An Optimal Multi-price Simultaneous Estimation Approach Based on Deep Learning and Genetic Algorithms

Shiva Ghasempour
PhD. Student in Financial Engineering, Department of Financial Management, Faculty of Management and Economy, Science and Research Branch, Islamic Azad University, Tehran, Iran
shivaghasempour@gmail.com

Shadi Shahverdiani
Assistant Professor, Department of Financial Management, Faculty of Humanities, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran and Visiting professor of Science and Research Branch, Islamic Azad University, Tehran, Iran. (corresponding author).
shshahverdiani@gmail.com

Amir-Reza Keyghobadi
Assistant Professor, Department of Accounting, Faculty of Economics and Accounting, Central Tehran Branch, Islamic Azad University, Tehran, Iran
a.keyghobadi@iauctb.ac.ir

Mahdi Madanchi Zaj
Assistant Professor, Department of Financial Management, Faculty of Management, Electronic Campus, Islamic Azad University, Tehran, Iran
madanchi@iauec.ac.ir

Submit: 16/08/2022 **Accept:** 17/04/2023

ABSTRACT

There has been an increase in the development of automated trading systems in numerous countries, including Iran. The greatest advantage of such systems is that they allow traders to make trading decisions at an increased pace and with more accuracy without having to rely on emotions. Estimating price is one of the most important aspects of algorithmic trading. Deep neural networks are preferred for estimation. Additionally, investors who rely on algorithmic trading have a huge advantage by having a model that estimates opening, maximum, minimum, and closing prices simultaneously. In this study, using short-term LSTM deep-long-term memory neural networks, and the genetic algorithm, these four prices are estimated simultaneously. In addition, the optimal feature was selected by considering 40 price, volume, volumetric and volumetric indicators. The proposed model is evaluated using five shares from the Tehran stock exchange during the period 2012-2021, namely Isfahan oil refining, Iran Khodro, and Amirkabir Kashan Steel, Eqtesad Novin Bank, Chin-Chin Industry and Cultivation, and Exir Pharmacy. Based on the results of this study, the proposed model has excellent simultaneous estimation performance and the average estimation error of all 4 prices is less than 8%, demonstrating that the proposed method has a lower estimation error.

Keywords: Algorithmic Trading, Deep Neural Networks, LSTM, Multi-Price Estimation, Genetic Algorithm

1. Introduction

Forecasting is the process of making quantitative estimates about the likelihood of future events based on information available now and in the past. Forecasting involves expressing present and past data in one-equation, multi-equation, or time series models, and predicting future events based on past data (Vijh et al., 2020). The forecasting of stock prices is an important part of financial research since well-forecast markets will maximize returns for investors. While various methods have been used to predict stock prices, none of them can address all the variables involved in estimating the stock price model and the effects of each. Getting the best results with the least amount of data is the key to successful stock market forecasting (Selvin et al., 2017).

Time series data are used for forecasting in single-variable quantitative methods. A time series is an arrangement of observations over time (Khajavi & Amiri, 2017). The forecasting of time series has attracted various researchers and numerous methods have been proposed. Many sciences, especially financial sciences, are interested in computational methods like those used in operations research to systematically investigate issues. A continuous optimization process is carried out with these algorithms until the best solution is found. An artificial intelligence algorithm is a part of an operations research algorithm. They provide the best results when there is uncertainty, infer the data, and discover its features. As opposed to statistical methods, they are not fundamentally model-based; instead, they are based on their own input data and do not require interpretation of model details (Shastri et al., 2019).

A number of artificial intelligence methods are available, such as neural networks, like perceptrons, recurrent neural networks, Naive-Bayes networks, back-propagating networks, LSTM networks, and support vector machines (SVMs). Neural networks can learn by using the process of data changes (training data) and the rules governing it and then generalize it on other data they have not seen before (test data) (Moghaddam et al., 2016).

Artificial neural networks are improving their ability to recognize patterns based on long-term data since deep learning was introduced. As deep neural networks (DNNs) are capable of learning much faster than normal neural networks (NNs), recent investigations into stock price forecasting have used DNNs. Financial markets have large amounts of time series data; therefore, a comprehensive neural network learning process is required. In deep neural networks, recursive neural networks (RNNs) are more commonly used for predicting stock prices. Yoshihara et al. (2014) and Dixon et al. (2015) were the first researchers to use RNNs to predict stock prices. Despite the advantages of RNN, it is difficult to train, especially in complex cases and cases with long time structures. In order to overcome this challenge, Long-Short-Term-Memory LSTM networks have been introduced, which perform simple linear operations on neuronal information and augment it with external information at the moment through gates. LSTM gates were proposed by Chow et al. (2014).

