Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms

Article *in* Intelligent Systems in Accounting Finance & Management · December 2019

DOI: 10.1002/isaf.1459

CITATIONS
28

READS
2,730

3 authors, including:

Gholamreza Mansourfar

7 PUBLICATIONS **39** CITATIONS

SEE PROFILE

J. Bagherzadeh

Urmia University

40 PUBLICATIONS **232** CITATIONS

SEE PROFILE

Received: 22 February 2019 | Revised: 20 September 2019 | Accepted: 20 September 2019

DOI: 10.1002/isaf.1459

RESEARCH ARTICLE

WILEY

Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms

Mahla Nikou[1] | Gholamreza Mansourfar[1] | Jamshid Bagherzadeh[2]

[1] Faculty of Economics and Management, Urmia University, Urmia, Iran

[2] Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran

Correspondence
Gholamreza Mansourfar, Faculty of Economics and Management, Urmia University. Urmia, Iran.
Email: g.mansourfar@urmia.ac.ir

Summary

Security indices are the main tools for evaluation of the status of financial markets. Moreover, a main part of the economy of any country is constituted of investment in stock markets. Therefore, investors could maximize the return of investment if it becomes possible to predict the future trend of stock market with appropriate methods. The nonlinearity and nonstationarity of financial series make their prediction complicated. This study seeks to evaluate the prediction power of machine-learning models in a stock market. The data used in this study include the daily close price data of iShares MSCI United Kingdom exchange-traded fund from January 2015 to June 2018. The prediction process is done through four models of machine-learning algorithms. The results indicate that the deep learning method is better in prediction than the other methods, and the support vector regression method is in the next rank with respect to neural network and random forest methods with less error.

KEYWORDS

artificial neural network, deep learning, prediction, random forest, support vector regression

1 | INTRODUCTION

Stock price indices are very significant among the world financial markets as the main criteria for evaluating the function of securities and stock market. They are achieved from accumulation of stock price movements of all companies or certain class of existing companies in exchange market (Wang, Wang, Zhang, & Guo, 2012). Therefore, as financial markets present a high degree of competition among participants (Parot, Michell, & Kristjanpoller, 2019), the study of stock price index prediction models would be very necessary for investors to turn the securities market to a profitable place. This leads to updating investors' knowledge in making proper decisions in selection of an appropriate portfolio, on the one hand, and the provision of international investment opportunities for investors, on the other hand.

Nevertheless, prediction of financial markets is a complicated task, in that financial time series are noisy, nonstationary, and irregular. Although there are many statistical and computational methods for prediction of these series (Atsalakis & Valavanis, 2009), it is still identified as a difficult problem on financial variables and especially price index. Predictions are not precise, but their rate of error depends on the algorithm used. The important point is that, according to researchers, the behaviour of most variables follows a nonlinear trend in financial markets (Thomaidis, 2006); therefore, it is likely that linear prediction does not yield appropriate results for examination of the future path of financial variables.

In the financial literature, price prediction methods are classified into four groups of technical analytical methods, fundamental analysis, and prediction based on time series and machine learning. Through discovering new patterns in historical data, machine learning intends to identify the underlying function based on which data are formed and realize the linear and nonlinear models that exist in these data (Kalyvas, 2001). In past years, algorithms such as artificial neural networks and support vector machine (SVM) were widely used for prediction of financial series and achievement of high precision in prediction (Das & Padhy, 2012; Guo, Wang, Liu, & Yang, 2014; Lu, Lee, & Chiu, 2009). The results of studies indicate the considerable superiority of

such methods over the traditional methods in stock price prediction (Aydin & Cavdar, 2015; Giovanis, 2009; Yim, 2002). For this reason, in many countries, such as the USA, Germany, France and the UK, researchers have succeeded in predicting stock market behaviour through various machine learning methods (Huang, Nakamori, & Wang, 2005; Tay & Cao, 2001; Wei, 2012). Machine learning methods are often considered to be a subfield of artificial intelligence. However, there are a large number of studies employing machine learning, as opposed to other artificial intelligence methodologies (Fisher, Garnsey, & Hughes, 2016).

Deep learning is one of the machine learning models that has recently been innovated. It is a branch of machine learning and artificial intelligence and a set of algorithms that try to model high-level conceptual concepts using learning in various levels and layers. Learning could be supervised, semi-supervised, or unsupervised (Bengio, Courville, & Vincent, 2013; LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). The most important advantages of deep learning are automatic learning of features, multilayer learning of features, high precision in results, high generalization power, and identification of new data. Therefore, the aim of this study is to forecast the close price of iShares MSCI United Kingdom through this method and to compare its prediction power with other machine learning methods.

2 | Theoretical Background

Although the initial concepts of machine learning emerged in the 1950s (Samuel, 1959), it was studied in the 1990s as an independent field (Michalski, Carbonell, & Mitchell, 2013). Machine learning as a highly applicable branch of artificial intelligence uses computers for simulation of human learning and building algorithms that could perform learning and prediction based on data (Anon., 1998; Portugal, Alencar, & Cowan, 2018). There are various types of machine learning algorithms, the most important of which include artificial neural networks, SVMs, random forests (RFs), and deep learning.

