Journal of Information Technology Research
Volume 15 • Issue 1

Gradient Boosting Machine and Deep Learning Approach in Big Data Analysis:

A Case Study of the Stock Market

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

Designing a system for analytics of high-frequency data (big data) is a very challenging and crucial task in data science. Big data analytics involves the development of an efficient machine learning algorithm and big data processing techniques or frameworks. Today, the development of the data processing system is in high demand for processing high-frequency data in a very efficient manner. This paper proposes the processing and analytics of stochastic high-frequency stock market data using a modified version of suitable gradient boosting machine (GBM). The experimental results obtained are compared with deep learning and auto-regressive integrated moving average (ARIMA) methods. The results obtained using modified GBM achieve the highest accuracy (R2 = 0.98) and minimum error (RMSE = 0.85) as compared to the other two approaches.

KEYWORDS

ARIMA, Big Data Analysis, Deep Learning, GBM, Gradient Boosting Machine, Machine Learning, Stock Market

1. INTRODUCTION

Fast data acquisition technology demands an appropriate analysis and prediction mechanism for its handling (Atsalakis & Valavanis 2009). Due to the advancement of internet technology and the processor's capability, the vast amount of data is generated at a fine interval of time (Pal & Kar, 2019). Therefore, fast absorption and processing techniques are required to handle the data generated at a very rapid rate. The advances of ICT (Information and Communication technology) and computing algorithms, open the horizon of collection and analysis of high-frequency data (data in a regular or irregular interval of time) (Pal & Kar, 2019). In the recent years, the developments of machine learning algorithms for data analytics (Mahdavinejad at el. 2018, Kumar 2017, 2018 (a, b)) play an essential role in providing an excellent and fast prediction over a vast amount of big high-frequency data (Calcagnile at el. (2018). The three big data attributes, i.e., three Vs. (velocity, volume, variety) are exhibited by the stock market stochastic dataset. Therefore, accurate forecasting or prediction of

DOI: 10.4018/JITR.2022010101

stock prices is the primary concern for the investors and companies operating in the stock market. Due to the non-stationary and non-linear time-series nature of stock market trends, the prediction of stock prices is a hugely challenging task in the financial market (Zhang et al. 2018). Economic time series analysis is a significant source of information for stock market prediction. Finding hidden patterns is the requirement of analysis and forecasting for the price actuations (Zhou et al. 2018). Existing frameworks for analysis and forecasting of high-frequency financial data sets can be classified into two categories (Zhou et al. 2018):

1. *Statistical models* in which advanced mathematical models and procedures can analyze the high-frequency dataset. As the analysis of financial data sets requires some underlying assumptions to be followed, therefore this category of methods can't be utilized to develop an intelligent system.
2. The use of *soft computing models* based on machine learning approaches to capture the dynamics of a financial dataset in the analysis, like in the stock price prediction (Mahdavinejad at el. 2018, Zhou et al. 2018).

In recent years, many stock market forecasting techniques have been proposed to predict the stochastic stock market data, but the accurate prediction of is still not a fully solved problem (Dai et al.,2013). Due to the slow and complexity in the processing of traditional and fundamental statistical methods, the prediction using analytical tools has a minimal application or obsolete in the analysis of high-frequency stochastic stock data.

The soft computing model has shown the better capability to handle the complex, Brownian, and nonlinear dataset of the stock market (Göçken et al., 2019). The proposed work is focusing on devising and applying the soft computing model or machine learning models in the new scenario. The proposed work selects the three best available models from three different paradigms. Auto-Regressive Integrated Moving Average (ARIMA) (Challa et al., 2018) for best statistical learning, Deep Learning for nonparametric machine learning model (Ding et al., 2015), June) and Gradient Boosting Machine (GBM) to ensemble tree-based machine learning model (Basak et al., 2019). The ARIMA is designed and developed by the 'Forecast' package in R-studio (Gandrud (2016). The Deep Learning and GBM are designed and developed by H2O package in R-studio that is capable and renowned to handle the big stochastic data.

ARIMA is widely regarded and efficient model used in the analysis and prediction of the stochastic stock market (Rathnayaka et al. 2015). It is a time series model which performs based on the past value of the datasets as well as previous error terms for the forecasting.

Deep learning is the most known supervised machine learning model to provide generalization, training stability, and stability with the big stochastic data. It is based on feed-forward architecture and gives the highest accuracy in the case of prediction. In this study, we have applied supervised deep learning model to optimize the predictive result (Fischer & Krauss, 2018). Gradient Boosting Machine (GBM) algorithm is an ensemble machine learning model that works to build a predictive tree (Basak et al., 2019). In this approach, GBM generates a new model that predicts the residual of the previously available models and report the final prediction by aggregating all (Khwaja et al., 2017).

1.1 Motivation and Contribution

Analysis of high frequency stochastic big data is a very challenging task, and it is never used in stock market prediction before. The accuracy in the prediction of the stock market is directly proportional to the gain and loss of investors. To maximize the profit of the customer investment, the accurate and timely forecast is very much desirable. So, motivated by this fact, we proposed the prediction model and its comparison with the best available models from three different paradigms to get accurate prediction and to have maximized gain.

The significant contributions of the proposed paper are as follows:

1. Implements generalized machine learning models to make an intelligent system, which will capable to produce the futuristic nature of the stock market.
2. Results thus obtained are compared with three most prominent model, including ARIMA, Deep Learning, and Gradient Boosting Machine (GBM) have been implemented to realize the problem and to understand the behavior of stock fluctuation and predict for the future.
3. Helps the developer or designer of the data processing system to develop a reliable and optimized intelligent-financial-forecasting system.

This paper is organized as follows: Section II presents the literature survey. Section III describes the proposed methodology. Section IV discusses the technical analysis of the models. Section V offers the technical design and implementation, the experimental results and its analysis are discussed in Section VI. Section VII presents the conclusion and future directions.

2. LITERATURE REVIEW

Available relevant literature from a decade has proved the nonlinear and volatile behavior of the stock market. Different researchers have used different techniques and tools to get more reliable and accurate prediction results. Recent studies show that mixed and hybrid methods provide better results as compared to a single analysis model for low-frequency datasets. This section presents a significant contribution to the proposed domain in a considerable period.