It is noteworthy that all previous approaches have been able to design an algorithmic trading system by predicting one price (usually the closing price) at a time. To predict several variables simultaneously, a model must deal with multiple inputs and produce multiple output values simultaneously. As a result, we developed an optimal deep recurrent neural network model based on a short-long-term memory network (LSTM) called LSTM-DRNN, and two generalized regression neural networks (GRNNs) with least squares boosting (LSBoosts) to estimate four open, high, low, and close prices simultaneously. Taking into account the high power of meta-heuristic algorithms and, in particular, the genetic algorithm, this algorithm is used to optimize hyperparameters and select optimal features.

The remainder of this paper is arranged as follows: The research background is reviewed in Section 2. The proposed methodology is discussed in Section 3. The results of the proposed model are discussed in Section 4. 5. Discussion and conclusions are presented in Section 5.

Research Background

The recent research in price prediction can be divided into shallow and deep neural networks. The LSTM network has been widely used in deep neural networks. Li et al. (2017) combined the LSTM model with the Bayesian method to extract market emotional factors to improve the predictive ability of the LSTM model. Zhao et al. (2017) estimated stock prices using an LSTM network with time-varying weights. Based on their results, they found that the relationship between the importance of data and their time series is linear, quadratic, and relatively quasilinear. In addition, a study that implemented the model on the CSI 300 index found that it was 83.91% accurate in estimating prices. Baek & Kim (2018) propose a new method for estimating the stock market index that includes two modules: LSTMs for preventing overvaluation and LSTMs for forecasting. Chung & Shin (2018) predict stock prices using a hybrid LSTM/genetic algorithm (GA) method. A high-frequency trading model was developed by Rundo (2019) using the LSTM network based on deep reinforcement learning. This research shows that this method is 85% accurate in estimating short-medium-term trends based on forex data of the EU/USD exchange rate. Li et al. (2020) presented a time-based feature-aware deep reinforcement learning model (TFJ-DRL), which improved signal representation learning and decision making through the integration of deep learning with reinforcement learning. The model was evaluated on real financial data with different price trends (upward, downward, and neutral) and the results indicate that the method is highly robust and interesting.

Based on an LSTM network, Ding et al. (2020) proposed a deep recurrent neural network model with multiple inputs and outputs. This model is capable of predicting the open, low, high, and close prices of a stock simultaneously. Results of this study showed that this algorithm was very effective at simultaneously estimating all three prices due to its high learning and estimation power. Michańków et al. (2022) used LSTM networks to predict the value of the BTC and S&P 500 indices, using data from 2013 to the end of 2020. They introduced an innovative loss function that improves the utility of the LSTM model's predictive capacity in algorithmic investment strategies. Banik et al. (2022) developed a decision support system applied by LSTM to allow swing traders to accurately analyses and predict future equity values.

Methodology

The proposed method of this study consists of s phases:

1- Preparing the time series

2- Normalization of data

4- Optimal price estimation

A schematic of the proposed method is shown in Fig. 1.

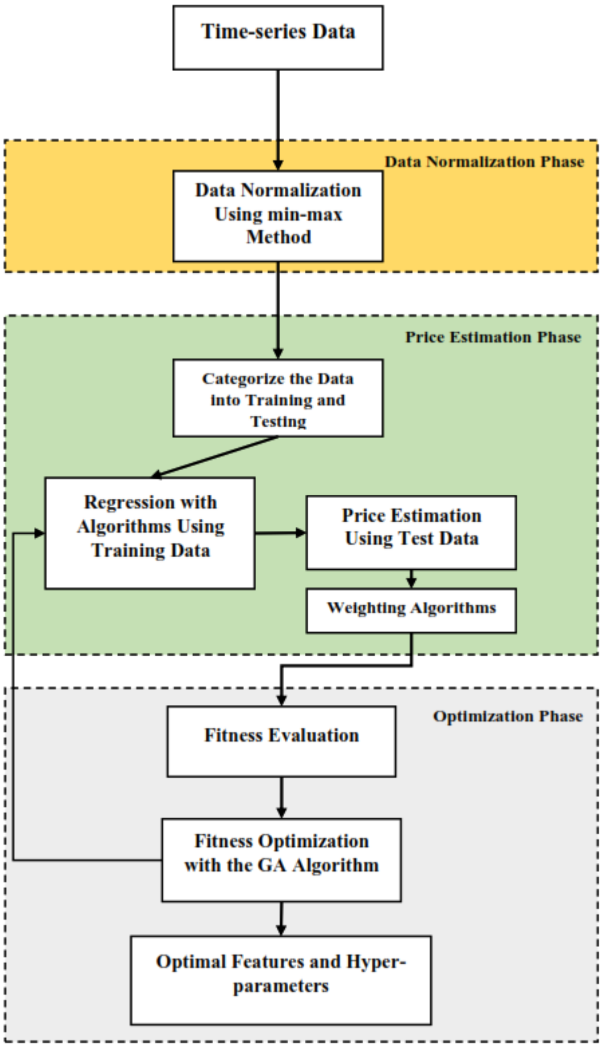

Fig 1. Outline the proposed algorithm.

