach with other neural network training algorithms, including Gradient Descent, Genetic Algorithm, Particle Swarm Optimization (PSO), and ICA. However, no technical indicators were used as network inputs, and only a single neural network architecture was adopted, which reduced prediction accuracy.

Seyed Mostafa Mirghafari and Mohammad Ali Rostegar [5] used the Ant Colony Optimization (ACO) algorithm to select the input variables of an adaptive neuro-fuzzy inference system (ANFIS) and subsequently employed the ANFIS model to predict stock prices. Their study focused on five leading companies listed on the Tehran Stock Exchange, namely Persian Gulf Petrochemical Industries, Iran Mobile Communications Company, Telecommunication Company of Iran, Mobarakeh Steel Company, and National Iranian Copper Industries. The input variables included technical indicators such as the Simple Moving Average (SMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), the previous day’s closing price, the daily crude oil price of Iran, and the daily exchange rate of the U.S. dollar against the Iranian rial.

Ahdiyeh Rahimi Gorkani [6], in her study “Identifying the Most Effective Model for Predicting Stock Prices of Companies Listed on the Tehran Stock Exchange Using Artificial Neural Networks,” utilized ANNs trained through error backpropagation, PSO, and ICA algorithms. Using daily data from the Tehran Stock Exchange, including the overall market index and daily prices of 14 selected stocks, the researcher predicted stock prices and evaluated the results. The findings revealed that PSO slightly outperformed ICA in terms of minimizing neural network training error. This study used fundamental data as network inputs and employed only a single neural network structure, which limited prediction accuracy.

Davoud Ahmadian and Omid Farkhondeh-Rooz [7], using stock data from 2008 to 2013 on the Tehran Stock Exchange, predicted next-day stock prices via feedforward neural networks trained with the backpropagation learning algorithm under three different input configurations. The results indicated that the best-performing networks were three-layer ANNs with varying numbers of hidden neurons, employing a Log-Sigmoid activation function in the first and second layers and a Tan-Sigmoid function in the third layer, significantly reducing mean squared error.

Atefeh Hasani Bafarani et al. [8] examined the application of a neuro-fuzzy model for predicting the stock prices of companies listed on the Tehran Stock Exchange. The study evaluated the predictive accuracy of the ANFIS model using data from all listed companies between 2006 and 2016, selecting 110 firms as the sample. The results confirmed that the proposed model had strong predictive capability for stock prices.

Jafar Babajani et al. [9], in their study "Stock Price Prediction in the Tehran Stock Exchange Using a Recurrent Neural Network Optimized by the Artificial Bee Colony Algorithm," proposed a model for stock price forecasting. Initially, stepwise regression–correlation analysis was used to select key variables influencing stock prices, which were then defined as model inputs. In the next stage, the Artificial Bee Colony (ABC) algorithm was used to optimize the weights of the recurrent neural network. The results showed that the ABC-optimized RNN achieved significantly higher prediction accuracy compared to other forecasting methods. However, as with previous studies, relying on a single predictor increased the prediction error.

2.2. Foreign Studies

Guresen et al. [10], in their article titled "Using Artificial Neural Network Models in Stock Market Index Prediction," examined the performance of multilayer perceptron (MLP) networks and hybrid neural network architectures. The researchers used MLP networks to forecast the Nasdaq index from 2008 to 2009 in a time-series framework. Their results indicated that neural network models have a strong capability to predict market movements. However, the limitation of this study lies in the use of a single predictor and the absence of technical analysis indicators, which could have enhanced model performance.

Hedayati Moghaddam et al. [11] applied neural networks to predict the Nasdaq index. The researchers constructed a time series based on closing price data and predicted the next day's closing price. In the first experiment, the model used historical price data from the previous four days, while in the second experiment, it used data from the previous nine days. The findings showed little difference in prediction accuracy between the two experiments, suggesting that increasing the number of past data points did not substantially improve the results. The study also analyzed the effect of varying the number of layers, activation functions, and training algorithms on neural network performance.

Qasemieh et al. [12] trained an MLP neural network using five metaheuristic algorithms: the Genetic Algorithm (GA), Cuckoo Search (CS), Improved Cuckoo Search (ICS), a hybrid GA–ICS algorithm, and Particle Swarm Optimization (PSO). The researchers concluded that, first, the PSO algorithm achieved the lowest error rate in training the MLP, followed by the hybrid GA–ICS algorithm. The study used 28 technical indicators as input variables to the neural network. However, one limitation was the lack of a feature selection process. As the number of input indicators increases, redundancy or irrelevance among inputs may occur, which can, in turn, increase neural network prediction error.

Nelson et al. [13] utilized historical stock price data and several technical analysis indicators with a Long Short-Term Memory (LSTM) neural network to predict stock prices. The results showed that LSTM networks outperformed MLP networks in terms of prediction accuracy.

