


# An Ensemble of Random Forest Gradient Boosting Machine and Deep Learning Methods for Stock Price Prediction

Lokesh Kumar Shrivastav, University School of Information, Communication, and Technology, Guru Gobind Singh Indraprastha University, New Delhi, India

 <https://orcid.org/0000-0002-7403-2887>

Ravinder Kumar, Shri Vishwakarma Skill University, India

## ABSTRACT

Stochastic time series analysis of high-frequency stock market data is a very challenging task for the analysts due to the lack availability of efficient tools and techniques for big data analytics. This has opened the door of opportunities for the developer and researcher to develop intelligent and machine learning-based tools and techniques for data analytics. This paper proposed an ensemble for stock market data prediction using the three most prominent machine learning-based techniques. The stock market dataset has a raw data size of 39364 KB with all attributes and processed data size of 11826 KB having 872435 instances. The proposed work implements an ensemble model comprises of deep learning, gradient boosting machine (GBM), and distributed random forest techniques of data analytics. The performance results of the ensemble model are compared with each of the individual methods (i.e., deep learning, gradient boosting machine [GBM], and random forest). The ensemble model performs better and achieves the highest accuracy of 0.99 and lowest error (RMSE) of 0.1.

## KEYWORDS

Big Data Analysis, Deep Learning, Distribute Random Forest, Ensemble Learning, Gradient Boosting Machine, Stock Market

## 1. INTRODUCTION

The accurate forecast or prediction of stock prices is specially focused issue for the investors and companies listed in the stock market. Non-stationary and non-linear time-series nature of stock prices makes the prediction results very complex and challenging (Cavalcante et al., 2016). Financial time series analysis is a very important source of information for stock market prediction (Oztekin et al., 2016). Finding hidden patterns is the requirement for analysis and prediction of the stock price actuations. The pre-assumption as given by very famous hypothesis Random Walk (Malkiel, B. G., 2003, Mankiw & Shapiro, 1985) and the Efficient Market (Jensen M. C., 1978) are stating that it is impossible to predict the nature of the stock market due to presence of randomness and nonlinearity in the dataset. These assumptions were verified by many different pursuance models in different interval of time (Atsalakis & Valavanis, 2009). The risk in investment into the stock market lies in the fact that the stock market price series are very dynamic, non-correlated, chaotic and noisy in nature. Therefore, the accurate prediction of stock prices is very crucial from the investor point of view as well as company point of view to maximize gains on the investments. Recent advancement

DOI: 10.4018/JITR.2022010102

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

in the field of soft computing has captured the attention of the researcher to analyze and predict the non-linear behavior of stock market in highly noisy environment.

Machine Learning Frameworks is usually deployed to forecast the price of the volatile stock market at the optimum level of the accuracy (Kumar et. al 2013a, 2013b). For this purpose, high frequency big data has been used for the experiments and estimation of accuracy. The volume, velocity, and variety of stock market datasets have tremendously increasing day by day. Therefore, it becomes the need of the day to develop a tool or a model to predict the behavior of the stock market under such a high volatility. The tree-based ensemble using machine learning techniques have achieved the popularity among the best available statistical model and the most efficient deep learning model.

This paper proposed an ensemble model that comprise of the Deep Learning, Gradient Boosting Machine (GBM) and Random Forest (RF) model. It is obvious from the literature that the Gradient Boosting Machine and Random Forest has already been combined to form an ensemble model. The performance of the proposed ensemble model is compared with the individual models as discussed below.

Deep learning is the well-known supervised machine learning model that provides generalization, training and stability with the stochastic big dataset. It is based on feed-forward neural architecture and results the highest prediction accuracy (Rusk N., 2015). In this study, a supervised deep learning model is used to optimize the predictive result.

Gradient Boosting Machine (GBM) model is an ensemble machine learning technique used to build predictive tree-based models (Friedman, J. H., 2002). Gradient boosting is an approach where new models are developed to predict the residuals or errors of prior models and then added together to make the final prediction.

Distributed Random Forest (DRF) has gained popularity as a powerful classification and regression tool to be used in stock market data analytics (Khaidem & Dey, 2016). DRF works by generating a classification forest of regression or classification trees as oppose to a single regression tree. Each tree individually is a weak learner and built a class of columns and rows. Variance can be optimized if the tree is more in number. The final value of the prediction is calculated by computing the average predicted values over all the trees.

In Ensemble machine learning methods, the multiple learning algorithms are used to obtain the enhanced prediction performance as compared to single learning algorithms (Zhang & Ma, 2012). Most of the latest popular prediction tools available in the literature use an ensemble technique for making the prediction. This study implements a unique ensemble of the most prominent models of Deep Learning, Gradient Boosting Machine (GBM) and Random Forecast (RF), where Gradient Boosting Machine (GBM) are both already an ensemble learners use boosting and bagging respectively. An ensemble model keeps a collection of weak learners and produces a single and strong learner model.

## **1.1 Motivation and Contribution:**

Analysis of high frequency stochastic big data is a challenging task where the prediction accuracy is rarely satisfactory. The accuracy in the prediction of the stock market is directly affects the gain and loss to the investors, so it is desirous to have accurate prediction in order to maximize gain and minimize losses. The usual analytics tools and techniques will not work as the stock market dataset has the characteristics of big data. This has motivated us to develop an ensemble of best of its class machine learning techniques for high frequency data prediction or forecasting.

The major contributions of this paper are as follows:

1. Proposed and develop an ensemble of best of its class machine learning techniques for high frequency data prediction or forecasting.
2. A comparative analysis of results obtained using proposed ensemble with other machine learning based prediction methods like: Deep Learning, Gradient Boosting Machine (GBM) and Distributed Random Forest (DRF).

3. The proposed methodology enables the organization to develop the prediction systems for predictive analysis of stock prices.

The rest of the paper is organized in the following Sections: Section II presents a review of significant and latest review of literature in the domain. Section III presents the proposed methodology. Section IV demonstrates the experimental setup and presents the discussion on the results. Finally, Section V presents the conclusion and future research directions.

## 2. REVIEW OF LITERATURE

Available relevant literature over a decade has proved the nonlinear behavior of the stock market data. Different researchers have used different techniques and tools to get more reliable and optimized prediction results. Recent studies show, that the combined and hybrid techniques are providing better results as compared to a single analysis model for low-frequency datasets.

A model was developed (Adebiyi et al., 2014) to study the comparative analysis of Autoregressive Integrating Moving Average (ARIMA) and Artificial Neural Network (ANN) model. These two models were individually applied on thirteen years (approx) dataset of Dell Stock Exchange, where first was statistical modeling and second was machine learning technique respectively. The Forecast Error of experimental result suggest that ANN model is better than the famous statistical ARIMA (1, 0, 0) model.

A very first hybrid model was developed on the belief that the single model can't optimized the result (Hnaity & Abbod, 2015). The study proposed five hybrid model as Ensemble Empirical Model Decomposition-Neural Network (EEMD-NN), EEMD-Bagging-NN, EEMD Cross-Validation-NN, EEMD-Cross-Validation-Bagging and Ensemble Empirical Model Decomposition-Neural Network (EEMD-NN-Proposed) with the application of Genetic Algorithm (GA). These single and ensemble both models were applied on daily stock close dataset collected in 30 years (approx) and collected from FTSE100. The result concluded that EEMD –NN-Proposed had highest accuracy and lowest RMSE: 25.31 than any of single or combined models used in the study.

