International Journal of
Mathematical Modelling & Computations
Vol. 12, No. 04, Fall 2022, 275- 297
DOI:10.30495/ijm2c.2022.1948716.1241

Comparison of the Efficiency of Statistical Learning Algorithms and Artificial Neural Networks to Predict Stock Prices

A. Sadat Najafi[a] and S. Sardar[b,*]

[a]*Department of Information Technology Management, Tehran North Branch, Islamic Azad University, Tehran, Iran,*
[b]*Department of Industrial Management, Tehran North Branch, Islamic Azad University, Tehran, Iran.*

Abstract. The importance of the capital market in economic development is undeniable through the effective management of capital and the optimal allocation of resources. In this study, according to capital market behaviors and research, Statistical Learning (SL) algorithms compared to Artificial Neural Networks (ANN) to analyze time-series data and predict stock prices have been investigated. In studies to compare methods or provide hybrid models, most statistical learning algorithms are limited and examined without the comparison of other algorithms. In this study, to eliminate this shortcoming by implementing and comparing statistical learning algorithms in the two categories of Regression Learner and Classification Learner, the most efficient algorithm has been identified based on the selected shares and based on the presented parameters. The first category (Regression Learner) includes Linear Regression, Interaction Linear Regression, Robust Linear Regression, Stepwise Linear Regression, Fine Tree, Medium Tree, Coarse Tree, Linear Support Vector Machine (SVM), Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Ensemble Boosted Trees, Ensemble Bagged Trees, Squared Exponential Gaussian Process Regression, Matern 5/2 Gaussian Process Regression, Exponential Gaussian Process Regression, Rational Quadratic Gaussian Process Regression. The second category (Classification Learner) includes Gaussian, Naive Bayes, K-nearest neighbors. The results show that Regression Learner methods are more effective in predicting the price of selected stocks.

Received: 01 January 2022, Revised: 04 October 2022, Accepted: 24 November 2022.

Keywords: Statistical learning algorithm, Artificial neural network, Data analysis, Capital market, Prediction.

Index to information contained in this paper

1. Introduction

Investing in the stock market requires decision-making on issues such as selection, timing, price, and amount of buying and selling shares [23], and obtaining information about the future state of the stock market prices. Therefore, if the future trend of the stock market predicts with appropriate methods, the investor can maximize the return on his investment. In addition to investors and stock market participants, financial managers prefer to have a mechanism that can help them in their decision-making.

For this reason, attention has paid to forecasting methods, and many studies and researches have done in this area, and stock market prediction has always caught the attention of many analysts and researchers [25]. The financial time series models

[*] Corresponding author. Email: s_sardar@iau-tnb.ac.ir

http://ijm2c.ctb.iau.ir

expressed by financial theories have been the basis for forecasting a series of data in the twentieth century. However, these theories are not directly applicable to predict the market values which have an external impact. The development of the multi-layer concept allowed ANN to be chosen as a prediction tool besides other methods [11]. ANN and Support Vector Regression (SVR) as two widely used machine learning algorithms for predicting stock prices and stock index values [25]. New statistic-based models provide a wide area of prediction equipment for different science areas [12].

Based on research on the nature and capital market behaviors, statistical learning algorithms and artificial neural networks are used to analyze time series of stock price data. They used for forecasting, but to compare methods or composition of combined models, limited statistical learning algorithms have been studied without comparing and predicting the prediction error of other algorithms.

2. Basic concept

In this section, we will give an overview of the concepts of the methods used in this paper.

2.1 *Regression*

Regression is a set of techniques for estimating relationships that are used to model and analyze the variables in question concerning the relationship between the dependent variable (response variable) and one or more independent variables (predictor variable).

2.1.1 Linear regression

In linear regression, an attempt is made to determine the best linear equation between the two variables [29] .In the general case, the matrix form of the linear regression model as follows:

$$y_i = x_i^T\beta + \epsilon_i \tag{1}$$

where $i = 1,\dots,n$ and y_i are dependent variables, x_i are independent variables, β_0 and β_1 are model parameters or regression coefficients, ϵ_i are error statements, β calculated by the Ordinary Least Square (OLS) method by minimizing the equation (2).

$$\sum_{i=1}^{n}(y_i + x_i^T\beta)^2 \tag{2}$$

2.1.2 Interaction linear regression

If the effect of an independent variable on dependent variable changes depending on the values of one or more other independent variables. In such a case, the linear regression model is called the Interaction Linear Regression. This can be estimated with an interaction term using the following regression equation:

$$Y = b_1X + b_2Z + b_3XZ + b_0 \tag{3}$$

where b_0 is the intercept, $b_1$is the effect of Xon Y, b_2 is the effect of Z on Y, and b_3 is the effect of XZ on Y [22].

2.1.3 Robust regression

When the data are skewed or have compelling observations, Robust Regression is used [32]. By replacing the least squares criterion with a robust criterion, M-estimate of β is:

$$\hat{\beta} = arg \ \min_{\beta} \sum_{i=1}^{n} \rho(\frac{y_i - x_i^T \ \beta}{\hat{\sigma}}) \tag{4}$$

where $\rho(0)$ is a robust loss function and $\hat{\sigma}$ is an error scale estimate.

2.1.4 Stepwise regression

In the Stepwise algorithm, the variables are entered into the regression model in stages, from the most crucial to the least important [19]. Stepwise regression is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. These include Forward and Backward selection. Forward selection begins with no variables selected (the null model). In the first step, it adds the most significant variable. At each subsequent step, it adds the most significant variable of those not in the model, until there are no variables that meet the criterion set by the user. Backward selection begins with all the variables selected, and removes the least significant one at each step, until none meet the criterion. Stepwise selection alternates between forward and backward, bringing in and removing variables that meet the criteria for entry or removal, until a stable set of variables is attained [8].

2.1.5 Decision tree (DT)

A Decision tree is a binary tree where each internal node is labeled with a variable, and each leaf is tagged with 0 or 1. The depth of a decision tree is the length of the longest path from the root to a leaf [21].

DT can be defined as a procedure that partitions/splits the input data into smaller and smaller subsets recursively. The splitting process is based on a set of thresholds delineated in each of the internal nodes in the tree. Internal nodes split input data from the root node into sub nodes, and split sub-nodes into further sub-nodes. Through this sequentially binary subdivision, the input data are classified in which the end of nodes called leaf nodes (leaves) represent the ultimate target classes.

2.1.6 Ensemble boosted and bagged trees

Bagging and Boosting are two families of the most commonly used integrated or so-called ensemble decision tree, which are based on manipulating training samples. Bagging is a kind of regular ensemble classifier technique in which several predictors are made independently and combined using some model averaging methods such as weighted average or majority vote. In contrast, Boosting is an Ensemble Learning (EL) technique in which the models are not built independently but sequentially, and successive predictors are used in an attempt to correct the errors generated by previous predictors. Boosted DTs are also members of the EL family. As the models are built in such methods, they are adapted to minimize the errors of the previous trees [13].

2.1.7 Regression tree

Regression tree analysis is when the predicted outcome can be considered a real number, and Classification tree analysis is when the predicted outcome is the class (discrete) to which the data belongs. Building a regression tree process are two steps. In first step the predictor space (that is, the set of possible values for X_1, X_2,..., X_p) divide into J distinct and non-overlapping regions, R_1,R_2,...,R_j, and in second step for every observation that falls into the region R_j, will be made the same prediction, which is simply the mean of the response values for the training observations in R_j. After regions, R_1,R_2,...,R_j, have been created, we predict the response for a given test observation using the mean of the training observations in the region to which that test observation belongs [9].