A very first model was developed by Enke et al. (2013) that uses the hybrid-approach of prediction. It works on the basis of the combined application of fuzzy clustering, differential evolution, and fuzzy inference to produce the indexing result. At first, the input is generated with the help of step-wise regression-analysis method. The model takes the sets of data which has the most robust prediction capability. In the second stage, the previously generated hybrid model applied by the use of extraction rules to produce an intermediate result.

At last, a fuzzy-inference in neural-network is used to create the final result. This combined experimental setup was applied to the dataset of the stock market. The simulated results provide the better root mean square (RMS) value as compared to regression neural network, linear regression models, multi-layered feed-forward neural network, and probabilistic neural network model. In spite of the better result, this study suggests improving the model with the use of the type-2 fuzzy sets, which contains extra power of expression and dynamic nature to handle the factors of uncertainties. This hybrid model was tested in the prediction of the volatile stock market, and the simulated results suggest that it is better than earlier existing models.

Another mixture model was proposed by Patel et al. (2015) which uses two-stage fusion techniques, at first stage Support Vector Regression (SVR) is used, and at the second stage, the combination of ANN, Random Forest, and SVR is used for prediction. The potential of this mixture model is compared with the single-stage modeling techniques in which RF, ANN, and SVR are used alone for modeling the prediction. The particular outcomes suggest that a two-stage hybrid or mixture model is superior to that of the single-stage prediction modeling techniques. They have recommended a mixture of these methods for further enhancement of the prediction results.

Chiang et al. (2016) proposed a very significant model that uses an adaptive-intelligent-stock-trading-decision-support-system. In this model, particle-swarm-optimization and ANN are applied to predict the future behavior of the volatile and Brownian's stock market. The proposed system has its types of limitation as to the use of technical indicators and patterns as an input pattern. This is a particular type of model.

A very significant work proposed by Chourmouziadis et al. (2016) suggested a two-fold- Fuzzy based prediction system. At the first stage, they used fuzzy-system in the short-term-trading that discard the overflowed confidence of classical data and use the detailed assessment. At the second stage, they applied a specific trading technique, and an "amalgam" between a compromised set of mainly picked unrequited particular indicators is used. This produces alarming indications and supplied these signals to previously settled and required fuzzy-system to provide the part of the portfolio that is to be invested. That short-term-fuzzy system is used to test the general index of the American Stock Exchange (ASE) for a longer time. This particular model has its type of limits as weights of the fuzzy rules. So, it is very tough to do a job, because the success rate of the model depends on the capability to select the required technical indications. The proposed strategy analyzes the nature of prediction for the short interval of time. It is a good strategy for the small size of the datasets with 66% accuracy. In testing period, it provides 81% accuracy for the traders for the prediction. DNN is a simple model, that recommends the other model as Deep Recurrent Neural Network, Deep Belief Network, Convolution Deep Neural Network, Deep Coding Network and other Network to get a more precise and accurate result for the large datasets.

Qiu et al. (2016) developed a model which is a mixture of ANN, Genetic Algorithm, and Simulated Annealing. It produces satisfactory results of prediction on the time of the test. They proposed eighteen input sets that can be used to predict the volatility of the stock market returns. It can be applied to minimize the dimension of the available input variables. They recommended that the application of ANN and other models will be more useful to predict the stock market returns.

Three-dimensional reduction-techniques were suggested by Zhong et al. (2017) that is based on fuzzy- robust-principal-component-analysis, principal-component-analysis, and kernel-based-principal-component-analysis. These techniques are used to simplify and rearrange the original data structure through the use of ANN and Dimension Reduction. It is essential to select a proper kernel for the excellent performance of KPCA. The suggested the mechanism to choose automatic kernel functions to get a better result. The simulated results indicated that mixture the ANNs with the PCA gives little better prediction accuracy than the rest of the other two mixture model.

Zhong et al. (2017) also proposed a mixture model that is a combination of 7 sets of features extraction methods, 3 data representation techniques as autoencoder, restricted Boltzmann machine, and principal component analysis, to design 3 layers of Deep Neural Networks (DNN) to forecast the futuristic nature of stock market return. It is applied on the Korean stock market index and found that the Deep Neural Network produces slightly better result in the training phase compare to linear autoregressive model. But these advantages were mostly disappeared in the testing phase. It works better in limited resource means low-frequency dataset. Its performance is doubtful in terms of high-frequency datasets.

At the first time a model was developed by Henrique et al., (2018) to handle high-frequency dataset which was collected for a small period of three months at the interval of a single minute. The data was a trend by the use of Support Vector Regression (SVR), and a random walk model compared the produced result with the same set of data. The experimental result suggests the SVR model is inferior to the Random Walk model in terms of prediction (RMSE). The period of three months is not sufficient to capture the nature of the high-frequency dataset.

Göçken et al. (2019) introduced an enhanced hybrid soft computing model to maximize the prediction capability of the stock market. In the study, they used a mixture of Neural Network, Regression Tree, Generalised Linear Model, Extreme Learning Machine, Gaussian Process Model, etc. to build an enhanced model to minimize the error rate and maximize the profit. The two and half year stock dataset was used, which was taken from three indices (ECILC, EREGL, and AFYON) respectively between the period of 17-Apr-2013 to 30-Nov-2015 for training as well as testing purpose. Harmony Search Optimization (HS) method is applied to optimize the parameters of the soft computing model. To normalized the Sigmoid dataset normalization was used. This model produces a satisfactory result as compared to existing models. It can further be enhanced by applying Harmony Search Optimization (HS) techniques

A new method was proposed by Wen et al. 2019 to reconstruct financial temporal time series by the use of high order structure as mofits. The experimental result suggests 4% to 7% improvement in the prediction capacity as compared to the deep learning model.

In spite of extensive research in the area, none of the researchers can produce a single established model which gives an optimized and précised model of computationally-intelligent-system for the stock market. Table 1 highlights the summary of the literature review in this research domain. This feedback demands the old, and new models can be applied and re-analyzed with the big high-frequency dataset for the futuristic analysis and prediction. So, this study proposes the most famous statistical as well as machine learning model should be applied to the big high-frequency data to open the new dimension of the prediction.