Time-Series Data

The data used in this study are daily trades of some selected stocks on the Tehran Stock Exchange. To this end, the stock price of the Isfahan oil refinery with the symbol of Shapna was chosen from among the petroleum products group. In addition, two other companies' data were used to validate the proposed approach. For this purpose, a large company of the automobile group, 2 medium-sized companies of the banking group and base metals, and 2 small companies from the food and pharmaceutical products group were selected. Information on these firms can be found in

Table 1. These data are collected from April 2012 to the end of March 2021.

The input variables studied in the present study are price data (Open, High, Low, and Close) along with trading volume and volume and non-volume indicators. These variables are used as features for the classification and price estimation process. Non-volume indicators depend on price data and volumetric indicators depend on volume and/or price. In this search, stochastics, oscillators, indexes, and indicators are summarized as indicators. Nonvolumetric indicators are presented in Table 2. A total of 15 nonvolumetric indicators are included. It should be noted that certain indicators have a period and their values are impacted by that period. This period generally consists of 9 days and 12 days. Based on these periods, there are a total of 29 nonvolumetric indicators. The volumetric indicators examined in this study are presented in Table 3. 4 of these indicators do not have a period and the VROC indicator has a period. There are thus six volumetric indicators in total. Generally, given the open, high, low and closed prices, and trading volume, along with 29 non-volume indicators and 6 volume indicators, a total of 40 features were taken into account in this research.

Table 1: Information on the companies examined in this research.

Frim Name	Symbol	Market Cap (Billion Rial)	Size
Isfahan Oil Refinery	Shapna	824,600	Big
Iran Khodro	Khodro	639,510,864	Big
Kashan Amirkabir Steel	Fajr	111,596	Medium
Eghtesad Novin Bank	Vanovin	288,937	Medium
Chin-Chin Industry and Cultivation	Chechin	16,674	Small
Exir Pharmacy	Delor	5,085	Small

Table 2: Nonvolumetric Indicators.

Indicator	Needed Data	Period
Accumulation/Distribution Oscillator (A/DO)	Open, High, Low, Close	×
Stochastic oscillator (SO)	High, Low, Close	✓
Acceleration Between Times (Acc)	Open, High, Low, Close	✓
Momentum Between Times (MOM)	Open, High, Low, Close	✓
Simple Moving Average (SMA)	Close	✓
Exponential Moving Average (EMA)	Close	✓
Modified Moving Average (MMA)	Close	✓
Moving Average Convergence/Divergence (MACD)	Open, High, Low, Close	×
Relative Strength Index (RSI)	Close	✓
Williams %R	High, Low, Close	✓
Bollinger Middle, Upper, and Lower Bands	Close	✓
Price Rate of Change (ROC)	Close	✓
Williams Accumulation/Distribution line (WADLine)	High, Low, Close	×
Average of the high, low, and closing prices (Typical Price)	High, Low, Close	×
Chaikin Volatility	High, Low	✓

Table 3: Volumetric Indicators.

Indicator	Needed Data	Period
Negative Volume Index (NVI)	Close, Volume	×
Positive Volume Index (PVI)	Close, Volume	×
On-Balance Volume (OBV)	Close, Volume	×
Price and Volume Trend (PVT)	Volume	×
Volume Rate of Change (VROC)	Volume	✓

3.1. Data Normalization

It is necessary to first normalize the data due to their difference in size. The min-max method is one of the most widely used methods for data normalization. It is expressed as follows:

$$x_i^N = \frac{x_i - x_{min}}{xmin_{max}}$$

where x_i^N are the research variables, x_{min} is their minimum value and x_{max} is their maximum value.

3.2. Optimal Price Estimation

In this research, in order to estimate prices simultaneously, the LSTM-Deep RNN (LSTM-DRNN) network model proposed by Ding and Qin (2020) has been used. LSTM neural networks were developed to prevent the rapid loss of information that occurs in recursive neural networks (RNNs). Unlike RNN neurons, LSTM neurons can store four states, including the current value and the last output, as well as the current value and last state of the memory neurons. The LSTM neural network is a special form of the RNN neural network that has three gates. These three gates are: "input gate", "forgetting gate" and "output gate" (Fig. 2).