In Coarse Tree we have few leaves to make coarse distinctions between classes (maximum number of splits is 4), In Medium Tree, we have medium number of leaves for finer distinctions between classes (maximum number of splits is 20) and in Fine Tree we have many leaves to make many fine distinctions between classes (maximum number of splits is 100).

2.1.8 Support vector machine

The Linear Support Vector Machine (LSVM) regression technique aims at finding a linear function $f(x)$ with the minimal norm value β that makes the function $f(x)$ to be as flat as possible using what is called primal formula and dual formula. The primal formula is briefly described using equations:

$$f(x) = \acute{x}\beta + b \tag{5}$$

$$J(\beta) = \frac{1}{2}\acute{\beta}\beta + C\sum_{n=1}^{N}(\varepsilon_n + \varepsilon_n^*) \tag{6}$$

while equations (7)-(10) illustrate the working principle for the dual formula used in linear SVM regression analysis [6].

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\,\acute{X}_{\imath}X_j + V \tag{7}$$

$$V = \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}y_i(\alpha_i^* - \alpha_i) \tag{8}$$

$$\beta = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)x_n \tag{9}$$

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\,(\acute{x}_n x) + b \tag{10}$$

2.1.9 Quadratic SVM

If the objective function is quadratic while the constraints are all linear, then the optimization problem is called a quadratic program. Standard quadratic programming should be used to ensure:

$$w(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\,\alpha_j y_i y_j x_i \times x_j \tag{11}$$

when the vector α^0 solution of the maximization problem is found in equation (11) [2].

2.1.10 Gaussian SVM

One of the most important design choices for SVM is the kernel function. The Gaussian function $G(x_i, x_j)$ in equation (13) is the kernel function:

$$G(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) = \exp\left(-r\left\|x_i - x_j\right\|^2\right) \quad (12)$$

where σ represents the width of the kernel. In the analysis, the Gaussian kernel function parameter G in (12) is adjusted to different values according to the following assumptions:

$r_{fG} = \sqrt{p/_4}$, for fine Gaussian,
$r_{mG} = \sqrt{p}$, for medium Gaussian, and
$r_{cG} = 4\sqrt{p}$, for coarse Gaussian,

where p is the number of features or the dimension size of x_i in (13).

$$f(x) = w^T \phi(x) + b \quad (13)$$

Different Gaussian kernels have different characteristics because they are used in different fields. Generally, fine Gaussian can classify more complex data, medium Gaussian can classify medium-complexity data, and coarse Gaussian can classify low-complexity data. [16].

2.1.11 Gaussian process regression

A Gaussian process is any collection of random variables where an arbitrary subset of variables has a joint Gaussian distribution. A Gaussian distribution is entirely specified by the mean covariance and similarly is completely specified by its mean function $m(x)$ and covariance function $\kappa(x, \acute{x})$.

That is, $m(x) = \mathbb{E}[\mathrm{f(x)}] = 0$ and, $\kappa(x, \acute{x}) = \mathbb{E}\left[\left(\mathrm{f(x)} - \mathrm{m(x)}\right)\left(\mathrm{f}\left(\acute{x} - m(\acute{x})\right)\right)\right] = K$ [N1].

2.1.12 Squared exponential kernel

The equation for the squared exponential kernel is given by:

$$\kappa(z, \acute{z}) = \varphi_1^2 \exp\left(-\frac{\|z - \acute{z}\|^2}{2\varphi_2^2}\right) \quad (14)$$

The hyper parameter $\varphi_1$describes the signal variance which determines the average distance of the data-generating function from its mean. The length scale φ_2 defines how far it is needed to move along a particular axis in input space for the function values to become uncorrelated. Formally, the lenghtscales determines the number of expected up crossings of the level zero in a unit interval by a zero-mean GP. The squared exponential kernel is infinitely differentiable, which means that the GPR exhibits a smooth behavior as limit of the Matern kernel, it is also a universal kernel.

2.1.13 Rational quadratic kernel

The equation for the rational quadratic kernel is given by:

$$\kappa(z, \acute{z}) = \varphi_1^2 \left(1 + \frac{\|z - \acute{z}\|^2}{2p\varphi_2^2}\right)^{-p}, p \epsilon \mathbb{N} \quad (15)$$

This kernel is equivalent to summing over infinitely many squared exponential kernels with different length scales. The parameter p determines the relative weighting of large-scale and small-scale variations. For $p \rightarrow \infty$, the rational quadratic kernel is identical to the squared exponential kernel.

The equation for the squared exponential automatic relevance determination (ARD) is given by:

$$\kappa(z, \acute{z}) = \varphi_1^2 \exp\left(-(z - \acute{z})^{\mathrm{T}} P^{-1} (z - \acute{z})\right) \tag{16}$$

$$P = \mathrm{diag}(\varphi_2^2, \dots, \varphi_{1+n_z}^2) \tag{17}$$

The ARD extension to the squared exponential kernel allows for independent lenghtscales $\varphi_2, \dots, \varphi_{1+n_z} \in \mathbb{R} > 0$ for each dimension of $z, \acute{z} \in \mathbb{R}^{n_z}$. The individual lenghtscales are typically larger for dimensions which are irrelevant as the covariance will become almost independent of that input [3].

2.1.14 Multiple and polynomial regression analysis

The Multiple Regression Analysis (MRA) is a method for analyzing linear relationships between a dependent variable and more than one independent variable. In an MRA, the independent variables affect the dependent variable; therefore, independent variables can be established when validity is obtained in relation to the dependent variable. To describe the relationships between independent and dependent variables, the constant and regression coefficients of each variable are calculated. The general multiple regression equation is expressed in equation (18).

$$Y = \alpha + \beta_1 X_a + \beta_2 X_b + \dots + \beta_k X_{ak} \pm e \tag{18}$$

where Y is the dependent variable, α is constant, $\beta_1 \dots \beta_k$ are the regression coefficients, $X_1 \dots X_k$ are the independent variables, and e is error. The MRA model assesses the validity of the association between the independent and dependent variables using the determination coefficient, R^2. The general determination coefficient equation is given by equation (19), and R^2 is the amount of change in the dependent variable related to the independent variables. However, the determination coefficient increases with the increase in the number of independent variables in the established model, and the adjusted determination coefficient R_{adj}^2 can be used to assess the validity R^2 and R_{adj}^2 are expressed in equations (19) and (20), respectively:

$$R^2 = 1 - \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} \tag{19}$$

$$R_{adj}^2 = 1 - \frac{n-1}{(n-p-1)(1-R^2)} \tag{20}$$

where $\sum(\hat{y}_i - \bar{y})^2$ is the amount of change in the dependent variable, $\hat{y}_i$ is the prediction value, $\bar{y}$ is the average of the experimental values, and y_i is the experimental value. In equation (20), n is the number of experimental values and p is the number of independent variables.

The polynomial regression equation is composed of a predicted variable, intersection, and square terms of the predicted variable. The general polynomial regression equation is given by following equation:

$$Y = \alpha + \sum \beta_i X_i + \sum \beta_{ij} X_{ij} + \sum \beta_{ij} X_i^2 \tag{21}$$

where Y is the predicted variable, α is a constant, $\beta_i - \beta_{ij}$ are the regression coefficients, and X_i and X_j are the input variables [14].