A single model strategy was suggested (Arévalo et al., 2016) that was based on Deep Neural Network strategy for real time evaluation. The study defined three parameters as Log Return, Pseudo Log Return and Trade Indicator and formulated and calculated these all terms. By the use of Deep Neural Network, the study predicts the next one minute Pseudo-Log-Return. The used architecture was chosen arbitrary which has one input layer, five hidden layer and one output layer. The Deep Neural Network was trained after every 50 epochs. This model concludes the evaluation with 66% of accuracy by Deep Learning.

A comparative analysis (Singh & Srivastava, 2017) was done with few famous machine learning frameworks as Deep Learning, Backpropagation Neural Network, Extreme Learning Machine and Radial Basis Neural Network to find the best model for the prediction of stock market. The CSI 300 future contract (IF1704) single minute dataset listed in Shanghai and Shenzhen Stock Exchange with three different sizes as small scale, medium scale and large scale has taken for the analysis. The training and testing dataset was partitioned into 90:10 ratios. The experimental result of the study concluded as Deep Learning is comparative better than the other three models. It also concludes that the performance estimation increases according to the size of samples.

An important comparative analysis (Chen et al., 2018) was done with 25 years and 9 months' datasets which was collected from S&P 500 index constituted from Thomson Reuters in the period of Dec-1989 to Sep-2015. Four very important models were applied as Long Short Term Network (LSTN), Deep Neural Network, Random Forest and Logistic Network models. The test case result of these models as Diebold and Mariano (DM), Pesaran Zimmermann (PT) and probability estimation of LSTN testing parameters concluded that LSTN is the better than the rest of the models.

A new model was introduced (Fischer & Krauss, 2018) that is based on the modification of traditional SVR named as Adaptive Support Vector Regression. Three types of data (5 minute: 1-Feb-2017 to 28-Feb-2017, 30 minute 1-Jan-2017 to 28-Feb-2017 and Daily: Date of listing to 31-Mar-2017) were collected (SH600006, SH6000016, SH6000036, SH6000056 from Shanghai Stock Exchange) and utilized for the analysis proposed model. To maintain the adaptive nature of the stock market, the Particle Swarm Optimization (PSO) is added in SVR to make it an adaptive SVR (ASVR). The performance of the model was compared with the two conventional models as Back Propagation Neural Network (BPNN) and Support Vector Regression (SVR) by the use of RMSE, MAPE and MAD. The RMSE of SH06 is 0.66 with BPNN, 0.75 with SVR and 0.65 with ASVR. The behavior of the ASVR is slightly better than its close competitor in the all experimented cases.

An automated software model (Guo et al., 2018) was developed to predict the stock market returns without any human intervention. The dataset was collected from 495 stocks listed in Shenzhen Growth Enterprise Market of 6 years and 9 months (approx.) with daily stock price in the time span of 25-Jan-2010 to 1- Oct-2016. The proposed classification model was implemented in WEKA by the use of Random Forest. The result was compared with SVM, ANN and kNN and found the proposed system was giving slightly better result than others in terms of Prediction Duration (PD), Return of the trade (ROT).

A novel model was proposed (Basak et al., 2019) to implement mixture of Deep Learning + (2D<sup>2</sup>). The dataset was collected from Google Stock Multimedia of NASDAQ for 2843 working days in the period of 19-Aug-2004 to 10-Dec-2015 to predict the stock return. The result of the proposed model DNN was 17.1% better than RBFNN and 43.4% better than the RNN in terms of deviation between the real and predicted value. But the model was not giving good result in terms of total return and RMSE compared with RBFNN.

Tree-based models (Zhang et al., 2018) were first time applied to the selection of technical indicator and used as a feature on a high-frequency dataset of Apple and Facebook. The data was collected from beginning to 3<sup>rd</sup> February 2017 with size 10 kB to 700 kB and 1180 to 10700 instances. The study applied the two most eminent methods as Random Forest (RF) and XGBoost on these datasets. The study finds the XGBoost model is the best model among the rest of models with an accuracy of 78%. The model accuracy can be improved by the use of other tree-based model or ensemble learning models.

A new model is proposed (Long et al., 2019) for high frequencies dataset of single minute of CSI 300 stock. The dataset was collected in period of 24-Dec-2013 to 7-Dec-2016 (approx. 3 years). The proposed model Multi Filter Neural Network (MFNN) applied on this high frequency 30 set dataset. The experimental result suggests the proposed model was 6.28% better than their close competitor RNN and CNN. The results of the model were also compared with LSTN, SVM, Logistic Regression, Random Forest and Linear Regression and found that the proposed model is far better than these traditional models by the use of performance estimators as Total Return, Return Rate, and Rate of Average Access Return etc.

In spite of huge research in the area, no researchers are able to produce a single established model that can optimize and precise model of computationally-intelligent-system for the stock market prediction. In addition to this, the applied dataset was either low frequency or it was taken for a short interval of time. Table 1 explores the summary of the literature review. These works had not been able to address the issue related to the accurate prediction for high frequency dataset. So, this study proposes an ensemble of the most famous machine learning models that can be applied on the high-frequency big dataset to open the new dimension of the analysis and prediction.

### 3. PROPOSED METHODOLOGY

The proposed work uses the H<sub>2</sub>O package to implement the models that is ideal for big data processing and provides optimum results with minimum resources. The three most reliable and result oriented

models as deep learning, gradient boosting machine and distributed random forest were selected to obtain the maximized predictive results. The prediction accuracy of proposed models is computed using Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Root Mean Square Logarithmic Error (RMSLE), Mean Residual Deviance (MRD) and  $R^2$ . Results obtained are further compared with the results of these three individual models to find the best model (<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/performance-and-prediction.html>).

Mean Squared Error (MSE) is the average squared difference between the real value and predicted value. It measures the quality of prediction and used for Gaussian distribution where the value closer to zero is better.

$$MSE = \frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2 \quad (1)$$

The Root Mean Square Error (RMSE) is evaluating parameter that decides how a model is behaving to capture the targeted result. The Root mean square is inversely proportional to the wellness of the model. It means lower RMSE value gives a better model.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (r - p)^2} \quad (2)$$

Mean Absolute Error (MAE) calculates the absolute difference between the real value and predicted value. It is a common error in the time series analysis and the value near to zero is better.

$$MAE = \frac{\sum_{i=1}^n |p_i - r_i|}{n} \quad (3)$$

Root Mean Squared Logarithmic Error (RMSLE) computes the ratio of a log of real and predicted values.

$$RMSLE = \sqrt{\frac{\sum_{i=1}^n \left( \ln \left( \frac{r_i + 1}{p_i + 1} \right) \right)^2}{n}} \quad (4)$$

Mean Residual Deviance (MRD) measures the goodness of fitness of a model and it is used in quintile distributions. The smaller positive real number is better.

