Petroleum
Business
Review

Machine Learning Application in Stock Price Prediction: Applied to the Active Firms in Oil and Gas Industry in Tehran Stock Exchange

Ali Mohammad Ghanbari[a*] and Hamid Jamshidi[b]

[a] *Assistant Professor, Accounting and Finance Department, Petroleum Faculty of Tehran, Petroleum University of Technology, Tehran, Iran; Email: aganbari@put.ac.ir*

[b] *M.S. Student in Finance, Accounting Department, Petroleum Faculty of Tehran, Petroleum University of Technology, Tehran, Iran; Email: jamshidi@put.ac.ir*

ARTICLE INFO

Keywords:

Stock Prediction

Machine Learning

Oil and Gas Industry

Received: April 02, 2019
Revised: May 16, 2019
Accepted: June 11, 2019

ABSTRACT

Stock price prediction is one of the crucial concepts in finance area. Machine learning can provide the opportunity for traders and investors to predict stock prices more accurately. In this paper, closing price is the dependent variable and first price, last price, opening price, oœdyy's hggh, oœd22's oow, volume, total index of Tehran Stock Exchange, Brent index, WTI index, and exchange rate are the independent variables. Seven different machine learning algorithms, including Bayesian linear, boosted tree, decision forest, neural networks, support vector, and ensemble regression are implemented to predict stock prices. The sample of the study is fifteen oil and gas companies active in the Tehran Stock Exchange. For each stock, the data were gathered from September 23, 2017 to September 23, 2019. Two metrics were employed for the performance of each algorithm: root mean square error and mean absolute error. By comparing the aforementioned metrics, the Bayesian linear regression had the best performance to predict stock price in the oil and gas industry on the Tehran Stock Exchange.

1. Introduction

The most significant information on the stock market for investors is stock price. Stock prices are fundamentally dynamic, nonlinear, and incendiary, indicating that investors should use a time series that is unstable and chaotic. Stock price dispersion is affected by macroeconomic elements such as political events, corporate policies, economic conditions, interest, inflation rates, investor expectations, traditional investors, choice, and physical and psychological factors of investors. Therefore, oœdyy's stock price predicting is not only very challenging but also of great interest to investors. Researchers have proposed different methods for predicting the stock and the securities market, and they can be divided into two classes: the statistical methods and the intelligent methods. Since the statistical methods and linear models do not consider all dimensions of stock prices and nonlinearity, researchers are encouraged to apply the intelligent methods. Machine learning and data mining methods can be applied to business intelligence (BI) systems so as to aid users for decision-making in plenty of real-life situations. One of the fascinating applications of BI is stock price prediction since it has the ability to consider more dimensions, nonlinearity, and behavior of prices in

* Corresponding Author

stock price forecasting. Making predictions about stock price is a hard problem. Algorithms which can forecast stock prices more accurately provides professionals who have access to stock prices with a significant financial incentive. One of this benefit is to eliminate wrong investments that will fail, decreasing the chance of major crisis and market disruptions. Another advantage is that a successful algorithm can be adapted to other domains with similar problem conditions (Moukalled, El-Hajj, & Jaber, 2019), (Langley, 2011), (Dua & Du, 2016), (Bhardwaj & Ansari, 2019), (Bontempi, Taieb, & Le Borgne, 2012). The main contribution of the current paper is the proposed machine learning algorithm which can predict stock prices in the oil and gas industry on Tehran Stock Exchange more accurately compared to other machine learning algorithms. The remainder of this paper is organized as follows. Literature review is provided in Section 2, and Section 3 is dedicated to methodology. Experimental results in Section 4 confirm the effectiveness, and especially the accuracy, of our developed algorithm. Finally, conclusions are presented in Section 5.

2. Literature Review

Kimoto et al. (1990) analyzed the application of feed-forward neural networks in stock price prediction back in 1990. The inputs to their prediction model included some macroeconomic factors such as foreign exchange and interest rate as well as technical indicators. They tested the proposed model for generating buying and selling signals of the TOPIX index for a 33 months horizon, from January 1987 to September 1989. Their results presented that the neural network algorithm was able to gain more profit over the buy-and-hold strategy (Kimoto, Asakawa, Yoda, & Takeoka, 1990).

Japanese researchers have found that the support vector performs better than other methods when comparing different machine learning algorithms using the NIKKEI 225 weekly stock market data. It may be argued that the most complete research in the field of stock price prediction using machine learning algorithms was carried out by Chen et al. (2007) entitled “Survey on oodkk aa rk Prddnnnnm ee hhods”. This research combines data and results from 100 scientific papers conducted in more than 50 countries to use neural networks and other algorithms for predicting stock markets and comparing different algorithms. The main result of this work emphasizes that support vector algorithms perform better than other machine learning algorithms (Chen, Yang, & Abraham, 2007).

Bekiros et al. (2007) compared recurrent neural networks (RNN) model and adaptive neuro-fuzzy inference system (ANFIS) model for forecasting the next day trend of NIKKEI and national association of securities dealers automated quotations (NASDAQ) indices. In both algorithms, past closing price was used for forccciii ng hh nxx dyy’s oos.. g pr.... To vvodd data snooping, they used the data from 1998 to 2002 for testing and the data from 1971 to 1998 for training. Their results show that the rate of return of the ANFIS model was more than that of the RNN, as well as the buy-and-hold strategy for both indices (Bekiros & Georgoutsos, 2007), (Patel, Shah, Thakkar, & Kotecha, 2015a).