3. PROPOSED METHODOLOGY

In this proposed work, a high-frequency Coca Cola dataset is used to validate the stock market prediction. The previous work had not been able to address the issue of related to the exact prediction with the given dataset. The proposed model initially applies the most acceptable time series model (ARIMA) on this high-frequency stochastic dataset to settle a standard base. This model is implemented using the "Forecast Library" available in the R-studio. At the next step, a very advanced and the most efficient package H_2O is used to applied Deep Learning and Gradient Boosting Machine model, and

Table 1. A Summary of Literature Review on Stock Price Forecast

References	Source of dataset	Target Output	Size of data (instances)	Timespan and frequency of data	Proposed model algorithms	Testing metrics
Enke et al. (2013)	US S&P 500 indices	Stock Return	361	Jan-1980 to Jan-2010 (Daily)	Fuzzy Clustering + Feature Selection + Fuzzy ANN	Lowered RMSE
Patel et al. (2015)	India CNX and BSE indices	Stock Return	2393	Jan-2003 to Dec-2012 (Daily)	(RF, ANN, SVR) +SVR	MAPE, MSE, MAE, rRmse
Chiang et al. (2016)	World 22 stock market indices	Stock Return	756	Jan-2008 to Dec-2010 (Daily)	ANN + Particle Swarm Optimization	Trading Simulation
Chourmouziadis et al. (2016)	Greece ASE General index	Portfolio Composition	3907	15-Nov-1996 to 5-Jun-2012 (daily)	Fuzzy System	Trading Simulation
Qiu et al. (2016)	Japan Nikkei 225 index	Stock Return	237	Nov-1993 to Jul-2013 (Monthly)	ANN + (Simulated Annealing, Genetic Algorithm)+ANN	MSE
Zhong et al. (2017)	US SPDR S&P 500 ETF (SPY)	Market Fluctuation	2518	1-Jun-2003 to 31-May-2013	ANN + Dimension Reduction	Statistical tests
Eunsuk et al. (2017)	Korea KOSPI 38 index	Stock Return	73,041	4-Jan2010 to 30 Dec-2014 (5-Minute)	DNN + Data Representation	RMSE, NMSE, MI MAE
Henrique et al. (2018)	Brazilian, American & Chinese stocks with three blue chip & 3 small-cap stock	Stock Return	Ten years historical datasets	1-Mar-2017 to 26-Mar-2017 (1-Minute)	SVR	RMSE
Göçken et al. (2019)	ECILC, EREGL and AFYON indices	Stock Return	2.5 years (Approx)	17-Apr-2013 to 30-Nov-2015	Hybrid Soft Computing Models	MAPE, MAE, RMSE
Wen et al. (2019)	Different stock indices	Stock Return	Small in size	High Frequency	Combining motif-based sequence reconstruction with CNN	Accuracy, Recall, and Precision
Proposed Work	Coca Cola listed in New York Stock Exchange (NYSE)	Stock Return	8 Lacks (Approx)	3-Jan 2000-31-12-2008 (1-Minute)	ARIMA, Deep Learning, GBM	RMSE, R^2

RF: Random Forest, ANN: Artificial Neural Network, SVR: Support Vector Regression, RMSE: Root Mean Square, MAPE: Mean Absolute Percentage Error, MSE: Mean Square Error, MAE: Mean Absolute Error, rRMSE: Relative RMSE, NMSE: Normalized MSE, ARIMA: Autoregressive Integrated Moving Average, GBM: Gradient Boosting Machine, MI: Mutual Information.

the results are compared with the ARIMA model. At the final stage, the prediction accuracy of these three models is analyzed in terms of RMSE (Root Mean Square Error) and R^2 to compare the results for the best model (lowest RMSE and highest R^2). The Root Mean Square (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:

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

R2 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 one means the full correlation between the real and the predicted dataset:

$$d_{i} = r_{i} - p_{i} \tag{2}$$

where d_i, r_i, and p_i is difference residuals data frame, real data frame, and predicted data frame respectively:

$$m = \frac{1}{n}\sum_{i=n}^{n} r_{i} \tag{3}$$

The total sum of the square is proportional to the variance of the dataset, we get:

$$ss_{tot} = \sum_{i}\left(r_{i} - m\right)^{2} \tag{4}$$

The regression sum of the square, we get:

$$ss_{reg} = \sum_{i}\left(p_{i} - m\right)^{2} \tag{5}$$

The sum of the square of residuals, we get:

$$ss_{res} = \sum_{i}\left(r_{i} - m\right)^{2} \tag{6}$$

Now calculate R^2 value by using equation Eqs. (4) and (5), we get:

$$R^{2} = 1 - \frac{ss_{reg}}{ss_{tot}} \tag{7}$$

Along with these evaluating parameters, the proposed study also uses some other evaluating parameter like (MSE, MAE, RMSLE, and Mean Residual Deviance).

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:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(p_i - r_i\right)^2 \tag{8}$$

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:

$$MAE = \frac{\sum_{i=1}^{n}\left|p_i - r_i\right|}{n} \tag{9}$$

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

$$RMSLE = \sqrt{\frac{\sum_{i=n}^{n}\left(\ln\left(\frac{r_{i+1}}{p_i + 1}\right)\right)^2}{n}} \tag{10}$$

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

The following notations variables were adopted and used for this study:

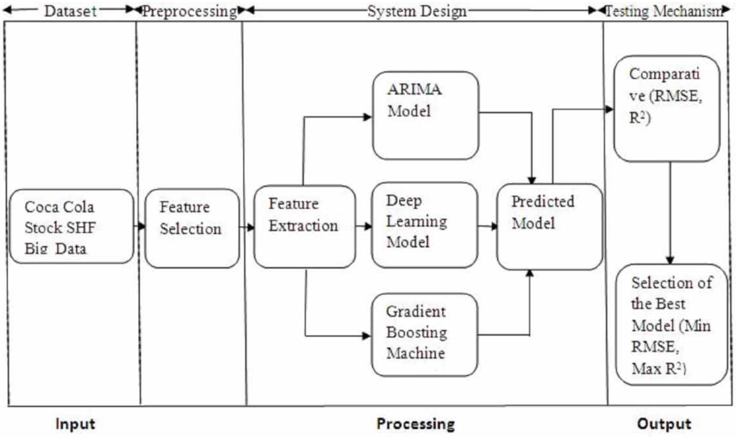
- *n*: total samples
- *r*: real sample value
- *p*: predicted value
- *m*: mean of real sample value

The flow graph of the proposed methodology is presented in Figure 1. The process is divided into four stages, namely data collection, pre-processing, feature extraction, and Prediction using two commonly used evaluating parameters.