$R^2$  is another evaluating parameter that explores the correlation between the real and the predicted datasets that grows in terms of unison. It varies between 0 and 1 where 0 means no correlation and 1 means the complete correlation between the real and the predicted dataset.

$$R^2 = 1 - \frac{SS_{reg}}{SS_{tot}} \quad (5)$$

The following notations variables were adopted and used in this work:

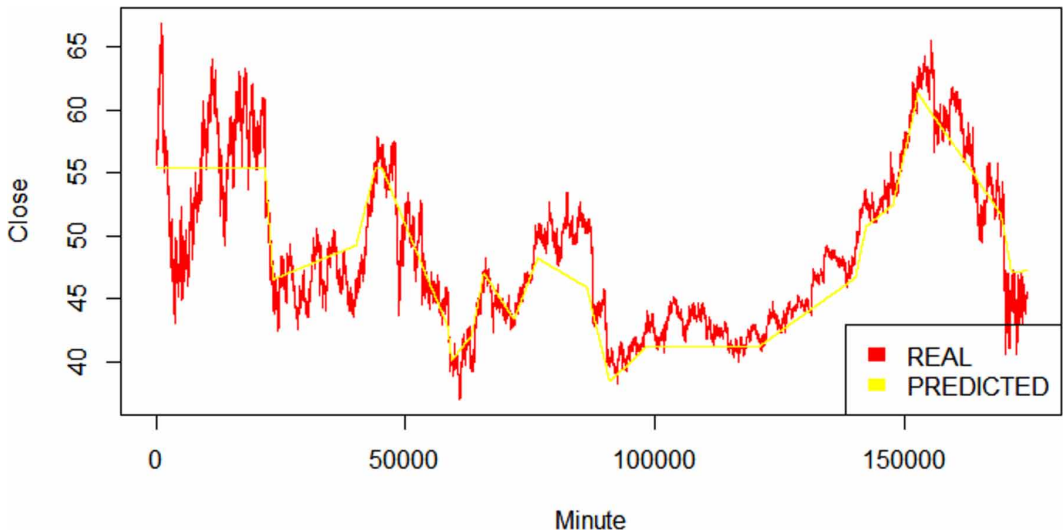
**Table 1. A summary of literature review on stock price forecast**

| References               | Source of dataset   | Targeted output          | Size of dataset or instances   | Time span and frequency of dataset   | Applied models  | Performance metrics  |
|--------------------------|---|--------------------------|--|--|---|--|
| Adebiyi et al., (2009)   | Dell Stock Exchange of NYSE   | Closing Price Prediction | 13 years (approx)<br>5680 instances  | 17-Aug-1988 to 25-Feb-2011<br>Closing daily  | ARIMA, ANN  | Forecast Error (FE)  |
| Al-Hnaity et al., (2015) | FTSE100 Index   | Closing Price Prediction | 30 years (approx)<br>Training: 7307<br>Testing: 250                                      | 2-Apr-1984 to 28-Feb-1914, Closing daily   | EEMD NN, EEMD Bagging NN, EEMD Cross Validation NN, EEMD Cross Validation Bagging and EEMD Bagging NN | MSE, RMSE  |
| Nino et al., (2016)      | TAQ database of NYSE (AAPL tick-by-tick transaction)                            | Stock Return             | 2 months (approx.)<br>1 minutes (high frequency dataset)                                 | 2-Sep-2008 to 7-Nov-2008   | DNN   | MSE, Directional Accuracy                                      |
| Singh et al., (2017)     | Google Stock Multimedia of NASDAQ   | Stock Return             | 2843 working days  | 19-Aug-2004 to 10-Dec-2015   | DNN + (2D <sup>2</sup> )  | R <sup>2</sup> , RMSE, MAPE etc.                               |
| Chen et al., (2018)      | CSI 300 future contract (IF1704) listed in Shanghai and Shenzhen Stock Exchange | Opening price prediction | Single minute high frequency dataset taken for small scale, medium scale and large scale | 20-Feb-2017 to 20-Apr-2017   | DNN, Backpropagation Neural Network, Extreme Learning Machine and Radial Basis Neural Network         | RMSE, MAPE, Directional Predictive Accuracy (DA)               |
| Fischer et al., (2018)   | S&P 500 index constituted from Thomson Reuters                                  | Stock Return             | 25 year 9 months (daily dataset)   | Dec-1989 to Sep-2015   | LSTN, DNN, Random Forest and Logistic Regression  | DM, PT, Statistical Estimation for probability of LSTN         |
| Guo et al., (2018)       | SH600006, SH6000016, SH6000036, SH6000056 from Shanghai Stock Exchange          | Stock Return             | Three types of data as 5 minutes, 30 minutes and daily data                              | 5 minute: 1-Feb-2017 to 28-Feb-2017, 30 minute 1-Jan-2017 to 28-Feb-2017<br>Daily: Listing to 31-Mar-2017, | BPNN, SVR<br>Adaptive SVR   | RMSE, MAPE, MAD  |
| Basaka et al., (2018)    | Apple and Facebook stock Return   | Stock Return             | 10kb-700kB in size   | Date of listing to 3-Feb 2017  | Random Forest, XGBoost  | Accuracy, Recall, Precision, Specificity, F-Score, Brier & AUC |
| Zang et al., (2018)      | 495 stocks listed in Shenzhen Growth Enterprise Market                          | Close price prediction   | 6 years and 9 months<br>Daily stock Price  | 25-Jan-2010 to 1-Oct-2016  | Xuanwu  | Prediction Duration (PD), Return of the trade                  |
| Long et al., (2019)      | CSI 300   | Stock Return             | 3 years (approx.)<br>High Frequency (1-min)  | 24-Dec-2013 to 7-Dec-2016  | MFNN = DNN + (2D <sup>2</sup> )<br>feature extraction   | Total Return, Return Rate, Rate of Average Access Return etc   |
| Proposed Work            | Coca Cola listed in New York Stock Exchange (NYSE)                              | Stock Return             | 8 Lacks (approx.)<br>High Frequency (1-min)  | 3-Jan -2000 to 31-Dec-2008 (1-Minute)  | DNN, GBM, DRF, Ensemble Learning  | MSE, RMSE, MAE, RMSLE,MRD, R <sup>2</sup>                      |

ARIMA: Autoregressive Integrated Moving Average; ANN: Artificial Neural Network, EEMD-NN: Ensemble Empirical Model Decomposition-Neural Network, EEMD-NN-Proposed: Ensemble Empirical Model Decomposition-Neural Network, DNN: Deep Neural Network, LSTM: Long Short Term Memory, BPNN: Backpropagation Neural Network, SVR: Support Vector Regression, GBM: Gradient Boosting Machine, DRF: Distributed Random Forest, rRMSE: Relative RMSE, NMSE: Normalized MSE, MI: Mutual Information, DM: Diebold and Mariano Testing, PT: Pesaran Timmermann Testing, RMSE: Root Mean Square Error, MAPE: Mean Absolute Percentage Error, MAD: Mean Absolute Deviation

n: total samples; r: real sample value; p: predicted value; m: mean of real sample value;  $SS_{reg}$ : the residual sum of square;  $SS_{tot}$ : the total sum of the square. The flow chart of the proposed work is shown in Figure 1 with two evaluating parameters that commonly used for all.

Figure 1. Flow chart of the proposed model



### 3.1 Data Pre-Processing

For Deep Learning and Gradient Boosting Machine, Random Forest and Ensemble Learning, H<sub>2</sub>O auto-handles the missing and categorical data without any intervention where the per column summaries of the parsed frame keeps the column type data.

### 3.2 Feature Extraction From Data

The original data has six attributes as date, time, open, high, low and close. These all attribute are independent and uncorrelated in nature. The data was collected on the basis of a single minute, so the combination of date and time will give a unique sequential number that is index or minute and the particular close will be dependent on this index value or minute value. So, in the present study, the index or sequential minute was added as a new feature. For this study, only index (minute) and close value of the particular minute is considered for the further processing.

