

# Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms

1. Anil Kumar Sikkakoli

Associate professor  
Dept Of CSE, TEC  
[sakmba.k@gmail.com](mailto:sakmba.k@gmail.com)

2. Srija Kuruba

Department Of CSE  
[srijakuruba2005@gmail.com](mailto:srijakuruba2005@gmail.com)

3. Neelima Mallavarapu

Department Of CSE  
[neelimamallavarapu123@gmail.com](mailto:neelimamallavarapu123@gmail.com)

4. Srinivasarao Muppalla

Department of CSE  
[muppallasrinivasarao750@gmail.com](mailto:muppallasrinivasarao750@gmail.com)

5. Saikiran Jammula

Department of CSE  
[saikiranc456@gmail.com](mailto:saikiranc456@gmail.com)

## ABSTRACT

The nature of stock market movement has always been ambiguous for investors because of various influential factors. This study aims to significantly reduce the risk of trend prediction with machine learning and deep learning algorithms. Four stock market groups, namely diversified financials, petroleum, non-metallic minerals and basic metals from Tehran stock exchange, are chosen for experimental evaluations. This study compares nine machine learning models (Decision Tree, Random Forest, Adaptive Boosting (Adaboost), eXtreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Naïve Bayes, K-Nearest Neighbors (KNN), Logistic Regression and Artificial Neural Network (ANN)) and two powerful deep learning methods (Recurrent

Neural Network (RNN) and Long short-term memory (LSTM). Ten technical indicators from ten years of historical data are our input values, and two ways are supposed for employing them. Firstly, calculating the indicators by stock trading values as continuous data, and secondly converting indicators to binary data before using. Each prediction model is evaluated by three metrics based on the input ways. The evaluation results indicate that for the continuous data, RNN and LSTM outperform other prediction models with a considerable difference. Also, results show that in the binary data evaluation, those deep learning methods are the best; however, the difference becomes less because of the noticeable improvement of models' performance in the second way


⋮  
⋮  
⋮  
**INDEX TERMS** Stock market, trends prediction, classification, machine learning, deep learning.

## I. INTRODUCTION

The task of stock prediction has always been a challenging problem for statistics experts and finance. The main reason behind this prediction is buying stocks that are likely to increase in price and then selling stocks that are probably to fall. Generally, there are two ways for stock market prediction. Fundamental analysis is one of them and relies on a company's technique and fundamental information like mar-

ket position, expenses and annual growth rates. The second one is the technical analysis method, which concentrates on previous stock prices and values. This analysis uses historical charts and patterns to predict future prices [1], [2].

Stock markets were normally predicted by financial experts in the past time. However, data scientists have started solving prediction problems with the progress of learning techniques. Also, computer scientists have begun using machine learning methods to improve the performance of prediction models and enhance the accuracy of predictions. Employing deep learning was the next phase in improving

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao .

prediction models with better performance [3], [4]. Stock market prediction is full of challenges, and data scientists usually confront some problems when they try to develop a predictive model. Complexity and nonlinearity are two main challenges caused by the instability of stock market and the correlation between investment psychology and market behavior [5].

It is clear that there are always unpredictable factors such as the public image of companies or political situation of countries, which affect stock markets trend. Therefore, if the data gained from stock values are efficiently preprocessed and suitable algorithms are employed, the trend of stock values and index can be predicted. In stock market prediction systems, machine learning and deep learning approaches can help investors and traders through their decisions. These methods intend to automatically recognize and learn patterns among big amounts of information. The algorithms can be effectively self-learning, and can tackle the predicting task of price fluctuations in order to improve trading strategies [6].

Since recent years, many methods have been improved to predict stock market trends. The implementation of a model combination with Genetic Algorithms (GA), Artificial Neural Networks and Hidden Markov Model (HMM) was proposed by Hassan *et al.* [7]; the purpose was transforming the daily stock values to independent groups of prices as inputs to HMM. The predictability of financial trend with SVM model by evaluating the weekly trend of NIKKEI 225 index was investigated by Huang *et al.* [8]. A comparison between SVM, Linear Discriminant method, Elman Backpropagation Neural Networks and Quadratic Discriminant method was their goal. The results indicated that SVM was the best classifier method. New financial prediction algorithm based on SVM ensemble was proposed by Sun *et al.* [9]. The method for choosing SVM ensemble's base classifiers was proposed by deeming both diversity analysis and individual prediction. Final results showed that SVM ensemble was importantly better than individual SVM for classification. Ten data mining methods were employed by Ou *et al.* [10] to predict value trends of Hang index from Hong Kong market. The methods involved Tree based classification, K-nearest neighbor, Bayesian classification, SVM and neural network. Results indicated that the SVM outperformed other predictive models. The value fluctuations by a developed Legendre neural network was forecasted by Liu *et al.* [11] by assuming investors' positions and their decisions by analyzing the prior data on the values. Indeed, they examined a random function (time strength) in the prediction model. Araújo *et al.* [12] proposed the morphological rank linear forecasting approach to compare its results with time-delay added evolutionary forecasting approach and multilayer perceptron networks.

From the above research background, it is clear that each of the algorithms can effectively solve stock prediction problems. However, it is vital to notice that there are specific limitations for each of them. The prediction results not only are affected by the representation of the input data but also

depend on the prediction method. Moreover, using only prominent features and identifying them as input data instead of all features can noticeably develop the accuracy of the prediction models.

Employing tree-based ensemble methods and deep learning algorithms for predicting the stock and stock market trend is a new area of research activities. In light of employing bagging and majority vote methods, Tsai *et al.* [13] used two different kinds of ensemble classifiers, such as heterogeneous and homogeneous methods. They also consider macroeconomic features and financial ratios from Taiwan stock market to examine the performance of models. The results demonstrated that with respect to the investment returns and prediction accuracy, ensemble classifiers were superior to single classifiers. Ballings *et al.* [14] compared the performance of AdaBoost, Random Forest and kernel factory versus single models involving SVM, KNN, Logistic Regression and ANN. They predict European company's prices for one-year ahead. The final results showed that Random Forest outperformed among all models. Basak *et al.* [15] employed XGBoost and Random Forest methods for the classification problem to forecast the stock increase or decrease based on previous values. Results showed that the prediction performances have advanced for several companies in comparison with the existing ones. For examining macroeconomic indicators to accurately predict stock market for one-month ahead, Weng *et al.* [16] improved four ensemble models, boosting regressor, bagging regressor, neural network ensemble regressor and random forest regressor. Indeed, another aim was employing a hybrid way of LSTM to prove that the macroeconomic features are the most successful predictors for stock market.

Moving on using deep learning algorithms, a deep neural network algorithm with the transaction records and public market data was investigated by Long *et al.* [17] to assess stock price trends. Their final results indicated that bidirectional LSTM could forecast the future of market for investors, and the technique attained the greatest performance. The employment of RNN and CNN algorithms was examined by Rekha *et al.* [18] to compare the accuracy of those with real values from stock markets. LSTM with an automatic encoder and LSTM with an embedded layer were utilized by Pang *et al.* [19] to acquire better stock market estimations. The result of experimental works indicated that LSTM with an embedded layer outperformed for the Shanghai composite index with 57.2% accuracy. The deep convolutional LSTM algorithm was employed by Kelotra and Pandey [20] to efficiently calculate stock market movements. They used a model with Rider-based monarch butterfly optimization method and gained the RMSE and MSE of 2.6923 and 7.2487. A forecasting LSTM model and an overfitting prevention LSTM module were suggested by Baek and Kim [21] to predict stock market. They showed that using the overfitting prevention module make results more accurate. Using a hybrid method of LSTM and GA was presented by Chung and Shin [22] to develop a new stock market prediction method. Their

results indicated that the method outperformed the benchmark model.