2.2 *Classification learner*

Classification Learner is used to helping machine learning outcomes by combining several methods. This method leads to better forecasting efficiency compared to other methods. The main idea is to learn a set of specialized instrument classifications to improve prediction accuracy.

The purpose of aggregation or reinforcement is to harmonize the grouping structure of the instruments. The possibility of training and testing simultaneously with flexible categories is one of the positive features of these methods. In the reinforcing model, in addition to random sampling with placement, the weight function is also used. Sampling will be different for heavier points, given a specific weight function. In practice, learning is done by assigning weights to different items in the data.

2.2.1 Naive bayes (NB)

In general, classifiers describe each instance using a set of feature-value pairs $(F_1, v_{k_1}), \dots, (F_m, v_{k_m})$. Probabilistic classifiers typically return the label L_j with the largest posterior probability, $P_j^* = P[L = L_j | F_1 = v_{k_1}, \dots, F_m = v_{k_m}]$. Any Naive Bayes classifiers, can be described by a set of parameters, which are typically learned from labeled training data. C_{ijk_i} denotes the number of training instances labeled by L_j, whose value for feature F_i is v_{k_i} and call them the information atoms. n_j, and n derive these information atoms. Given the NB assumption, we can derive a formula for each posterior probability, P_j^* and define its maximum likelihood estimator $\widehat{P_j}$, in terms of the C_{ijk_i} where the normalization constant $\alpha = P[F_1 = v_{k_1}, \dots, F_m = v_{k_m}]$ ensures that the probabilities summation is 1 (as shown in equations (22) and (23)) [28].

$$P_j^* = P[L = L_j | F_1 = v_{k_1}, \dots, F_m = v_{k_m}] = \frac{P[L = L_j] \times \prod_{i=1}^{m} P[F_i = v_{k_i} | L = L_j]}{P[F_1 = v_{k_1}, \dots, F_m = v_{k_m}]} \quad (22)$$

$$\widehat{P_j} = \frac{1}{\alpha}\left(\frac{n_j}{n}\right)\prod_{i=1}^{m}\left(\frac{C_{ijk_i}}{n_j}\right) \quad (23)$$

2.2.2 K-nearest neighbors

K-nearest neighbor classifier is to classify unlabeled observations by assigning them to the class of the most similar labeled examples. Characteristics of observations are collected for both training and test dataset [33].

If D is training dataset and made up of $(x_i)_{i \in [1,|D|]}$ training samples. The examples are described by a set of features F and any numeric features have been normalised to the range $[0.1]$. Each training example is labelled with a class label $y_i \in Y$. Our objective is to classify an unknown example q. For each $x_i \in D$ we can calculate the distance between q and x_i as follows:

$$d(q, x_i) = \sum_{f \in F} w_f \, \delta(q_f, x_{if}) \quad (24)$$

There are a large range of possibilities for this distance metric, a basic version for continuous and discrete attributes would be:

$$\delta(q_f, x_{if}) = \begin{cases} 0 & f \text{ discrete and } q_f = x_{if} \\ 1 & f \text{ discrete and } q_f \neq x_{if} \\ |q_f - x_{if}| & f \text{ continuous} \end{cases} \quad (25)$$

The k nearest neighbours are selected based on this distance metric. Then there are a variety of ways in which the k nearest neighbors can be used to determine the class of q.

The most straightforward approach is to assign the majority class among the nearest neighbors to the query. It will often make sense to assign more weight to the nearer neighbors in deciding the class of the query. A fairly general technique to achieve this is distance weighted voting where the neighbors get to vote on the class of the query case with votes weighted by the inverse of their distance to the query.

$$Vote(y_j) = \sum_{c=1}^{k} \frac{1}{d(q, X_c)^n} 1(y_j, y_c) \tag{26}$$

Thus, the vote assigned to class y_j by neighbour X_c is 1 divided by the distance to that neighbour, i.e., $1(\mathrm{y}_j, \mathrm{y}_c)$ returns 1 if the class labels match and 0 otherwise [4].

2.2.3 Gaussian classification

Given data points x_i from a domain $\mathcal{X}$ with corresponding class labels $y \in \{-1,1\}$ one would like to predict the class membership probability for a test point x_*. This is achieved by using a latent function f whose value is mapped into the unit interval by means of a sigmoid function sig: $\mathbb{R} \to [0,1]$ such that the class membership probability $\mathbb{P}(y = +1|\mathrm{x})$ can be written as $\mathrm{sig}(f(x))$. The class membership probability must normalize $\sum_y \mathbb{P}(y|\mathrm{x}) = 1$, which leads to $\mathbb{P}(y = +1|\mathrm{x}) = 1 - \mathbb{P}(y = -1|\mathrm{x})$. If the sigmoid function satisfies the point symmetry condition $sig(t) = 1 - sig(-t)$, the likelihood can be compactly written as [18]:

$$\mathbb{P}(y|\mathrm{x}) = sig(y.f(x)) \tag{27}$$

2.3 *Artificial neural network (ANN)*

An Artificial Neural Network is a nonlinear mapping tool that relates a set of inputs to a set of outputs. It can learn this mapping using a set of training data and then generalize the obtained knowledge to a new set of data. ANNs have a variety of applications as a classifier, one of the most commonly used ANNs is the Multi-Layer Perceptron (MLP) network. There are three types of layers in any MLP: the output layer, the input layer, and the hidden layer. Each layer comprises of n nodes ($n \geq 1$), and each node in any layer is connected to all the nodes in the neighboring layers. Each node can also connected to a constant number which is called bias. These connections have their individual weights called synaptic weights, and multiplied by the node values of the previous layer [1].

With the help of weight modification, the network can learn or train to map patterns in the input to the target value on the output. The procedure used for weight adaptation is called the learning or training algorithm. ANN is good for the complex system, which underlies the process, not completely disordered properties [27].

In the Neural Network, with the corresponding inputs and outputs, the weight of the network is adjusted in such a way that the network error, which is the difference between the desired output and the network output, is minimized. This minimization done in different ways, such as Levenberg-Marquardt (LM), Bayesian Regularization (BR), and Scaled Conjugate Gradient (SCG).

For solving nonlinear least-squares problems, Levenberg-Marquardt Algorithm (LMA) usually use as a standard algorithm. A combination of gradient descent and Gauss-Newton methods appears in this algorithm. In many cases, the LMA can guarantee problem-solving through its adaptive behavior. If back-propagation (BP) expressed as gradient descent, the algorithm becomes slow, and doesn't give an optimal solution. On the other hand, if BP expressed as Gauss-Newton, the algorithm has the highest probability to give an optimal solution. In this algorithm, the Hessian calculation approximation shows in equation (28), and the calculation of its gradient expresses in equation (29):

$$H = J^T J \tag{28}$$

$$g = j^T e \tag{29}$$

where Jacobian matrix represents by J, and e indicate a vector of a network error. The LMA behaves as Newton expresse by the following equation:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{30}$$

where x is a new weight that calculate as gradient function, and x as current weight using the Newton algorithm.

The most popular iteration algorithm to solve the problem of large systems of linear equations is Conjugate Gradient. The iteration of the conjugate gradient show in equation (31):

$$x_k = x_{k-1} + \alpha_k d_{k-1} \tag{31}$$

where k is the iteration index, α is the step length at k iteration, and d is the search direction.