Abbasi et al. (2008) analyzed the Iran Khodro Corporooon’s sodkk pr prrrrrr on Tehran Stock Exchange by applying an adaptive neuro-fuzzy inference system. Their results suggested that the pattern and behavior of the stock price could be predict with a low level of error (Abbasi & Abouec, 2008).

Jandaghi et al. (2010) used fuzzy neural networks and autoregressive integrated moving average (ARIMA) to predict SAIPA auto-mkknng oompnny’s sodkk preee. Their results confirmed the preference of nonlinear fuzzy-neural networks to the classic linear model and verified the capabilities of the fuzzy-neural networks to predict stock price (Jandaghi, Tehrani, Hosseinpour, Gholipour, & Shadkam, 2010).

Hadavandi (2010) et al. presented an elite system based on fuzzy genetic systems and artificial neural networks that predicted stock prices. In their model, open price, closing price, highest daily price, and lowest daily price were considered as the independent variables, and prediction of the next dyy’s last price was regarded as the dependent variable of the model. They used 50 selected stocks from Tehran Stock Exchange. Their results showed that this approach worked better than previous methods (Hadavandi, Shavandi, & Ghanbari, 2010).

Patel et al. (2015) also reported on using a combination of different machine learning algorithms to elevate the prediction performance. In the proposed two-stage algorithm, support vector regression (SVR) was first employed to forecast the value of technical indicators in *n* days. Random forest (RF), ANN and SVR were used in the second stage for forecasting closing price in *n* days using the technical indicators predicted from the first stage. Their results suggested that this two-stage combination model should be able to gain superior performance to the single-stage algorithm (Patel, Shah, Thakkar, & Kotecha, 2015b).

Yu et al. (2016) developed a new sigmoid-based mixed discrete-continuous differential evolution algorithm for stock performance prediction and ranking using the fundamental and technical data of the stock. 483 stocks listed in Shanghai A share market from Q1 2005 to Q4 2012 were used for developing and testing the algorithm. Their results revealed that the proposed algorithm could make portfolios which significantly outperformed the benchmark (Yu, Hu, & Tang, 2016).

Bohn (2017) combined sentiment analysis, fundamental analysis, and technical analysis and compared a set of machine learning algorithms for long-term stock prediction. He used about 1500 stocks which appeared in the S&P 500 between 2002 and 2016 for 23 experiments. Regression models were built, and ranks were induced based on the algorithm predictions for each week of testing and validation. He evaluated the model performance using the Spearman rank correlation coefficient between the actual rank and predicted rank. The results showed that the neural network algorithm combined with iterative feature selection could match the performance of a model developed with human expertise from an investment firm (Bohn, 2017).

Chong et al. (2017) analyzed deep neural network algorithm for stock price prediction. He assumed that a properly tuned deep neural network algorithm was able to extract features from a large set of raw data without relying on the past knowledge of predictors to predict stock price movement with a reasonable degree of accuracy. For the raw input data, 380-dimensional lagged stock price (38 stocks and 10 lagged prices) were considered. Three unsupervised algorithms were tested for feature selection: restricted Boltzmann machine, principal component analysis (PCA), and autoencoder. As the research aimed to test deep neural network algorithm for high frequency trading, the time interval between each observation of the stock price data was only five minutes. The deep neural network algorithm was trained to predict stock price movement five minutes ahead. Mean absolute error (MAE), normalized mean square error (NMSE), and standard root mean square error (RMSE) were employed for the evaluation of the performance of the models. The results showed that the deep neural network algorithm achieved performance similar to a simple linear autoregressive model (Chong, Han, & Park, 2017).

Ghasemiyeh et al. (2017) predicted prices on Tehran Stock Exchange using metaheuristic algorithms which consist of improved cuckoo search genetic algorithm (GA), particle swarm optimization (PSO), improved cuckoo search, cuckoo search, and hybrid artificial neural networks. Twenty eight important variables of value-added knowledge related to stock indices were determined as the inputs to this network, and then the actual values were gained. The results of the proposed algorithm showed that particle swarm optimization had a superior performance in predicting stock price compared to the other algorithms (Ghasemiyeh, Moghdani, & Sana, 2017).

To predict stock prices, Garakani et al. (2018) trained an ANN with post-propagation algorithm, independent component analysis (ICA), frog leaping, genetic algorithm, and particle swarm optimization algorithm by using daily price of 14 stocks selected from Tehran Stock Exchange as well as daily index data. This study aimed to specify the best evolutionary algorithm used in stock prediction algorithms. The ANN algorithm yielded superior performance (Garakani & Branch, 2018).

Vatanparast et al. (2019) presented an Levenberg-Marquardt back propagation (LMBP) neural network based on time series with respect to the open price, the highest price, the lowest price, the package price, and the volume of transactions. In the study, 315 days of stock prices were chosen to create 10 samples, and the test set included stock prices from day 316 to day 320 and used the LM-BP neural network. The results showed that stock price prediction based on the LM-BP neural network and over-point estimation by counting the intervals resulted in better results than the existing methods (Vatanparast & Mohammadi, 2019).