As far as stock markets are usually nonlinear and chaotic systems, nonlinear algorithms are preferred to linear algorithms in predicting stock behaviour. The most important nonlinear models that have been widely used in recent years in financial markets and achieved desired results are artificial neural network models. Whitcomb for the first time sought to answer the question of whether neural networks are capable of identifying nonlinear laws in time series and unknown laws in asset price movement and stock price variations (Schwartz & Whitcomb, 1977). Artificial neural networks usually function better than regression methods. For example, the findings of Hill, Marquez, O'Connor, and Remus (1994) on stock price prediction indicated the superiority of this method over other models. Moshiri and Cameron (2000) predicted the inflation in Canada based on various time series models of autoregressive–moving average (ARMA), vector autoregression (VAR), and Bayesian VAR, structural models, and artificial neural network models on the basis of a multilayer feedforward neural network. The results indicate that neural network models can predict the competitive models and in some cases are even more efficient. Using a neural network model, Yim (2002) predicted the return of the daily index of Brazil stock and compared the results of prediction using root-mean-squared error (RMSE), mean absolute error (MAE), and the Chung and Hendry model with the results of generalized autoregressive conditional heteroskedasticity and ARMA and structural prediction models and showed the superiority of the artificial neural network.

Kumar and Thenmozhi (2005) carried out some studies on the efficiency of each of the ARMA, RF, SVM, and artificial neural network methods in prediction of the S&P CNX NIFTY index rate and compared the function of these three nonlinear models and one linear model with each other. The results indicated that the SVM could yield more precise results. Zhang, Shi, Zhang, and Shi (2006) predicted the stock exchange price trend of Shanghai using an SVM and concluded that SVM has a high prediction capability and the combined model of an SVM and intelligent models has even better results than the SVM model. Lee (2009) dealt with prediction of the NASDAQ index through a combined support vector regression (SVR) model and its comparison with artificial networks. In this study, SVR was combined with F-score and supported sequential forward search and utilized for index change by 29 technical indices as a set of complete features. The results of the study indicated the superiority of the combined SVR estimator model over the neural network. Giovanis (2009) investigated the stock price in the Athens stock market and then forecasted the stock price values. He concluded that the neural network model had a lower rate of error and benefitted from relative superiority. Kara, Boyacioglu, and Baykan (2011) predicted the stock price movement direction of Istanbul through fuzzy-neural network models and SVM. The prediction performance of the fuzzy-neural network was 74.5% and that of the SVM was 52.71%, and so the fuzzy-neural network functioned better than the SVM. Aydin and Cavdar (2015) studied the USD–TRY relation, gold price, and Borsa Instanbul (BIST) 100 using an artificial neural network and VAR and compared the results obtained from the two methods with each other. The results showed that the neural network approach functions better than VAR in prediction.

Gao (2016) forecasted the stock market using a recurrent neural network (RNN) through long short-term memory (LSTM). This study intended to investigate the feasibility and efficiency of LSTM in stock market forecasting. Based on the results, the average precision and accuracy of the LSTM model in forecasting six shares was 54.83%, whith the highest and lowest precision being 59.5% and 49.75% respectively. Bao, Yue, and Rao (2017) presented a new deep learning framework where wavelet transforms, stacked auto encoders, and LSTM methods were combined for stock price prediction. For examination of the performance of the proposed model, six market indices and their related future indices were selected. The results indicated that the proposed model is better in terms of the precision of prediction and profitability compared with other similar models. Hiransha, Gopalakrishnan, Menon, and Soman (2018) forecasted the National Stock Exchange (NSE) of India stock market using four deep learning models: multilayer perceptrons (MLPs), RNNs, LSTM, and convolutional neural networks (CNNs). Data used in this study were

the closing prices of two different stock markets: the NSE and the New York Stock Exchange. The results indicated that deep learning models are better than ARIMA. Moreover, CNNs function better than other deep learning models. Using real-life credit card data linked to 711,397 credit card holders from a large bank in Brazil, Sun and Vasarhelyi (2018) confirmed that—compared with machine-learning algorithms of logistic regression, naive Bayes, traditional artificial neural networks, and decision trees—deep neural networks have a better overall predictive performance. The recent success of deep networks is partially attributable to their ability to learn abstract features from raw data. This motivated Galeshchuk and Mukherjee (2017) to investigate the ability of deep learning for the change direction prediction of foreign exchange rates by using CNNs over three different exchange indexes. They concluded that trained deep networks achieve satisfactory out-of-sample prediction accuracy.

3 | DATA AND METHOD

3.1 | Data

The data used in this study include the close price of daily data of iShares MSCI United Kingdom from January 2015 to June 2018. The iShares MSCI United Kingdom fund (NYSEARCA: EWU) is an exchange-traded fund that provides investors with access to a selected equities market by tracking the total performance return of the MSCI United Kingdom Index as its target benchmark. The MSCI United Kingdom Index is composed of 111 large- and mid-cap company stock holdings that together represent nearly 85% of the total free float-adjusted market capitalization of UK. This index utilizes the standards and methodologies of MSCI Global Investable Market Indexes, which attempt to cover a wide range of regional, market capitalization, sector and style segments of the international equity markets. EWU follows the same standards as its benchmark indices. As of August 2015, EWU has provided investors with an annualized total return of 5.9% since the fund's inception in 1996.

Research data has been collected from the Yahoo Finance site. For price index prediction using a deep learning algorithm and other machine learning methods and for comparison of the results for selection of the best algorithm, coding has been done in Python 3.5.1 software. The error evaluation criteria include MAE, mean-square error (MSE), and RMSE.

4 | Method

4.1 | Artificial neural network

The concept of artificial neural networks was proposed for the first time by McCulloch and Pitts (1943) and later on was widely used by researchers in experimental modelling of nonlinear processes. The perceptron is one type of neural network, having single-layer and multilayer forms. By presenting the back-propagation (BP) algorithm McClelland, Rumelhart, and Hinton (1986) succeeded in the development of a multilayer feedforward neural network (MLP). Neural networks have wide applications in financial and investment areas, such as prediction of bankruptcy, decision-making, and financial planning.