Yang et al. [14] designed a deep neural network model with 40 input variables using 10 days of stock price data including opening, closing, highest, and lowest prices and predicted the next day's closing price.

Coupra et al. [2] applied neural networks with different activation functions and various numbers of neurons across network layers to predict stock prices. The input variables included the opening price, highest price, and lowest price, while the network output represented the closing price. The researchers found that increasing the number of neurons did not necessarily improve accuracy and that using a maximum of 10 neurons in the hidden layer was optimal for prediction performance.

Selvamuthu et al. [1] investigated the impact of different neural network training algorithms Levenberg–Marquardt, Scaled Conjugate Gradient, and Bayesian Regularization on stock price prediction using price data. Their findings revealed that the Levenberg–Marquardt algorithm yielded better results and required less computation time than the other algorithms.

Heydari et al. [3] trained an MLP neural network using the Lion–Ant Optimization (LAO) algorithm [15]. The researchers concluded that the LAO algorithm performed better than other metaheuristic optimization algorithms such as PSO, GA, Differential Evolution, and Population-Based Incremental Learning (PBIL).

3. STATISTICAL SAMPLE

The data required for stock price prediction and evaluation of the proposed method were collected from the website of the Tehran Stock Exchange Technology Management Company (www.tsetmc.com) over a ten-year period from November 2009 to November 2019 (Hasani Bafarani et al., 2018). The study sample comprised the top five most active companies listed on the Tehran Stock Exchange during the first quarter of 2019, as identified by the Exchange, provided that they possessed complete price data for the preceding ten years. The selected companies used in this study are presented in Table 1 [5].

Table 1. Sample Companies Used for Evaluating the Proposed Method

No.	Symbol	Company Name	Industry Group
1	Foolad	Esfahan’s Mobarakeh Steel Company	Basic Metals
2	F.Melli	National Iranian Copper Industries Company	Basic Metals
3	Shapna	Esfahan Oil Refining Company	Petroleum Products
4	K.GolGohar	Gol Gohar Mining and Industrial Company	Metal Ore Extraction
5	V.BankMellat	Bank Mellat	Banks and Credit Institutions
6	Overall Index	–	–

4. PROPOSED METHOD

In this study, 42 technical analysis indicators were extracted based on the daily stock prices. In the first stage, the Genetic Algorithm (GA) in combination with a Multilayer Perceptron (MLP) neural network was employed to select the most relevant indicators (feature selection phase). In the second stage, using the selected indicators, a hybrid learning model was developed for stock price prediction. The proposed hybrid framework demonstrates a lower prediction error compared to existing methods.

Figure 1 illustrates a summary of the workflow of the proposed method.