Bhardwaj et al. (2019) inspected advancements in economic market predictions. By looking at different predictive models using 100 stocks selected from New York Stock Exchange, they discovered that logistic-regression had the capacity to predict and analyze market movement direction more precisely than the other existing methods.

Different models, such as random forest and ARIMA have additionally turned out to be well known in stock market prediction. Random forest demonstrated its fruitful application in classification work, and ARIMA in time series prediction and financial related applications. K-nearest neighbors (K-NN) model is also applied to the experiments and shows some good results in predicting stock market directions (Bhardwaj & Ansari, 2019).

As a result, the questions of the study are as follows:

Q1. Which algorithm has better performance compared to the others?

Q2. Does the ensemble learning algorithm override a single algorithm?

These questions will be answered using the results in the conclusion section.

3. Methodology

The stock price prediction was performed using Bayesian linear, support vector, neural network, decision forest, boosted decision tree regression, and ensemble learning using Microsoft Azure ML Software. First, these algorithms are described in detail, and then two statistical metrics, namely root mean square error and mean absolute error, will be explained. After that, we will concentrate on the sampling and specify variables which are incorporated in the algorithms.

3.1. Algorithms

a. Bayesian linear regression

In statistics, the Bayesian linear regression is a linear regression approach in which statistical analysis is performed within the Bayesian inference framework. When the errors of the linear regression model follow a normal distribution, considering a prior distribution on the model parameters, it uses a posteriori distribution derived from Bayes' law. Consider a standard linear regression problem, in which for i = 1, 2, ,, n we determine the mean of the conditional distribution of y_I for a k × 1 predictor vector, i.e. x_i^T using Eq. (1):

$$y_i = x_i^T.\beta + \varepsilon_i \quad (1)$$

where ss a k × 1 vector, and ε_i is an independent and identically normally distributed random variable defined as:

$$\varepsilon_i \sim N(0, \partial^2) \quad (2)$$

This corresponds to the following likelihood function:

$$\tau(y|x, \sigma^2) \sim (\sigma^2)^{\frac{-n}{2}} \exp\left(\frac{-1}{2\sigma^2}(y - x\beta)^T (y - x\beta)\right) \quad (3)$$

The ordinary least squares solution is used to estimate the coefficient vector using the Moore–Penrose pseudo-inverse:

$$\hat{\beta} = (x^t.x)^{-1}x^t y \quad (4)$$

where x is the n × k design matrix, each row of which is a predictor vector x_i^T and y is the n-vector column $[y_1 \; y_n]^T$.

The least squares solution uses a frequentist approach where beta values are determined only by the available data. In the Bayesian inference method, the data with additional information are investigated in the form of a prior probability distribution, and the latter is used to predict the model using Bayes rule, prior distributions, and exponential function (Castillo, Schmidt-Hieber, & Van der Vaart, 2015).

b. Neural network regression

One of the most common algorithms that use a perceptron as a base is neural networks, sometimes known as a multilayer perceptron. By combining perceptron multilayers, the algorithm can create more complex classes and decision boundaries that are nonlinear. As more layers of a perceptron are added, the layers and boundaries that separate them become more complex.

This algorithm is trained in a way similar to a perceptron, but since there are multiple perceptrons in different layers, weights are updated from the last step (output nodes); also, this update is performed using a mathematical function known as the backpropagation function (Liu et al., 2017).

c. Decision forest regression

Decision trees are nonparametric algorithms which perform a sequence of simple tests for each sample; they traverse a binary tree data structure until a leaf node is reached.

This algorithm has the following features:

- It is efficient in memory usage and computation during prediction and training.
- It shows nonlinear decision margins.
- It performs integrated classification and feature selection and is robust in the presence of noisy features.

This regression algorithm includes an ensemble of decision trees. Each tree in a regression decision forest outputs a Gaussian distribution as a prediction. Aggregation is conducted over the ensemble of trees to find a Gaussian distribution nearest to the fusion distribution for all the trees in the algorithm (Rokach, 2016).

d. Support vector regression

Support vector machine algorithm is a type of learning system, which is also used for classification and estimating the data fitting function in the regression problems such that the lowest error in the grouping data or fitting function. The algorithm is based on statistical learning theory, which uses the principle of minimizing the structural error and leads to an overall optimal solution. The goal of support vector regression of the support vector machine algorithm is to recognize the function *f*(*x*) for the training patterns *x* so as to determine the maximum margin from the training values of *y*. In other words, SVR is an algorithm that fits a curve with a thickness of epsilon to the data so that the least error in the test data is achieved.

The primary SVR problem can be defined as follows:

$$\begin{aligned} &\text{Minimize } \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \hat{\xi}_i) \\ &\text{Subject to } \mathrm{w}.x_i + b - y_i \le \epsilon + \xi_i \\ &\qquad y_i - \mathrm{w}.x_i - b \le \epsilon + \hat{\xi}_i \\ &\qquad \xi_i, \hat{\xi}_i > 0 \end{aligned} \tag{5}$$

where *w* is a dimensional weight vector. Constant *C* which is higher than zero specifies the trade-off between the differences in decision function, where the upper limit of deviation that is more hhnn can still be tolerated.