The nodes in feedforward networks are located in consecutive layers with a one-directional relation. Upon the arrival of an input pattern at the network, the first layer calculates the output value and delivers it to the next layer. The next layer receives this number as the input and transfers its output values to the next layer. The BP algorithm includes some calculations whereby the error due to the difference between network output and real value is returned to the network and the network parameters are regulated such that with the next similar input pattern it presents better output with lower error value (Haykin, 1999).

Neural network learning is done through weights balance with the aim of minimizing an error function using the balance of initial weights (Lawrence, 1997; Tan, 2009). The interneuron connections in most neural networks are such that the neurons of mid layers receive their input value from among all neurons of its underlying layer (usually the input neurons). In this way, the signals in a neural network move from the input layer to the upper layers and at the end reach the outer layer and the network output which is called feed forward.

4.2 | Support vector machine

The other nonlinear model widely used in financial markets in recent years and which has achieved desirable results is the SVM. The SVM has been used for prediction of time series in nonstationary status of variables, unjustifiability of classic methods, or time series complexity (Tian, 2015). SVM models are classified into SVM and SVR. SVRs are a certain type of SVM used for future price prediction (Das & Padhy, 2012). The SVM has the capability in the prediction process to eliminate irrelevant and scattered data and improve the precision of prediction (Wei, 2012). The SVM is based on the structural minimization of risk taken from statistical training theory.

The SVM in financial data modelling is applicable as long as there are no strong assumptions. The basis for the SVM is the linear classification of data, where attempts are made to select a line with higher reliability. Solving the equations of the optimal line for data is done through quadratic programming methods, which are known methods in solving constraining problems. Before linear classification, data are transferred by a phi function to a wider space to enable the machine to classify the highly complex data. Briefly speaking, this algorithm uses a nonlinear mapping to convert the main data to a higher dimension and utilizes a linear optimality for separation of the hyperplane (Han, Pei, & Kamber, 2011).

4.3 | Random Forest

This algorithm was introduced in 2001 for the first time by Leo Breiman and is based on a decision trees model known as classification and regression trees (CART; Breiman, Friedman, Olshen, & Stone, 1984).

Decision trees that are used to forecast classified variables are known as classification trees, since they place the samples in classes or categories, and the decision trees used for prediction of continuous variables are called regression trees. However, the group nature of the RF algorithm leads to compatibility with changes and eliminates instability.

Two main features in construction of RFs are bagging (Breiman, 1996) and random selection in each node. The bagging method is an algorithm based on bootstrapping and composition concepts for improving machine learning. In machine learning, group algorithms combine several weak learners to achieve a strong learner, which prevents data overfitting. In bagging, good results are produced when the base classifications are part of unstable learning algorithms (such as a decision tree with a neural network) such that small changes in the training data lead to main variations in the model constructed by that algorithm.

The characteristic of random features is that, in each node of each tree, a small group of input features is randomly selected and for node division the best feature with the highest information efficiency is selected for tree growth rather than searching all features. The number of these features is less than the number of main features. Each tree in the RF grows using the CART decision tree algorithm to its maximum size without any pruning. Breiman (2001) utilized $\left[\log_2^M\right]+1$ features in each node, where M is the total number of input features. More details of RFs can be seen in Booth, Gerding, and McGroarty (2014) and Breiman (2001).

4.4 | Deep learning

Deep learning is generally considered as a subset of machine learning and was first introduced in 2005 and has seriously been taken into account since 2012. The basis of deep learning is on learning representation of knowledge and features in model layers (Bengio, 2009). Using new facilities and technologies, the deep learning idea, inspired by the human brain structure (Olshausen & Field, 1996), has considerably succeeded in most areas related to artificial intelligence and machine learning. In fact, deep learning means the investigation of new methods for artificial neural networks (Collobert, 2011; Gomes, 2014); in fact, it is a new attitude toward the idea of neural networks. Neural networks have an internal hidden layer, and a network with several internal hidden layers is called a deep neural network (Bengio, 2009; Schmidhuber, 2015).

In deep learning, iteration and the creation of higher depths are used to train computers some practices and a deep investigation of the subject matter is attempted. Deep neural network models include autoencoder neural networks, deep belief networks, CNNs, and RNNs. CNNs are unique to visual data and have wide application in image and video recognition and natural language processing (Collobert & Weston, 2008). Moreover, RNNs are appropriate for timed data (Brax, 2000). Therefore, since the data used in this study are time series, RNNs have been used for prediction. RNNs are neural networks with one or more recurrent loops. These networks are in fact created for processing sequential signals and have one type of memory that records the data observed so far. Theoretically, these networks could record and utilize data in a long sequence; however, it is not the case in practice, and they are so limited in this regard such that they just record data related to a few past steps. The main feature of an RNN is its hidden state, which stores the data of a sequence. Moreover, it is not necessary to have an output and input in each time step.

In a standard RNN structure, the accessible content scope is very limited in practice. The problem is that the effect of a given input on the hidden layer, and consequently on the output of the network, decreases exponentially and vanishes, which is known as the vanishing gradient problem (Pascanu, Mikolov, & Bengio, 2013). RNNs are called recurrent since the output of each layer depends on the calculations of its previous layers; in other words, these networks have memory that stores the information related to the observed data. These networks are in fact various copies of ordinary neural networks that are arranged beside each other and each transmitting a message to another.

Training in neural networks includes four stages.