Overall, regarding the above literature, prior studies often concentrated on macroeconomic or technical features with recent machine learning methods to detect stock index or values movement without considering appropriate preprocessing methods.

Tehran's stock market has been greatly popular lately due to the remarkable growth of the main index in the last decade. The important reason behind that is privatizing most of the state-owned in the Iranian constitution firms under the general policies of article 44. The shares of lately privatized firms can be bought by ordinary people under particular conditions. The market has some special features compared to other country's stock markets; for example, dealing price limitation that is  $\pm 5\%$  of opening price for every index in each trading day. This matter hampers scatter market shocks and irregular market fluctuations, political matters, etc. over a particular time and could form the market smoother. However, the effect of fundamental parameters on the market is considerable and the prediction task of future movements is not easy [23].

This study employed stock market groups (that are important for traders) to investigate the task of predicting future trends. In spite of remarkable progress in Tehran stock market in the recent decade, there has been not adequate papers on the stock price predictions and trends via novel machine learning algorithms. However, a paper has been published recently by Nabipour *et al.* [23] where they employed tree-based models and deep learning algorithms to estimate future stock prices from 1 day ahead to 30 days ahead as a regression problem. The experimental results indicated that LSTM (as the superior model) could successfully predict values (from Tehran Stock Exchange) with the lowest error.

In this research, we concentrate on comparing prediction performance of nine machine learning models (Decision Tree, Random Forest, Adaboost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression and ANN) and two deep learning methods (RNN and LSTM) to predict stock market movement. Ten technical indicators are utilized as inputs to our models. Our study includes two different approaches for inputs, continuous data and binary data, to investigate the effect of preprocessing; the former uses stock trading data (open, close, high and low values) while the latter employs preprocessing step to convert continuous data to binary one. Each technical indicator has its specific possibility of up or down movement based on market inherent properties. The performance of the mentioned models is compared for the both approaches with three classification metrics, and the best tuning parameter for each model (except Naïve Bayes and Logistic Regression) is reported. All experimental tests are done with ten years of historical data of four stock market groups (petroleum, diversified financials, basic metals and non-metallic minerals), that are totally crucial for investors, from Tehran stock exchange. We believe that this study is a new research paper that incorporates multiple machine learning and deep learning methods

to improve the prediction task of stock groups' trend and movement.

This paragraph is organized to show the structure of our paper. Section 2 defines our research data with some statistical data, and two approaches supposed for input values. Eleven prediction models, including nine machine learning and two deep learning algorithms, are introduced and discussed in Section 3. The final results of prediction are presented in Section 4 with analyzing, and Section 5 concludes our paper.

## II. RESEARCH DATA

In this study, ten years of historical data of four stock market groups (petroleum, diversified financials, basic metals and non-metallic minerals) from November 2009 to November 2019 is employed, and all data is gained from www.tsetmc.com website. Figures 1-4 show the number of increase or decrease cases for each group during ten years.

In the case of predicting stock market movement, there are several technical indicators and each of them has a specific ability to predict future trends of market; however, we choose ten technical indicators in this paper based on previous studies [24]–[26]. Table 10 (in Appendix section) shows technical indicators and their formulas, and Table 11 (in Appendix section) indicates summary statistics of the

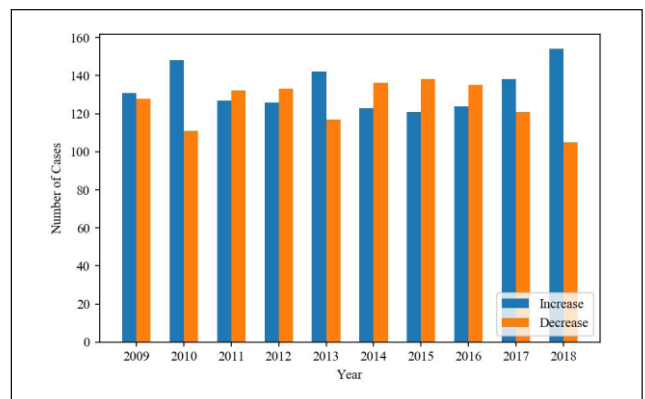


FIGURE 1. The number of increasing and decreasing cases (trading days) in each year for the diversified financials group.

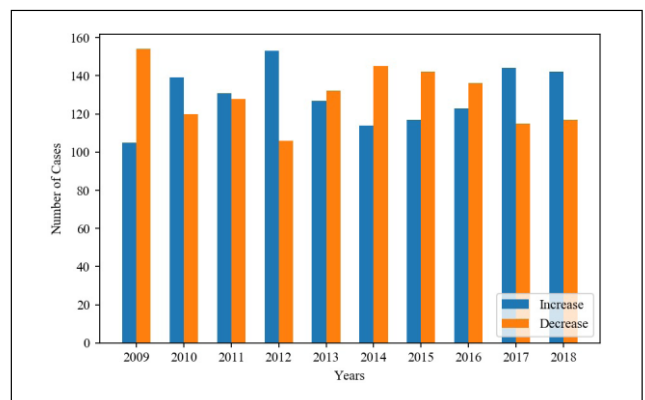
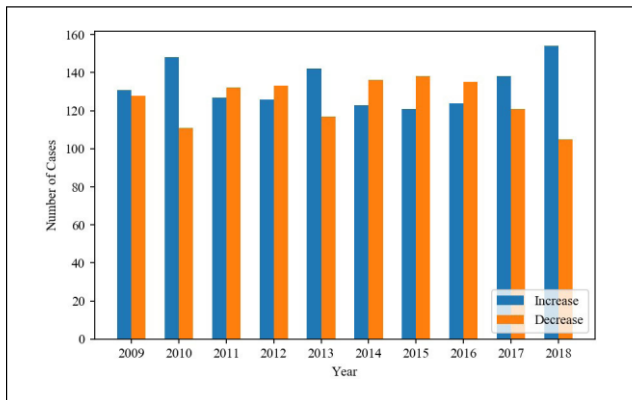
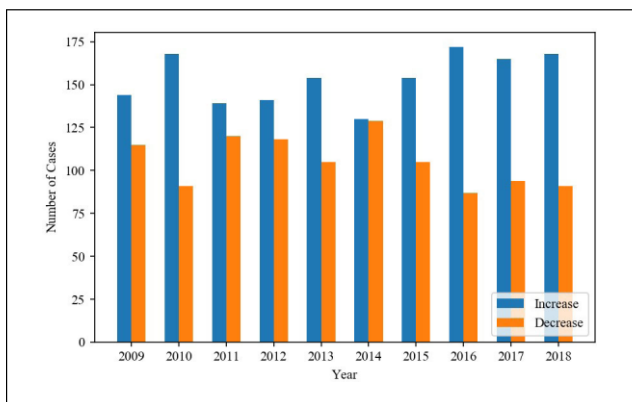


FIGURE 2. The number of increasing and decreasing cases (trading days) in each year for the petroleum group.



**FIGURE 3.** The number of increasing and decreasing cases (trading days) in each year for the diversified financials group.



**FIGURE 4.** The number of increasing and decreasing cases (trading days) in each year for the basic metals group.

indicators of four stock groups. The inputs for calculating indicators are open, close, high and low values in each trading day.

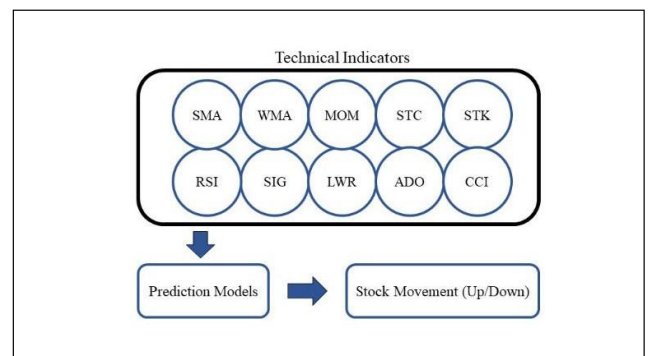
This paper involves two approaches for input information. continuous data is supposed to be based on actual time series, and binary data is presented with a preprocessing step to convert continuous data to binary one with respect to each indicator nature.