In dual formulations, the optimization problem of SVR is represented by:

$$\begin{aligned} &\max\Big[\tfrac{-1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \\ &\hat{\alpha}_j)k(x_i, x_j) + \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)y_i - \\ &\varepsilon\sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)\Big] \\ &\text{Subject to } [\sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i) = 0,\ 0 \le \alpha_i \le \\ &C, 0 \le \hat{\alpha}_i \le C] \end{aligned} \tag{6}$$

where $k(x_i, x_j)$ denotes the kernel function, which is defined as $k(x_i, x_j) = \varphi(x_i).\varphi(x_j)$, nn whhhh φ a mapping from the data space to the feature space. Using the Lagrange multiplier and optimal conditions, the regression function can be explicitly formulated as follows (Awad & Khanna, 2015):

$$y(x) = \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)k(x_i, x) + b \tag{7}$$

e. Boosted decision tree

Boosting is one of several classic algorithms for creating ensemble algorithms, along with random forests and bagging. Boosted decision trees apply an efficient implementation of the multiple additive regression trees (MART) gradient boosting technique which is for regression problems. It builds each regression tree in a step-wise fashion while recruiting a predetermined loss function to evaluate the error in each step and correct it for the next one. Thus, the prediction model is actually an ensemble of weaker prediction models (Si et al., 2017).

f. Ensemble learning

Ensemble learning is a machine learning paradigm where multiple models (often called "waak rrrr nrrs)) rr trained to solve the same problem and are combined to achieve better results.

- Bagging, which often considers homogeneous weak learners, trains them independently in parallel and combines them following some type of deterministic averaging process.
- Boosting, that often considers homogeneous weak learners, trains them sequentially in a very adaptive way, in which a base model depends on the previous ones, and combines them following a deterministic strategy (Krawczyk, Minku, Gama, Stefanowski, & Woźnkkķ 2017..

3.2. Statistical metrics

For evaluating each algorithm, the resultant root mean square error and mean absolute error of them are compared. In the following paragraphs, these two metrics are described:

a. Root mean square error

Root mean square error is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far data points are from the regression line, and RMSE is a measure of how these residuals are spread out. In other words, it tells how the data is concentrated around the line of the best fit. Root mean square error is commonly used in climatology, predicting, and regression analysis to verify experimental results.

The root mean square error is expressed in:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

where *yi* is the observed values (known results), *yi* represents the predicts (expected values or unknown results), and *n* stands for the number of observations.

The bar above the squared differences is the mean (similar to $\bar{x}$).

When standardized observations and predicts are used as the inputs to RMSE, there is a relationship between the correlation coefficient. For example, if the correlation

coefficient is one, the RMSE will be zero, because all the points lie on the regression line.

b. Mean absolute error

MAE measures the average magnitude of the errors in a set of predictions without considering their direction. It is the average of the absolute differences between the prediction and the actual observation where all individual differences have an equal weight.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (9)$$

where y_i is the observed values (known results), $\hat{y}_i$ represents the predicts (expected values or unknown results), and *n* stands for the number of observations (Veretelnikova & Elantseva, 2016).

The population of the current study is 45 oil and gas companies listed on the Iranian Stock Market. The survey of the companies showed that 23 companies are listed in over-the-counter, and 22 companies are listed on the stock exchange. Because the over-the-counter market and stock exchange market have different regulations, variables affect them differently; thus, to make it possible to compare the results of the algorithms, we concentrate on the stock exchange market. Due to the inactivity of some companies, the deficient data on some others (Khorasan, Noori, and Pars Petrochemicals), and different value chains, finally, the following 15 firms which are active in the oil and gas industry are selected:

- Shazand Petrochemical (Shazand)
- Jam Petrochemical (Jam)
- Khark Petrochemical (Shekhark)
- Pardis Petrochemical (Shapdis)
- Fanavaran Petrochemical (Shefan)
- Mobin Petrochemical (Mobin)
- Kermanshah Petrochemical (Kermasha)
- Shiraz Petrochemical (Shiraz)
- Parsian Expansion of Oil and Gas (Parsan)
- Tabriz Oil Refinery (Shabriz)
- Khalij Fars Petrochemical (Fars)
- Bandar Abbas Oil Refinery (Shebandar)
- Isfahan Oil Refinery (Shapna)
- Tehran Oil Refinery (Shatran)
- Iranians Group Petrochemical (Petrol)

To implement machine learning algorithms, this paper used the stock price data of the sample set for two years from September 23, 2017 to September 23, 2019.