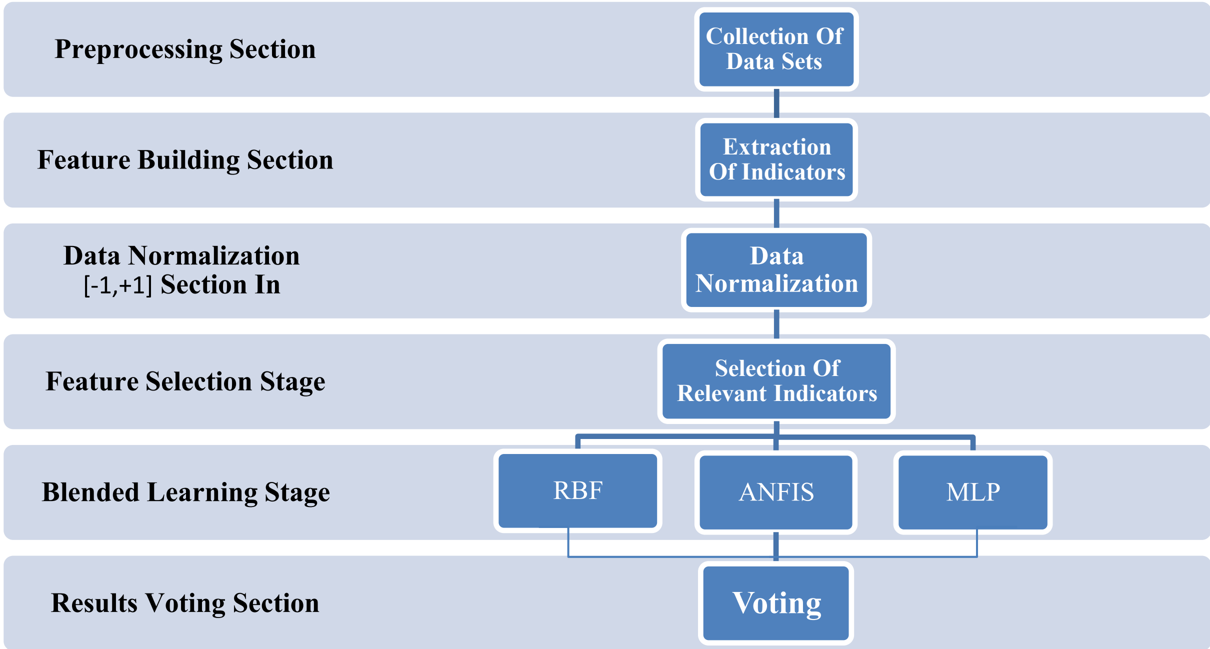

Fig. 1. Workflow of the Proposed Method

One of the most critical aspects of designing a neural network architecture is dataset preparation and preprocessing. Data normalization ensures that the importance of each variable does not depend on its unit of measurement. Various normalization methods exist, among which linear normalization has been employed in this study. Linear normalization maps the data into a specified range $[\boldsymbol{a}, \boldsymbol{b}]$, calculated using Equation (1).

$$x_{norm} = (b-a)\left(\frac{(x - x_{min})}{(x_{max} - x_{min})}\right) + a \tag{1}$$

In this study, all data were scaled within the range [-1, +1] based on the minimum and maximum values of the entire dataset. Here, $x$represents the variable, while $x_{\max}$and $x_{\min}$denote the maximum and minimum values, respectively.

Table 2 presents the list of technical indicators extracted and utilized in this research, along with their corresponding formulas and time periods.

Table 2. List of Technical Indicators Used in This Study with Their Formulas and Time Periods

Time Period	Formula	Indicator Name
5,10,14,21,50,100,200	$\frac{C_t + C_{t-1} + \cdots + C_{t-n+1}}{n}$	**Simple Moving Average (SMA)**
5,10,14,21,50,100,200	$(C_t - EMA(n)_{t-1}) * \left(\frac{2}{n+1}\right) + EMA(n)_{t-1}$	**Exponential Moving Average (EMA)**
5,10,14,21,50,100,200	$RS = \frac{Avg(Gain)}{Avg(Loss)}$ $RSI = 100 - \frac{100}{1+RS}$	**Relative strength Index (RSI)**
5,10,14,21,28	$\frac{C_t}{C_{t-1}} * 100$	**Momentum**

5,10,14,21,28	$\left(\frac{C_t - L_t(n)}{H_t(n) - L_t(n)}\right) * 100$	**Stochastic**
5,10,14,21,28	$\left(\frac{H_t(n) - C_t}{H_t(n) - L_t(n)}\right) * -100$	**Williams R**
5,10,14,21,28	$\frac{\left(\frac{H + L + C}{3}\right) - SMA}{0.015 * MeanDeviation}$	**Commodity Channel Index (CCI)**
[12,26]	$EMA(n) - EMA(M)$	**Moving Average Convergence Divergence (MACD)**

4.1. Feature Selection Phase

Feature selection methods have become an integral component of the learning process when dealing with high-dimensional data. Feature selection can be defined as the process of identifying relevant features and eliminating irrelevant or redundant ones, with the goal of selecting a subset of features that best represent the problem while minimizing performance degradation. To address the challenge posed by a large number of features, dimensionality reduction techniques are essential, as they can enhance the efficiency and effectiveness of the learning process.

In this study, the Genetic Algorithm (GA) in combination with a Multilayer Perceptron (MLP) neural network was employed to identify the most optimal features. The optimal features are defined as those that minimize the mean squared error (MSE) of the MLP neural network. Figure 2 illustrates the workflow of the feature selection process.

In this implementation, the Genetic Algorithm was run for 20 iterations, with an initial population size of 40 chromosomes, a crossover probability of 70%, and a mutation probability of 3%. For the MLP neural network, 10 neurons were used in the hidden layer. Moreover, all input data were normalized within the range [-1, +1] prior to training.

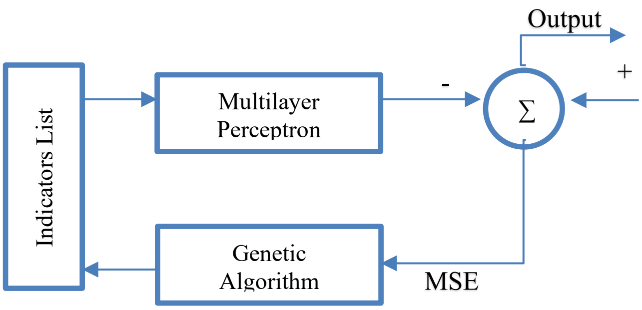

Fig. 2. Workflow of the Feature Selection Process

4.2. Hybrid Learning Phase

Hybrid learning is a type of learning approach in which multiple classifiers are trained to predict a single target. Generally, the outcomes of ensemble or hybrid learning algorithms outperform those of individual algorithms. In this study, three different types of neural networks were employed for prediction: the Multilayer Perceptron (MLP), the Neuro-Fuzzy Network, and the Radial Basis Function (RBF) Neural Network. A brief description of each classifier is provided below.