#### A. CONTINUOUS DATA

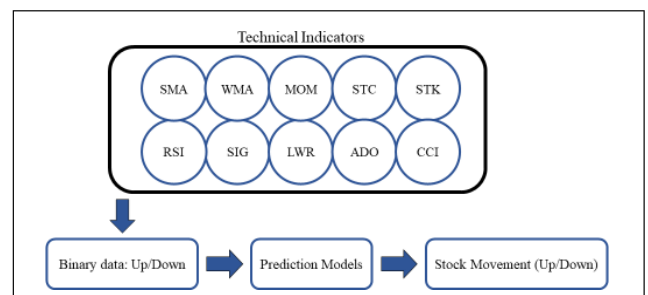
In this method, input values to prediction models are computed from formulas in Table 10 for each technical indicator. The indicators are normalized in the range of (0, +1) before using to prevent overwhelming smaller values by larger ones. Figure 5 shows the process of stock trend prediction with continuous data.

#### B. BINARY DATA

In this approach, a new step is added to convert continuous values of indicators to binary data based on each indicator's nature and property. Figure 6 indicates the process of stock trend prediction with binary data. Here, binary data is intro-



**FIGURE 5.** Predicting stock movement with continuous data.



**FIGURE 6.** Predicting stock movement with binary data.

duced by +1 as the sign of upward trend and -1 as the sign of downward trend.

Details about the way of calculating indicators are presented here [26]–[28]:

The average of values in a particular range computes SMA indicator, and this helps investors to decide if a price will go further in the same trend. WMA indicator provides us a weighted average of the last n prices, where the weighting falls with every previous price.

- SMA and WMA: if current value is below the moving average then the trend is -1, otherwise it is +1.

MOM indicator computes the speed of the fall or rise in values and it is a handy indicator of strength (or weakness) in estimating prices.

- MOM: if the value of MOM is positive then the trend is +1, otherwise it is -1.

STCK indicator is a momentum to liken a specific closing price of a stock to its range over a certain period. The oscillator's sensitivity to market movements is decreased by adjusting that period or by a moving average of values. The relative position of the closing values is calculated by STCD compared to the amplitude of price oscillations in a particular period. This is assumed that as prices grow, the closing price tends towards the prices which fit to the higher part of the area of price movements in the previous period and when prices drop, the reverse is true. LWR indicator is a kind of momentum indicator that estimates overbought and oversold levels. Occasionally LWR indicator is employed to discover entry and exit times for investors. MACD indicator

demonstrates the association between two moving averages of a stock's price as another kind of momentum indicator. Investors regularly use this to buy the stock after the MACD indicator goes beyond its signal line and sell the shares in the opposite situation. ADO indicators is typically employed to observe the flow of money into or out of stock. Investors ordinarily use ADO line to find buying or selling time of stock or confirm the strength of a movement.

- STCK, STCD, LWR, MACD and ADO: if the current value (time  $t$ ) is more than the previous price (time  $t-1$ ) then the trend is  $+1$ , otherwise it is  $-1$ .

RSI indicator is another momentum that assesses the magnitude of recent price changes to evaluate oversold or overbought situations for stock values. RSI indicator is indicated as an oscillator (a line graph that moves between two extremes) and changes between 0 to 100.

- RSI: its value is between 0 and 100. If the RSI value surpasses 70 then the trend is  $-1$ , and if the value goes below 30 then the trend is  $+1$ . For values between 30 and 70, if the current value (time  $t$ ) is larger than the prior value (time  $t-1$ ) then the trend is  $+1$ , otherwise it is  $-1$ .

CCI indicator is utilized as a momentum-based oscillator to decide on what time a stock price is getting a condition of being overbought or oversold. CCI indicator also estimates the difference between the history of average price and the current value. This indicator defines the time of entry or exit for investors by computing special signals.

- CCI: if values surpass 200 then the trend is  $-1$  and if values go below  $-200$  then the trend is  $+1$ . For values between  $-200$  and 200, if the current value (time  $t$ ) is larger than the prior value (time  $t-1$ ) then the trend is  $+1$ , otherwise it is  $-1$ .

### III. PREDICTION MODELS

In this study, we use nine machine learning methods (Decision Tree, Random Forest, Adaboost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression and ANN) and two deep learning algorithms (RNN and LSTM).

#### A. DECISION TREE

Decision Tree is a common supervised learning approach employed for both regression and classification problems. The goal of technique is forecasting a target by using easy decision rules shaped from the dataset and related features. Being easy to interpret or able to solve problems with different outputs are two advantages of using this model; on the contrary, constructing over-complex trees that cause over-fitting is a typical disadvantage. A schematic illustration of Decision Tree is indicated in Figure 7.

#### B. RANDOM FOREST

Great number of decision trees make a random forest model. The model basically averages the forecast result of trees, which is named a forest. Also, the algorithm includes three random ideas, selecting training data randomly when forming

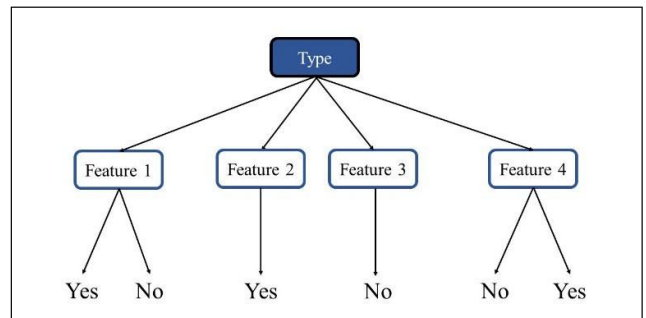


FIGURE 7. Schematic illustration of Decision tree [23].

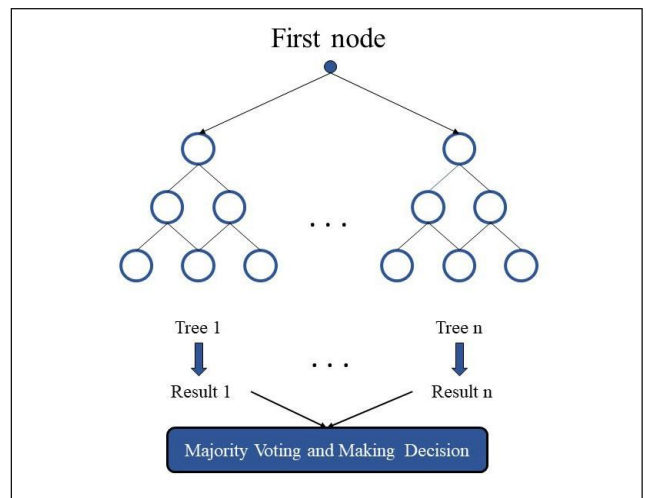


FIGURE 8. Schematic illustration of Random forest [23].

trees, randomly choosing some subsets of variables when dividing nodes and deeming only a subset of all variables for splitting every node in each basic decision tree. Every basic tree learns from a random sample of the dataset during the training process of a random forest. A schematic illustration of the model is shown in Figure 8.

#### C. ADABOOST

The process of converting some weak learners to a powerful one is named Boosting method. AdaBoost is a specific type of Boosting that is an ensemble model to progress the predictions of every learning technique. The goal of boosting is to train weak learners sequentially for adjusting their previous predictions. This model is a meta-predictor which starts by fitting a model on the basic dataset before fitting additional copies of it on the same dataset. During the process of training, samples' weights are modified based on the current forecasting error; therefore, the consequent model focuses on tough items.