### 3.3 Machine Learning Algorithms

#### 3.3.1 Deep Learning Model

Deep learning is one of the most powerful computational model that is a combination of the many processing layers and capable to capture the data with multi-levels of the abstraction (Chong et al., 2017). It finds the intricate structure that presents in the dataset by the use of the back-propagation algorithm and ensures to change inside parameter of present depends on previous, for the betterment of the targeted model. The model developed by the H<sub>2</sub>O is purely supervised learning, fast and memory efficient model (Candel et al., 2016). The model was already used in the same domain by

many researchers on low-frequency datasets. The algorithm is already discussed in the previous work with H<sub>2</sub>O package development.

### 3.3.2 Gradient Boosting Machine

Gradient Boosted Machine is an ensemble model and it is very much capable to handle the regression task. It is easy to interpret with the adaptability characteristics that provide very precise results (Dietterich & Kong 1995). The predictor value can be produce in every iteration that is based on previous iteration and ensure to provide the optimal result by the use of average weight. In every stage, overall performance can be boosted by the use of invoking an additional classifier. The modified version of boosting can be classify as non linear classification that optimize the accuracy of the tree without effecting its speed. It provides easily distributable and parallelizable feature with effortless environment for model tuning and selection. This version of Gradient Boosting Machine (GBM) that is capable to handle the bigdata with optimal accuracy, is rararly used in the stock market prediction domain. But the efficiency of the model is very significant in the current senerio of big data. The modified Gradient Boosting Model designed (<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html>) and developed (Malohlava & Candel 2017) by H2O explained as follows.

Algorithm 2: Gradient Boosting Machine

1. Initialization  $f_{k0}=0, k=1, 2, \dots, K$

2. Repeat  $m=1$  to  $M$

$$a. P_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}} \text{ for all } k=1, 2, \dots, K$$

b. Repeat  $k=1$  to  $K$

i. Calculate  $r_{ikm} = y_{ik} - p_k(x_i), i=1, 2, \dots, N$

ii. Fit regression tree to the targets  $r_{ikm}, i=1, 2, \dots, N$

$$iii. \text{ Calculate } r_{ikm} = \frac{K-1}{K} \left( \frac{\sum_{x_i \in R_{jkm}} r_{ikm}}{\sum_{x_i \in R_{jkm}} |r_{ikm}| (1 - |r_{ikm}|)} \right), \text{ where } j=1, 2, \dots, J_m$$

$$iv. \text{ Upgrade } f_{km}(x) = f_{k,m-1} + \sum_{j=1}^{J_m} r_{ikm} I(x \in R_{jkm})$$

3. Result  $f_k(x) = f_{kM}(x), \text{ where } k=1, 2, \dots, K$

### 3.3.3 Distributed Random Forest

The Distributed Random Forest (DRF) is a very competent technique of the regression and classification (Geurts & Wehenkel, 2006). This modified version has much more capacity to train categorical variables (Guillame-Bert & Teytaud, 2018). The computation of the algorithm is divided into workers and manager. The communication between the worker and manager can be done by the use of network. For the particular dataset, DRF creates a forest of regression tree in parallel manner. In this, the splitting rule (described in algorithm) is based on the most discriminative thresholds that are selected from random subset of candidate feature. It uses bagging function to build the individual tree. The modified algorithm (Niculescu-Mizil, & Caruana, 2005, <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drff.html>) is based on H<sub>2</sub>O package as described here:

**Algorithm 3(a): Distributed Random Forest (Finding optimal super split function):**

repeat (a, y, i) in q(j)



```

h = sampletwonode(i)
is h closed node then continue
is candidatefeature(j, h, p) false then continue
b= bag(i, p)
is b = 0 then also continue
t= (a + vh)/2
s' = score( t)
is s' > sh then
sh = s'
th = t
end
Hh=label(y) weighted by b
vh = a
end repeat
return( th, sh)
where
Hh : h is label leaf of histogram already traversed h∈[1,l]
vh:: last tested threshold for the leaf h∈[1,l].
q(j): list sorted as the attribute j
th: best threshold of leaf h∈[1,l] i.e. initially null.
sh: score of th=0
Algorithm 3(b): Distributed Random Forest (Training of dataset):
1. Create a decision tree that has only a root node.
// The root is the only open leaf of the tree.
2. Perform to initialize the mapping starting from sample index to
node index.
// Consequently all samples will be assigned to the particular
root.
3. Perform optimal supersplit function.
//Each splitter returns a partial optimal supersplit and the
optimal supersplit is picked as global by the tree builder by
analyzing the result of the splitters.
4. Modify the structure of the tree by replacing optimal
supersplit.
5: Perform to find the best supersplit by evaluation of the
splitters.
6. Update the mapping from sample index to node index by the use
of active leaves calculation.
7. Replace the evaluation-conditions to all the splitters for
further updates from sample index to node .index.
8. Stop growth of leaves if not enough records available for
close.
9. If one leaf remains open at least then go to step 3.
10. Pass this optimal Distributed Random Tree to the manager.
It provides average multiple decision trees that are based different random samples of rows and
columns.

```

### 3.3.4 Ensemble Machine Learning Model

The concept of ensemble learning was developed in 1992 and modified as super learner in 2007. The proposed ensemble machine learning model or super learner that finds the optimal hybridization of

prediction model by the use of boosting, bagging and stacking methods (Sagi & Rokach, 2018).). Boosting is a process to increase accuracy, decrease variance, flexible in use and applied in Gradient Boosting Machine. It is, not robust against noisy data. Bagging is a process that also increase accuracy, reduces variance and applied in Random Forest. It is robust against the noisy data. Stacking is the process to make a mixture model of strong learners and produce an optimal combination by the use of Meta learner algorithm. This is a supervised machine learning model that can be used in any kind of classification and regression problems. The algorithm of the ensemble machine learning model by H2O (<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html>) is as follows:

Algorithm 4: Ensemble Learning

Input: List of L base models that is selected as the base model.

//In the proposed ensemble considers deep Learning, gradient boosting machine and random forest are takes as base models.

1. Set up an ensemble machine model.

a. Specify a list of L base models

//Base Models: Deep Learning, Gradient Boosting Machine and Random Forest

b. Specify a Meta learning model.

//It is an automatic machine learning process in which the optimal hybrid model can be produced.

2. Train the ensemble model

a. Train each of the L base algorithms on the training set.

//Each individual model will be trained separately with the provided datasets to capture the individual performance.

b. Perform k-fold cross-validation on the each individual model.

//Each individual model trained with the k-fold cross validation and the result will be collected.

c. Perform to collect N x L matrix from cross-validated and predicted values from the individual model.

//N x L matrix can be formed where N is training set of all individual model and L is the list of the models used.

d. Train the Meta learning algorithm on the available level-one data provided in N x L matrix.

//The "ensemble machine model" that is the combination of the L base machine models is ready for the prediction on new dataset.

3. Predict on new data.

a. Perform ensemble model predictions,

//At first all individual prediction will be performed from the individual base model.

b. Feed these predictions to ensemble machine mode.

//Now the Metal earner or ensemble machine model is ready to predict result on the new testing dataset.