4.3. Multilayer Perceptron (MLP) Neural Network

A neural network consists of a set of neurons arranged in multiple layers, forming a specific architecture based on interconnections between neurons across layers. Each neuron can act as a nonlinear mathematical function; thus, a network composed of multiple neurons can represent a complex nonlinear system. In a neural network, each neuron operates independently, and the overall network behavior results from the collective behavior of all neurons. The MLP network is one of the most widely implemented types of neural networks and consists of three layers: input, hidden, and output layers [11, 16].

4.4. Neuro-Fuzzy Network

The neuro-fuzzy system is a type of artificial neural network that emerged in the early 1990s. Since it combines the structure of neural networks with the reasoning capability of fuzzy logic, it benefits from the strengths of both methods within a unified framework. Compared to traditional artificial neural networks, neuro-fuzzy systems tend to train faster and achieve higher accuracy due to the tunable parameters of the fuzzy system.

4.5. Radial Basis Function (RBF) Neural Network

The RBF neural network is an artificial neural network that uses radial basis functions as activation functions. The network's output is a linear combination of radial basis functions applied to the input parameters and neurons. These networks are widely used for function approximation, prediction, and classification tasks.

Table 3 presents the configuration parameters used for the three neural network models in this study.

Table 3. Neural Network Parameter Settings

Neural Network Type	Parameter	Value
1. Multilayer Perceptron (MLP) Neural Network	1-1 Number of hidden layers	2
	1-2 Number of neurons in hidden layer	3
	1-3 Activation function in hidden layer	tansig
2. Neuro-Fuzzy Neural Network	2-1 Number of input membership functions	2
	2-2 Type of input membership function	Gaussmf
	2-3 Type of output membership function	linear
3. Radial Basis Function (RBF) Neural Network	3-1 Width of radial basis functions	1
	3-2 Maximum number of neurons	10

5. EXPERIMENTAL RESULTS

5.1. Evaluation Metrics

To assess the performance of the proposed method, the evaluation metrics listed in Table 4 were used [10,11,12].

Table 4. Evaluation Metrics

Formula	Metric Name
$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert fo_i - fe_i\rvert$	**Mean Absolute Error (MAE)**
$MSE = \frac{1}{N}\sum_{i=1}^{N}(fo_i - fe_i)^2$	**Mean Squared Error (MSE)**
$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(fo_i - fe_i)^2}$	**Root Mean Squared Error (RMSE)**

$R = \frac{\sum_{i=1}^{N}(fo_i - \overline{fo_\iota})(fe_i - \overline{fe_\iota})}{\sqrt{\sum_{i=1}^{N}(fo_i - \overline{fo_\iota})^2(fe_i - \overline{fe_\iota})^2}}$	**Regression Coefficient (R)**

In these evaluation metrics, $f_e$represents the actual values, and $f_o$represents the predicted values. The terms $\bar{f}_e$and $\bar{f}_o$denote the mean of actual values and mean of predicted values, respectively, while $N$is the total number of data points.

6. RESULTS ANALYSIS AND COMPARISON

Table 5 presents a comparison of results from Qasemiyeh et al. [12], who trained a Multilayer Perceptron (MLP) neural network using the Particle Swarm Optimization (PSO) algorithm, and Heidari et al. [3], who trained an MLP using the Ant Lion Optimization (ALO) algorithm. The third column shows the results of the proposed method using five selected features for the entire dataset. The best values are highlighted in bold. As observed, the proposed method achieved the best performance across all datasets.

Table 5. Summary Comparison of Normalized Results for All Data

Proposed Method with Selection of 5 Features				**Neural Network Trained with Antlion Optimization (ALO) Algorithm**				**Neural Network Trained with Particle Swarm Optimization (PSO) Algorithm**				**Symbol**
R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	
0.9994	**0.0159**	**0.0002**	**0.0106**	0.9772	0.0981	0.0096	0.0764	0.9764	0.1016	0.0103	0.0813	Foolad
0.9988	**0.0193**	**0.0003**	**0.0125**	0.9057	0.1814	0.0329	0.1338	0.9814	0.0786	0.0061	0.0624	F.Melli
0.9969	**0.0406**	**0.0016**	**0.0171**	0.9689	0.1291	0.0166	0.0913	0.9784	0.1077	0.0116	0.0763	Shapna
0.9994	**0.0168**	**0.0002**	**0.0105**	0.9672	0.1242	0.0154	0.0997	0.9784	0.1038	0.0107	0.0836	K.GolGohar
0.9961	**0.0200**	**0.0004**	**0.0123**	0.9336	0.0822	0.0067	0.0648	0.9421	0.0800	0.0064	0.0608	V.BankMellat
0.9988	**0.0193**	**0.0003**	**0.0141**	0.9697	0.0963	0.0092	0.0790	0.9525	0.1272	0.0162	0.0972	Overall Index

Tables 6 and 7 (not shown here) compare results for the training and testing datasets, respectively. The best values are highlighted in bold. Reviewing these tables confirms that the proposed method using five selected features consistently outperforms the other approaches.