1. Preparation of training data: the higher the number of data, the better the training. In this stage, the data preprocessing is one, the aim of which is elimination of incomplete data and undesired variations and data simulation. For example, normalization (e.g. transmission to a zero–one range) is one of the data preprocessing methods.
2. Selection of an appropriate network architecture. In this stage, the appropriate numbers of neurons, layers, and type of network will be determined; the number of layers and neurons is determined by trial and error.
3. Network training. In this stage, the training cycle is iterated until achievement of the desired results.

Error or cost is the difference between network output and desired output, which is calculated through cost/loss function. Some examples of cost functions are mean squared, cross-entropy, hinge, and Softmax. Then optimization will be performed for modification and update of weights to achieve the least error. Some of the optimization algorithms in deep neural networks are SGD, SGD + Momentum, RMSprop, Adagrad, Adadelta, and Adam. In these networks, the weight modification is done through BP and the training rate is determined so that the size of steps moves toward least error.

4. Training improvement methods. These methods include batch normalization, dropout, and transfer learning.

The calculations in RNNs are as follows. First, the hidden layer in an RNN changes according to:

$$a_h^t = \sum_{i=1}^{I} \omega_{ih} x_i^t + \underbrace{\sum_{h'=1}^{H} \omega_{h'h} b_{h'}^{t-1}}_{\substack{\text{recurrent relations} \\ \text{(in addition to MLP)}}}$$

- ω_{ih} is the weight between the ith input and hth neuron from the hidden layer

- x_i^t is the ith input of network in moment t
- $\omega_{h'h}$ is the weight between neurons of the hidden layer.

The BP training changes according to:

$$\delta_h^t = \theta'(a_h^t)\left(\sum_{k=1}^{K}\delta_k^t\omega_{hk} + \underbrace{\sum_{h'=1}^{H}\delta_{h'}^{t+1}\omega_{hh'}}_{\text{recurrent relations}}\right)$$

(in addition to MLP)

4.5 | Long short-term memory

LSTM is a type of model or structure for sequential data that has emerged from development of RNNs and improved by Gers, Schmidhuber, and Cummins (2000). Long-term memory refers to learned weights and short-term memory refers to internal states of cells. LSTM was created for the vanishing gradient problem in RNNs whose main change is replacement of the RNN mid layer with a block that is called an LSTM block (Hochreiter & Schmidhuber, 1997). The main feature of LSTM is the possibility of long-term affiliation learning, which was impossible with RNNs. To forecast the next time step, it is required to update the weight values in the network, which requires maintenance of the initial time step data. An RNN could just learn a limited number of short-term affiliations; however, long-term time series, such as 1000 time steps, cannot be learned by RNNs; in contrast, LSTMs could properly learn these long-term affiliations (Schmidhuber, 2015). The LSTM structure includes a set of recurrent subnetworks, called memory blocks. Each block includes one or more autoregressive memory cells and three multiple units of 'input, output, and forgetting' that present the analogues of continuous writing, reading, and regulation of the cells' functions. Moreover, there are various types of LSTM blocks, including stacked LSTMs, encoder–decoder LSTMs, bidirectional LSTMs, CNN LSTMs, and generative LSTMs (Brownlee, 2017).

When a computational problem can be solved using a certain algorithm, the very next issue, after solvability, is the time and space complexity of the solution method. The interesting point is the behaviour of the algorithm when applied to larger problems (Rojas, 1996).

If d is the length of input vector and h is the number of neurons in the hidden layer, then the memory space needed for an LSTM cell is $O(d \times h)$. As the result of the next cell $(t+1)$ is replaced in the same memory for the old values, the amount of memory needed in LSTM is $O(d \times h)$. Normally, space complexity is not considered a primary issue, since the computational models are provided with an infinitely large memory. What is most interesting for applications is time complexity; therefore, when speaking about the complexity of an algorithm, it is actually about its time complexity (Rojas, 1996).

In an LSTM unit, there are a total of eight matrix multiplications and three vector multiplications. As each matrix multiplication is $O(d \times h)$ and vector multiplication is $O(d)$ or $O(h)$, the total time complexity of an LSTM unit is $O(d \times h)$.

In these systems, learning time is not important since the learning is completed before the running of the actual system and there is enough time available for it. What is important is the accuracy of learning and reducing errors.

5 | RESULTS

The closing price of iShares MSCI United Kingdom was analysed from January 2015 to June 2018 using four machine learning methods. Table 1 presents the descriptive statistics for the closing price of the index. Based on the Jarque–Bera test, these data do not have a normal distribution because, concerning the probability value of statistics $(P = 0.00001)$ at a reliability level of 95%, the zero assumption on data normality is rejected. The index closing price diagram is shown in Figure 1.

TABLE 1 Descriptive statistics

Mean	Median	Maximum	Minimum	Std. Dev.	Skewness	Kurtosis	Jarque–Bera	Probability
33.85622	34.04000	39.80000	27.98000	2.533603	0.127012	2.142125	29.31752	0.000000