Scaled Conjugate Gradient Algorithm (SCGA) is the second order of Conjugate Gradient Algorithm which can minimize the purpose function on several variables. SCGA uses step size scaling techniques to avoid consumption of learning iteration time and exhibits the superliner convergence on almost all problems as an advantage [5].

To reduce overfitting, a mathematical technique known as Bayesian regularization developed to convert nonlinear systems into well-posed problems. In general, the training step aims to reduce the sum squared error of the model output and target value. Bayesian regularization adds an additional term to this equation;

$$F = \beta E_D + \alpha E_w \tag{32}$$

where F is the objective function, E_D is the sum of squared errors, E_w is the sum of the square of the network weights, and α, and β are objective function parameters. In the Bayesian network, the weights are considered random variables, and thus, their density function is written according to the Bayes' rules:

$$P(w|D, \alpha, B, M) = \frac{P(D|w, \beta, M)P(w|\alpha, M)}{P(D|\alpha, \beta, M)} \tag{33}$$

where w is the vector of network weights, D represents the data vector, and M is the neural network model being used.

In Bayesian regularized networks, overly complex models are penalized as unnecessary linkage weights are effectively driven to zero. The network will calculate and train on the nontrivial weights, also known as the adequate number of parameters, which will converge to a constant as the network grows. The inherent noise and volatility of stock markets introduce a high probability of overfitting and overtraining for general back propagation networks. These more parsimonious networks reduce the chance of overfitting while eliminating that, requires a validation step, and thus increasing the available data for training, introduces a high probability of overfitting and overtraining for general back propagation networks [30].

3. Preliminaries

3.1 *Data analysis*

The software used is MATLAB, and all calculations performed in Windows10 based on the hardware specifications presented in Table 1.

Table 1. Hardware specifications (Desktop PC).

Processor Model	Intel Core i5 - 7400
Core Counts	6
Frequency	3.1 GHz
Cashes	9 MB
RAM	16 GB

3.2 *Companies selection*

In order to select suitable companies for research, data of companies listed on the Tehran Stock Exchange in the year 1397 AH (from March 21, 2018, to March 19, 2019) from the website of Tehran Securities Exchange Technology Management Company[1] received and were examined based on the following criteria:

1. Top companies in terms of number and volume of transactions.
2. Companies selected from different industries.
3. Selected companies are not major shareholders of other company.
4. More than five years have passed since the admission and offers of companies' shares in the Tehran Stock Exchange.

As a result, Tamin Petroleum & Petrochemical Investment, Saipa group, Isfahan Oil Refining, Chadormalu Mining & Industrial, and Tejarat Bank companies were selected.

3.3 *Data collection*

After selection companies, the data related to each share includes the symbol name, date, first price, highest price, lowest price, final price, value, volume, number, period, yesterday price, price of the last transaction since beginning of acceptance in Stock Exchange received from the website of Tehran Securities Exchange Technology Management Company by using a program written with Python (version 3.7.4) for web scraping in CSV[2] file format.

The period of collected data is daily and concept and unit of them includes:

First price: The first price means the price at which the first share transaction took place and its unit is Rial.

Highest price: The highest price means the maximum price at which the share was traded on the day and its unit is Rial.

Lowest price: The lowest price means the minimum price at which the share was traded on the day and its unit is Rial.

Trading volume: Trading volume, refers to the amount of shares traded in a day.

Value of transactions: The value of transactions is obtained from the product of the volume of transactions and the daily price of the shares, the value of transactions and its

[1] http://tsetmc.com

[2] Comma Separated Values

unit is Rial.

Last trade price: The last traded price is the last price at which the trade occurred on the day and its unit is Rial.

Opening price: The opening price is the price that is announced at the beginning of the official trading session for each security on the stock market board.

Trading volume: Trading volume is the amount of securities traded in a day.

Close price: If the trading volume of the share during the day is equal to or more than its base volume, the closing price of the share will be equal to the weighted average of the trading price of the symbol during the trading session, otherwise, the closing price will be calculated as follows:

$$p_1 = p_0 + \left[\frac{N}{M}(VWAP - P_0)\right] \tag{34}$$

where in:
p_1: The final price of the day
p_0: The closing price of the previous day
$VWAP$: Weighted average of traded stock prices
N: The volume of transactions in the normal market
M: Base volume
Period: The time period of transactions is included in the data "D", which means daily.

3.4 *Preparing data*

After reviewing the data, because of the article aims is to predict next day price, the last transaction on the next day selected as the response variable and other variables as the predictor variable.

3.5 *Data analysis*

In the first stage, 70% of the data used for training, 15% for validation, and 15% for testing, and three methods, LM, BR, and SCG, were used to train Neural Networks. Then, statistical learning algorithms in two categories of Regression Learner and Classification Learner implemented on selected shares.

The first category (Regression Learner) includes Linear Regression, Interaction Linear Regression, Robust Linear Regression, Stepwise Linear Regression, Fine Tree, Medium Tree, Coarse Tree, Linear Support Vector Machine (SVM), Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Ensemble Boosted Trees, Ensemble Bagged Trees, Squared Exponential Gaussian Process Regression, Matern 5/2 Gaussian Process Regression, Exponential Gaussian Process Regression, Rational Quadratic Gaussian Process Regression. The second category (Classification Learner) includes Gaussian, Naive Bayes, K-nearest neighbors.

After implementing the methods, the best algorithms were compared based on the mentioned indicators.

3.6 *Parameters*

After implementing the methods, the best algorithms were measured based on R-Squared, MSE, and RMSE criteria. An explanation of statistical parameters presents in Table 2. Mathematically, the mean least-squares error (RMSE) for the data y_i where $i = 1, \dots, n$ with the mean $\hat{y}_i$ is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{35}$$

The results of the optimal method are presented for each symbol separately.

Table 2. Explanation of parameters.

Note	Description	Parameter
Smaller values indicate better results, and models	The root of the mean squared error is always positive and its unit corresponds to the response unit	RMSE
The square values of R are close to one; if this criterion is equal to 0.99, it means that the set of variables used in the model explains 0.99 of the model varieties	The coefficient of determination is always a value between zero, and one, which compares the Trained model with a model whose response is constant and equal to the average response answer. If the model is worse than this fixed model, R becomes a negative square	R-Squared
Smaller values of MSE are more desirable	MSE is square of RMSE	MSE

In addition to the above criteria, to evaluate the performance of the optimal model, we use the graph of predicted values in comparison with real values.

4. Results and discussion

4.1 *Tamin petroleum & petrochemical investment company*

Data of Tamin Petroleum & Petrochemical Investment Company include 1356 records in 16272 information fields from July 9; 2013, to September 22; 2019. The answer variable diagram is as follows:

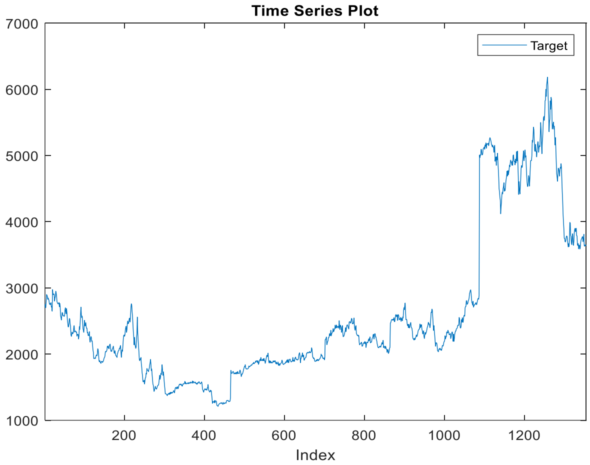

Figure 1. Graph of variable response values.