Table 6. Summary Comparison of Results for Training Data (Normalized)

Proposed Method with Selection of 5 Features				**Neural Network Trained with Antlion Optimization (ALO) Algorithm**				**Neural Network Trained with Particle Swarm Optimization (PSO) Algorithm**				**Symbol**
R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	
0.9994	**0.0158**	**0.0002**	**0.0106**	0.9767	0.0975	0.0095	0.0756	0.9755	0.1020	0.0104	0.0817	Foolad
0.9989	**0.0180**	**0.0003**	**0.0119**	0.9088	0.1819	0.0330	0.1327	0.9815	0.0780	0.0060	0.0614	F.Melli
0.9984	**0.0289**	**0.0008**	**0.0164**	0.9688	0.1278	0.0163	0.0894	0.9802	0.1016	0.0103	0.0744	Shapna
0.9994	**0.0167**	**0.0002**	**0.0104**	0.9690	0.1228	0.0151	0.0986	0.9790	0.1021	0.0104	0.0823	K.GolGohar
0.9963	**0.0202**	**0.0004**	**0.0124**	0.9352	0.0805	0.0064	0.0640	0.9504	0.0785	0.00617	0.0599	V.BankMellat
0.9988	**0.0190**	**0.0003**	**0.0139**	0.9683	0.0976	0.0095	0.0802	0.9535	0.1265	0.0160	0.0967	Overall Index

Table 7. Summary Comparison of Results for Test Data (Normalized)

Proposed Method with Selection of 5 Features				**Neural Network Trained with Antlion Optimization (ALO) Algorithm**				**Neural Network Trained with Particle Swarm Optimization (PSO) Algorithm**				**Symbol**
R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	R	RMSE	MSE	MAE	
0.9993	**0.0159**	**0.0002**	**0.0106**	0.9784	0.0994	0.0098	0.0781	0.97847	0.1005	0.0101	0.0802	Foolad
0.9986	**0.0221**	**0.0004**	**0.0140**	0.8983	0.1804	0.0325	0.1365	0.98139	0.0801	0.0064	0.0646	F.Melli
0.9932	**0.0597**	**0.0035**	**0.0189**	0.9689	0.1322	0.0174	0.0959	0.97443	0.1209	0.0146	0.0808	Shapna
0.9994	**0.0170**	**0.0002**	**0.0107**	0.9630	0.1275	0.0162	0.1022	0.97727	0.1078	0.0116	0.0867	K.GolGohar
0.9963	**0.0194**	**0.0003**	**0.0122**	0.9308	0.0860	0.0073	0.0666	0.91541	0.0833	0.0069	0.0628	V.BankMellat
0.9988	**0.0198**	**0.0003**	**0.0145**	0.9727	0.0931	0.0086	0.0762	0.95039	0.1290	0.0166	0.0982	Overall Index

The regression coefficient plots for the proposed method for the Overall Index (Total Index) are illustrated in Figure 3, showing the results for the training data, test data, and the entire dataset.