3.1 Data Collection and Preparation

A high-frequency stock market data is collected from 03-Jan-2000, 9:38 am to 31-Dec-2008 15:59 pm in the fine time interval of a 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.

Figure 1. Flow graph of the Proposed Model



3.2 Data Preprocessing and Feature Extraction

For the smooth extraction of features pre-processing mechanism is used to minimize the irrelevant or blank data to produce a more accurate and precise result. R-Studio has used to handle new data or the data with ‘NA’ value.

3.2.1 Data Partition

The dataset is partitioned into two parts, where the first part is used for training and the second part is used for testing the modal. The training dataset has a size of 697949 rows, and six columns and testing dataset have a capacity of 174486 rows and six columns. The datasets used for experiment and analysis is independent dataset; it means there are no correlations among data. It is observed that the increment or decrement of any value does not disturb the other benefits. 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.

3.3 Summary of Dataset

The summary is a general-purpose function in R-language that thoroughly analyzes the central tendencies of the datasets as minimum, maximum, mean, and median of datasets, as shown in Table 2.

3.4 The Measure of Central Tendency and Time Series of Trained Dataset has a Wide Variation

The range of variation in the indexes of the stock market dataset is highly fluctuating in a non-linear way. This variation is because, of factors like social issues, company’s management and government policies, etc. So, it is difficult to predict by looking at past patterns entirely. Therefore, advance analysis tooled might be developed to analyze and predict the prediction frameworks using statistical tools for better prediction. Figure 2 shows the variation about its mean of the stock market prices during closing for the used dataset.

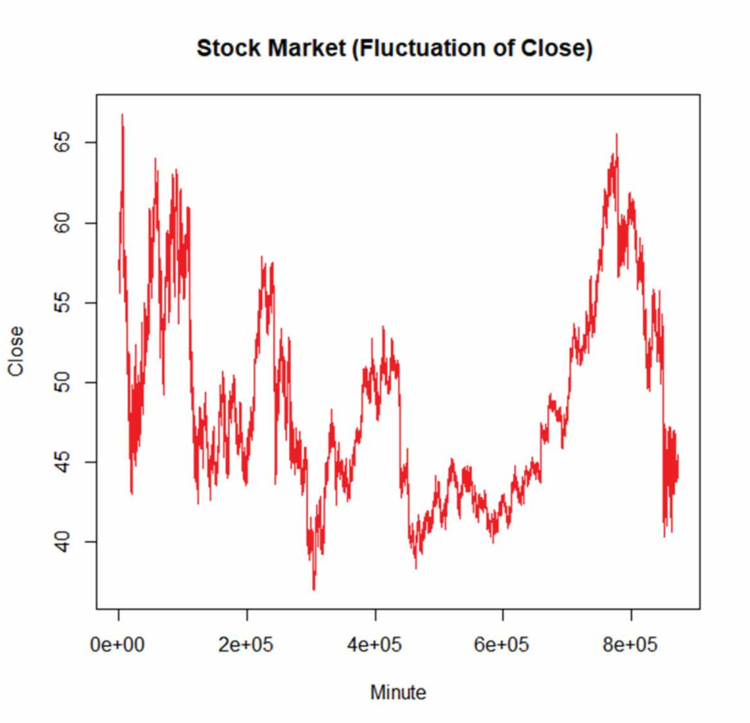
4. MACHINE LEARNING ALGORITHMS

This section presents the machine learning algorithms used for prediction analysis of results.

Table 2. Summary of Dataset

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

Figure 2. Variation about the mean of the stock market prices during closing for the used dataset



4.1 ARIMA Model

This model is the most popular and benchmark model of the era. It was designed and developed by Box and Jenkins in 1970. After its development, it started to control all statistical and soft computing technique due to continuous modification available in the original packages. Due to its tremendous prediction capabilities in the area of time-series datasets (Debiyi et al. 2013), it became a benchmark model for any introduced model. This method is the composition of many activities with time series mechanism. It already noticeable results in short term forecasting. ARIMA (p, q, d) model where p is AR order, q is a degree of the differencing and d is MA order. The algorithm developed for the analysis of stock market data is presented as follows:

Algorithm

```
1. Take training dataset as Close.
2. Find log (Close) to make it saturated.
   // This kind of dataset can't be taken for further analysis
   by the system. So, to reduce the //non-linearity nature of the
   dataset, we have taken the log of the Close dataset to make it
   a narrow //deviated dataset.
3. Find diff(log(Close))
   //It was realized that after taking the log of Close, the
   deviation has decreased up to the limit, but it //has still a
   large range of non-linearity, that has to be reduced further.
   For this //purpose, the differentiation of the log of the Close
   the dataset was performed, which is shown in //the Fig. 5.
4. Test for saturation by the use of ACF and PCF.
   //The differentiation (log (Close)), provides the saturated
   and denser dots in the two //dimensional plotting system. Now,
   this thicker and narrowed fluctuated rage is good enough //for
   the further procedure of the ARIMA model. Now, we are in need
   to plot ACF and PCF to get //the lag inside the available
   pattern. For a better analysis of the dataset and further
   procedure, //we applied ACF (Autocorrelation Function) and
   PCF (Partial autocorrelation Function) to find //the
   correlation between the time series and available lags in the
   Close attribute of stock //market.
5. Develop the time series of the diff(log(Close))
   //This high-frequency stochastic dataset can be analyzed in the
   better manner by the help of time //series as shown in Fig. 6.
6. Train the model by the use of auto ARIMA available in the R-package.
   //Auto ARIMA is modified and advanced version of the ARIMA
   model mainly applied and used //to auto fit the dataset without
   knowing any parameter of the available model as shown in Fig. 7.
7. Forecast the model for testing phase.
   The R-Language will do //in this case, forecasting by the help
   of available Close price //dataset. Once the network is
   trained, the performance will be enhanced automatically by
   support// experience with the fitted model.
8. Transform the forecasted data into an actual numeric value.
9. Perform the RMSE and $R^2$ to analyze the deviation from the actual value.
```