#### D. XGBoost

XGBoost is a recent ensemble model based on decision trees. This employs the rules of Boosting for weak learners similarly. XGBoost was presented for better performance and

speed compared to other tree-based models. Regularization for preventing overfitting, In-built cross-validation capability, proficient handling of missing data, catch awareness, parallelized tree building and tree pruning are significant benefits of XGBoost method.

**E. SVC**

Support Vector Machines (SVMs) are a set of supervised learning approaches that can be employed for classification and regression problems. The classifier version is named SVC. The method's purpose is finding a decision boundary between two classes with vectors. The boundary must be far

from any point in the dataset, and support vectors are the sign of observation coordinates with a gap named margin. SVM is a boundary that best separates two classes with employing

a line or hyperplane. The decision boundary is defined in Equation 1 where SVMs can map input vectors  $x_i \in R^d$  into a high dimensional feature space  $\Phi(x_i) \in H$ , and  $\Phi$  is mapped by a kernel function  $K(x_i, x_j)$ . Figure 9 shows the schematic illustration of SVM method.

$$f(x) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \cdot K(x, x_i) + b\right) \quad (1)$$

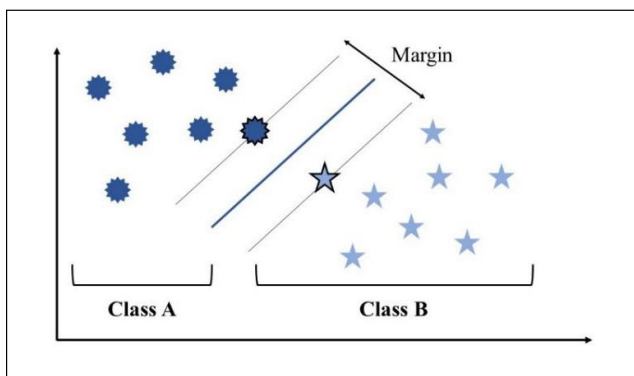
SVMs can perform a linear or non-linear classification efficiently, but for non-linear, they must use a kernel trick which map inputs to high-dimensional feature spaces. SVMs convert non-separable classes to separable ones by kernel functions such as linear, non-linear, sigmoid, radial basis function (RBF) and polynomial. The formula of kernel functions is shown in Equations 2-4 where  $\gamma$  is the constant of radial basis function and  $d$  is the degree of polynomial function. Indeed, there are two adjustable parameters in the sigmoid function, the slope  $\alpha$  and the intercepted constant  $c$ .

$$RBF : K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2)$$

$$Polynomial : K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (3)$$

$$Sigmoid : K(x_i, x_j) = \tanh(\alpha x_i^T x_j + c) \quad (4)$$

SVMs are often effective in high dimensional spaces and cases where the number of dimensions is greater than the



**FIGURE 9.** Schematic illustration of SVM.

number of samples, but to avoid over-fitting in selecting regularization term and kernel functions, the number of features should be much greater than the number of samples.

**F. NAÏVE BAYES**

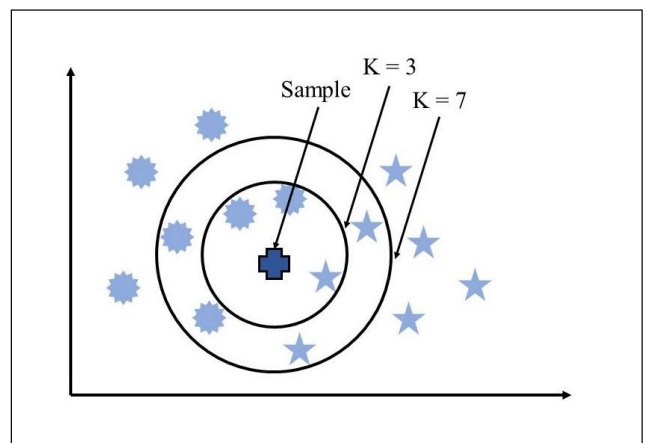
Naïve Bayes classifier is a member of probabilistic classifiers based on Bayes' theorem with strong independence assumptions between the features given the value of the class variable. This method is a set of supervised learning algorithms. The following relationship is stated in Equation 5 by Bayes' theorem where  $y$  is class variable, and  $x_1$  through  $x_n$  are dependent feature vectors.

$$P(y | x_1, \dots, x_n) = P(y) \prod_{i=1}^n \frac{P(x_i | y)}{P(x_1, \dots, x_n)} \quad (5)$$

Naive Bayes classifier can be highly fast in comparison with more sophisticated algorithms. The separation of the class distributions means that each one can be independently evaluated as a one-dimensional distribution. This in turn helps for alleviating problems from the dimensionality curse.

**G. KNN**

Two properties usually are suggested for KNN, lazy learning and non-parametric algorithm, because there is not any assumption for underlying data distribution by KNN. The method follows some steps to find targets: Dividing dataset into training and test data, selecting the value of  $K$ , determining which distance function should be used, choosing a sample from test data (as a new sample) and computing the distance to its  $n$  training samples, sorting distances gained and taking  $k$ -nearest data samples, and finally, assigning the test class to the sample on the majority vote of its  $k$  neighbors. Figure 10 shows the schematic illustration of KNN method.



**FIGURE 10.** Schematic illustration of KNN.

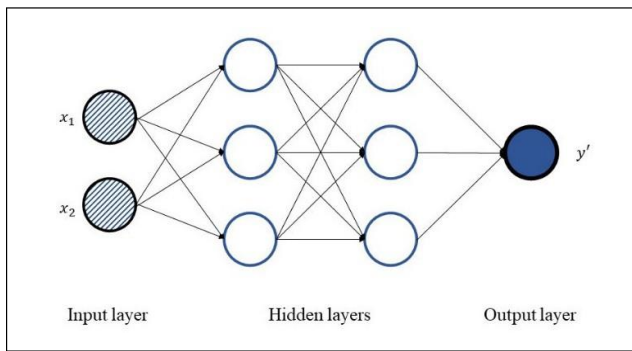
**H. LOGISTIC REGRESSION**

Logistic regression is used to assign observations to a separated set of classes as a classifier. The algorithm transforms its output to return a probability value with the logistic

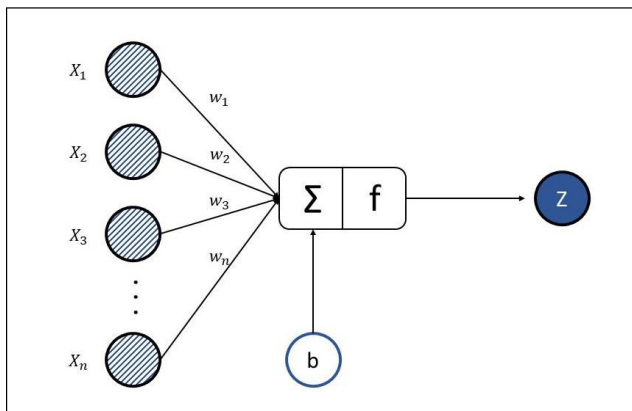
sigmoid function, and predicts the target by the concept of probability. Logistic Regression is similar to Linear Regression model, but the Logistic Regression employs sigmoid function, instead of logistic one, with more complexity. The hypothesis behind logistic regression tries to limit the cost function between 0 and 1.

**I. ANN**

ANNs are a prominent subset of machine learning algorithms that are usually single or multi-layer nets which fully connected together. Figure 11 is shown as an example of ANN with an output and input layer and also two hidden layers. Each node (in a layer) is connected to all other nodes (in the next layer). By the rise in the number of hidden layers, it is able to form the network deeper.