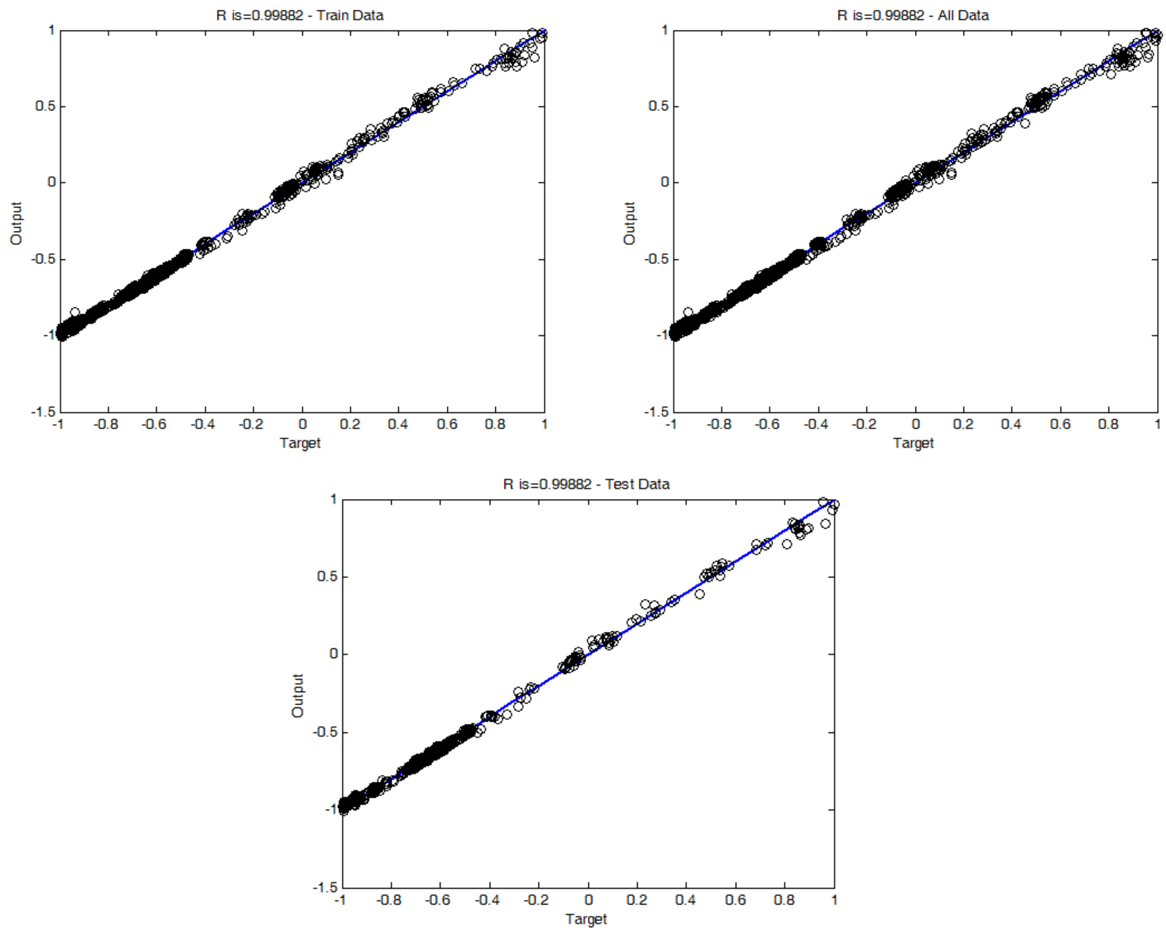

Fig. 3. Regression coefficient plot for the Overall Index (Total Index) for the training data, test data, and the entire dataset (normalized) using the proposed method.

In Figures 4 and 5, the regression coefficient plots for the Overall Index for the training data, test data, and the entire dataset (normalized) are shown for the multi-layer perceptron neural network trained using the Particle Swarm Optimization algorithm and the Antlion Optimization algorithm, respectively.