For each of the stocks listed above, the following data were added to the dataset; it is worth mentioning that to avoid over and under fitting and the curse of dimensionality, we implement principal component analysis (PCA) method to select appropriate variables which have more capacity to predict stock prices and can explain dependent variables more than others (Ajayi & oo ugou,, 1996 Aprr gss & rrrrrr r 2009 Byyzz, Tekiner, Zeng, & Keane, 2018; Cutler, Poterba, & Summers, 1988; Fama, 1965, 1995; Garakani & Branch, 2018; Hamao, Masulis, & Ng, 1990; Hui, 2019; Jiang, 2019; Lo & MacKinlay, 1988; McQueen & Roley, 1993; Warner, Watts, & Wruck, 1988). The algorithms are trained on the closing price; the training dataset included 80% of the data, and the remaining 20% was utilized to test the dataset (Bhardwaj & Ansari, 2019). The following variables are normalized and then applied to the algorithms:

- First price
- Last price
- Opening price
- Todyy's hggh
- Todyy's oow
- Volume
- Closing price

The algorithms employed the Tehran Stock Exchange index as a measure of the overall state of Iran's economy. Due to the dependence of Iran's economy, and consequently firms with an oil and gas nature, on oil prices, the WTI index and Brent index were considered. The exchange rate is really crucial to petrochemicals and refineries because it can influence the import and export of their commodities (Abbasi & Abouec, 2008; Deshpande, 2017; Fama, 1965; Garakani & Branch, 2018; Ghasemiyeh et al., 2017; Hadavandi et al., 2010; Lo & MacKinlay, 1988; McQueen & Roley, 1993; Moukalled et al., 2019; Warner et al., 1988). Furthermore, for the total index of Tehran Stock Exchange, the Brent index, WTI index, and exchange rate, the daily data have been introduced into the algorithms.

- Total index of Tehran Stock Exchange
- Brent index
- WTI index
- Exchange rate (Monfared & Aknn, 2017)

3.3. Data Analysis

The score module was used to view the prediction results of the algorithms. The Microsoft Azure ML software makes it easy for the researcher to compare the results of the algorithms with each other by providing the Evaluate Module. To determine the best algorithm, we considered RMSE and MAE metrics, and the selected algorithm had the lowest RMSE and MAE. By

implementing Bayesian linear regression, boosted decision tree regression, decision forest regression, neural network regression, support vector machine regression, ensemble learning (boosted tree) regression, ensemble learning (bagged tree) regression algorithms on all 15 stocks and determining the best algorithm for each stock, finally, the algorithm that had the highest frequency in terms of samples was introduced as the best algorithm.

4. Results

We applied 7 algorithms to each of the 15 stocks and then compared the obtained results. Tables 1–15 list the obtained results, and Tables 16 and 17 summarize the results.

Table 1. Shazand Petrochemical.

Shazand Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	17.608208	1	25.67419	1
Boosted decision tree	33.8274	3	55.459207	3
Decision forest	31.812155	2	45.308166	2
Neural network	1419.61	7	1704.239	7
Support vector machine	63.959	5	79.583	5
Ensemble (boosted)	195.41	6	217.51	6
Ensemble (bagged)	35.732	4	62.961	4

Table 2. Jam Petrochemical.

Jam Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	68.707208	1	103.117209	1
Boosted decision tree	83.650514	3	119.107878	2
Decision forest regression	91.914502	4	137.972813	4
Neural network regression	7426.602	7	7978.57	7
Support vector machine	107.26	5	275.06	5
Ensemble (boosted)	518.98	6	551.48	6
Ensemble (bagged)	79.952	2	128.14	3

Table 3. Khark Petrochemical.

Khark Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	147.16644	1	233.360042	1
Boosted decision tree	228.77528	2	401.808666	2
Decision forest regression	350.81204	4	609.28896	3
Neural network regression	9828.4740	7	11926.29	7
Support vector machine	572.29	5	675.25	5
Ensemble (boosted)	1474.6	6	1677.5	6
Ensemble (bagged)	316.42	3	636.87	4

Table 4. Pardis Petrochemical.

Pardis Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	75.160747	1	103.8645	1
Boosted decision tree	115.693673	3	201.692498	3
Decision forest regression	135.758509	4	236.679212	4
Neural network regression	4656.374	7	5352.4	7
Support vector machine	351.63	5	388.03	5

Pardis Petrochemical	MAE	Rank	RMSE	Rank
Ensemble (boosted)	531.31	6	598.04	6
Ensemble (bagged)	114.68	2	181.69	2

Table 5. Fanavaran Petrochemical.

Fanavaran Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	287.982545	1	397.501269	1
Boosted decision tree	404.319737	2	631.817912	4
Decision forest regression	414.621638	4	627.937977	3
Neural network regression	12572.57	7	14800.58	7
Support vector machine	807.42	5	978.06	5
Ensemble (boosted)	1888.1	6	2084.03	6
Ensemble (bagged)	413.33	3	599	2

Table 6. Shiraz Petrochemical.

Shiraz Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	34.793945	1	46.383802	1
Boosted decision tree	53.498294	2	91.704558	3
Decision forest regression	64.08798	4	108.185954	4
Neural network regression	2165.473	7	2584.61	7
Support vector machine	127.46	5	147.77	5
Ensemble (boosted)	218.52	6	257.96	6
Ensemble (bagged)	56.949	3	86.895	2

Table 7. Kermanshah Petrochemical.

Kermanshah Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	37.896904	1	52.588578	1
Boosted decision tree	60.228657	4	99.085645	3
Decision forest regression	56.158662	2	83.605048	2
Neural network regression	2000.676665	7	2303.2041	7
Support vector machine	105.6	5	126.38	5
Ensemble (boosted)	234.04	6	271.47	6
Ensemble (bagged)	58.622	3	109.2	4

Table 8. Mobin Petrochemical.