**FIGURE 11.** Schematic illustration of ANN [23].



**FIGURE 12.** An illustration of relationship between inputs and output for ANN [23].

Figure 12 indicates the relationship between inputs and output for ANNs. A node gets the weighted sum of the input values then add the result to a bias. A non-linear function commonly is used to calculate an outcome that is the output of the node, which makes another input for the next layer. The process of a network goes from the input layer to the output layer, and the final output is computed by implementing this method for all nodes in the network. The learning procedure

of weights and biases is linked with all nodes to train the neural network.

Equation 6 demonstrates the relationship between weights, biases and nodes. The weighted sum of inputs passed through a non-linear activation function from a layer to another one. It can be supposed as a vector, where n is the number of inputs for the final node, f is activation function,  $X_1, X_2 \dots$  and  $X_n$  are inputs,  $w_1, w_2 \dots$  and  $w_n$  are weights and z is the final output.

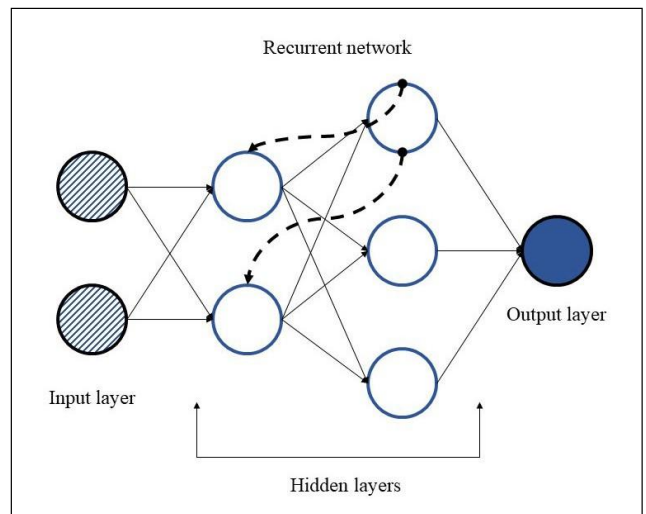
$$Z = f(x.w + b) = f\left(\sum_{i=1}^n x_i w_i + b\right) \quad (6)$$

The training process is finalized with some rules by computing weights and biases. Randomly initializing the weights and biases for each node, implementing a forward pass by the current weights and biases, computing each node output, comparing the final output with the real target, and adjusting the weights and biases consequently by gradient descent with the backpropagation technique.

**J. RNN**

A very significant version of neural networks is recognized as RNN, which is widely employed in different problems. In a typical neural network, the input passes through some layers, and output is created. It is proposed that two consecutive inputs are totally independent; however, the condition is not true in all processes. For instance, to forecast stock market at a certain period, it is vital to observe the prior samples.

RNN is named recurrent due to it does the same task for each item of a sequence when the output is related to the previous computed values. As another important point, RNN has a specific memory, which stores previous computed information for a long time. In theory, RNN can use information randomly for long sequences, but in real practices, there is a limitation to look back just a few steps. Figure 13 shows the architecture of RNN.



**FIGURE 13.** An illustration of recurrent network [23].

**K. LSTM**

LSTM is a particular type of RNN with an extensive range of uses such as document classification, time series analysis, voice and speech recognition. Opposite to feedforward networks, the predictions (created by RNNs) are dependent on prior estimations. In experimental works, RNNs are not applied broadly due to include a few lacks that result in impractical estimations.

Without investigation of too much detail, LSTM solves the problems by employing assigned gates for forgetting old information and learning new ones. LSTM layer is made of four neural network layers that interact in a specific method. A usual LSTM unit involves three different parts, a cell, an output gate and a forget gate. The main task of cell is recognizing values over random time intervals and the task of controlling the information flow into the cell and out of it belongs to the gates.

**L. MODELS' PARAMETERS**

Since stock market data are time-series information, there are two approaches for training dataset of prediction models. Because of the recurrent nature of RNN and LSTM models, the technical indicators of one or more days (up to 30 days) are considered and rearranged as input data to be fed into the models. For other models except RNN and LSTM, ten technical indicators are fed to the model. Output of all models is the stock trend value with respect to input data. For recurrent models, output is the stock trend value of the last day of the training sample.

All models (except Naïve Bayes) have one or several parameters known as hyper-parameters which should be adjusted to obtain optimal results. In this paper, one or two parameters of every model (except Decision Tree and Logistic Regression which fixed parameter(s) is used) is selected to be adjusted for an optimal result based on numerous experimental works. In Tables 1-3, all fixed and variable parameters of tree-based models, traditional supervised models, and neural-network-based models are presented, respectively.

**IV. EXPERIMENTAL RESULTS**

**A. CLASSIFICATION METRICS**

F1-Score, Accuracy and Receiver Operating Characteristics-Area Under the Curve (ROC-AUC) metrics are employed to evaluate the performance of our models. For Computing F1-score and Accuracy, Precision and Recall must be evaluated by Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These values are indicated in Equations 7 and 8.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

By calculation of above equations, F1-Score and Accuracy are defined in Equations 9 and 10.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

**TABLE 1.** Tree-based models parameters.

Model	Parameters	Value(s)
Decision Tree	Max Depth	10
Bagging Classifier	Max Depth	10
	Estimator	Decision Tree
	Number of Trees	50, 100, 150, ... , 500
Random Forest	Max Depth	10
	Number of Trees	50, 100, 150, ... , 500
Adaboost	Max Depth	10
	Estimator	Decision Tree
	Number of Trees	50, 100, 150, ... , 500
	Learning Rate	0.1
Gradient Boosting	Max Depth	10
	Number of Trees	50, 100, 150, ... , 500
	Learning Rate	0.1
XGBoost	Max Depth	10
	Number of Trees	50, 100, 150, ... , 500
	Objective	Logistic Regression for Binary Classification

**TABLE 2.** Traditional supervised models parameters.

Model	Parameters	Value(s)
SVC	Kernels	Linear, Poly (degree = 3), RBF, Sigmoid
Naïve Bayes	C	1.0
	Gamma	$1/((\text{num}_i \times \text{variance}_i)^{\gamma})$ f: features
	Algorithm	Gaussian
KNN Classifier	Number of Neighbors	1, 2, 3, ... , 100
	Algorithm	K-dimensional Tree
Logistic Regression	Weights	Uniform
	Leaf Size	30
	Metric	Euclidean Distance (L <sub>2</sub> )
	Tolerance	10 <sup>-7</sup>
Model	C	1.0
	Penalty	Euclidean Distance (L <sub>2</sub> )
	Parameters	Value(s)
SVC	Kernels	Linear, Poly (degree = 3), RBF, Sigmoid
	C	1.0
	Gamma	$1/((\text{num}_i \times \text{variance}_i)^{\gamma})$ f: features

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

Among classification metrics, Accuracy is a good metric, but it is not enough for all classification problems. It is often necessary to look at some other metrics to make sure that a model is reliable. F1-Score might be a better metric to employ if results need to achieve a balance between Recall and Precision, especially when there is an uneven class distribution.

**TABLE 3.** ANN, RNN and LSTM parameters.

ANN Parameters	
Parameters	Value(s)
Hidden Layer Neuron Count	20, 50, 100, 200, 500
Activation Function	ReLU, Sigmoid, Tanh
Optimizer	Adam: learning rate = 0.001 $\beta_1=0.9, \beta_2=0.999$
Training Stop Condition	Early stopping: Monitoring parameter = validation data accuracy Patience = 100 epochs
Max Epochs	10000
RNN and LSTM Parameters	
Parameters	Value(s)
Hidden Layer Neuron Count	500
Number of Training Days	1, 2, 5, 10, 20, 30
Neuron Type	RNN/LSTM
Activation Function	Tanh, Softmax
Optimizer	Adam: learning rate = 0.00005 $\beta_1=0.9, \beta_2=0.999$
Training Stop Condition	Early stopping: monitoring parameter = validation data accuracy patience = 100 epochs
Max Epochs	10000

ROC-AUC is another powerful metric for classification problems, and is calculated based on the area under ROC-AUC curve from prediction scores.