The LSTM network has a sigmoid neural network layer (Eq. (2)) and a pair of multiplication operations to select whether information is transmitted. Each output element of the sigmoid layer is a real number between [0,1] that represents the weight of the transmitted information. In addition, a layer containing the tanh activation function according to Eq. (3) is used to update the neurons:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad (2)$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (3)$$

Forgetting gates are used to determine rejected information by capturing h_{t-1} and x_t information and returning a value between 0-1 to the neuron state C_t. Eq. (4) shows how to calculate the probability of forgetting:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad (4)$$

where h_{t-1} represents the output of the previous neuron and x_t represents the input of the current neuron.

The input gate determines how much new information is added to the neuron state. First, the input layer, which contains the sigmoid activation function, determines which information needs to be updated, and then the *tanh* layer generates the cc candidate vectors, and the state of the neurons is updated according to Eq. (5):

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \qquad (5)$$

in which i_t and $\hat{C}_t$ are obtained from the following:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (6)$$

$$\hat{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \qquad (7)$$

The output gate is used to control how much of the current neuron state and control states are filtered:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad (8)$$

$$h_t = O_t \times tanh(C_t) \qquad (9)$$

Fig. 3 shows the LSTM-DRNN model, which is capable of simultaneously estimating the four open, high, low, and close prices.

In addition to the LSTM-DRNN model, two models of generalized regression neural networks (GRNNs) are used to estimate prices, as well as a regression method based on ensemble learning of least squares boosting (LSBoost). The estimated value of each price is equal to the weighted average value of the prices estimated by these algorithms:

$$\hat{P}(t)_i = \sum_{j=1}^{3} V_j \hat{P}(t)_{i,j}\,, \qquad i = 1,2,3,4.$$

Where $P(t)$ represents the estimated price, and V represents the weight of the algorithms. The weight of each algorithm is obtained through optimization.

An integrated optimization process is used to optimize the feature selection, and price estimation phases simultaneously using the genetic algorithm. The hyper-parameters of this research include the parameters of regression algorithms, the weight of each of these algorithms, along with 40 features studied. These hyper-parameters along with their upper (UB) and lower bounds (LB) are shown in Table 4. For attributes, the lower limit is 0 and the upper limit is 1, and these variables are considered binary, so that a value of 0 means that the attribute is not selected and a value of 1 means that it is selected. According to Table 4, 4 parameters are related to regression algorithms, 40 parameters are related to features and 3 parameters are related to the weight of each regression algorithms. Thus, in total, 47 hyper-parameters will be obtained from the optimization process.

Selecting the appropriate fitness is very important during the optimization process. The estimation error value is considered as a fitness in this study. Therefore, the optimal problem of the present study is defined as follows:

$$\max: \frac{\sum_{j=1}^{4} \overline{\mathrm{RMSE}_j}}{\overline{\mathrm{price}}}$$
$$\text{s.t.} \quad \mathrm{LB} \leq X_{\mathrm{GA}} \leq \mathrm{UB}, \quad \sum_{i=1}^{3} W_j = 1 \tag{11}$$

where the value of $\overline{\mathrm{RMSE}}$is also equal to the mean value of the difference between the actual price and the estimated price of each of the four prices and three algorithms, and $\overline{\mathrm{price}}$is equal to the average value of the real price. For example, for the opening price we have:

$$\mathrm{RMSE}_{Open} = \sqrt{\frac{\sum_{j=1}^{n} \mathrm{Open}_j - \sum_{i=1}^{3} \sum_{j=1}^{n} W_i \hat{\mathrm{Open}}_{ij}}{n}}$$

Where Open is the actual price, $\hat{\mathrm{Open}}$is the estimated value, and n is the number of observations (number of trading days). In addition, X_{GA}in Eq. (11) contains all 47 decision variables. There is also a constraint that the sum of the weights of the classification algorithms and the sum of the weights of the regression algorithms is equal to 1.

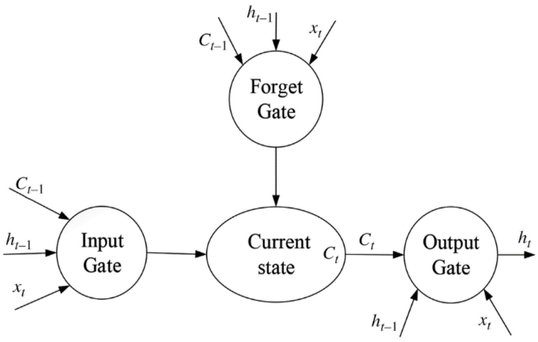

Fig 2: LSTM unit structure.