Mobin Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	29.247705	1	45.679307	1
Boosted decision tree	54.121579	2	94.426606	2
Decision forest regression	58.729658	4	111.126342	4
Neural network regression	2899.61	7	3498.24	7
Support vector machine	102.04	5	124.81	5
Ensemble (boosted)	286.5	6	318.08	6
Ensemble (bagged)	55.457	3	106.29	3

Table 9. Tabriz Oil Refinery.

Tabriz Oil Refinery	MAE	Rank	RMSE	Rank
Bayesian linear regression	97.94808	1	169.3511	1
Boosted decision tree	170.5052	2	298.1761	2
Decision forest regression	203.7205	4	355.1779	4
Neural network regression	6198.17	7	7293.25	7
Support vector machine	471.55	5	533.41	5
Ensemble (boosted)	625.71	6	743.35	6
Ensemble (bagged)	175.78	3	306.84	3

Table 10. Parsian Expansion of Oil and Gas.

Parsian Expansion of Oil and Gas	MAE	Rank	RMSE	Rank
Bayesian linear regression	15.371004	1	21.706794	1
Boosted decision tree	25.224086	2	40.238857	2
Decision forest regression	29.462171	4	41.111174	3
Neural network regression	1244.55	7	1491.134	7
Support vector machine	78.33	5	90.29	5
Ensemble (boosted)	153.45	6	167.24	6
Ensemble (bagged)	29.046	3	45.337	4

Table 11. Khalij Fars Petrochemical.

Khalij Fars Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	27.33869	1	43.681532	2
Boosted decision tree	43.157556	3	84.581747	4
Decision forest regression	43.873312	5	83.701845	3
Neural network regression	824.9552	7	1179.256	7
Support vector machine	30.343	2	40.342	1
Ensemble (boosted)	235.86	6	257.35	6
Ensemble (bagged)	43.349	4	101.62	5

Table 12. Bandar Abbas Oil Refinery.

Bandar Abbas Oil Refinery	MAE	Rank	RMSE	Rank
Bayesian linear regression	48.86023	1	88.149434	1
Boosted decision tree	74.93462	2	115.299635	2
Decision forest regression	90.97734	4	134.885884	4
Neural network regression	3075.539	7	3556.84	7
Support vector machine	222.26	5	257.46	5
Ensemble (boosted trees)	444.17	6	481.51	6
Ensemble (bagged trees)	81.351	3	120.11	3

Table 13. Esfahan Oil Refinery.

Esfahan Oil Refinery	MAE	Rank	RMSE	Rank
Bayesian linear regression	22.478648	1	31.734827	1
Boosted decision tree	51.57414	2	80.101396	2
Decision forest regression	53.38526	3	81.421921	3
Neural network regression	2932.163	7	3897.82	7
Support vector machine	173.49	5	207.08	5

Esfahan Oil Refinery	MAE	Rank	RMSE	Rank
Ensemble (boosted trees)	301.71	6	335.23	6
Ensemble (bagged trees)	61.304	4	100.72	4

Table 14. Tehran Oil Refinery.

Tehran Oil Refinery	MAE	Rank	RMSE	Rank
Bayesian linear regression	20.918141	1	30.802937	1
Boosted decision tree	43.799851	3	80.13327	5
Decision forest regression	44.000222	4	64.276922	3
Neural network regression	1338.79	7	1508.42	7
Support vector machine	49.514	5	61.253	2
Ensemble (boosted trees)	220.61	6	236.44	6
Ensemble (bagged trees)	40.353	2	66.54	4

Table 15. Iranians Group Petrochemical.

Iranians Group Petrochemical	MAE	Rank	RMSE	Rank
Bayesian linear regression	11.979611	3	16.139384	2
Boosted decision tree	11.794116	1	17.029451	4
Decision forest regression	11.797079	2	16.063623	1
Neural network regression	289.514944	7	363.69261	7
Support vector machine	13.418	4	16.803	3
Ensemble (boosted trees)	68.587	6	73.004	6
Ensemble (bagged trees)	13.859	5	19.031	5

Table 16. Comparing algorithms by means of RMSE.

Top three algorithms in terms of RMSE	Frequency	Rank
Bayesian linear regression	13	1
Boosted decision tree	7	2
Decision forest	6	3

Table 17. Comparing algorithms by means of MAE.

Top three algorithms in terms of MAE	Frequency	Rank
Bayesian linear regression	14	1
Boosted decision tree	8	2
Ensemble learning (bagged)	8	2

We used the data on 15 firms that are active in the oil and gas industry on the Tehran Stock Exchange from September 23, 2017 to September 23, 2019. By applying these data to seven different algorithms, that is, Bayesian linear regression, decision forest, boosted decision tree, neural network, support vector machine, and ensemble regression, and utilizing Microsoft Azure ML software and its reported metrics, namely the RMSE and the MAE, the Bayesian linear regression had a higher frequency than the others considering the lowest RMSE (13 out of 15) and the lowest MAE (14 out of 15).

5. Conclusions

As mentioned earlier, the goal of this paper was to determine the best algorithm that can predict the stock prices in oil and petrochemical companies on Tehran Stock Exchange. According to the results, the Bayesian linear regression algorithm outperformed the other methods. Now, we discuss the questions mentioned above.

Q1. Which algorithm has better performance compared to the others?

According to the results, Bayesian linear algorithm has the best performance in terms of the minimum RMSE (13 out of 15) and the minimum MAE (14 out of 15).