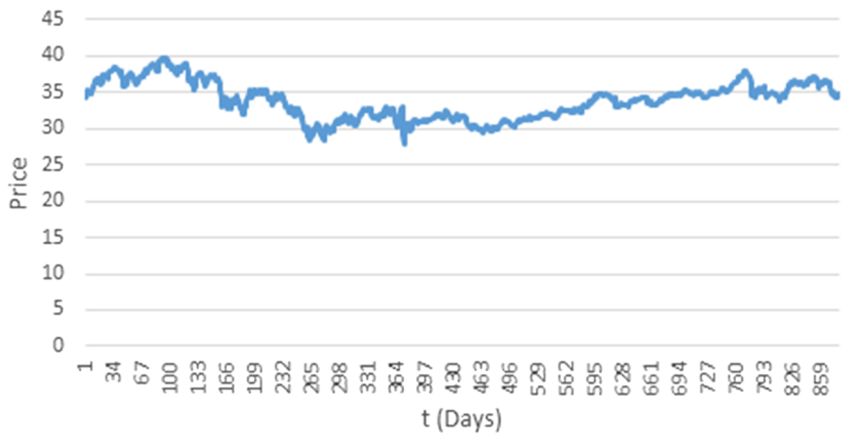

FIGURE 1 The closing price of iShares MSCI United Kingdom index

5.1 | Results of prediction with artificial neural network

In this study, an MLP neural network was used, and 80% of the data were utilized for training and 20% for testing the results. Ten neurons representing lags of the closing price of iShares MSCI United Kingdom index were used in the input layer, a rectified linear unit transfer function was used in the hidden layers, and a linear transfer function in was used the output layer. In order to find the best prediction model, the number of hidden layers and neurons in hidden layers are given to the program and the functional indices are calculated based on each of them. Moreover, for this purpose, the AdamOptimizer algorithm and MSE cost function were used, and BP was used for weight modification. The best model is achieved by iterating the tests on various values of parameters with an epoch value equal to 100 and a batch size equal to 10. In order to determine the number of hidden layers in the network, it is concluded that a network with a hidden layer has less error. The results indicate that rarely does a network with more than three hidden layers improve prediction issues (Kaastra & Boyd, 1996). In order to determine the appropriate number of data processing units, the number of neurons in the first hidden layer was changed between 5 and 15, and in the second hidden layer, depending on the first hidden layer, some values were examined and for prevention of error the aforementioned process was iterated several times (see Table 2).

By iteration of tests on different values of parameters, the results indicate that for iShares MSCI United Kingdom, the best model is a single hidden-layer network with a 10–12–1 structure. This structure has a minimum error mean in which 10, 12, and 1 refer to the number of neurons in the input layer, hidden layer, and output layer respectively.

5.2 | Results of prediction with support vector regression

After data examination and preprocessing, the data were divided into training and test data and the inputs of the model are similar to the neural network method. It should be noted that 695 data out of 869 data are for training and 174 data for test. In order to construct the SVR model, it is required to optimize the parameters of the model. SVR utilizes various kernel functions for data processing, including different types of radial basis, polynomial, annular, and linear functions. Rüping (2001) showed that the radial basis function kernel function acts well in various types of time series and learning algorithms. Moreover, in this method, the *k*-fold cross-validation method is used to estimate the efficiency of the machine learning model. The common values for *k*-fold are considered as 5 and 10 (Hastie, Tibshirani, & Friedman, 2009). Moreover, it is required to specify the optimum values of *C*, epsilon, and gamma parameters. To this end, the following values were selected:

C: $2^{-5}, 2^{-4}, \ldots, 2^{9}, 2^{10}$
epsilon: $2^{-10}, 2^{-9}, \ldots, 2^{-1}, 2^{0}$
gamma: $2^{-20}, 2^{-19}, \ldots, 2^{9}, 2^{10}$

The optimum values of parameters *C*, epsilon, and gamma obtained are given in Table 3. Consequently, it is determined that the most appropriate SVR model is the one with *k*-fold 10.

5.3 | Results of prediction with random Forest

In this method, data are divided into training and test groups as in previous methods, with 80% of data considered for training and the rest

TABLE 2 Determination of the number of hidden layers and appropriate number of neurons in the hidden layer

First layer	MAE	MSE	RMSE	Second layer	MAE	MSE	RMSE
5	0.51129	0.46978	0.676355	5	0.388393	0.301373	0.540229
6	0.393792	0.300025	0.543319	6	0.33738	0.216018	0.459981
7	0.450537	0.384993	0.612582	7	0.381501	0.254027	0.491368
8	0.370184	0.277556	0.5073	8	0.43648	0.355222	0.589392
9	0.444537	0.351343	0.585778	9	0.323808	0.20477	0.44431
10	0.468681	0.405391	0.614578	10	0.392794	0.311001	0.534902
11	0.405735	0.313126	0.543371	11	0.352541	0.249768	0.492202
12	0.342613	0.207035	0.454131	12	0.3717	0.26475	0.508939
13	0.400007	0.276403	0.520171				
14	0.434626	0.306795	0.553386				
15	0.347109	0.244483	0.489031				

TABLE 3 Determination of optimum parameters of SVR

k-fold	MAE	MSE	RMSE	Optimal gamma	C	Epsilon
5	0.266410819	0.149248498	0.386326932	0.125	128	0.0625
10	0.240020103	0.116047328	0.3406572	0.125	4	0.03125

for test. The research data are single-variable time series; therefore, the inputs of model are selected on a 10-lag basis. First, the optimal number of trees in the forest is determined; in this study we examined 100–1000 trees in steps of 100 trees. Then the number of *x*-ratio variables for the decision tree is determined, selected here from 0.1 to 1 in steps of 0.1 and the RF regressor is implemented for each *x*-ratio and the values of the target variable for out-of-bag (OOB) samples is estimated. The optimal *x*-ratio value with lower OOB error is determined. In the next step, the RF regressor model is created based on the number of trees and optimal *x*-ratio. The results obtained are presented in Table 4. Based on the results, the number of trees in the forest and the optimal *x*-ratio should have lower OOB and test error;

TABLE 4 Determination of RF optimal parameters

n-trees	MAE	MSE	RMSE	Optimal *x*-ratio
100	0.296496556	0.155479333	0.394308677	0.8
200	0.2973361	0.153507168	0.391782986	0.8
300	0.296117561	0.152496081	0.390493848	0.9
400	0.294691597	0.151731948	0.389482008	0.8
500	0.29645561	0.152896251	0.391011462	0.8
600	0.294768275	0.152013321	0.389887752	0.8
700	0.295439408	0.152220109	0.390144206	0.8
800	0.295599609	0.152755575	0.390828096	0.9
900	0.295587397	0.153428457	0.391697861	0.9
1000	0.295868256	0.153137577	0.391323467	0.9