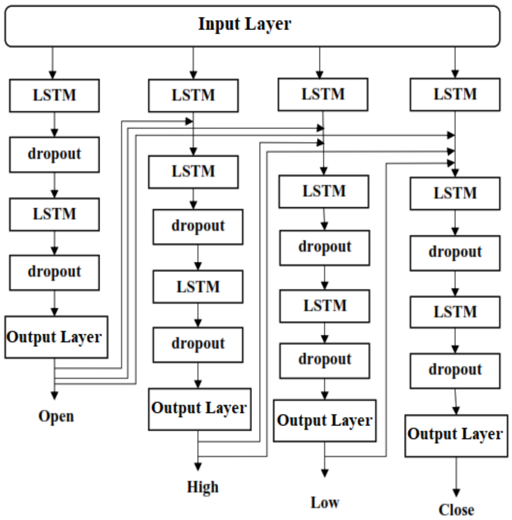

Fig 3: Proposed LSTM-DRNN network structure.

Table 4: Hyper-parameters and their LB and UB

Algorithm	Parameter	LB	UB
LSTM	No, of hidden layers	10	100
GRNN	Spread	0.01	1
LSBoost	No. of learning cycles	100	1000
	Learning rate	0.01	0.2
--	Attributes	0	1
--	Weights of regression algorithms	0.01	1

Results

By carrying out the integrated optimization process, the Shepna convergence diagram was obtained as indicated in Fig. 4. As illustrated in this figure, the algorithm could converge in 44 iterations. The optimum hyperparameter values can also be found in Table 5. The algorithm optimum weight values are also shown in Table 6. As we can see from this table, the LSTM algorithm has the greatest weight.

The optimal estimation of all 4 prices for the last 150 records is shown in Fig. 5. As can be seen from this figure, the estimated prices obtained from the optimal algorithm have a high accuracy, especially for the closing price. The mean percentage error of the estimate for each of the 4 prices is given in Table 7. This table shows that the lowest amount of error is related to the closing price, and the highest error is related to the highest price; however, the overall average estimation error of all 4 prices is less than 8%, indicating high accuracy estimation by the proposed method for all 4 prices simultaneously.

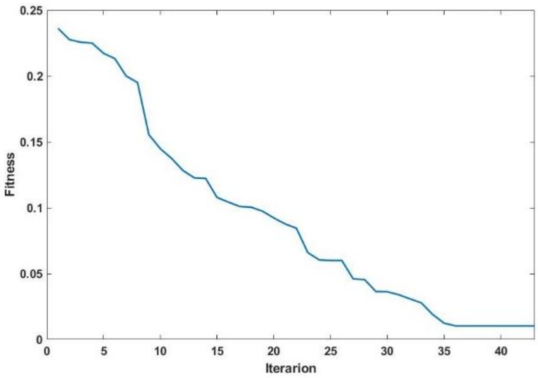

Fig 4: Optimizing process convergence diagram for Shepna.

Table 5: The optimum value of hyper-parameters for Shapna.

Algorithm	Parameter	Optimal Value
LSTM	No, of hidden layers	18
GRNN	Spread	0.027
LSBoost	No. of learning cycles	372
	Learning rate	0.026

Table 6: The optimum weight of algorithms for Shapna.

Algorithm	Optimal Value
LSTM	0.609
GRNN	0.326
LSBoost	0.064

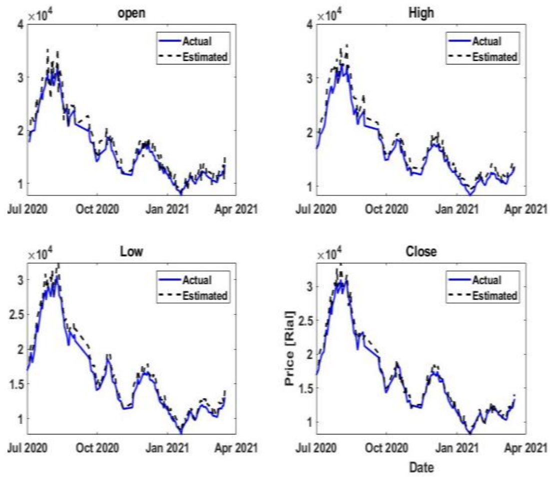

Fig 5: Optimum estimated prices for Shapna.

Table 7: Average percent error of Shapna's optimum estimate.

Price	Total average error (%)
Open	5.080
High	6.823
Low	4.064
Close	3.319

To validate the proposed integrated optimal method, it was evaluated for an additional 5 shares of Khodro, Fajr, Vanovin, Ghechin, and Delor. The optimum parameters obtained for these shares are presented in Table 8-10. Based on these tables, the LSTM algorithm is the most weighted. As a result, we can conclude that among the algorithms of the study, artificial neural networks and deep neural networks have the best performance. Furthermore, the number of optimal features for Khodro equal to 16, for Fajr and Delor is equal to 15, for Ghachin equal to 11, for Vanovin equal to 17 and for Shapna equal to 19. As shown in Table 10, certain features are common to all 6 shares studied and indicate the importance of these features in the optimization process. These features can be found in Table 11. According to this table, 11 common features have been selected, including prices and trade volume.