**B. RESULTS**

For training machine learning models, we implement the following steps: normalizing features (just for continuous data), randomly splitting the main dataset into train data and test data (30% of dataset was assigned to the test part), fitting the models and evaluating them by validation data (and “early stopping”) to prevent overfitting, and using metrics for final evaluation with test data. The creating deep models is different from machine learning when the input values must be three dimensional (samples, time\_steps, features); so, we use a function to reshape the input values. Also, weight regularization and dropout layer are employed to prevent overfitting here. All coding process in this study is implemented by python3 with Scikit Learn and Kears library.

Based on extensive experimental works by deeming the approaches, the following outcomes are obtained:

In the first approach, continuous data for the features is used, and Tables 4-6 show the result of this method. For each model, the prediction performance is evaluated by the three metrics. Also, the best tuning parameter for all models (except Naïve Bayes and Logistic Regression) is reported. For achieving a better image of experimental works, Figure 14 is made to indicate the average of F1-score based on average running time through the stock market groups. It can be seen that Naive-Bayes and Decision Tree are least accurate

**TABLE 4.** Tree-based models with best parameters for continuous data.

Stock Group	Prediction Model							
	Decision Tree				Random Forest			
	F1-score	Accuracy	ROC AUC	ntrees	F1-score	Accuracy	ROC AUC	ntrees
Div. Fin.	0.6993	0.6846	0.6838	1	0.7200	0.7218	0.7224	50
Metals	0.7164	0.6538	0.6347	1	0.7553	0.7077	0.6898	100
Minerals	0.6658	0.6513	0.6519	1	0.7464	0.7282	0.7271	100
Petroleum	0.6459	0.6641	0.6632	1	0.7042	0.7308	0.7288	250
	Adaboost				XGBoost			
	F1-score	Accuracy	ROC AUC	ntrees	F1-score	Accuracy	ROC AUC	ntrees
Div. Fin.	0.7266	0.7231	0.7205	250	0.7213	0.7167	0.7167	100
Metals	0.7553	0.7051	0.6904	250	0.7577	0.7064	0.6906	150
Minerals	0.7277	0.7064	0.7046	100	0.7196	0.7013	0.7005	50
Petroleum	0.7148	0.7218	0.7217	50	0.6964	0.7115	0.7107	250

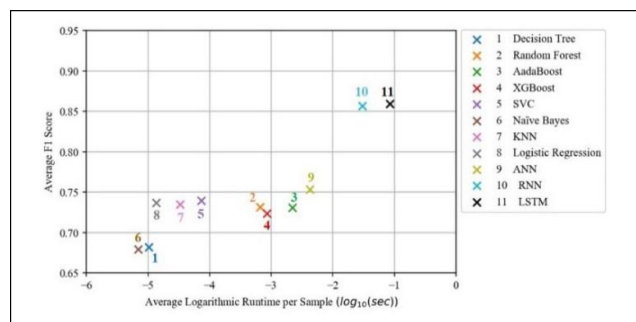
**TABLE 5.** Supervised models with best parameters for continuous data.

Stock Group	Prediction Model						
	SVC				Naïve Bayes		
	F1-score	Accuracy	ROC AUC	Kernel	F1-score	Accuracy	ROC AUC
Div. Fin.	0.7312	0.7154	0.7143	Poly	0.6866	0.6782	0.6780
Metals	0.7833	0.7269	0.7029	RBF	0.7223	0.6846	0.6819
Minerals	0.7529	0.7282	0.7248	Linear	0.6658	0.6692	0.6743
Petroleum	0.6917	0.7051	0.7045	RBF	0.6429	0.6795	0.6771
	KNN				Logistic Regression		
	F1-score	Accuracy	ROC AUC	Neighbors	F1-score	Accuracy	ROC AUC
Div. Fin.	0.7244	0.7141	0.7136	47	0.7321	0.7167	0.7136
Metals	0.7859	0.7359	0.7171	21	0.7710	0.7167	0.6965
Minerals	0.7353	0.7167	0.7415	41	0.7529	0.7282	0.7248
Petroleum	0.6929	0.7103	0.7092	17	0.6911	0.7077	0.7067

(approximately 68%) while RNN and LSTM are top predictors (roughly 86%) with a considerable difference compared to other models. Indeed, the running time of those superiors is more than other algorithms.

**TABLE 6.** Neural-network-based models with best parameters for continuous data.

Stock Group	Prediction Model											
	ANN				RNN				LSTM			
	F1-score	Accuracy	ROC AUC	Activation Func./epochs	F1-score	Accuracy	ROC AUC	ndays/epochs	F1-score	Accuracy	ROC AUC	ndays/epochs
Div. Fin.	0.7590	0.7500	0.7495	ReLU/245	0.8620	0.8643	0.8664	20/842	0.8638	0.8643	0.8664	20/773
Metals	0.7932	0.7359	0.7091	ReLU/90	0.8571	0.8282	0.8238	20/772	0.8781	0.8716	0.8770	5/398
Minerals	0.7671	0.7462	0.7437	ReLU/233	0.8810	0.8716	0.8770	5/402	0.8810	0.8716	0.8770	5/402
Petroleum	0.6932	0.7128	0.7116	Tanh/148	0.8279	0.8224	0.8221	10/373	0.8335	0.8314	0.8331	10/358

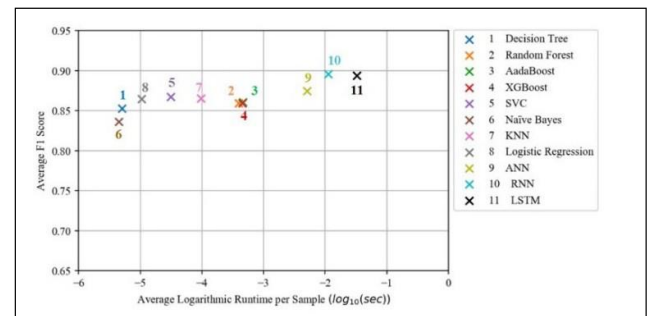


**FIGURE 14.** Average of F1-Score based on average logarithmic running per sample for continuous data.

In the second approach, binary data for the features is employed, and Tables 7-9 demonstrate the result of this way. The structure and experimental works here are similar to the first approach except inputs where we use an extra layer to convert continuous data to binary one based on the nature and property of the features. Similarly, for better understanding, Figure 15 is made to show the average of F1-score based on average running time through the stock market groups. It is

**TABLE 7.** Tree-based models with best parameters for binary data.

Stock Group	Prediction Model							
	Decision Tree				Random Forest			
	F1-score	Accuracy	ROC AUC	ntrees	F1-score	Accuracy	ROC AUC	ntrees
Div. Fin.	0.8421	0.8462	0.8460	1	0.8508	0.8538	0.8538	450
Metals	0.8738	0.8474	0.8364	1	0.8794	0.8513	0.8360	400
Minerals	0.8660	0.8667	0.8668	1	0.8671	0.8679	0.8680	100
Petroleum	0.8278	0.8346	0.8349	1	0.8402	0.8449	0.8457	150
	AdaBoost				XGBoost			
	F1-score	Accuracy	ROC AUC	ntrees	F1-score	Accuracy	ROC AUC	ntrees
Div. Fin.	0.8538	0.8564	0.8564	400	0.8523	0.8551	0.8551	50
Metals	0.8792	0.8513	0.8365	450	0.8788	0.8526	0.8403	50
Minerals	0.8674	0.8679	0.8680	300	0.8668	0.8679	0.8681	150
Petroleum	0.8413	0.8462	0.8470	50	0.8407	0.8436	0.8451	100