TABLE 5 The result of determining optimal number of neurons

No. of neurons	MAE	MSE	RMSE
50	0.238623737	0.116605098	0.340227102
100	0.214368733	0.088005688	0.296063146
150	0.21382158	0.088199563	0.296829412
200	0.215998857	0.091932023	0.3028901
250	0.211789958	0.084693724	0.290793214
300	0.204745693	0.078053354	0.279368997
350	0.2122883	0.092136384	0.303410223
400	0.212813319	0.090775564	0.30122451

TABLE 6 Determination of optimal dropout rate *p*

Dropout rate *p*	MAE	MSE	RMSE
0.5	0.211620	0.093981	0.306561
0.6	0.212588	0.093990	0.306584
0.7	0.213330	0.094644	0.307618
0.8	0.210350	0.093969	0.306543

TABLE 7 Summary of results obtained for the close price of iShares MSCI United Kingdom

	ANN	SVR	RF	LSTM
MAE	0.342613	0.24002	0.294692	0.210350
MSE	0.207035	0.116047	0.151732	0.093969
RMSE	0.454131	0.340657	0.389482	0.306543

ANN: artificial neural network.

therefore, the appropriate number of trees and optimal *x*-ratio are 400 and 0.8 respectively.

5.4 | Results of prediction with deep learning algorithm

In this study, RNNs are used from deep neural network models. Moreover, the LSTM block has been used due to the vanishing gradient problem in these networks. From two states of LSTM, stateful LSTM will be used because it is applicable in cases where the results are interrelated. In this method, 80% of data are considered for training and 20% for testing. Similar to previous methods, 10 neurons were used in the input layer. After data preprocessing, such as making data stationary and data scaling between −1 and + 1, the RNN model was created with the LSTM block. To this end, first, the optimal values of the number of neurons should be determined. Batch size is considered equal to 1 and epoch value equal to 100. Moreover, in this method, the AdamOptimizer algorithm has been used for weight modification and the MSE cost function; and we used the dropout method to prevent overfitting in training the deep neural networks. Dropout has a tuneable hyperparameter *p*

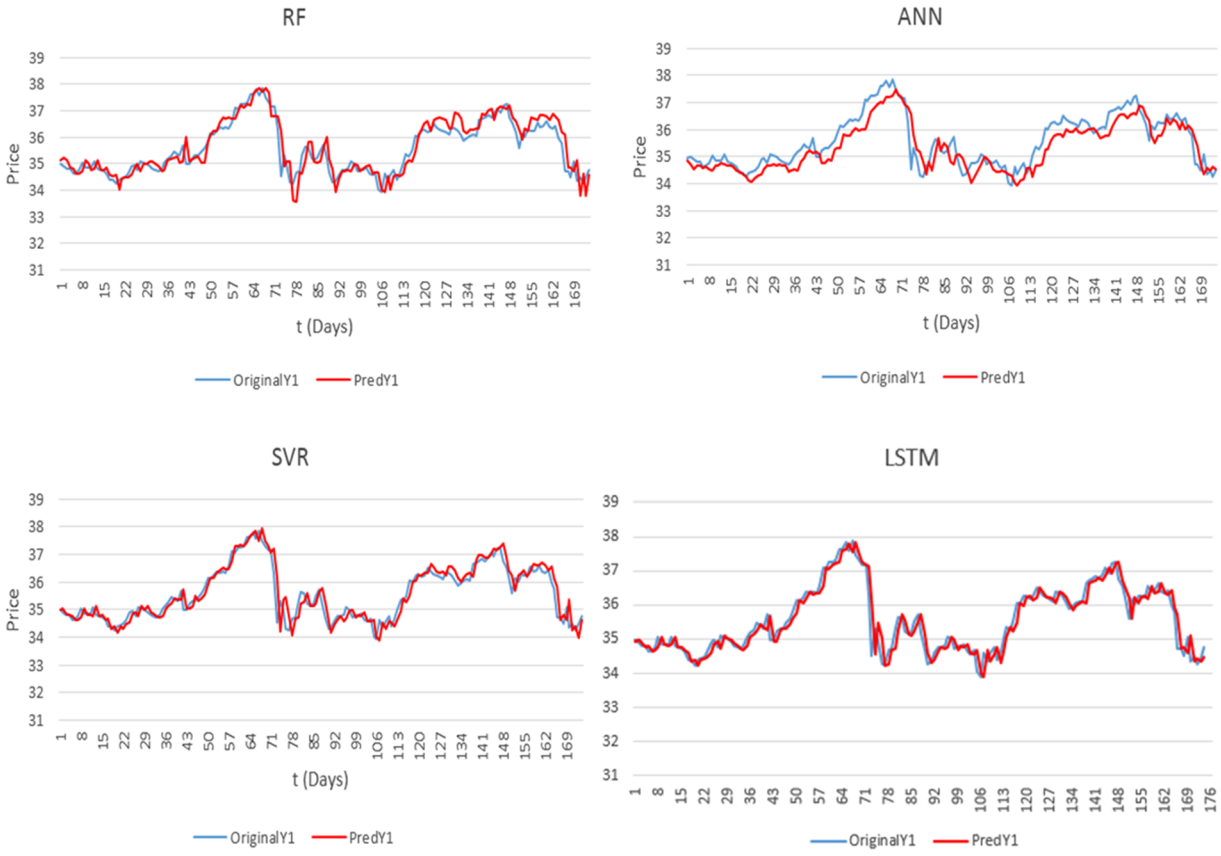

FIGURE 2 Prediction results with artificial neural network (ANN), RF, SVR, and LSTM