At first step, attempts to build Neural Networks and Classification Learner were not successful. In the next step, Regression Learner methods applied to the data set. Table 3 shows the results of this method.

Table 3. Comparison between Regression Learner methods.

Model	**RMSE**	**Prediction speed (obs/sec)**	**Training time (sec)**
Linear Regression	86.171	5200	11.187
Intreraction Linear Regression	2127.900	4200	19.902
Robust Linear Regression	85.017	1300	4.7301
Stepwise Linear Regression	85.583	31000	63.896
Fine Tree	104.670	20000	17.905
Medium Tree	100.870	29000	16.898
Coarse Tree	116.590	64000	16.445
Linear SVM	88.704	3900	88.313
Quadratic SVM	1652.800	16000	406.960
Cubic SVM	12363.000	53000	427.370
Fine Gaussian SVM	237.780	46000	407.830
Medium Gaussian SVM	120.950	35000	408.300
Coarse Gaussian SVM	97.339	46000	408.750
Ensemble Boosted Trees	152.040	13000	414.690
Ensemble Bagged Trees	93.192	8300	417.170
Squared Exponential Gaussian Process Regression	176.230	12000	514.550
Matern 5/2 Gaussian Process Regression	145.940	9800	507.670
Exponential Gaussian Process Regression	98.210	15000	676.360
Rational Quadratic Gaussian Process Regression	109.720	8200	730.940

Examining the results, we find that the Robust Linear Regression method is the best method in terms of RMSE with the results presented in Table 4.

Table 4. Optimal model results.

RMSE	85.017
R	0.99
MSE	7227.9
MAE	42.159
Prediction speed	1300 obs/sec
Training time	4.7301 sec

Figure 2 is the graph of the results of the optimal model. This diagram shows the prediction values with light points and the actual values observed with dark points. The match of the points indicates the accuracy of the forecast.

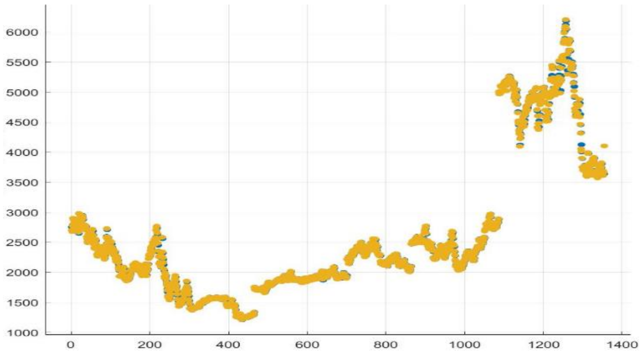

Figure 2. Model diagram of predicted values compared to actual values.

4.2 *Saipa group*

Data of Saipa Company includes 3839 records in 46068 information fields from 16 May; 2001 to 31; September 2019. The answer variable diagram is as follows:

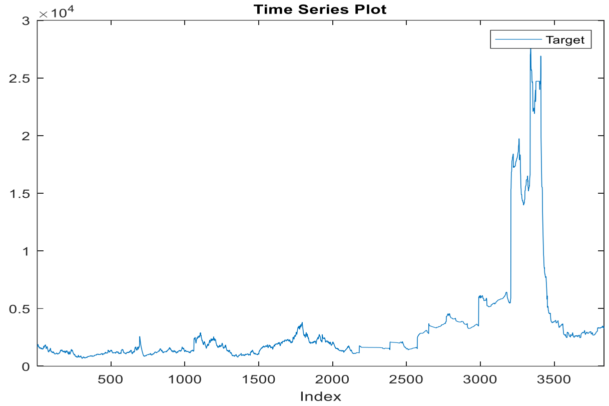

Figure 3. Graph of variable response values.

In the first stage, according to the optimal results, the structure of the resulting Neural Network is illustrated in Figure 4.

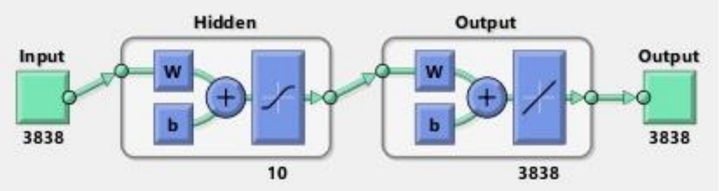

Figure 4. ANN structure.

The best model training algorithm, in this case, is LM, and the results presents in Table 5.

Table 5. Neural Network results.

Epoch	29 iteration
Time	0:00:00
Performance	3.22E+21
Gradient	1.41E+21
MSE Training	2147483647
MSE Validation	2147483647
MSE Testing	2147483647
R Training	0.233682
R Validation	0.560898
R Testing	-0.0867748

In the next step, Classification Learner methods were not successful. Then we applied Regression Learner methods to the data set. Table 6 shows a comparison between these methods.

Table 6. Comparison between Regression Learner methods.

Model	RMSE	Prediction speed (obs/sec)	Training time (sec)
Linear Regression	275.860	26000	14.619
Intreraction Linear Regression	2222.000	16000	12.474
Robust Linear Regression	271.61	38000	50.358
Stepwise Linear Regression	277.070	67000	66.594
Fine Tree	300.190	50000	19.568
Medium Tree	309.280	200000	18.140
Coarse Tree	393.490	140000	17.788
Linear SVM	276.690	34000	237.910
Quadratic SVM	593.130	29000	817.540
Cubic SVM	20878.000	91000	978.440
Fine Gaussian SVM	974.320	44000	818.900
Medium Gaussian SVM	389.230	51000	819.890
Coarse Gaussian SVM	294.340	53000	820.820
Ensemble Boosted Trees	364.530	33000	827.680
Ensemble Bagged Trees	285.080	21000	831.280
Squared Exponential Gaussian Process Regression	655.770	7400	1586.400
Matern 5/2 Gaussian Process Regression	616.190	4800	1540.800
Exponential Gaussian Process Regression	291.640	7100	2247.100
Rational Quadratic Gaussian Process Regression	356.070	3500	27856.600

Examining the results, we find that the Robust Linear Regression method is the best method in terms of RMSE with the following results.

Table 7. Optimal model results.

RMSE	271.61
R	1
MSE	73772
MAE	52.624
Prediction speed	38000 obs/sec
Training time	50.358 sec

The results of comparing the two optimal methods are presents in Table 8.

Table 8. Results of comparing the results of optimal models.

Method	ANN	Regression Learner	Amount of difference
Optimal model	LM	**Robust Linear Regression**	
MSE	2147483647	73772	2147409875
R	0.233682	1	-0.766318
RMSE	46340.95	271.61	46069.34

As shown in Figure 5, the matching of the points indicates the high degree of accuracy of the prediction of the Robust Linear Regression method.