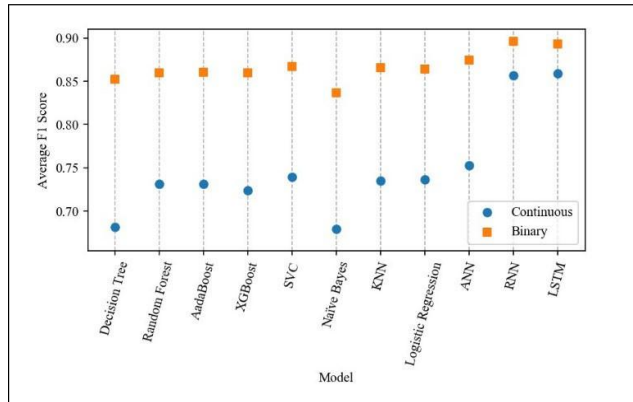
**FIGURE 15.** Average of F1-Score based on average logarithmic running per sample for binary data.

clear that there is a significant improvement in the prediction performance of all models in comparison with the first approach, and this achievement is obviously shown in Figure 16. There is no change in the inferior methods (Naive-Bayes and Decision Tree with roughly 85% F1-score) and the superior predictors (RNN and LSTM with approximately 90% F1-score), but the difference between them becomes less by binary data. Also, the prediction process for all models is faster in the second approach.

As a prominent result, deep learning methods (RNN and LSTM) show a technical skill to forecast stock movement in both approaches, especially for continuous data when the performance of machine learning models is so weaker than binary method. However, the running time of those is always more than others due to use large amount of epochs and prices from some days before.

**TABLE 8.** Supervised models with best parameters for binary data.

Stock Group	Prediction Model						
	SVC				Naïve Bayes		
	F1-score	Accuracy	ROC AUC	Kernel	F1-score	Accuracy	ROC AUC
Div. Fin.	0.8553	0.8590	0.8588	Linear	0.8351	0.8410	0.8406
Metals	0.8872	0.8679	0.8645	Poly	0.8466	0.8295	0.8354
Minerals	0.8721	0.8718	0.8718	Linear	0.8313	0.8372	0.8375
Petroleum	0.8544	0.8641	0.8630	Poly	0.8327	0.8423	0.8416
	KNN				Logistic Regression		
	F1-score	Accuracy	ROC AUC	Neighbors	F1-score	Accuracy	ROC AUC
Div. Fin.	0.8607	0.8551	0.8563	13	0.8526	0.8564	0.8562
Metals	0.8894	0.8641	0.8502	60	0.8837	0.8603	0.8510
Minerals	0.8649	0.8667	0.8668	27	0.8680	0.8667	0.8666
Petroleum	0.8473	0.8526	0.8532	21	0.8532	0.8641	0.8626



**FIGURE 16.** The average of F1-Score with continuous and binary data for all models.

Overall, it is obvious that the whole of algorithms predict well as they are trained with continuous values (up to 67%), but the models' performance is remarkably improved when they are trained with binary data (up to 83%). The result behind this improvement is interpreted as follows: an extra layer is employed in the second approach, and the duty of the layer is comparing each current continuous value (at time t) with previous value (at time t-1). So the future up or down trend is identified and when binary data is given as the input values to the predictors, we enter data with a recognized trend based on each feature's property. This critical layer is able to convert non-stationary values in the first approach to trend deterministic values in the second one, and algorithms must find the correlation between

**TABLE 9.** Neural-network-based models with best parameters for binary data.

Stock Group	Prediction Model						
	ANN				LSTM		
	F1-score	Accuracy	ROC AUC	Activation Func./epochs	F1-score	Accuracy	ROC AUC
Div. Fin.	0.8691	0.8756	0.8750	Sigmoid/111	0.8994	0.8986	0.8991
Metals	0.8925	0.8718	0.8645	Tanh/6	0.8801	0.8819	0.8871
Minerals	0.8733	0.8705	0.8704	Tanh/305	0.8900	0.8846	0.8842
Petroleum	0.8646	0.8731	0.8722	ReLU/19	0.8852	0.8910	0.8899
	RNN				LSTM		
	F1-score	Accuracy	ROC AUC	ndays/epochs	F1-score	Accuracy	ROC AUC
Div. Fin.	0.9024	0.9012	0.9016	5/68	0.8994	0.8986	0.8991
Metals	0.9011	0.8819	0.8872	5/233	0.8801	0.8819	0.8871
Minerals	0.8894	0.8897	0.8895	2/284	0.8900	0.8846	0.8842
Petroleum	0.8852	0.8936	0.8923	2/115	0.8852	0.8910	0.8899

input trends and output movement as an easier prediction task.

Despite careful tries to find valuable researches on the similar stock market, no important paper could be reported and compared; therefore, this deficiency is presented as a novelty of this study, which will be a baseline for future articles.

## V. CONCLUSIONS

The purpose of this study was the prediction task of stock market movement by machine learning and deep learning algorithms. Four stock market groups, namely diversified financials, petroleum, non-metallic minerals and basic metals, from Tehran stock exchange were chosen, and the dataset was based on ten years of historical records with ten technical features. Also, nine machine learning models (Decision Tree, Random Forest, Adaboost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression and ANN) and two deep learning methods (RNN and LSTM) were employed as predictors. We supposed two approaches for input values to models, continuous data and binary data, and we employed

**TABLE 10.** Selected Technical Indicators (n is 10 here) [23].

<p>Simple n-day moving average (SMA) = <math display="block">\frac{C_t + C_{t-1} + \dots + C_{t-n+1}}{n}</math></p> <p>Weighted 14-day moving average (WMA) = <math display="block">\frac{n \times C_t + (n-1) \times C_{t-1} + \dots + C_{t-n+1}}{n + (n-1) + \dots + 1}</math></p> <p>Momentum (MOM) = <math display="block">C_t - C_{t-n+1}</math></p> <p>Stochastic K% (STCK) = <math display="block">\frac{C_t - LL_{t-t-n+1}}{HH_{t-t-n+1} - LL_{t-t-n+1}} \times 100</math></p> <p>Stochastic D% (STCD) = <math display="block">\frac{K_t + K_{t-1} + \dots + K_{t-n+1}}{n} \times 100</math></p> <p>Relative strength index (RSI) = <math display="block">100 - \frac{100}{1 + \left( \frac{\sum_{i=1}^{n-1} UP_{t-i}}{\sum_{i=1}^{n-1} DW_{t-i}} \right)}</math></p> <p>Signal(n)<sub>t</sub> (SIG) = <math display="block">MACD_t \times \frac{2}{n+1} + Signal(n)_{t-1} * \left(1 - \frac{2}{n+1}\right)</math></p> <p>Larry William's R% (LWR) = <math display="block">\frac{HH_{t-t-n+1} - C_t}{HH_{t-t-n+1} - LL_{t-t-n+1}} \times 100</math></p> <p>Accumulation/Distribution oscillator (ADO) = <math display="block">\frac{H_t - C_t}{H_t - L_t}</math></p> <p>Commodity channel index (CCI) = <math display="block">\frac{M_t - SM_t}{0.015D_t}</math></p>
<p><b>While:</b></p> <p><math>C_t</math> is the closing price at time t  <math>L_t</math> and <math>H_t</math> is the low price and high price at time t respectively  <math>LL_{t-t-n+1}</math> and <math>HH_{t-t-n+1}</math> is the lowest low and highest high prices in the last n days respectively  <math>UP_t</math> and <math>DW_t</math> means upward price change and downward price change at time t respectively</p> <p><math>EMA(K)_t = EMA(K)_{t-1} \times \left(1 - \frac{2}{k+1}\right) + C_t \times \frac{2}{k+1}</math></p> <p>Moving average convergence divergence (MACD)<sub>t</sub> = <math>EMA(12)_t - EMA(26)_t</math></p> <p><math>M_t = \frac{H_t + L_t + C_t}{3}</math></p> <p><math>SM_t = \frac{\sum_{i=0}^{n-1} M_{t-i}}{n}</math></p> <p><math>D_t = \frac{\sum_{i=0}^{n-1}  M_{t-i} - SM_t }{n}</math></p>