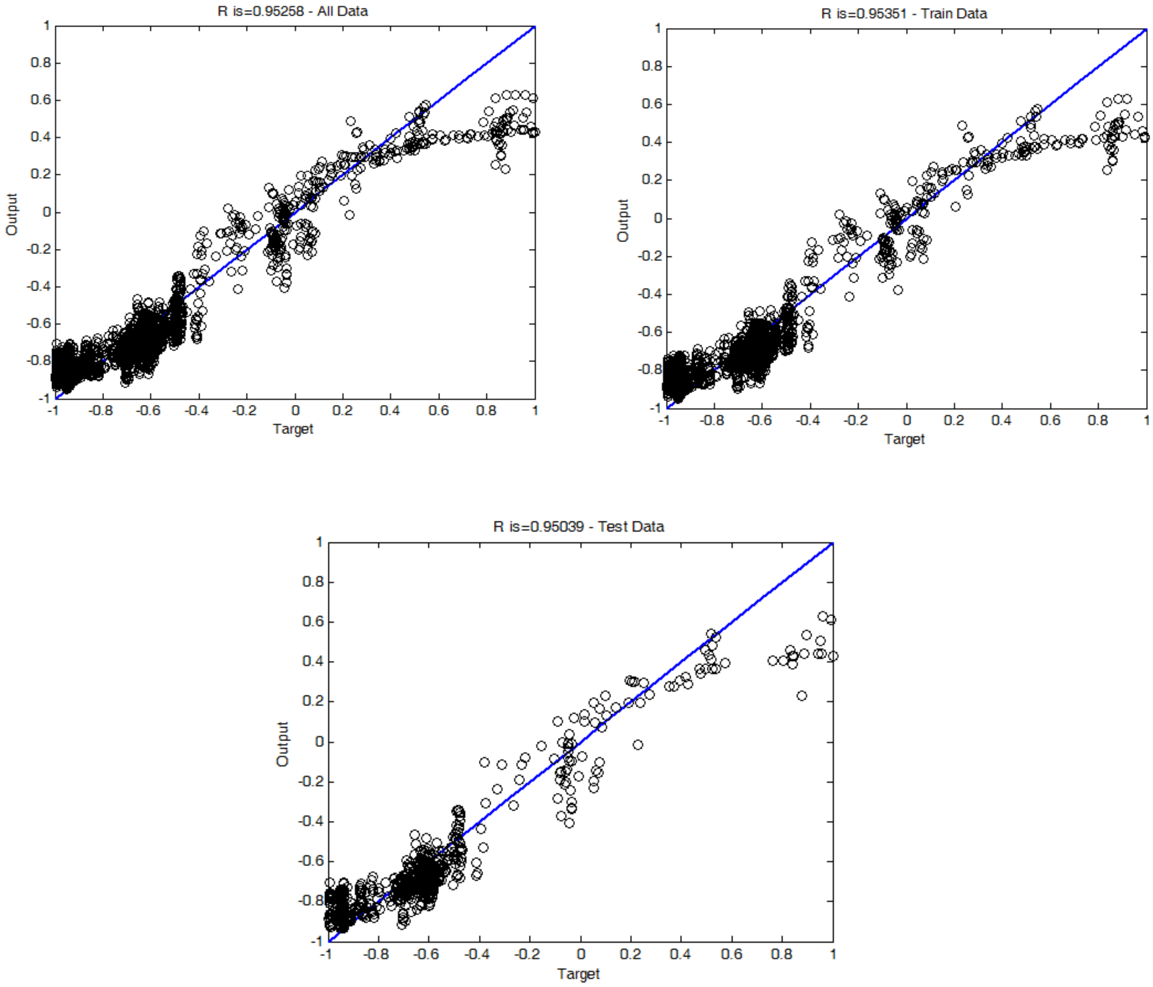

Fig. 4. Regression coefficient plot for the Overall Index (Total Index) for the training data, test data, and the entire dataset (normalized) using the multi-layer perceptron neural network trained via the Particle Swarm Optimization algorithm.

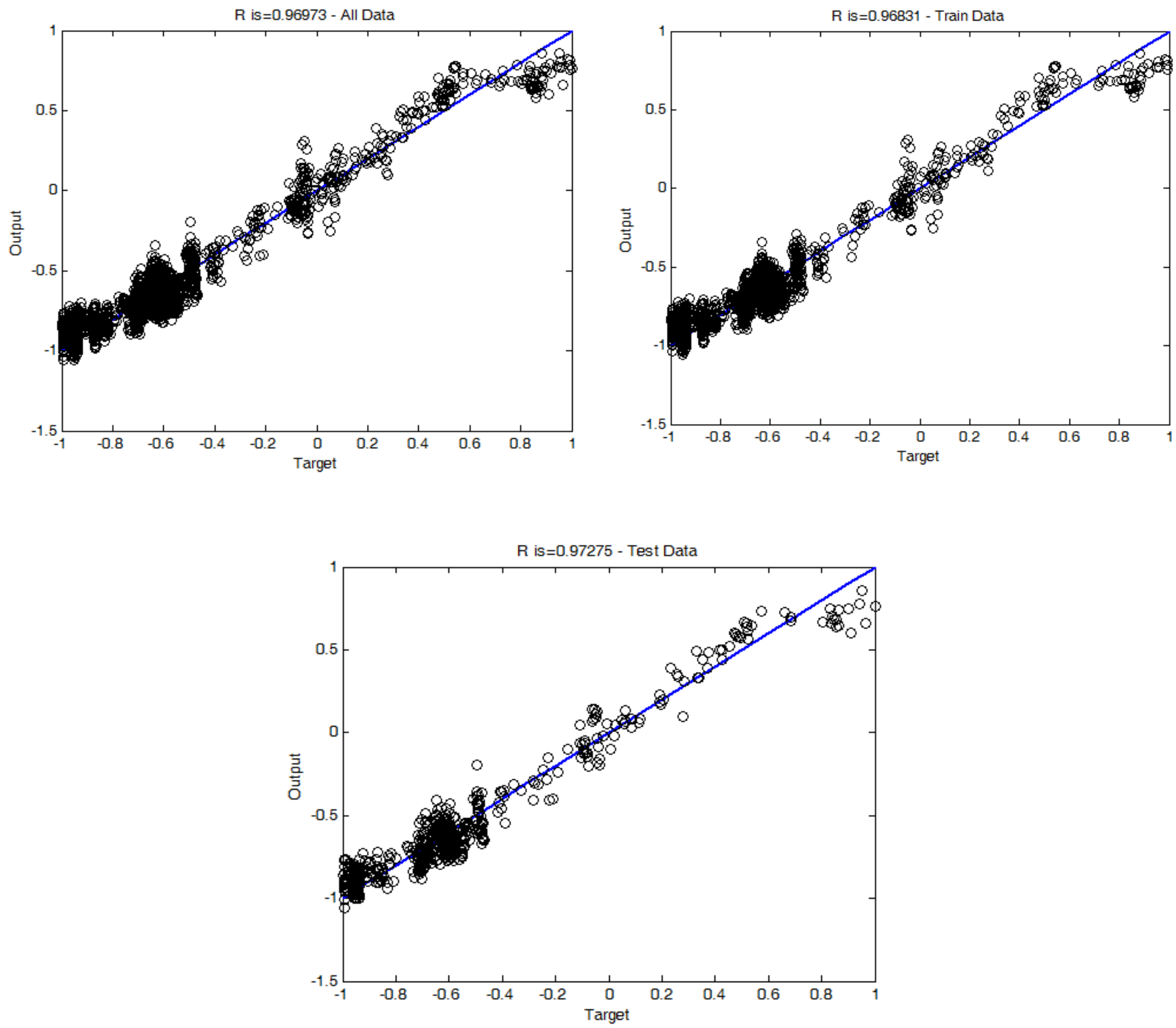

Fig. 5. Regression coefficient plot for the Overall Index (Total Index) for the training data, test data, and the entire dataset (normalized) using the multi-layer perceptron neural network trained via the Ant Lion Optimization algorithm.