### 3.4 Framework And Experimental Setup

Experiments have been performed on Coca Cola stock dataset listed in New York Stock Exchange (NYSE) on core i7 processor with 2.50 GHz speed using H<sub>2</sub>O package in R-studio. A high frequency stock market data is collected from 03-Jan-2000, 9:38am to 31-Dec-2008 15:59pm in fine time interval of minute. The volume of data is very high i.e. 872435 instances and 11826 KB in size. The data is collected under the headings "Index", "Date", "Time", "Open", "High", "Low" and "Close" attributes. In this proposed work only two attributed have been used namely index as minute and close for prediction. The combination of the date and time forms a unique identification inside the

**Table 2. Summary of dataset used in this work**

| Minute           | Close           |
|------------------|-----------------|
| Min. : 1         | Min. : 37.02    |
| 1st Qu. : 217953 | 1st Qu. : 43.93 |
| Median : 436269  | Median : 47.22  |
| Mean : 436138    | Mean : 48.70    |
| 3rd Qu. : 654173 | 3rd Qu. : 52.62 |
| Max. : 872434    | Max. : 66.88    |

dataset, can be represented by Index or Minute. For the smooth extraction of features pre-processing mechanism is used minimize the irrelevant or blank data to produce more accurate and precise result. R-Studio is used to handle the blank data or the data with 'NA' value. Summary is a general purpose function in R-language that completely analyses the central tendencies of the datasets as minimum, quartile where 1st (lower or 25% quartile and 3rd (upper or 75% quartile), median, mean and max of datasets as shown in the Table 2.

Out of given six attributes of the dataset, Index and Close attributes are chosen for analysis and prediction for the sake of simplicity and ease of understanding. The dataset is partitioned into two parts, where first part is used for training and second part is used for testing the modal. The training dataset has size of 697949 rows and 2 columns and testing dataset has size of 174486 rows and 2 columns. The datasets used for experiment and analysis is independent dataset, it means there are no correlations among adjacent dataset. It is observed that the increment or decrement of any value does not disturb the other values.

## 4. EXPERIMENTAL RESULTS AND DISCUSSION

The results of the regression models are shown in Figure 2-5 and its comparison of performance estimation are given in Table 2. The model was developed with default parameter where fold value five is taken. In general, the accuracy of the model is directly proportional to its  $R^2$  values and inversely proportional to its RMSE values.

### 4.1 Results of Deep Learning

The experimental result confirms that deep learning with Gaussian distribution and with given fold, is not able to capture the nature the dataset in good manner. The performance parameter of the Deep Learning also explores deviation from the real dataset where Root Mean Square Error (RMSE) is 2.531699 and  $R^2$  value of 0.8339508 is obtained. The results are presented in the Figure 2 and Table 3 respectively.

### 4.2 Results of Gradient Boosting Machine

Ensemble Gradient Boosting Machine is better model to capture the nature of nonlinear kind of dataset. Root Mean Square Error (RMSE): 0.8464796,  $R^2$ : 0.9811871 and its predictive capability are presented in Figure 3 and Table 3 respectively.

### 4.3 Results of Random Forest model

Distributed Gradient Boosting Machine is the most adequate model to capture the dataset. The predictive capability as well as it performance parameters (Root Mean Square Error (RMSE): 0.8464796 and  $R^2$ : 0.9811871) are Figure 4 and Table 3 respectively.

Figure 2. Prediction results using Deep Learning (Real Close vs. Predicted Close)

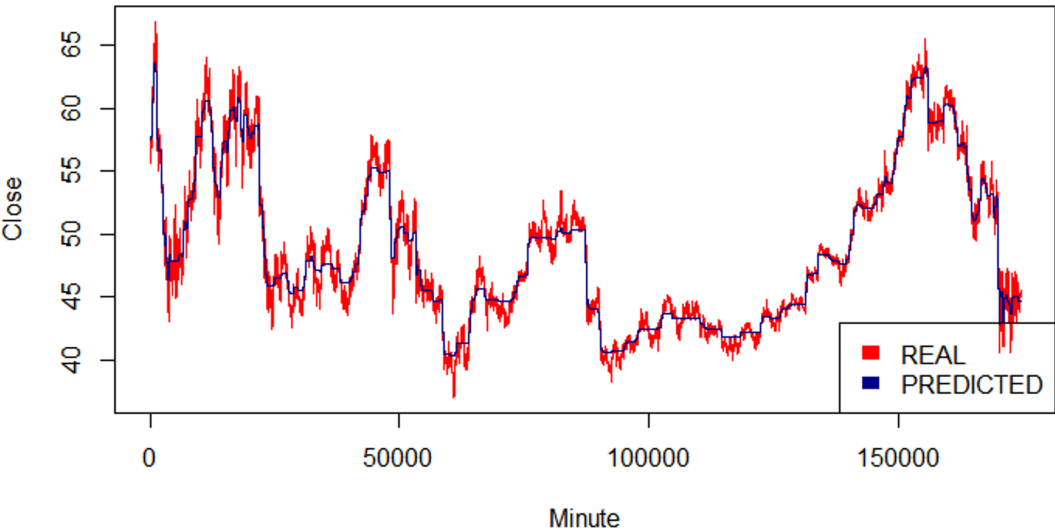
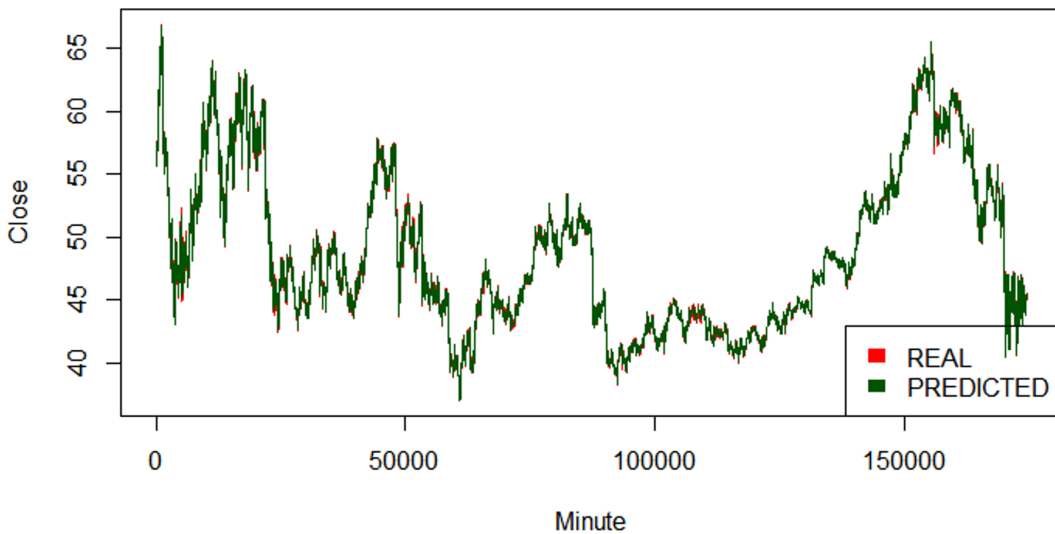


Figure 3. Prediction results using Gradient Boosting Machine (Real Close vs. Predicted Close)