The estimated price of these shares is also shown in Fig. 6-10. As can be seen from these figures, the estimated prices obtained from the optimal algorithm have a high accuracy, especially for the closing price. This indicates that in terms of estimation, the proposed method is valid. Table 12 provides the average percentage of estimate error for each of the five shares. As this table shows, the price estimation error is below

10%. Therefore, this method can be regarded as a general method, since for each type of time series it is capable of estimating optimally and simultaneously.

Discussion and Conclusions

With the advent of the Deep Learning Process (DNN), a new era of artificial neural networks has been created, which have remarkable model recognition capabilities for longstanding data. In this research, an optimal method of multi-price estimation was presented, which is able to optimally and simultaneously estimate 4 opening prices, the lowest and the highest price and the closing price. Regression models depend on the number and type of features of the problem. Due to the fact that there are different feautures for different problems and time data, and it is not possible to consider a series of general features for all problems. This study considered 40 price, volume, volumetric and nonvolumetric indicators as features. Additionally, estimation algorithms have a series of parameters that require different experiment and testing to adjust their values. So designing a system that automatically adjusts those values is very practical and interesting. One of the best approaches to such problems is to use the optimization process with metaheuristic algorithms and a genetic algorithm at the top. This algorithm, with a search space and a fitness function, can obtain the optimal values of hyperparameters. The proposed method was used and evaluated for the Tehran Stock Exchange, which is a one-way market. The results of this study showed that the proposed method has a very good performance in terms of simultaneous multi-price estimation, and therefore it can be considered as a very good and suitable algorithm for automated trading in stock markets.

Table 8: The optimum value of hyper-parameters.

Algorithm	Parameter	Optimal Value				
		Khodro	Fajr	Ghechin	Vanovin	Delor
LSTM	No, of hidden layers	18	16	13	23	22
GRNN	Spread	0.034	0.032	0.026	0.046	0.027
LSBoost	No. of learning cycles	419	562	309	528	296
	Learning rate	0.026	0.017	0.038	0.029	0.023

Table 9: The optimum weight of algorithms.

Algorithm	Optimal Value				
	Khodro	Fajr	Ghechin	Vanovin	Delor
LSTM	0.532	0.563	0.597	0.527	0.599
GRNN	0.342	0.309	0.316	0.266	0.301
LSBoost	0.126	0.128	0.087	0.207	0.100

Table 10: The selected optimal features.

Features category	Optimal Features					
	Shapna	Khodro	Fajr	Ghechin	Vanovin	Delor
Price	open	open	open	open	open	open
	high	high	high	high	high	high
	low	low	low	low	low	low
	close	close	close	close	close	close
Volume	volume	volume	volume	volume	volume	volume
Nonvolumetric indecators	SO12	ADO	SO12	SO12	SO12	SO12
	ACC12	SO9	ACC9	ACC9	ACC9	ACC12
	MOM9	SO12	ACC12	ACC12	ACC12	MOM12
	MOM12	ACC9	MOM12	MMA12	MMA12	MACD
	MMA9	ACC12	MACD	MOM9	MOM9	RSI12
	MACD	MOM12	RSI9	MOM12	MOM12	ROC12
	RSI12	MACD	RSI12	MACD	MACD	Bollinger

Features category	Optimal Features					
	Shapna	**Khodro**	**Fajr**	**Ghechin**	**Vanovin**	**Delor**
						Upper9
	Bollinger Middle12	RSI12	ROC12	RSI12	RSI9	Williams12
	Bollinger Upper12	ROC12	Bollinger Lower9	ROC9	RSI12	
	ROC9			ROC12	ROC9	
	ROC12			WADLine		
				Williams9		
Valumetric indicators	NVI	PVI	VROC12	OBV	PVI	NVI
	PVI	VROC12		VROC12	VROC12	VROC12
	VROC12					
No. of features	**19**	**16**	**15**	**11**	**17**	**15**

Table 11: The selected common optimal features.