Q2. Does the ensemble learning algorithm override a single algorithm?

As demonstrated by the results, the ensemble learning algorithm does not necessarily outperform a single one. In some cases, the ensemble algorithm produced results better than a single algorithm; however, regarding all the implemented algorithms, a single algorithm (Bayesian linear regression) performed better than the others.

According to Tables 16 and 17, the RMSE of the Bayesian linear algorithm has the best performance, and in one case decision forest algorithm and support vector regression algorithm have better performance compared to the other algorithms; therefore, we can conclude that the Bayesian linear algorithm is suitable for identifying price behavior in oil and gas industry on Tehran Stock Exchange. It is worth noting that the neural networks algorithm, without any exception, has the weakest performance in identifying the past behavior of stock prices and predicting the future stock prices. Regarding the ensemble learning, the bagged algorithm has better performance than the boosted one. After the Bayesian linear algorithm, the boosted decision tree had the best performance in identifying the past behavior of stock prices and predicting the future prices compared to the other algorithms.

On the other hand, regarding the MAE, in 14 out of 15 algorithms, the Bayesian linear regression algorithm yielded the minimum MAE; therefore, considering this metric, the Bayesian linear algorithm has the best performance in specifying a behavioral pattern of stock prices and predicting the future prices. Again, in this case, the boosted ensemble algorithm has weaker performance than the bagged algorithm. The neural network algorithm has the maximum MAE and consequently resulted in the weakest prediction of the future stock prices. After the Bayesian linear algorithm, the boosted decision tree algorithm offered the minimum MAE; therefore, it has better performance compared to the other algorithms. Before implementing the algorithms, we expected that the boosted decision tree algorithm would lead to higher performance than the decision forest algorithm, but, in some cases, reverse findings were obtained.

In a nutshell, the Bayesian linear regression produced the best results (the minimum RMSE and the minimum MAE) compared to the other algorithms. Taking account of the above findings, it will be appropriate to use the Bayesian linear algorithm for identifying the past behavior of stock prices and predicting the future stock prices of the companies in oil and gas industry on Tehran Stock Exchange.

Moreover, the results of the present paper confirm that the SVR algorithm outperforms the neural networks algorithm on Tehran Stock Exchange; this finding is in agreement with the work of Huang and Nakamori which applied machine learning algorithms to 150 selected stocks from New York Stock Exchange (W. Huang, Nakamori, & Wang, 2005). Nevertheless, because we employed the Bayesian linear regression despite the above research, our results demonstrate that the Bayesian linear regression is more effective in predicting stock prices than support vector regression; similar results are also obtained elsewhere (B. Huang, Ding, Sun, & Li, 2018; Wright, 2008; Zuo & Kita, 2012) .

6. Research Suggestions

The following suggestions can be helpful for the scholars to research in the field of machine learning:

- Our results recommend using the Bayesian linear regression to predict stock price in the oil and gas industry on the Tehran Stock Exchange.
- Considering the significant researches which compare the support vector regression and the neural networks, the support vector regression should be selected rather than the neural networks to predict stock prices in the oil and gas industry on the Tehran Stock Exchange.
- Machine learning algorithms can be applied to over-the-counter (OTC) market so as to observe the behavior of the stocks of the oil and gas industry in this market.
- Utilizing classification category of machine learning algorithms to group stocks with different behaviors and levels of risk in the oil and gas industry on the Tehran Stock Exchange will be useful.

References

Abbasi, E., & Abouec, A. (2008). Stock price forecast by Using neuro-fuzzy Inference System. Paper Presented at The Proceedings of World Academy of Science, Engineering and Technology.

Ajyy R. A., & oo ugou,, .. (1996). nn the Dynamic Relation Between Stock Prices and Exchange Rates.

Journal of Financial Research, 19(2), 193–207.

Apergis, N., & Miller, S. M. (2009). Do structural oil-Market Shocks Affect Stock Prices? Energy Economics, 31(4), 569–575.

Awad, M., & Khanna, R. (2015). Support Vector regression. In Efficient Learning Machines (pp. 67–80): Springer.

Bekiros, S. D., & Georgoutsos, D. A. (2007). Evaluating direction-of-change forecasting: Neurofuzzy Models vs. Neural Networks. Mathematical and Computer Modelling, 46(1–2), 38–46.

Beyaz, E., Tekiner, F., Zeng, X.-j., & Keane, J. (2018). Comparing Technical and Fundamental indicators in Stock Price Forecasting. Paper Presented at the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS).

Bhardwaj, N., & Ansari, M. A. (2019). Prediction of Stock Market Using Machine Learning Algorithms.

Bohn, T. A. (2017). Improving long Term Stock Market prediction with Text Analysis.

Bontempi, G., Taieb, S. B., & Le Borgne, Y.-A. (2012). Machine learning strategies for Time Series Forecasting. Paper Presented at The European Business Intelligence Summer School.

Castillo, I., Schmidt-Hieber, J., & Van Der Vaart, A. (2015). Bayesian Linear Regression with Sparse Priors. The Annals of Statistics, 43(5), 1986–2018.

Chen, Y., Yang, B., & Abraham, A. (2007). Flexible Neural Trees Ensemble for Stock Index Modeling. Neurocomputing, 70(4–6), 697–703.