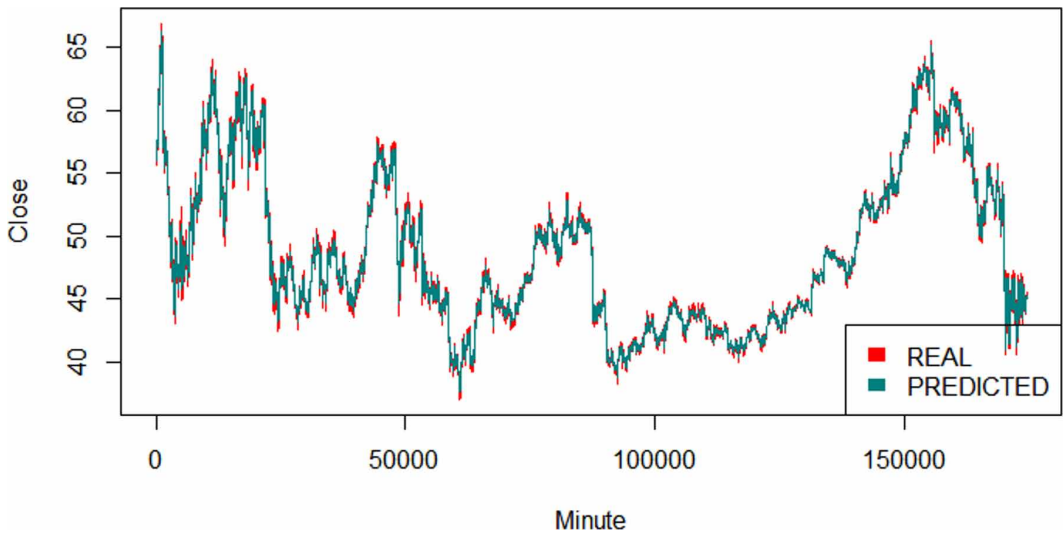
#### 4.4 Results of Ensemble model

The Ensemble model is the combination of these all three models. So, its predictive capability as well as performance parameter (RMSE: 0.175822, R2: 0.9992056) both are better than Deep Learning and GBM but comparable with Random Forest model as presented in the Figure5 and Table 3 respectively.

#### 4.4 Performance comparison

This section presents the comparative analysis of outcomes of the well-known models discussed in the literature (with same preprocessing). The outcome is also compared with the performance of well-known algorithms like Deep Learning and Gradient Boosting Machine (GBM), Distributed Random Forest (RF) and Ensemble Learning Model. All these models were trained by using “Gaussian

Figure 4. Prediction results obtained using Distributed Random Forest (Real Close vs. Predicted Close)



distribution” at particular fold in H<sub>2</sub>O package. The result shows that the prediction capability of the Ensemble Learning Model is far better than Deep Learning Model, as well as that of Gradient Boosting Machine but is comparable with that of Random Forest. The result of the study and proposed work presented in this paper were analyzed in terms of MSE (Mean-Square Error), RMSE (Root Mean-Square Error), MAE (Mean Absolute Error), RMSLE (Root Mean Squared Logarithmic Error) and R<sup>2</sup> used for all three model (refer Table 1, 2, 3). Five different models outputs have been used for the comparative analysis (Minute, real Close, Deep Close (prediction using Deep Learning model), GBM Close (predicted value produced by GBM), DF Close (prediction using Distributed Random Forest Model RF) and the Ensemble Close (prediction by the Ensemble Learning model) on testing dataset. It is clear from the observations that the “RF close” and “Ensemble Close” are very close to the “real Close” as shown in Figure 5 and Table 4.

The prediction results obtained using all four models have been compared against the actual values and is shown in Figure 6. The red line shows the actual value of the stock data, yellow line represents the prediction using Deep Learning, dark green line shown the prediction results obtained using GBM and dark blue line indicates the result obtained using RF and finally the dark cyan represents the prediction results of the proposed Ensemble Learning model.

The comparison analysis of the Root Mean Square and R-Square values of all the four models is presented in Figure 7 and 8. The yellow bar represents the Deep model, whereas dark green bar represents the result obtained using GBM model, dark blue bar presents DRF and dark cyan shows the output obtained using Ensemble learning models. It is also observed that the output obtained using DRF and Ensemble have the comparable RMSE and R-square value.

The prediction results (in terms of R-Square) obtained using proposed method are compared against the model developed by Basak et. al, 2018. The results thus obtained shows that the proposed method is far better than that of the model proposed by Basak in the literature

## 5. CONCLUSION AND FUTURE RESEARCH SCOPE

The present work establishes a direction for the analysis of stochastic high-frequency big data, which is rarely utilized in the terms of the stock market prediction. The proposed ensemble model utilizes the most prominent Deep Learning, Gradient Boosting Machine (GBM) and Distributed Random

Table 3. Comparative Performance evaluation Parameter (Deep Learning, GBM, RF, Ensemble Learning)

| Performance Parameters | Deep Learning | Gradient Boosting Machine (GBM) | Random Boosting (RF) | Ensemble Learning |
|------------------------|---------------|---------------------------------|----------------------|-------------------|
| MSE                    | 6.4095        | 0.7165277                       | 0.003300014          | 0.03091336        |
| RMSE                   | 2.531699      | 0.8464796                       | 0.05744575           | 0.175822          |
| MAE                    | 1.773624      | 0.6315757                       | 0.0345164            | 0.1293437         |
| RMSLE                  | 0.04884439    | 0.01707818                      | 0.00115312           | 0.003549565       |
| Mean Residual Deviance | 6.4095        | 0.7165277                       | 0.003300014          | 0.03091336        |
| R <sup>2</sup>         | 0.8339508     | 0.9811871                       | 0.9999149            | 0.9992056         |

Forest Model (DRFM) for stock market data prediction. It is clear from the validation results that the ensemble model (minimum RMSE and highest R-square value that is approx. 1) is the far better than of the Deep Learning and GBM. But the DRF performance is comparable with the proposed model. This is indicated that the predicted results obtained using Ensemble and DRF are very close to the original values of the stock prices. It is evident from the outcomes that an ensemble of the tree-based model has far better capability to predict high-frequency big data and it contradicts the perception of unpredictability of stock market price. This is recommended that ensemble models are very effective tools for the prediction and generalization of high frequency big data.

Figure 5. Prediction results obtained using Ensemble Learning Model (Real Close vs. Predicted Close)

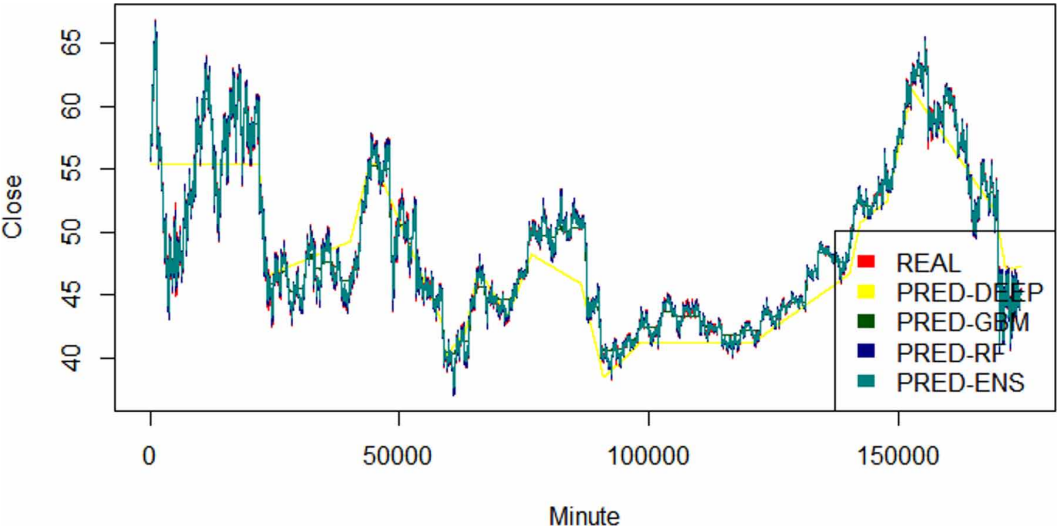


Figure 6. Comparative analysis of Prediction (Deep Learning, GBM, DRF, and Ensemble)