Features category	Optimal features
Price	open
	high
	low
	close
Volume	volume
Nonvolumetric indecators	SO12
	ACC12
	MOM12
	MACD
	RSI12
Valumetric indicators	VROC12
No. of features	**11**

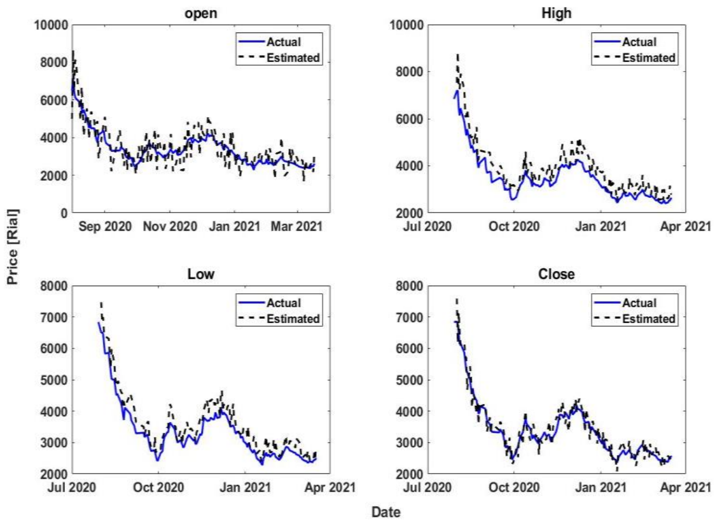

Fig. 6. Optimum estimated prices for Khodro.