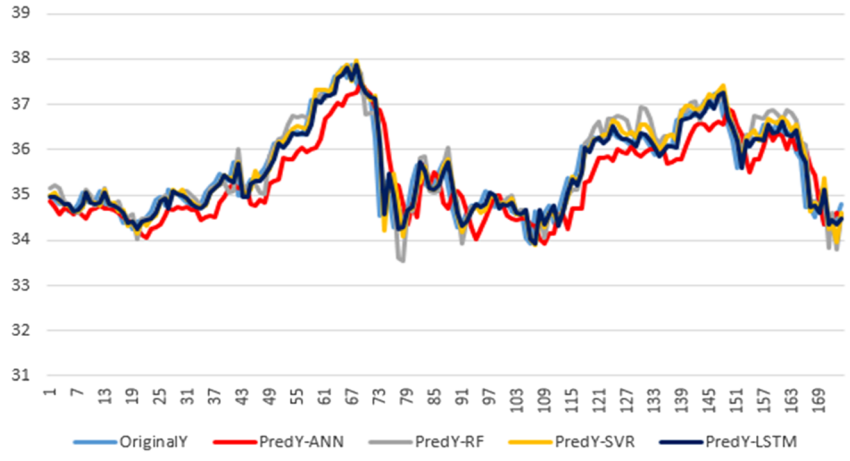

FIGURE 3 iShares MSCI United Kingdom index prediction with machine learning methods

(the probability of retaining a unit in the network). $p = 1$ implies no dropout, and low values of p mean more dropout. Typical values of p for hidden units are in the range of 0.5 to 0.8 (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Iterating the tests on different values of parameters will yield the best model with less error, the results of which are presented in Tables 5 and 6. Concerning the results, the optimum value of the number of neurons is considered as 300. Then we tried different values of p between 0.5 to 0.8, and the results showed that the best accuracy is achieved with $p = 0.8$ Table 7.

Figure 2 presents the graphs for each of the aforementioned methods on predicted test data and real test data. The red lines indicate the predicted test data and the blue lines indicate the real test data. As seen, the error in forecasting the close price of iShares MSCI United Kingdom is less in deep learning than with the other methods. After deep learning, SVR functions better than the other methods (Figure 3).

6 | CONCLUSIONS

Stock market prediction is so difficult due to its nonlinear, dynamic and complicated nature; yet, it is so important. Successful prediction has some interesting benefits that usually affect the decision of a financial trader on the purchase or sale of a financial instrument. One of the main factors considered by investors in making investment and stock buy and sell decisions is the stock price index. This study has been carried out with the aim of predicting the close price of iShares MSCI United Kingdom. To this end, four data mining techniques (i.e. artificial neural network, SVR, RF, and LSTM) were implemented using coding in Python and their functions were compared with the daily close price of iShares MSCI United Kingdom from January 2015 to June 2018. The analysis and summary of results obtained is as follows.

The results of the study show that the recurrent network method with an LSTM block functions better in prediction of the close price of iShares MSCI United Kingdom than the other methods, and then the SVR method has higher precision than neural network and RF. Hence, the following recommendations can be made:

- For predicting the close price of iShares MSCI United Kingdom, the deep learning method is better than the other methods tested; therefore, investors, managers, and market analysers are recommended to use this method. On the other hand, proper prediction of price indices could have a significant effect on profitability and making proper decisions on portfolio of investors and stakeholders. The results of this study could help investment companies and other users in evaluation of the profitability in selling and buying stock.
- Researchers are recommended to use combined models such as a combined support vector regression model with a genetic algorithm and other combined models taken from machine learning algorithms. Moreover, for further research, it is recommended to investigate different types of LSTM models, such as stacked LSTMs, encoder–decoder LSTMs, bidirectional LSTMs, CNN LSTMs, and generative LSTMs models in prediction of stock price. Finally, due to the use of time series data in this study, the role of other influential factors on stock price prediction were not investigated; therefore, researchers are recommended to consider the role of other factors in future studies and compare the results obtained with the results of this study.

ORCID

Gholamreza Mansourfar https://orcid.org/0000-0002-3076-0241

REFERENCES

Anon. (1998). Glossary of terms. *Machine Learning*, *30*(2–3), 271–274.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques—Part II: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932–5941.

Aydin, A. D., & Cavdar, S. C. (2015). Comparison of prediction performances of artificial neural network (ANN) and vector autoregressive (VAR) models by using the macroeconomic variables of gold prices, Borsa Istanbul (BIST) 100 index and US dollar–Turkish lira (USD/TRY) exchange rates. *Procedia Economics and Finance*, *30*, 3–14.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, *12*(7), e0180944. https://doi.org/10.1371/journal.pone.018094

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Booth, A., Gerding, E., & McGroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*, *41*(8), 3651–3661.

Brax, C. (2000). Recurrent neural networks for time-series prediction. MSc dissertation, University of Skövde, Skövde, Sweden.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth.

Brownlee, J. (2017). *Long Short-Term Memory Networks with Python Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery, EBook: Brownlee.

Collobert, R. (2011). Deep learning for efficient discriminative parsing. In G. Gordon, D. Dunson, & M. Dudík (Eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (pp. 224–232). Fort Lauderdale, FL, USA, vol. 15. PMLR: JMLR W&CP.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 160–167). New York: ACM.