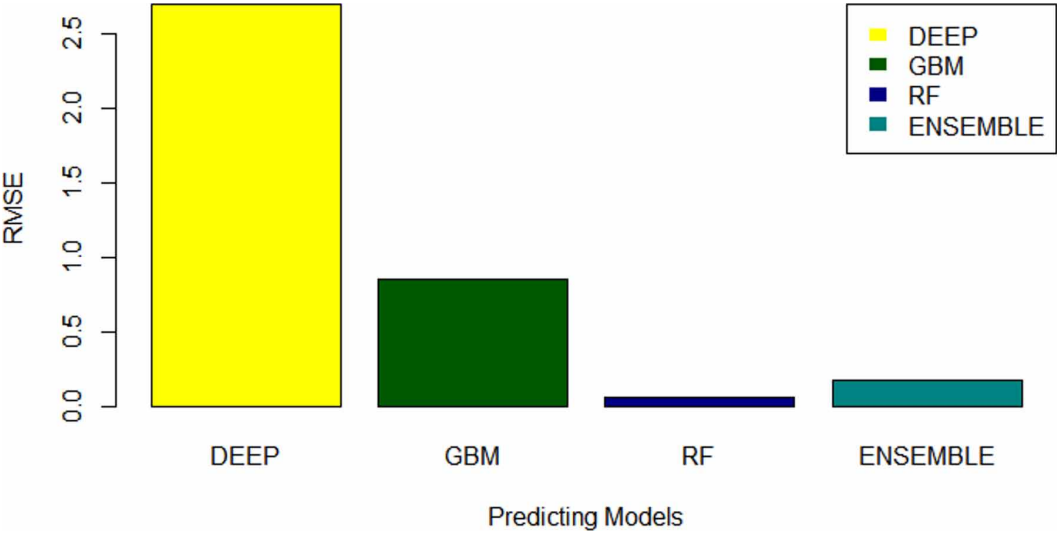


Table 4. Real vs. predicted result captured for 20 days

| Index | Minute | Real Close | PRED-DEEP | PRED-GBM | PRED-RF  | PRED-ENSEMBLE |
|-------|--------|------------|-----------|----------|----------|---------------|
| 1     | 9      | 57.5000    | 63.17802  | 57.53813 | 57.53375 | 57.57667      |
| 2     | 25     | 57.0625    | 63.16585  | 57.53813 | 57.22500 | 57.32958      |
| 3     | 27     | 57.2500    | 63.16433  | 57.53813 | 57.19000 | 57.30157      |
| 4     | 30     | 57.1250    | 63.16205  | 57.53813 | 57.22875 | 57.33258      |
| 5     | 31     | 57.3125    | 63.16129  | 57.53813 | 57.35375 | 57.43262      |
| 6     | 40     | 57.3125    | 63.15444  | 57.53813 | 57.34859 | 57.42849      |
| 7     | 45     | 57.3750    | 63.15064  | 57.53813 | 57.26749 | 57.36358      |
| 8     | 51     | 57.1250    | 63.14608  | 57.53813 | 57.26155 | 57.35883      |
| 9     | 55     | 57.2500    | 63.14304  | 57.53813 | 57.24813 | 57.34808      |
| 10    | 56     | 57.2500    | 63.14228  | 57.53813 | 57.19125 | 57.30257      |
| 11    | 63     | 57.1250    | 63.13695  | 57.53813 | 57.07375 | 57.20853      |
| 12    | 66     | 57.0625    | 63.13467  | 57.53813 | 57.02125 | 57.16652      |
| 13    | 67     | 57.0625    | 63.13391  | 57.53813 | 57.01625 | 57.16251      |
| 14    | 68     | 57.0000    | 63.13315  | 57.53813 | 57.05750 | 57.19553      |
| 15    | 69     | 57.0625    | 63.13239  | 57.53813 | 57.05875 | 57.19653      |
| 16    | 80     | 56.8750    | 63.12402  | 57.47822 | 56.82688 | 56.99871      |
| 17    | 89     | 56.8750    | 63.11718  | 57.47822 | 56.87000 | 57.03322      |
| 18    | 94     | 56.7500    | 63.11338  | 57.47822 | 56.83500 | 57.00521      |
| 19    | 95     | 56.7500    | 63.11262  | 57.47822 | 56.78375 | 56.96420      |
| 20    | 96     | 56.8125    | 63.11186  | 57.47822 | 56.78000 | 56.96119      |

Figure 7. Comparative RMSE (Deep Learning, GBM, RF, and Ensemble)

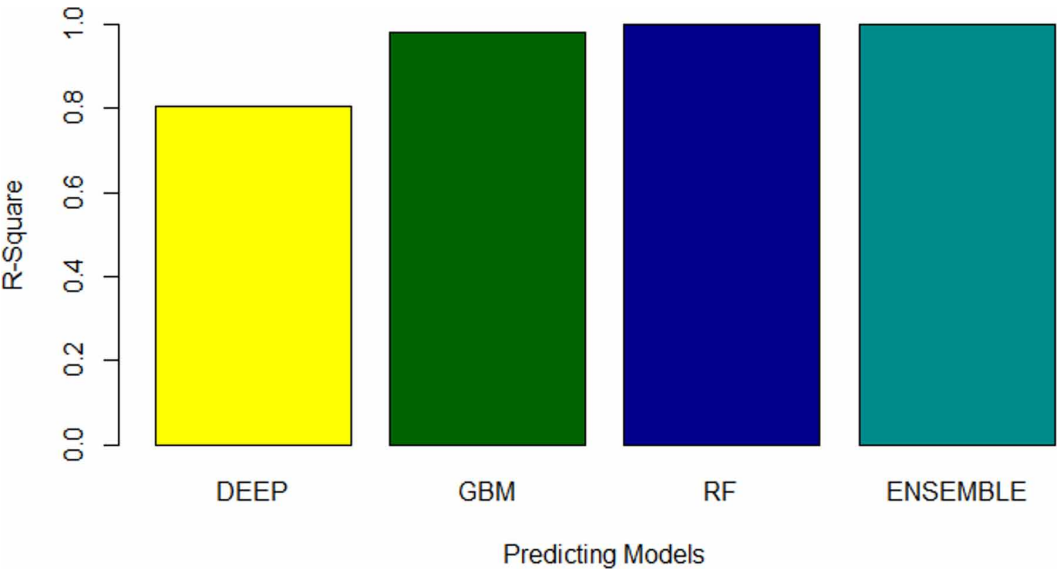
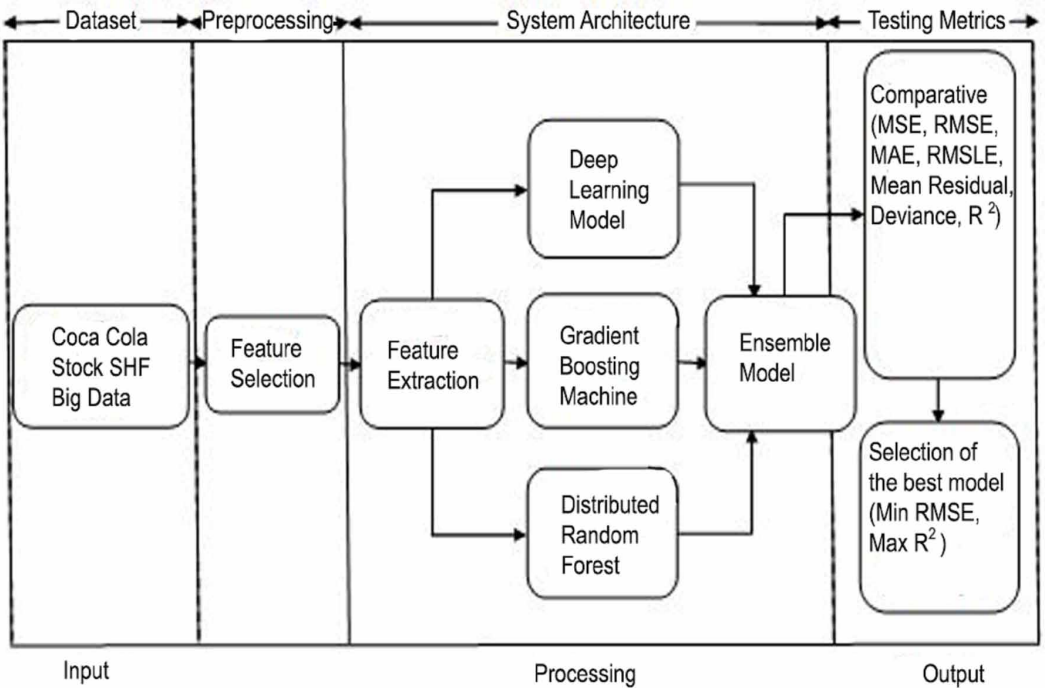