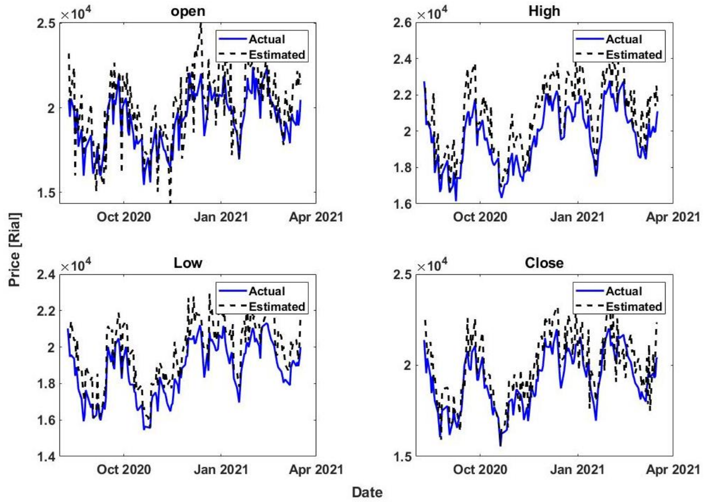

Fig. 7. Optimum estimated prices for Fajr

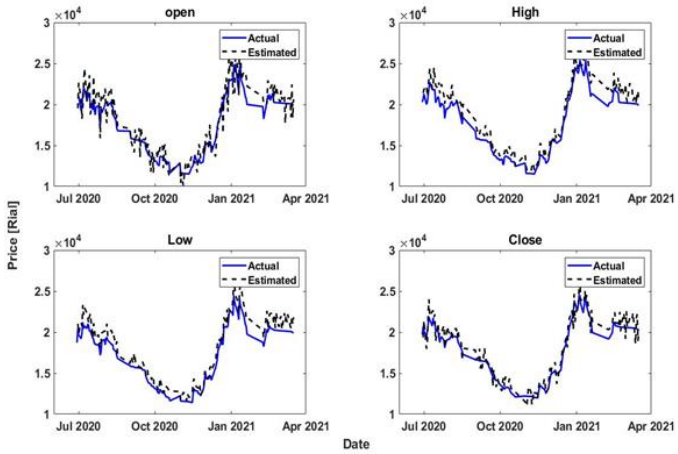

Fig. 8. Optimum estimated prices for Ghechin

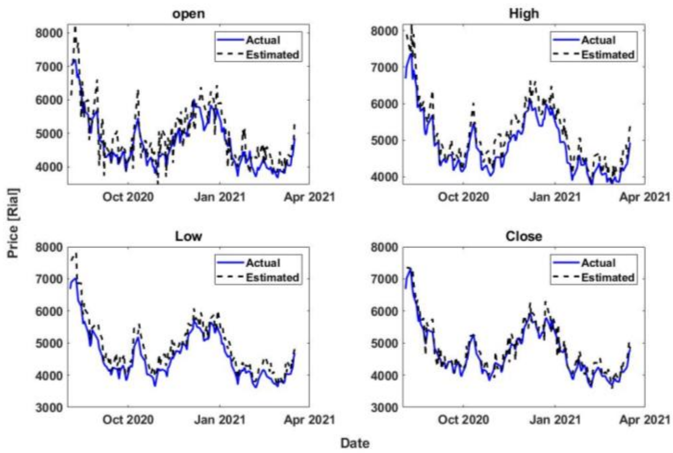

Fig. 9. Optimum estimated prices for Vanovin

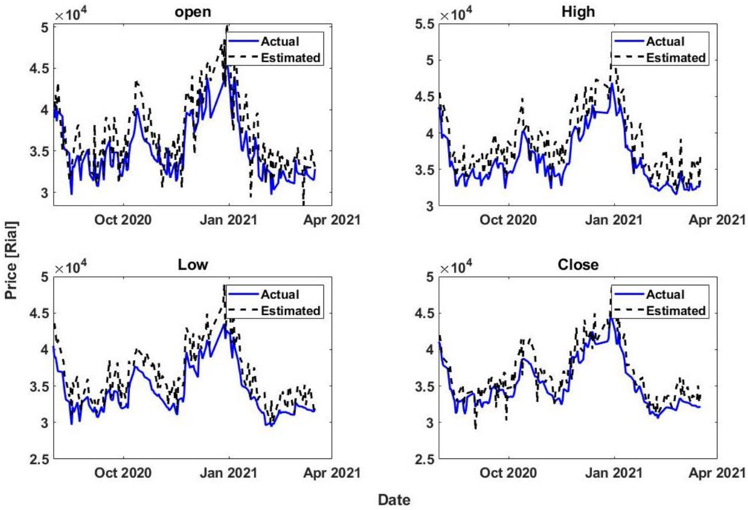

Fig. 10. Optimum estimated prices for Delor

Table 12: Average percent error of optimum estimate

Price	Total average error (%)				
	Khodro	Fajr	Ghechin	Vanovin	Delor
Open	3.760	8.093	6.356	6.264	6.097
High	9.003	6.702	8.267	7.457	9.361
Low	8.604	5.365	6.782	6.253	5.7854
Close	3.663	4.283	5.157	4.278	5.529

References

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, *113*, 457-480.

Batchelor, R., & Kwan, T. Y. (2007). Judgemental bootstrapping of technical traders in the bond market. *International Journal of Forecasting*, *23*(3), 427-445.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, *10*(10), 3765.

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, *11*(6), 1307-1317.

Dixon, M., Klabjan, D., & Bang, J. H. (2015, November). Implementing deep neural networks for financial market prediction on the Intel Xeon Phi. In *Proceedings of the 8th Workshop on High Performance Computational Finance* (pp. 1-6).

Gomber, P., & Zimmermann, K. (2018). Algorithmic trading in practice. In *The Oxford Handbook of Computational Economics and Finance*. Oxford University Press.

Hollis, T., Viscardi, A., & Yi, S. E. (2018). A comparison of LSTMs and attention mechanisms for forecasting financial time series. *arXiv preprint arXiv:1812.07699*.

Hota, H. S., Handa, R., & Shrivas, A. K. (2017). Time series data prediction using sliding window based RBF neural network. *International*

Journal of Computational Intelligence Research, *13*(5), 1145-1156.

Khajavi, S., & Amiri, F. S. (2017). Prediction of Stock Price using Particle Swarm Optimization Algorithm and Box-Jenkins Time Series. *International Journal of Finance & Managerial Accounting*, *2*(7), 25-31.

Kissell, R. (2021). *Algorithmic trading methods: Applications using advanced statistics, optimization, and machine learning techniques*. Academic Press.

Lei, K., Zhang, B., Li, Y., Yang, M., & Shen, Y. (2020). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Systems with Applications*, *140*, 112872.

Li, J., Bu, H., & Wu, J. (2017, June). Sentiment-aware stock market prediction: A deep learning method. In *2017 international conference on service systems and service management* (pp. 1-6). IEEE.

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, *7*, 108014-108022.

Li, Y., Zhu, Z., Kong, D., Han, H., & Zhao, Y. (2019). EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*, *181*, 104785.

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.

Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic trading. *Computer*, *44*(11), 61-69.

Rundo, F. (2019). Deep LSTM with reinforcement learning layer for financial trend prediction in FX high frequency trading systems. *Applied Sciences*, *9*(20), 4460.

Seyfert, R. (2018). Automation and affect: A study of algorithmic trading. In *Affect in Relation* (pp. 197-217). Routledge.

Wen, Y., & Yuan, B. (2018, March). Use CNN-LSTM network to analyze secondary market data. In *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence* (pp. 54-58).

Yan, H. (2019). *The real effects of algorithmic trading* (Doctoral dissertation, Duke University).

Yoshihara, A., Fujikawa, K., Seki, K., & Uehara, K. (2014, December). Predicting stock market trends by recurrent deep neural networks. In *Pacific rim international conference on artificial intelligence* (pp. 759-769). Springer, Cham

Zhao, Z., Rao, R., Tu, S., & Shi, J. (2017, November). Time-weighted LSTM model with redefined labeling for stock trend prediction. In *2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI)* (pp. 1210-1217). IEEE.