4.3 *Isfahan oil refining company*

The data of Isfahan Oil Refining Company includes 1940 observations in 23280 information fields from July 30; 2008, to September 22; 2019. The answer variable diagram is as follows:

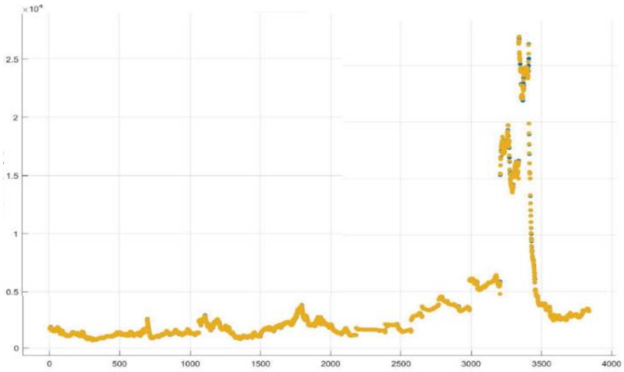

Figure 5. Model diagram of predicted values compared to actual values.

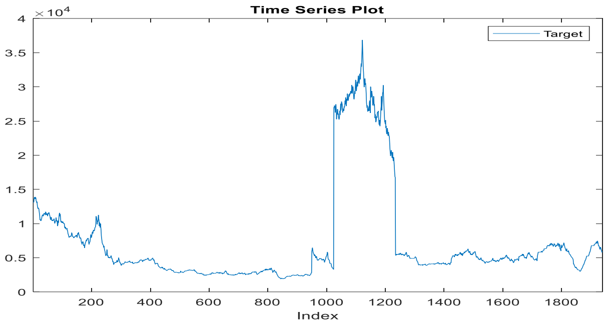

Figure 6. Graph of variable response values.

In this symbol, the attempt to build Neural Networks, and the classification learning algorithm was not successful. In the next step, Regression Learner methods applied to the data set. Table 9 shows the results of this method.

Table 9. Comparison between Regression Learner methods.

Model	RMSE	Prediction speed (obs/sec)	Training time (sec)
Linear Regression	667.030	10000	12.780
Intreraction Linear Regression	8996.900	10000	10.925
Robust Linear Regression	663.05	21000	15.251
Stepwise Linear Regression	663.350	50000	55.618
Fine Tree	824.250	33000	17.813
Medium Tree	733.270	110000	17.215
Coarse Tree	830.670	51000	19.151
Linear SVM	665.750	29000	109.600
Quadratic SVM	2651.600	30000	419.830
Cubic SVM	121640.000	68000	484.250
Fine Gaussian SVM	1841.100	49000	420.720
Medium Gaussian SVM	922.370	56000	421.210
Coarse Gaussian SVM	747.910	31000	422.070
Ensemble Boosted Trees	854.890	21000	427.160
Ensemble Bagged Trees	709.310	15000	430.010
Squared Exponential Gaussian Process Regression	1532.600	11000	700.020
Matern 5/2 Gaussian Process Regression	1218.100	8100	726.900
Exponential Gaussian Process Regression	752.980	10000	1009.300
Rational Quadratic Gaussian Process Regression	872.120	7300	1278.800

The results show that the Robust Linear Regression method is the best method in terms of RMSE .The results show in Table 10.

Table 10. Optimal model results.

RMSE	663.05
R	0.99
MSE	444930
MAE	169.82
Prediction speed	21000 obs/sec
Training time	15.2516 sec

Figure 7 is the optimal model diagram; the high accuracy of the prediction is evident from the matching of light (prediction) and dark (Real) dots.

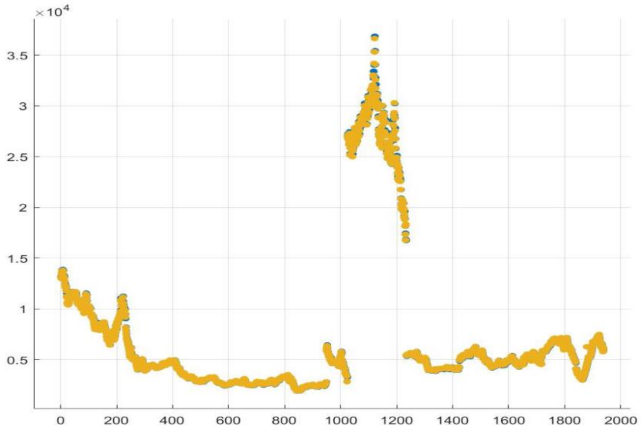

Figure 7. Model diagram of predicted values compared to actual values.

4.4 *Chadormalu mining & industrial company*

The data of Chadormalu Mining & Industrial Company includes 3354 observations in 40284 information fields from October 18; 2003, to September 22; 2019.

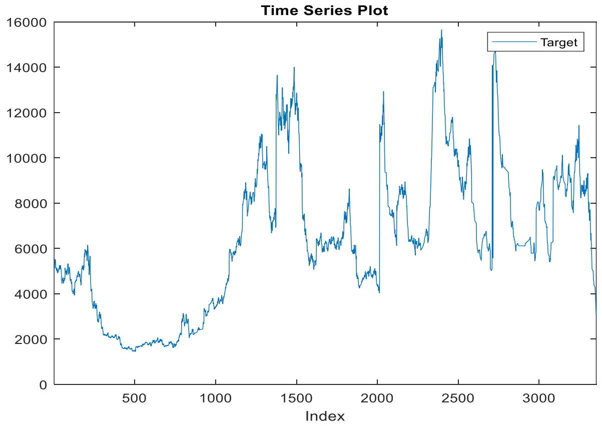

Figure 8. Graph of variable response values.

In the first stage, according to the optimal results, the structure of the resulting Neural Network is as follows. Neural Networks are applied with ten middle or hidden layers.

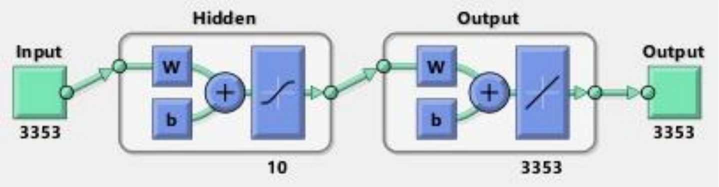

Figure 9. ANN structure.

The best model training algorithm, in this case, is LM. The resulting Neural Network results describe in Table 11.

Table 11. Neural Network results.

Epoch	23 iteration
Time	0:00:00
Performance	9.25E+20
Gradient	1.77E+21
MSE Training	2147483647
MSE Validation	2147483647
MSE Testing	2147483647
R Training	0.447696
R Validation	-0.174105
R Testing	0.0242507

Next, the Classification Learner methods were not successful. Then Regression Learner methods were applied to the data set. Table 12 shows a comparison between these methods.

Table 12. Comparison between Regression Learner methods.

Model	RMSE	Prediction speed (obs/sec)	Training time(sec)
Linear Regression	349.110	21000	20.636
Intreraction Linear Regression	9778.500	10000	35.488
Robust Linear Regression	397.31	28000	34.829
Stepwise Linear Regression	350.200	48000	82.751
Fine Tree	418.490	30000	32.068
Medium Tree	364.040	43000	31.324
Coarse Tree	371.090	55000	30.485
Linear SVM	356.380	27000	103.460
Quadratic SVM	938.990	23000	777.030
Cubic SVM	51104.000	81000	848.270
Fine Gaussian SVM	625.950	72000	777.080
Medium Gaussian SVM	422.510	100000	777.930
Coarse Gaussian SVM	348.750	87000	778.760
Ensemble Boosted Trees	469.770	38000	788.720
Ensemble Bagged Trees	370.270	26000	789.050
Squared Exponential Gaussian Process Regression	336.890	11000	1421.200
Matern 5/2 Gaussian Process Regression	340.780	6000	1471.000
Exponential Gaussian Process Regression	329.85	7400	70.764
Rational Quadratic Gaussian Process Regression	351.340	4700	2702.000

Examining the results, we find that the Exponential Gaussian Process Regression method is the best method in terms of RMSE with the following results.