4.2 Deep Learning

Deep learning is one of the most powerful computational model that is a combination of the many processing layers and capable of capturing the data with multi-levels of the abstraction (Zhou & Troyanskaya, 2015). It finds the intricate structure that presents in the dataset by the use of the back-propagation algorithm and ensures to change it's an inside parameter of present depends on previous, for the betterment of the targeted model. The model developed by the H_2O is purely supervised learning, fast and memory-efficient model. The H2O package (Candel et al., 2019) tool used in this paper and also used by many researchers for low-frequency datasets analytics (Heaton et al. 2017, Fischer et al., 2018) using is presented as follows:

Algorithm

```
1. The high-frequency stochastic datasets are taken for the training.
2. The training datasets is given to deep learning module, which
   is based on a multilayer feed-forward neural network and
   trained the dataset by the use of stochastic gradient descent
   by the use of the back-propagation algorithm.
3. This network keeps a significant number of hidden layers with
   rectifier, tanh, and activation functions, etc.
4. All the computed node trains the global model parameter on its
   local dataset by the use of multithreading and participates in
   forming a global model with model averaging across the network.
5. This global model produces the optimal trained prediction result.
```

4.3 Gradient Boosting Machine

Gradient Boosted Machine is an ensemble model, and it is very much capable of handling the regression task. It is easy to interpret with the adaptability characteristics that give exact results (Landary et al., 2016). The predictor value can be produced in every iteration that is based on the previous iteration and ensures 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. This original algorithm is designed and developed by Friedman in (2002). The modified version of boosting can be classified as a nonlinear classification that optimizes the accuracy of the tree without affecting its speed. It provides an easily distributable and parallelizable feature with the effortless environment for model tuning and selection. This version of GBM that is capable of handling the big data with optimal accuracy is rarely used in the stock market prediction domain. But the efficincy of the model is very significant in the current senerio of big data. The modified Gradient Boosting Model (Malohlava et al., 2019) designed and developed by H2O explained as follows:

```
Input: Training set $\left\{\left(x_i, y_i\right)\right\}_{i=1}^{n}$
  //Here $x_i$ is minutes and $y_i$ is Close value of processed training
  dataset, i =1 to n=872435, and //differentiable loss function
  $L\left(y, F\left(x\right)\right) = \frac{1}{2}\left(observed - predicted\right)^2$
  Algorithm:
  1. Initialize the model with a constant value
  2. $F_0\left(x\right) = arg_\gamma min\sum_{i=1}^{n} \mathrm{L}\left(\mathrm{y}, ^3\right)$
     //Here L is loss function, y is observed, $\gamma$ predicted the
     value of the dataset. In this stage, we will get //mean value
     of the Close.
  3. For m=1 to M
     //It indicates the no. of the tree starting from 1 to M,
     where M=872435.
     1. Compute the pseudo residuals
        $r_{im} = -\left[\frac{\partial \mathrm{L}\left(\mathrm{y_i}, \mathrm{F}\left(\mathrm{x_i}\right)\right)}{\partial \mathrm{F}\left(\mathrm{x_i}\right)}\right]_{F\left(x\right)=F_{m-1}\left(x\right)} \quad for\, i = 1, \ldots., n$
        //Here r is residual; i is the sample no. and m is the tree
        that we want to build.
        2. Fit a regression tree to the $r_{im}$ , $h_m\left(x\right)$ to pseudo
           residual, i.e., train it using the train set $\left\{\left(x_i, y_i\right)\right\}_{i=1}^{n}$ .
           //It decides the roots and leaves.
        3. Compute multiplier $\gamma_m$ by solving the following one-
           dimensional optimization problem
           $\gamma_m = arg_\gamma min\sum_{i=1}^{n}\mathrm{L}(\mathrm{y}_\mathrm{i},\mathrm{F}_{\mathrm{m}-1}\left(\mathrm{x}_\mathrm{i}\right) + \gamma h_m\left(x_i\right)$
           //Now we will calculate $\gamma_m$ with the same procedure.
        4. Update $F_m\left(x\right) = F_{m-1}\left(x\right) + \aleph h_m\left(x_i\right)$
           //In this stage, old prediction will be updated by the old
           learning prediction. Here $\aleph$ is new and //small learning rate.
    4. Output $F_m\left(x\right)$
        //We will get the optimized predicted datasets.
```

5. EXPERIMENTAL SET-UP AND DISCUSSION OF RESULTS

Experiments have been performed on Coca Cola stock dataset listed in New York Stock Exchange (NYSE) on core i5 processor using Forecast Library and H2O package of R-studio. The results obtained for the regression of stochastic dataset of Coca Cola are shown in Figure 2-5, and its performance estimation values are presented in Table 1-3. Generally, the accuracy of the model is directly proportional to its R^2 values and inversely proportional to its RMSE values.

5.1 Results of the ARIMA Model

The experimental result suggests that p=0.7026 (log (Close)) and p=0.01(diff (log(Close))) both are stationary because it is less than the threshold where p value check availability and no availability of the null hypothesis. This concludes the dataset was non-seasonal, and the result can be fine-tuned. The ARIMA (p, q) is differentiated from the process ARIMA (p, d, q) that depends on model Selection, parameter estimation, and diagnostics and potential improvement. The model developed according to the algorithm, produces ARIMA (0, 0,1) model with non zero mean with MA(1). This signifies that the presented model is not so efficient to capture the dataset in terms of high frequency stochastic big data. The deviation between real and predicted dataset and evaluating parameter (RMSE = 10.67176 and R2=-1.988473) both suggest that the ARIMA model is not so adequate to capture the big high-frequency dataset as shown in Figure 3 and Table 3 respectively.

5.2 Results of Deep Learning

The experimental result confirms the deviation between the real and predicted datasets of the testing phase. The red line indicates the real datasets where as yellow line indicates the predicted datasets. In spite of the deviation, the model of deep learning is far better than the previously analyzed model ARIMA. The performance parameter of Deep Learning explores how the model deviated from the real dataset. Root Mean Square Error (RMSE): 2.531699 and R^2: 0.8339508 values, all indicates that the model is a far better model than the ARIMA model as shown in Figure 4 and Table 4, respectively.