Chong, E., Han, C., & Park, F. C. (2017). Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, data Representations, and Case Studies. Expert Systems with Applications, 83, 187–205.

Cutler, D. M., Poterba, J. M., & Summers, L. H. (1988). What Moves Stock Prices? In: National Bureau of Economic Research Cambridge, Mass., USA.

Deshpande, R. (2017). Semi-Strong Form of Market Efficiency: Does all Critical Information Affect Stock Price Valuations? Indian Journal of Research in Capital Markets, 15.

Dua, S., & Du, X. (2016). Data Mining and Machine Learning in Cybersecurity: CRC Press.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. The Journal of Business, 38(1), 34–105.

Fama, E. F. (1995). Random Walks in Stock Market Prices. Financial Analysts Journal, 51(1), 75–80.

Garakani, A. R., & Branch, S. T. (2018). Stock Price Prediction Using Multilayer Perceptron Neural Network by Monitoring Frog Leaping Algorithm. Journal of Intelligent Computing Volume, 9(1), 15.

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.

Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010). Integration of Genetic Fuzzy Systems and Artificial Neural Networks for Stock Price Forecasting. Knowledge-Based Systems, 23(8), 800–808.

Hamao, Y., Masulis, R. W., & Ng, V. (1990). Correlations in Price Changes and Volatility Across International Stock Markets. The Review of Financial Studies, 3(2), 281–307.

Huang, B., Ding, Q., Sun, G., & Li, H. (2018). Stock Prediction Based on Bayesian-LSTM. Paper Presented at The Proceedings of the 2018 10th International Conference on Machine Learning and Computing.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting Stock Market Movement Direction with Support Vector Machine. Computers & Operations Research, 32(10), 2513–2522.

Hui, M. (2019). Empirical Analysis of The Impact of Macro Factors on Stock Prices.

Jandaghi, G., Tehrani, R., Hosseinpour, D., Gholipour, R., & Shadkam, S. A. S. (2010). Application of Fuzzy-neural Networks in Multi-ahead Forecast of Stock Price. African Journal of Business Management, 4(6), 903.

Jiang, Q. (2019). Comparison of Black–Scholes Model and Monte-Carlo Simulation on Stock Price Modeling. Paper Presented at the 2019 International Conference on Economic Management and Cultural Industry (ICEMCI 2019).

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock Market Prediction System with Modular Neural Networks. Paper Presented at the 1990 IJCNN International Joint Conference on

Neural Networks.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Wonnkkk, .. (2017). Ensmnb Laarnnng for aaaa Stream Analysis: A Survey. Information Fusion, 37, 132–156.

Langley, P. (2011). The Changing Science of Machine Learning. Machine Learning, 82(3), 275–279.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A Survey of Deep Neural Network Architectures and Their Applications. Neurocomputing, 234, 11–26.

Lo, A. W., & MacKinlay, A. C. (1988). Stock Market Prices do not Follow Random Walks: Evidence from a Simple Specification Test. The Review of Financial Studies, 1(1), 41–66.

McQueen, G., & Roley, V. V. (1993). Stock Prices, News, and Business Conditions. The Review of Financial Studies, 6(3), 683–707.

oo nfrr dd, ,,, & Aknn (2017). Th Roooooship Between Exchange Rates and Inflation: The Case of Iran. European Journal of Sustainable Development, 6(4), 329.

Moukalled, M., El-Hajj, W., & Jaber, M. (2019). Automated Stock Price Prediction Using Machine Learning. Paper Presented at The Proceedings of The Second Financial Narrative Processing Workshop (FNP 2019), September 30, Turku Finland.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015a). Predicting Stock and stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. Expert Systems with Applications, 42(1), 259–268.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015b). Predicting Stock Market Index Using Fusion of Machine Learning Techniques. Expert Systems with Applications, 42(4), 2162–2172.

Rokach, L. (2016). Decision Forest: Twenty Years of Research. Information Fusion, 27, 111–125.

Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., & Hsieh, C.-J. (2017). Gradient Boosted Decision Trees for High Dimensional Sparse output. Paper Presented at The Proceedings of the 34th International Conference on Machine Learning-Volume 70.

Vatanparast, M., & Mohammadi, S. (2019). Stock Price Prediction Based on LM-BP Neural network and Over-point Estimation by Counting Time Intervals: Evidence from the Stock Exchange.

Veretelnikova, E. L., & Elantseva, I. L. (2016). Selection of Factor for Root Mean Square Minimum Error Criterion. Paper Presented at the 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE).

Warner, J. B., Watts, R. L., & Wruck, K. H. (1988). Stock Prices and Top Management Changes. Journal of Financial Economics, 20, 461–492.

Wright, J. H. (2008). Bayesian Model Averaging and Exchange Rate forecasts. Journal of Econometrics, 146(2), 329–341.

Yu, L., Hu, L., & Tang, L. (2016). Stock Selection with a Novel Sigmoid-Based Mixed Discrete-Continuous Differential Evolution Algorithm. IEEE Transactions on Knowledge and Data Engineering, 28(7), 1891–1904.

Zuo, Y., & Kita, E. (2012). Stock Price Forecast Using Bayesian Network. Expert Systems with Applications, 39(8), 6729–6737.