Table 13. Optimal model results.

RMSE	329.85
R	0.99
MSE	127010
MAE	120.04
Prediction speed	7400 obs/sec
Training time	70.764 sec

The results of comparing the two optimal methods present in Table 14.

Table 14. Results of comparing the results of optimal models.

Method	ANN	Regression Learner	Amount of difference
Optimal model	LM	Gaussian Exponential GPR	
MSE	2147483647	127010	2147356637
R	0.447696	0.99	-0.542304
RMSE	46340.95	329.85	46011.1

Figure 10 is the optimal model (Exponential Gaussian Process Regression) diagram. The high correlation between the light and dark dots indicates the high accuracy of the prediction.

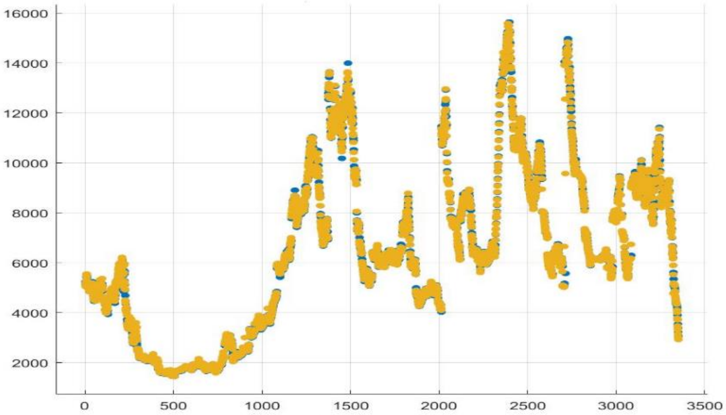

Figure 10. Model diagram of predicted values compared to actual values.

4.5 *Tejarat bank*

The data of Tejarat Bank includes 3354 records in 40248 information fields from October 18; 2003, to September 22; 2019.

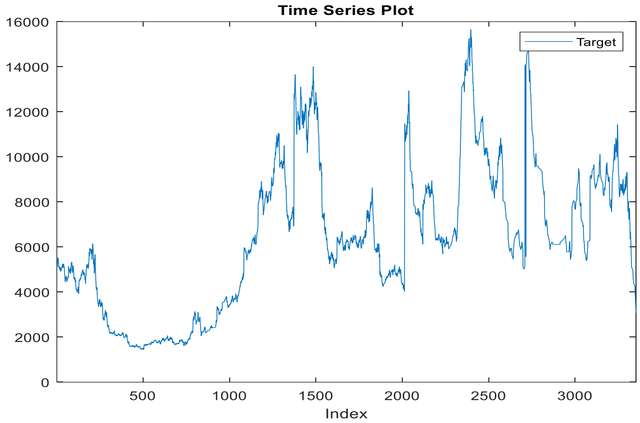

Figure 11. Graph of variable response values.

In the first stage, Neural Networks are applied with ten middle or hidden layers. According to the optimal results, the structure of the resulting Neural Network is as follows.

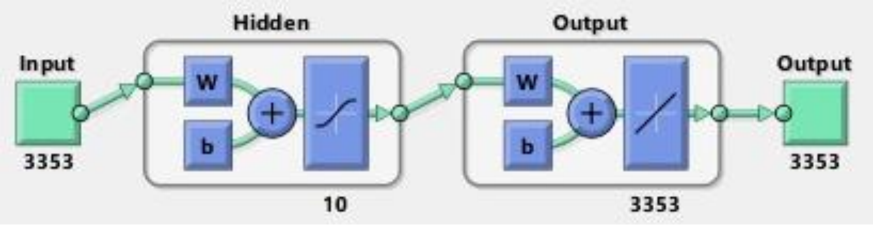

Figure 12. ANN structure.

The best model training algorithm, in this case, is LM. The results of the Neural Network describe in table 15.

Table 15. Neural Network results.

Epoch	164 iteration
Time	0:00:01
Performance	1.88E+20
Gradient	5.99E+20
MSE Training	2147483647
MSE Validation	2147483647
MSE Testing	2147483647
R Training	0.826179
R Validation	0.0330894
R Testing	0.0294807

In this case, too, Classification Learner methods did not work. The results of the Regression Learner methods present in Table 16.

Table 16. Comparison between Regression Learner methods.

Model	RMSE	Prediction speed (obs/sec)	Training time(sec)
Linear Regression	351.960	6000	33.009
Intreraction Linear Regression	13455.000	5000	47.075
Robust Linear Regression	394.81	32000	46.138
Stepwise Linear Regression	338.750	33000	118.270
Fine Tree	398.640	36000	44.515
Medium Tree	370.030	68000	41.561
Coarse Tree	372.370	22000	46.207
Linear SVM	358.550	21000	134.360
Quadratic SVM	983.430	31000	933.150
Cubic SVM	89217.000	57000	1050.500
Fine Gaussian SVM	635.280	48000	935.040
Medium Gaussian SVM	423.050	70000	936.400
Coarse Gaussian SVM	350.690	42000	937.720
Ensemble Boosted Trees	467.940	31000	946.890
Ensemble Bagged Trees	365.800	15000	951.570
Squared Exponential Gaussian Process Regression	373.470	7200	1890.600
Matern 5/2 Gaussian Process Regression	343.130	4400	2072.700
Exponential Gaussian Process Regression	323.78	5100	2763.9
Rational Quadratic Gaussian Process Regression	343.750	3100	3583.300

Examining the results, we find that the Exponential Gaussian Process Regression method is the best method in terms of RMSE with the following results:

Table 17. Optimal model results.

RMSE	323.78
R	0.99
MSE	104380
MAE	117.83
Prediction speed	5100 obs/sec
Training time	2763.9 sec

The results of comparing the two optimal methods are present in Table 18.

Table 18. Results of comparing the results of optimal models.

Method	ANN	Regression Learner	Amount of difference
Optimal model	LM	**Exponential GPR**	
MSE	2147483647	104380	2147379267
R	0.826179	0.99	-0.163821
RMSE	46340.95	323.78	46017.17

Figure 13 is the optimal (Exponential Gaussian Process Regression) model diagram. The high accuracy of the prediction is evident from the matching of light (prediction) and dark (Real) dots.

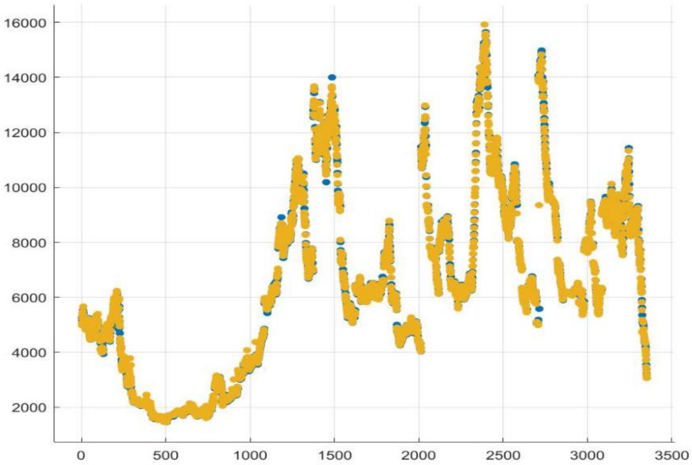

Figure 13. Model diagram of predicted values compared to actual values.