Figure 3. Shows the deviation between real and predicted datasets

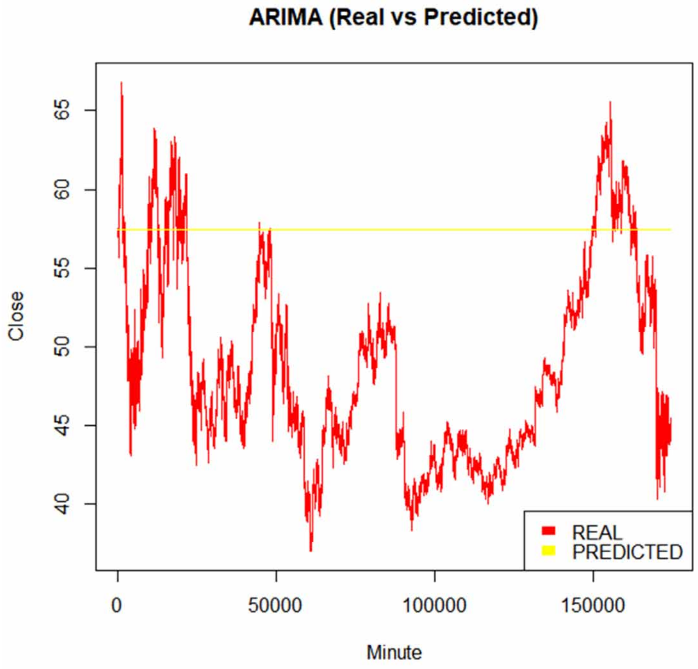


Table 3. ARIMA Performance Parameter

Performance parameter	Result
Root Mean Square Error (RMSE)	10.67176
R-Squared Value	-1.988473

5.3 Results of Gradient Boosting Machine

Ensemble Gradient Boosting Machine is an adequate model to capture the nature of this kind of dataset. The red line indicates the real datasets where the blue line indicates the predicted datasets. Root Mean Square Error (RMSE) values are 0.8464796 and R^2: 0.9811871, all three suggests that the model is far times better than the statistical ARIMA model as well as its close competitor Deep Learning model as shown in Figure 5 and Table 5 respectively.

5.4 Comparison of Performance

This section provides the comparative the outcomes of the models that were previously used in literature (with the same preprocessing) to compare the performance of ARIMA, Deep Learning, and Gradient Boosting Machine (GBM). The ARIMA is a base statistical model, and it was selected to settle a base where the dataset was trained and predicted by the help of “Forecast” package in R-studio. The Deep Learning and Gradient Boosting Machine were trained by the use of “Gaussian distribution” in H_2O package. RMSE comparative convergence graph is commonly drawn for single instance that describes

Figure 4. Prediction using Deep Learning (Real Close vs. Predicted Close)

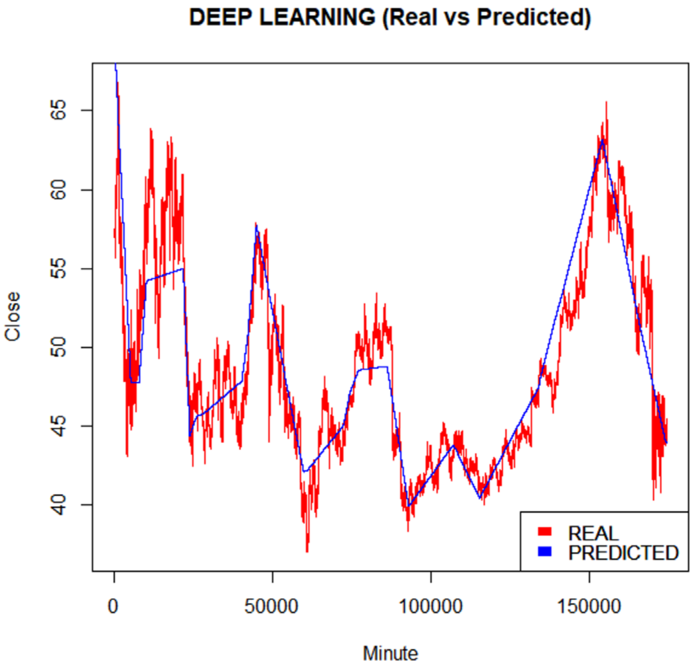


Table 4. Deep Learning Performance Parameter

Performance Parameter	Result
MSE	6.4095
RMSE	2.531699
MAE	1.773624
RMSLE	0.04884439
Mean Residual Deviance	6.4095
R-Squared Value	0.8339508

how the RMSE error (real dataset -mean (predicted dataset)) may develop according to time during testing. It also explains that ARIMA produces the highest error and GBM produce lowest according to time, as shown in Figure 6. Overall, the prediction capability of the Gradient Boosting Machine is far better than the statistical model ARIMA and better than the Deep Learning Model implemented in this paper or any other cited paper used for this study. This study applied these models and has reported performance parameter as RMSE and R^2 used for all three models (please refer to Table 1, 2, 3). For overall comparative study, this study used five attributes as "Minute" that represents the

Figure 5. Prediction using Gradient Boosting Machine (Real Close vs. Predicted Close)

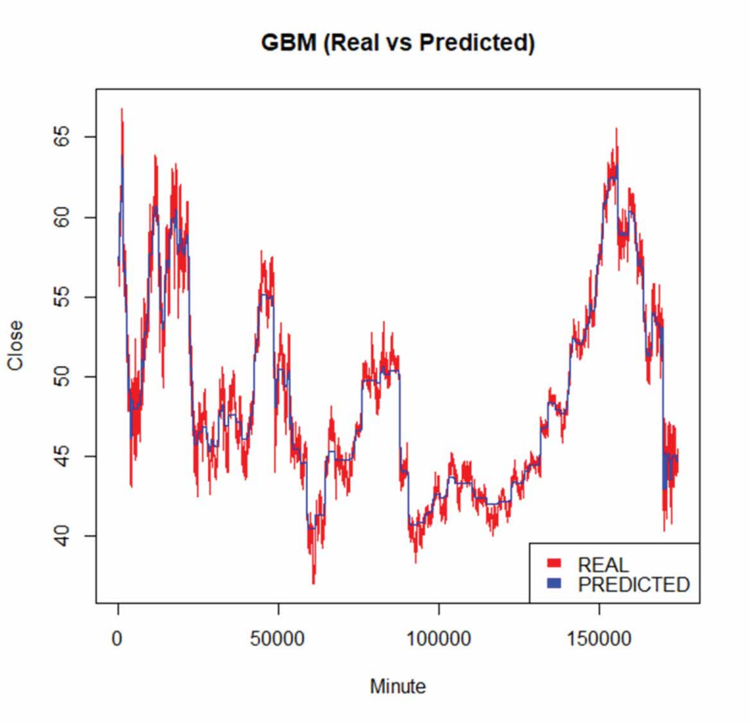


Table 5. GBM Performance Parameter

Performance Parameter	Result
MSE	0.7165277
RMSE	0.8464796
MAE	0.6315757
RMSLE	0.01707818
Mean Residual Deviance	0.7165277
R-Squared Value	0.9811871

particular minute, “real Close” that represents the actual Close value, “ARIMA Close” represents the predicted value produced by ARIMA model, Deep Close represents the expected value produced by Deep Learning model and GBM Close represents the expected value provided by Gradient Boosting Machine (GBM) model of testing datasets. It is clear that “GBM close” is very near to the real Close, as shown in Figure 6 and Table 7.