**TABLE 11.** Summary statistics of indicators [23].

Feature	Max	Min	Mean	Standard Deviation
<b>Diversified Financials</b>				
SMA	6969.46	227.5	1471.201	1196.926
WMA	3672.226	119.1419	772.5263	630.0753
MOM	970.8	-1017.8	21.77033	126.5205
STCK	99.93224	0.159245	53.38083	19.18339
STCD	96.9948	14.31843	53.34332	15.28929
RSI	68.96463	27.21497	50.18898	6.471652
SIG	310.5154	-58.4724	16.64652	51.62368
LWR	99.84076	0.06776	46.61917	19.18339
ADO	0.99986	0.000682	0.504808	0.238426
CCI	270.5349	-265.544	14.68813	101.8721
<b>Basic Metals</b>				
SMA	322111.5	7976.93	69284.11	60220.95
WMA	169013.9	4179.439	36381.48	31677.51
MOM	39393.8	-20653.8	1030.265	4457.872
STCK	98.47765	1.028891	54.64576	16.41241
STCD	90.93235	12.94656	54.64294	13.25043
RSI	72.18141	27.34428	49.8294	6.113667
SIG	12417.1	-4019.14	803.5174	2155.701
LWR	98.97111	1.522349	45.36526	16.43646
ADO	0.999141	0.00097	0.498722	0.234644
CCI	264.6937	-242.589	23.4683	99.14922
<b>Non-metallic Minerals</b>				
SMA	15393.62	134.15	1872.483	2410.316
WMA	8081.05	69.72762	985.1065	1272.247
MOM	1726.5	-2998.3	49.21097	264.0393
STCK	100.00	0.154268	54.71477	20.2825
STCD	96.7883	13.15626	54.68918	16.37712
RSI	70.89401	24.07408	49.67247	6.449379
SIG	848.558	-127.47	37.36441	123.9744
LWR	99.84573	-2.66648	45.28523	20.2825
ADO	0.998941	0.00036	0.501229	0.238008
CCI	296.651	-253.214	20.06145	101.9735
<b>Petroleum</b>				
SMA	1349138	16056.48	243334.2	262509.8
WMA	707796.4	8580.536	127839.1	138101
MOM	227794	-136467	4352.208	26797.25
STCK	100.00	0.253489	53.78946	22.0595
STCD	95.93565	2.539517	53.83312	17.46646
RSI	75.05218	23.26627	50.02778	6.838486
SIG	71830.91	-33132	3411.408	11537.98
LWR	99.74651	-1.8345	46.23697	22.02162
ADO	0.999933	0.000288	0.498381	0.239229
CCI	286.7812	-284.298	14.79592	101.8417

three classification metrics for evaluations. Our experimental works showed that there was a significant improvement in the performance of models when they use binary

data instead of continuous one. Indeed, deep learning algorithms (RNN and LSTM) were our superior models in both approaches.

## APPENDIX

See Tables 10 and 11.

## REFERENCES

- [1] J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Penguin, 1999.
- [2] T. Turner, *A Beginner's Guide To Day Trading Online*, 2nd ed. New York, NY, USA: Simon and Schuster, 2007.
- [3] H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, M. M. Selim, and K. Muhammad, "A local and global event sentiment based efficient stock exchange forecasting using deep learning," *Int. J. Inf. Manage.*, vol. 50, pp. 432–451, Feb. 2020.
- [4] W. Long, Z. Lu, and L. Cui, "Deep learning-based feature engineering for stock price movement prediction," *Knowl.-Based Syst.*, vol. 164, pp. 163–173, Jan. 2019.
- [5] J. B. Duarte Duarte, L. H. Talero Sarmiento, and K. J. Sierra Juárez, "Evaluation of the effect of investor psychology on an artificial stock market through its degree of efficiency," *Contaduría y Administración*, vol. 62, no. 4, pp. 1361–1376, Oct. 2017.
- [6] Lu, Ning, *A Machine Learning Approach to Automated Trading*. Boston, MA, USA: Boston College Computer Science Senior, 2016.
- [7] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of HMM, ANN and GA for stock market forecasting," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 171–180, Jul. 2007.
- [8] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.
- [9] J. Sun and H. Li, "Financial distress prediction using support vector machines: Ensemble vs. Individual," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2254–2265, Aug. 2012.
- [10] P. Ou and H. Wang, "Prediction of stock market index movement by ten data mining techniques," *Modern Appl. Sci.*, vol. 3, no. 12, pp. 28–42, Nov. 2009.
- [11] F. Liu and J. Wang, "Fluctuation prediction of stock market index by legendre neural network with random time strength function," *Neurocomputing*, vol. 83, pp. 12–21, Apr. 2012.
- [12] C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, "Predicting stock returns by classifier ensembles," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2452–2459, Mar. 2011.
- [13] R. D. A. Araãjo and T. A. E. Ferreira, "A Morphological-Rank-Linear evolutionary method for stock market prediction," *Inf. Sci.*, vol. 237, pp. 3–17, Jul. 2013.
- [14] M. Ballings, D. Van den Poel, N. Hespels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, Nov. 2015.
- [15] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers," *North Amer. J. Econ. Finance*, vol. 47, pp. 552–567, Jan. 2019.
- [16] B. Weng, W. Martinez, Y.-T. Tsai, C. Li, L. Lu, J. R. Barth, and F. M. Megahed, "Macroeconomic indicators alone can predict the monthly closing price of major U.S. indices: Insights from artificial intelligence, time-series analysis and hybrid models," *Appl. Soft Comput.*, vol. 71, pp. 685–697, Oct. 2018.
- [17] J. Long, Z. Chen, W. He, T. Wu, and J. Ren, "An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in chinese stock exchange market," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106205.
- [18] G. Rekha, D. Bhanu Sravanthi, S. Ramasubbareddy, and K. Govinda, "Prediction of stock market using neural network strategies," *J. Comput. Theor. Nanoscience*, vol. 16, no. 5, pp. 2333–2336, May 2019.
- [19] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *J. Supercomput.*, vol. 76, no. 3, pp. 2098–2118, Mar. 2020.
- [20] Kelotra, A. and P. Pandey, "Stock market prediction using optimized deep-convLSTM model," *Big Data*, vol. 8, no. 1, pp. 5–24, 2020.
- [21] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Syst. Appl.*, vol. 113, pp. 457–480, Dec. 2018.
- [22] H. Chung and K.-S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, p. 3765, 2018.
- [23] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, and E. Salwana, "Deep learning for Stock Market Prediction," *Entropy*, vol. 22, no. 8, p. 840, Aug. 2020.
- [24] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5311–5319, May 2011.
- [25] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2162–2172, Mar. 2015.
- [26] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, Jan. 2015.
- [27] R. Majhi, G. Panda, B. Majhi, and G. Sahoo, "Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 10097–10104, Aug. 2009.
- [28] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017.

•••