Figure 8. Comparative R-Square (Deep Learning, GBM, RF and Ensemble)





## REFERENCES

- Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*.
- Al-Hnaity, B., & Abbod, M. (2015, July). A novel hybrid ensemble model to predict FTSE100 index by combining neural network and EEMD. In *2015 European Control Conference (ECC)* (pp. 3021-3028). IEEE.
- Arévalo, A., Niño, J., Hernández, G., & Sandoval, J. (2016, August). High-frequency trading strategy based on deep neural networks. In *International conference on intelligent computing* (pp. 424-436). Springer. doi:10.1007/978-3-319-42297-8\_40
- Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques - Part II: Soft computing methods. *Expert Systems with Applications*, 36(3), 5932–5941. doi:10.1016/j.eswa.2008.07.006
- Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47, 552–567. doi:10.1016/j.najef.2018.06.013
- Beltrametti, L., Fiorentini, R., Marengo, L., & Tamborini, R. (1997). A learning-to-forecast experiment on the foreign exchange market with a classifier system. *Journal of Economic Dynamics & Control*, 21(8-9), 1543–1575. doi:10.1016/S0165-1889(97)00035-3
- Beltrametti, L., Fiorentini, R., Marengo, L., & Tamborini, R. (1997). A learning-to-forecast experiment on the foreign exchange market with a classifier system. *Journal of Economic Dynamics & Control*, 21(8-9), 1543–1575. doi:10.1016/S0165-1889(97)00035-3
- Candel, A., Parmar, V., LeDell, E., & Arora, A. (2016). *Deep learning with H2O*. H2O. AI Inc.
- Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., & Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55, 194–211. doi:10.1016/j.eswa.2016.02.006
- Chen, L., Qiao, Z., Wang, M., Wang, C., Du, R., & Stanley, H. E. (2018). Which artificial intelligence algorithm better predicts the Chinese stock market? *IEEE Access : Practical Innovations, Open Solutions*, 6, 48625–48633. doi:10.1109/ACCESS.2018.2859809
- Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83, 187–205.
- Dietterich, T. G., & Kong, E. B. (1995). *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Technical report, Department of Computer Science, Oregon State University. <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. doi:10.1016/j.ejor.2017.11.054
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367–378. doi:10.1016/S0167-9473(01)00065-2
- Gerlein, E. A., McGinnity, M., Belatreche, A., & Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54, 193–207. doi:10.1016/j.eswa.2016.01.018
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.
- Guillame-Bert, M., & Teytaud, O. (2018). *Exact Distributed Training: Random Forest with Billions of Examples*. arXiv preprint arXiv:1804.06755.
- Guo, Y., Han, S., Shen, C., Li, Y., Yin, X., & Bai, Y. (2018). An adaptive SVR for high-frequency stock price forecasting. *IEEE Access : Practical Innovations, Open Solutions*, 6, 11397–11404. doi:10.1109/ACCESS.2018.2806180

- Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science*, 4(3), 183–201. doi:10.1016/j.jfds.2018.04.003
- Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, 6(2/3), 95–101. doi:10.1016/0304-405X(78)90025-9
- Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*.
- Kumar, R., Chandra, P., & Hanmandlu, M. (2013, December). Local directional pattern (LDP) based fingerprint matching using SLFNN. In *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)* (pp. 493–498). IEEE. doi:10.1109/ICIIP.2013.6707640
- Kumar, R., Chandra, P., & Hanmandlu, M. (2013, December). Fingerprint matching using rotational invariant image based descriptor and machine learning techniques. In *2013 6th International Conference on Emerging Trends in Engineering and Technology* (pp. 13–18). IEEE. doi:10.1109/ICETET.2013.4
- Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164, 163–173. doi:10.1016/j.knosys.2018.10.034
- Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *The Journal of Economic Perspectives*, 17(1), 59–82. doi:10.1257/08953300321164958
- Malohlava, M., & Candel, A. (2017). *Gradient boosting machine with H2O*. Academic Press.
- Mankiw, N. G., & Shapiro, M. D. (1985). Trends, random walks, and tests of the permanent income hypothesis. *Journal of Monetary Economics*, 16(2), 165–174. doi:10.1016/0304-3932(85)90028-5
- Niculescu-Mizil, A., & Caruana, R. (2005, August). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 625–632). ACM. <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html>
- Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, 253(3), 697–710. doi:10.1016/j.ejor.2016.02.056
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4), 2162–2172. doi:10.1016/j.eswa.2014.10.031
- Rusk, N. (2015). Deep learning. *Nature Methods*, 13(1), 35. doi:10.1038/nmeth.3707
- Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, 8(4), e1249. Retrieved June 07, 2019, from <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html>
- Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, 76(18), 18569–18584. doi:10.1007/s11042-016-4159-7
- Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble machine learning: Methods and applications*. Springer Science & Business Media.
- Zhang, J., Cui, S., Xu, Y., Li, Q., & Li, T. (2018). A novel data-driven stock price trend prediction system. *Expert Systems with Applications*, 97, 60–69. doi:10.1016/j.eswa.2017.12.026

*Ravinder Kumar received the M.Tech. degree in Computer Science & Engineering in 1998 from GJ University of Science and Technology, Hisar, India. Currently, he is Assistant Professor with Ansal Institute of Technology, Gurgaon and submitted Ph.D. to USICT GGSIPU Delhi, India. His research interest is in the image processing and biometrics.*

*Lokesh Kumar Shrivastav, pursuing Ph.D. in Computer Science Engineering from USICT, GGSIPU, Dwarka, New Delhi under the supervision of Prof. (Dr.) Ravinder Kumar. M. Tech. (CSE) from Amity University, Noida, Uttar Pradesh MCA from IGNOU, New Delhi B. Sc. (H) Physics, Sitamarhi, Bihar Member of International Association of Academicians (IAASSE) Assistant Professor(Guest), ARSD College, University of Delhi, New Delhi.*