The comparative chart using RMSE values for all three models, is presented in Figure 8. It is clear from Figure 7 that Gradient Boosting Machine performs much better in comparison to the other two prediction techniques.

Figure 6. Comparison of RMSE Convergence Graph for Single Instance (ARIMA, DEEP, and GBM)

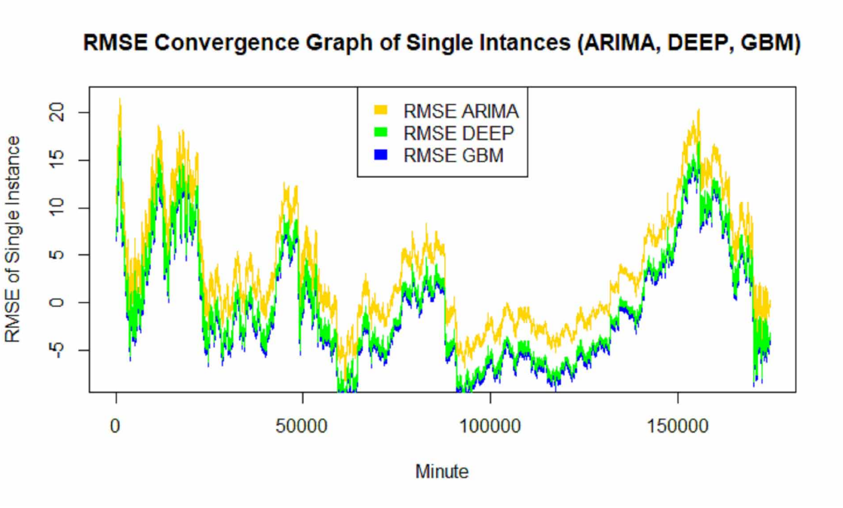
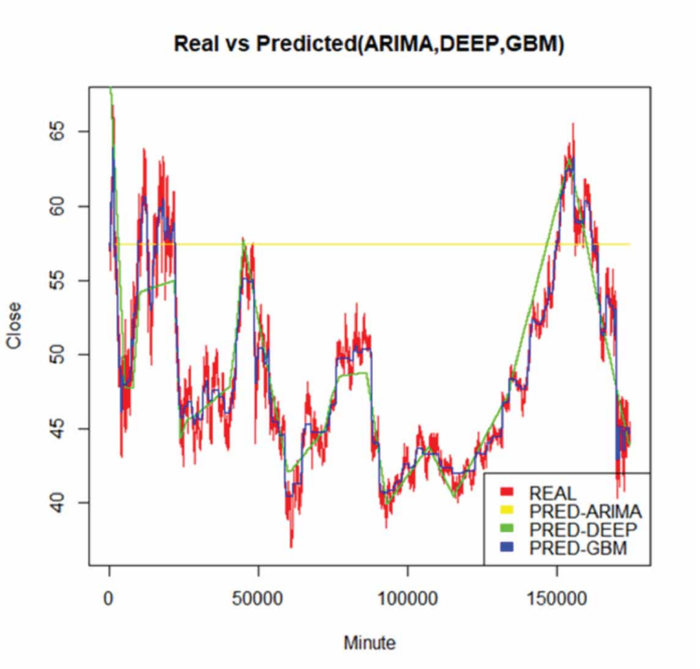


Figure 7. Comparison of Prediction Results (Real Close, ARIMA Close, DEEP LEARNING Close, and GBM)



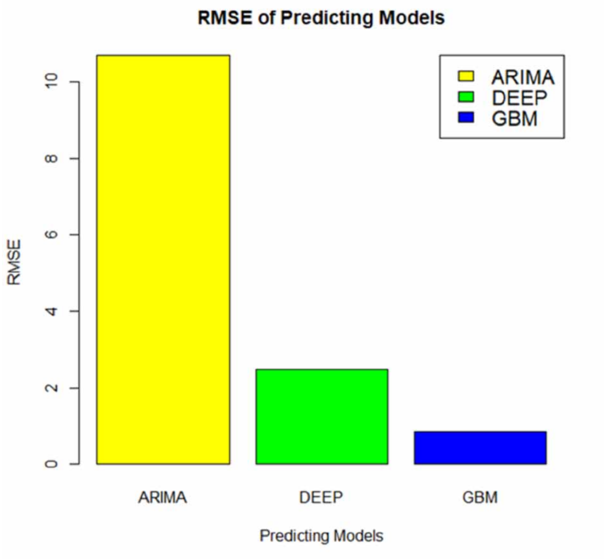
6. CONCLUSION AND FUTURE RESEARCH SCOPE

The present work establishes a direction to analyze a stochastic high-frequency dataset, which is rarely utilized in terms of the stock market. Three most prominent techniques have been implemented to understand the variation of the models. The analyzed result suggest GBM (minimum RMSE and highest R^2 value that is approx. 1) is the best among the other two models. It means the GBM predicted

Table 6. Real vs. predicted result captured for 20 days for (Real Close, ARIMA Close, DEEP Learning Close, and GBM)