Das, S. P., & Padhy, S. (2012). Support vector machines for prediction of futures prices in Indian stock market. *International Journal of Computer Applications*, *41*(3), 22–26.

Fisher, I. E., Garnsey, M. R., & Hughes, M. E. (2016). Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, *23*(3), 157–214.

Galeshchuk, S., & Mukherjee, S. (2017). Deep networks for predicting direction of change in foreign exchange rates. *Intelligent Systems in Accounting, Finance and Management*, *24*(4), 100–110.

Gao, Q. (2016). Stock market forecasting using recurrent neural network. Doctoral dissertation, University of Missouri, Columbia, MO.

Gers, F. A., Schmidhuber, J., & Cummins, F. A. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*, 2451–2471.

Giovanis, E. (2009). Application of ARCH–GARCH models and feed-forward neural networks with Bayesian regularization in capital asset pricing model: the case of two stocks in Athens Exchange stock market. https://doi.org/10.2139/ssrn.1325842

Gomes, L. (2014). Machine-learning maestro Michael Jordan on the delusions of big data and other huge engineering efforts. IEEE Spectrum, October 20. https://spectrum.ieee.org/robotics/artificial-intelligence/machinelearning-maestro-michael-jordan-on-the-delusions-of-big-data-and-other-huge-engineering-efforts

Guo, Z., Wang, H., Liu, Q., & Yang, J. (2014). A feature fusion based forecasting model for financial time series. *PLoS ONE*, *9*(6), e101113. https://doi.org/10.1371/journal.pone.0101113

Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Amsterdam: Elsevier.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (pp. 485–585). New York, NY: Springer.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall International: Englewood Cliffs, NJ.

Hill, T., Marquez, L., O'Connor, M., & Remus, W. (1994). Artificial neural network models for forecasting and decision making. *International Journal of Forecasting*, *10*(1), 5–15.

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, *132*, 1351–1362.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*, *32*(10), 2513–2522.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, *10*(3), 215–236.

Kalyvas, E. (2001). Using neural networks and genetic algorithms to predict stock market returns. MSc thesis, University of Manchester.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. *Expert Systems with Applications*, *38*(5), 5311–5319.

Kumar, M., & Thenmozhi, M. (2005). Forecasting stock index movement: a comparison of support vector machines and random forest. In *Proceedings of Ninth Indian Institute of Capital Markets Conference*, Mumbai, India.

Lawrence, R. (1997). Using neural networks to forecast stock market prices. University of Manitoba. https://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, M. C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904.

Lu, C. J., Lee, T. S., & Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115–125.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations* (pp. 3–44). Cambridge, MA: MIT Press.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.) (2013). *Machine Learning: An Artificial Intelligence Approach*. Berlin: Springer.

Moshiri, S., & Cameron, N. (2000). Neural network versus econometric models in forecasting inflation. *Journal of Forecasting*, *19*(3), 201–217.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Parot, A., Michell, K., & Kristjanpoller, W. D. (2019). Using artificial neural networks to forecast exchange rate, including VAR–VECM residual analysis and prediction linear combination. *Intelligent Systems in Accounting, Finance and Management*, *26*(1), 3–15.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In S. Dasgupta, & D. McAllester (Eds.), *International Conference on Machine Learning* (Vol. 28) (pp. III-1310–III-1318). Miami, Florida, USA: PMLR.

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, *97*, 205–227.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction* (pp. 149–182). Berlin: Springer.

Rüping, S. (2001). SVM kernels for time series analysis. Technical Report 2001,43, Universität Dortmund, Sonderforschungsbereich 475. Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, Dortmund.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schwartz, R. A., & Whitcomb, D. K. (1977). Evidence on the presence and causes of serial correlation in market model residuals. *Journal of Financial and Quantitative Analysis*, *12*(2), 291–313.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, T., & Vasarhelyi, M. A. (2018). Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management*, *25*(4), 174–189.

Tan, C. (2009). Financial time series forecasting using improved wavelet neural network. MSc thesis, University of Copenhagen, Copenhagen.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309–317.

Thomaidis, N. S. (2006). Efficient statistical analysis of financial time-series using neural networks and GARCH models. https://ssrn.com/abstract=957887

Tian, G. (2015). Research on stock price prediction based on optimal wavelet packet transformation and ARIMA–SVR mixed model. *Journal of Guizhou University of Finance and Economics*, *6*(6), 57–69.

Wang, J. J., Wang, J. Z., Zhang, Z. G., & Guo, S. P. (2012). Stock index forecasting based on a hybrid model. *Omega*, *40*(6), 758–766.

Wei, Z. (2012). A SVM approach in forecasting the moving direction of Chinese stock indices. MSc thesis, Lehigh University, Bethlehem, PA.

Yim, J. (2002). A comparison of neural networks with time series models for forecasting returns on a stock market index. In T. Hendtlass, & M. Ali (Eds.), *Developments in Applied Artificial Intelligence: 15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE 2002 Cairns, Australia, June 17–20, 2002 Proceedings* (pp. 25–35). Berlin: Springer.

Zhang, Z. Y., Shi, C., Zhang, S. L., & Shi, Z. Z. (2006). Stock time series forecasting using support vector machines employing analyst recommendations. In J. Wang, Z. Yi, J. M. Zurada, B. L. Lu, & H. Yin (Eds.), Lecture Notes in Computer Science*Advances in Neural Networks—ISNN 2006* (Vol. 3973) (pp. 452–457). Berlin: Springer.

How to cite this article: Nikou M, Mansourfar G, Bagherzadeh J. Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intell Sys Acc Fin Mgmt*. 2019;1–11. https://doi.org/10.1002/isaf.1459