5. Conclusion

In this paper, to examine the five selected symbols of the Tehran Stock Exchange, we used Artificial Neural Network methods and various Statistical Learning to predict future prices. The results indicate the optimal performance of Regression Learner methods compared to Artificial Neural Networks and Classification Learners. Comparison of prediction values with real values indicates the high accuracy of prediction of Regression Learner methods and can use in cases where the Neural Network is not responsive. In cases where the ANN is responsive, the prediction accuracy of the Regression Learner is higher. According to the results of this study, Regression Learner methods, especially Robust Linear Regression and Exponential Gaussian Process Regression, in addition to accuracy in terms of speed and computational efficiency, are significant in predicting capital market capitalization and can be used in research and algorithmic trading systems design, and also comparing models or presenting hybrid models.

References

[1] M. Akbari, H. Homaei and M. Heidari, An intelligent fault diagnosis approach for gears and bearings based on wavelet transform as a preprocessor and artificial neural networks, International Journal of Mathematical Modelling & Computations, **4** (2014) 309–329.

[2] O. Altay, M. Ulas and K. E. Alyamac, Prediction of the fresh performance of steel fiber reinforced self-compacting concrete using quadratic SVM and weighted KNN models, IEEE Access, **8** (2020) 92647–92658.

[3] T. Beckers, An introduction to Gaussian process models, Electrical Engineering and Systems Science, (2021), doi:10.48550/arXiv.2102.05497.

[4] P. Cunningham and S. J. Delany, k-Nearest neighbour classifiers, ACM Computing Surveys, **54 (6)** (2007), Article 128, doi:10.1145/3459665.

[5] Y. C. Du and A. Stephanus, Levenberg-Marquardt neural network algorithm for degree of arteriovenous fistula stenosis classification using a dual optical photoplethysmography sensor, Sensors, **18** (2018) 1–26, doi:10.3390/s18072322.

[6] A. M. Farayola, A. N. Hasan and A. Ali, Optimization of PV systems using data mining and regression learner MPPT techniques, Indonesian Journal of Electrical Engineering, **10** (2018) 1080–1089.

[7] M. Farrell and A. Correa, Gaussian process regression models for predicting stock trends, Computer Science, **10** (2007) 3414.

[8] P. L. Flom and D. L. Cassell, Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use, NESUG Conference, Baltimore, Maryland, (2007).

[9] J. Gareth, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics, Springer, (2014).

[10] E. Gholamian and S. M. Davoodi, Predicting the direction of stock market prices using random forest, Financial Engineering and Portfolio Management, **9** (2018) 301–322.

[11] E. Guresen, G. Kayakutlu and T. U. Daim, Using artificial neural network models in stock market index prediction, Expert Systems with Applications, **38 (8)** (2011) 10389–10397.

[12] S. Hajiali, M. Doroudian and H. Borna, Understanding Behavior of Antineoplastic Molecules Based on MLR Models, International Journal of Mathematical Modelling & Computations, **1** (2011) 263–269.

[13] H. Jafarzadeh, M. Mahdianpari, E. Gill, F. Mohammadimanesh and S. Homayouni, Bagging and boosting ensemble classifiers for classification of multispectral, hyperspectral and PolSAR data: A comparative evaluation, Remote Sensing, **13 (21)** (2021) 4405, doi:10.3390/rs13214405.

[14] Y. Kim and H. Oh, Comparison between multiple regression analysis, polynomial regression analysis, and an artificial neural network for tensile strength prediction of BFRP and GFRP, Materials, **14 (17)** (2021) 4861.

[15] M. Kuss, Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning Darmstadt, Technische Universität, Ph.D. Thesis, (2006).

[16] S. L. Lin, Application of machine learning to a medium Gaussian support vector machine in the diagnosis of motor bearing faults, Electronics, **10 (18)** (2021) 2266, doi:10.3390/electronics10182266.

[17] Y. Lin, H. Guo and J. Hu, An SVM-based approach for stock market trend prediction, International Joint Conference on Neural Networks (IJCNN), Dallas, Texas, USA, (2013) 4–9.

[18] H. Nickisch and C. Rasmussen, Approximations for binary Gaussian process classification, Journal of Machine Learning Research, **9** (2008) 2035–2078.

[19] R. Noori, K. Ashrafi and A. Ajdarpour, Comparison of ANN and PCA based multivariate linear regression applied to predict the daily average concentration of CO: A case study of Tehran, Journal of the Earth and Space Physics, **34** (2008) 135–152.

[20] R. Raee, A. Nikahd and M. Habib, M., The Index Prediction of Tehran Stock Exchange by Combining the Principal Components Analysis, Support Vector Regression and Particle Swarm Optimization, Journal of Financial Management Strategy, Alzahra University, **4** (2017) 1–23.

[21] R. Rivest, Learning decision lists, Machine Learning, **2** (1987) 229–246.

[22] C. Robinson, S. Tomek and R. Schumacker, Tests of moderation effects: Difference in simple slopes versus the interaction term, Multiple Linear Regression Viewpoints, **39** (2013) 16–24.

[23] A. Sadat Najafi and S. Sardar, Application of econometric modeler for predicting stock prices in the capital market, Financial Engineering and Portfolio Management, **11 (45)** (2020) 223–246.

[24] D. Selvamuthu, V. Kumar and A. Mishra, Indian stock market prediction using artificial neural networks on tick data, Financial Innovation, **16** (2019) 5–16.

[25] D. Shah, H. Isah and F. Zulkernine, Stock market analysis: A review and taxonomy of prediction techniques, International Journal of Financial Studies, **7 (2)** (2019) 7–26.

[26] K. Shailaja and M. Manjunath, Prediction and clustering techniques used in the development of stock forecasting model, International Journal of Innovative Technology and Exploring Engineering, **9 (3)** (2020) 2278–3075.

[27] G. Sundar and K. Satyanarayana, Stock prediction scrutiny using artificial neural network, International Journal of Recent Technology and Engineering, **7 (623)** (2019) 2277–3878.

[28] D. Szafron, R. Greiner, P. Lu, D. Wishart, C. Macdonell, J. Anvik, B. Poulin, Z. Lu and R. Eisner, Explaining naïve Bayes classifications, Computer Science, (2003), doi:10.7939/R36D5PH6N.

[29] B. Taremi and B. Zeighami, Biostatistics, Payame Noor University, 17th Edition, (2020), (In Persian).

[30] J. Ticknor, A bayesian regularized artificial neural network for stock market forecasting, Expert Systems with Applications, **40 (14)** (2013) 5501–5506.

[31] G. Vaishnavi, V. Shriya and K. Ashwini, Stock market prediction using linear regression and support vector machines, International Journal of Applied Engineering Research, **14** (2019) 1931–1934.

[32] C. Yu and W. Yao, Robust linear regression: A review and comparison, Communications in Statistics -Simulation and Computation, **46 (8)** (2017) 6261–6282.

[33] Z. Zhang, Introduction to machine learning: K-nearest neighbors, Annals of Translational Medicine, **4 (11)** (2016) 218.