Minute	Real Close	ARIMA Close	Deep Close	GBM Close
9	57.5000	57.34334	60.80565	57.65178
11	57.5000	57.40275	60.80411	57.65178
13	57.1875	57.40275	60.80257	57.65178
18	57.3750	57.40275	60.79873	57.65178
27	57.2500	57.40275	60.79181	57.65178
32	57.4375	57.40275	60.78797	57.65178
33	57.1875	57.40275	60.78720	57.65178
35	57.3125	57.40275	60.78566	57.65178
39	57.4375	57.40275	60.78258	57.65178
43	57.3750	57.40275	60.77951	57.65178
45	57.3750	57.40275	60.77797	57.65178
46	57.2500	57.40275	60.7772	57.65178
49	57.2500	57.40275	60.7749	57.65178
53	57.4375	57.40275	60.77182	57.65178
56	57.2500	57.40275	60.76952	57.65178
57	57.1875	57.40275	60.76875	57.65178
60	570000	57.40275	60.76644	57.60631
62	57.125	57.40275	60.7649	57.60631
63	57.125	57.40275	60.76413	57.60631
73	56.875	57.40275	60.75645	57.55966

Figure 8. Comparison of RMSE values prediction results using ARIMA, DEEP LEARNING, and GBM



results are approaching the original observation of the data set. The work suggests and provides the fundamental obstacles and futuristic dimension and guidelines in directions of financial data analysis research. The simulated result achieved the high prediction accuracy of the stock market using the big high-frequency datasets. The experimental result suggests that the tree base learning model gives far better performance than the other two models. This study also indicates that more in-depth, comparative, and ensemble tree-based analysis and simulation is required in future research to build a fully optimal-intelligent-system to predict the stock market behaviors more precisely and accurately.

REFERENCES

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

Calcagnile, L. M., Bormetti, G., Treccani, M., Marmi, S., & Lillo, F. (2018). Collective synchronization and high-frequency systemic instabilities in financial markets. *Quantitative Finance*, *18*(2), 237–247. doi:10.1080/14697688.2017.1403141

Candel, A., & LeDell, E. (2019, Jan.). Deep learning with h2o. H2O. *AI.*

Challa, M. L., Malepati, V., & Kolusu, S. N. R. (2018). Forecasting risk using autoregressive integrated moving average approach: Evidence from S&P BSE Sensex. *Financial Innovation*, *4*(1), 24. doi:10.1186/s40854-018-0107-z

Chiang, W. C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications*, *59*, 195–207. doi:10.1016/j.eswa.2016.04.025

Chourmouziadis, K., & Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications*, *43*, 298–311. doi:10.1016/j.eswa.2015.07.063

Debiyi, 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*, *2014*, 1–7. https://doi.org/10.1155/2014/614342

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015, June). Deep learning for event-driven stock prediction. *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Enke, D., & Mehdiyev, N. (2013). Stock Market Prediction Using a Combination of Stepwise Regression Analysis, Differential Evolution-based Fuzzy Clustering, and a Fuzzy Inference Neural Network. *Intelligent Automation & Soft Computing, 19*(4), 636–648. 10.1080/10798587.2013.839287

Fischer, T., & Krauss, C. (2018). Deep learning with large short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.

Gandrud, C. (2016). *Reproducible research with R and R studio*. Chapman and Hall/CRC.

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2019). Stock price prediction using hybrid soft computing models incorporating parameter tuning and input variable selection. *Neural Computing & Applications*, *31*(2), 577–592.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1), 3–12. https://doi.org/10.1002/asmb.2209

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. https://doi.org/10.1016/j.jfds.2018.04.003

Khwaja, A. S., Zhang, X., Anpalagan, A., & Venkatesh, B. (2017). Boosted neural networks for improved short-term electric load forecasting. *Electric Power Systems Research*, *143*, 431–437.

Kumar, R. (2017). Fingerprint matching using rotational invariant orientation local binary pattern descriptor and machine learning techniques. *International Journal of Computer Vision and Image Processing*, *7*(4), 51–67.

Kumar, R. (2018). (a). A Review of Non-Minutiae Based Fingerprint Features. *International Journal of Computer Vision and Image Processing*, *8*(1), 32–58.

Kumar, R. (2018). (b). A Robust Biometrics System Using Finger Knuckle Print. In *Handbook of Research on Network Forensics and Analysis Techniques* (pp. 416–446). IGI Global.

Landry, M., Erlinger, T. P., Patschke, D., & Varrichio, C. (2016). Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. *International Journal of Forecasting*, *32*(3), 1061–1066. https://doi.org/10.1016/j.ijforecast.2016.02.002

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, *4*(3), 161–175.

Malohlava, M., Candel, A., Bartz, A., Roark, H., & Parmar, V. (2019). *Gradient Boosting Machine with H2O: Seventh Edition Gradient Boosting Machine with H2O*. Retrieved from http://h2o.ai/resources/

Pal, S. S., & Kar, S. (2019). Time series forecasting for stock market prediction through data discretization by fuzzistics and rule generation by rough set theory. *Mathematics and Computers in Simulation*.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using the fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031

Qiu, M., Song, Y., & Akagi, F. (2016). Application of Artificial Neural Network for the Prediction of Stock Market Returns The Case of the Japanese Stock Market. *Chaos, Solitons, and Fractals*, *85*, 1–7. https://dx.doi.org/10.1016/j.chaos.2016.01.004

Rathnayaka, R. K. T., Seneviratna, D. M. K. N., Jianguo, W., & Arumawadu, H. I. (2015, October). A hybrid statistical approach for stock market forecasting based on Artificial Neural Network and ARIMA time series models. In *2015 International Conference on Behavioral, Economic, and Socio-cultural Computing (BESC)* (pp. 54-60). IEEE.

Wen, M., Li, P., Zhang, L., & Chen, Y. (2019). Stock Market Trend Prediction Using High-order Information of Time Series. *IEEE Access*. 10.1109/access.2019.2901842

Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., & Philip, S. Y. (2018). Improving stock market prediction via complex information fusion. *Knowledge-Based Systems*, *143*, 236–247.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, *67*, 126–139. https://doi.org/10.1016/j.eswa.2016.09.027

Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, *12*(10), 931.

Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, *12*(10), 931.

Zhou, X., Pan, Z., Hu, G., Tang, S., & Zhao, C. (2018). Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*.

Ravinder Kumar has completed his Ph.D. from GGSIPU Delhi India in Biometrics. He is presently Associate Professor in CSE/IT at Shri Vishwakarma Skill University, Gurgaon.

Lokesh Kumar Shrivastav is presently pursuing Ph.D. in Computer Science Engineering from USICT, GGSIPU, Dwarka, New Delhi under the supervision of Prof. (Dr.) Ravinder Kumar. His area of interest is predictive analysis of big data and machine learning.