7. CONCLUSION AND RECOMMENDATIONS

Daily stock price prediction is highly challenging due to the nonlinear and chaotic nature of stock price movements. In this study, we proposed a model for predicting stock prices and indices that reduces prediction errors. In the first phase, the model uses a genetic algorithm combined with a multi-layer perceptron neural network to select the most relevant features. The selected features are those that minimize the mean squared error in the multi-layer perceptron network. In the second phase, these selected features are input into three different neural network predictors, including a multi-layer perceptron, a neuro-fuzzy network, and a radial basis function network. Finally, the outputs of the three networks are averaged to produce the final prediction. Experimental results demonstrate that the proposed model achieves lower prediction errors compared to existing methods. Future research can focus on deep neural networks, including their training processes and architectural optimization.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Selvamuthu, D., Kumar, V., & Mishra, A. (2019). Indian stock market prediction using artificial neural networks on tick data. Financial Innovation, 5(1), 16. https://doi.org/10.1186/s40854-019-0131-7

[2] Chopra, S., Yadav, D., & Chopra, A. N. (2019). Artificial neural networks based Indian stock market price prediction: Before and after demonetization. Journal of Swarm Intelligence and Evolutionary Computation, 8(174), 2.

[3] Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., & Mafarja, M. (2020). Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks. In Nature-Inspired Optimizers (pp. 23–46). Springer, Cham. https://doi.org/10.1007/978-3-030-12127-3_3

[4] Ahmadkhan Beigi, S., & Abdolvand, N. (2017). Stock price prediction using a hybrid approach of artificial neural networks and imperialist competitive algorithm based on chaos theory. Financial Management Strategy, 5(3).

[5] Mirghafari, S., & Rostgar, M. A. (2017). A fuzzy hybrid model for stock return prediction using fuzzy neural networks and the ant colony algorithm. In Proceedings of the 2nd National Conference on Soft Computing, Gilan-Roudsar, University of Gilan.

[6] Rahimi Garkani, A. (2017). Identifying the most effective model for predicting stock prices of companies listed on the structured securities market using artificial neural networks. In Proceedings of the 2nd International Conference on Knowledge-Based Research in Computer Engineering and Information Technology, Tehran, Majlesi University.

[7] Ahmadian, D., & Farkhandeh Rooz, O. (2017). Neural network methods for predicting the overall stock market index during 2016–2017. In Proceedings of the 1st International Conference on Management Patterns in the Age of Progress, Tehran, Islamic Government Research Institute – NAJA Social Studies, University of Tehran.

[8] Hasani Bagherani, A., Arab Bagherani, M., & Esmaeilian, G. (2018). Examining the application of neuro-fuzzy models in predicting stock prices of companies listed on the Tehran Stock Exchange. In Proceedings of the 1st National Conference on Accounting and Management, Natanz, Islamic Azad University, Natanz Branch.

[9] Babajani, J., Taghva, M. R., Bolou, G., & Abdollahi, M. (2019). Stock price prediction on the Tehran Stock Exchange using recurrent neural networks optimized with the artificial bee colony algorithm. Financial Management Strategy, 7(2).

[10] Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market

index prediction. Expert Systems with Applications, 38(8), 10389–10397. https://doi.org/10.1016/j.eswa.2011.02.068

[11] Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. Journal of Economics, Finance and Administrative Science, 21(41), 89–93. https://doi.org/10.1016/j.jefas.2016.07.002

[12] Ghasemiyeh, R., Moghdani, R., & Sana, S. S. (2017). A hybrid artificial neural network with metaheuristic algorithms for predicting stock price. Cybernetics and Systems, 48(4), 365–392. https://doi.org/10.1080/01969722.2017.1285162

[13] Nelson, D. M., Pereira, A. C., & de Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 1419–1426). IEEE. https://doi.org/10.1109/IJCNN.2017.7966019

[14] Yong, B. X., Rahim, M. R. A., & Abdullah, A. S. (2017, August). A stock market trading system using deep neural network. In Asian Simulation Conference (pp. 356–364). Springer, Singapore. https://doi.org/10.1007/978-981-10-6463-0_31

[15] Mirjalili, S. (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

[16] Heidari, A. A., Faris, H., Aljarah, I., & Mirjalili, S. (2019). An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Computing, 23(17), 7941–7958. https://doi.org/10.1007/s00500-018-3424-2