Speculation of Stock Marketing Using Advanced Recursive Techniques

Neeraj Kumar, Amity Institute of Information Technology, Amity University, Noida, India

Ritu Chauhan, Artificial Intelligence and IoT Lab, Center for Computational Biology & Bioinformatics Amity University, India*

ABSTRACT

In the current scenario, the economic status of countries is dependent on stock markets. However, predicting the future prices of any stock is a multifaceted task, as the nature of data is complex and unstructured in nature, which is difficult understand. The focus of the study relies on applying deep neural techniques with regression-based application to discover knowledge from financial databases. The authors have applied LSTM, an advanced version of RNN, and regression-based methods such as ARIMA for predicting future prices of stocks. The study was supported by implementing the techniques on real-world data that was captured from SBI for 6 years. The data has significant opening and closing prices of stock markets. To implement the current study approach, the authors have utilized Python language, where result predicts various performance parameters such as MAE, MSE, RMSE, and bias for both LSTM as well as ARIMA. The performance matrix of LSTM and ARIMA were compared for MAE (mean absolute error) for LSTM, which is 4.32, whereas for ARIMA is 3.83. Also, MSE (mean squared error) value for LSTM is 29.52, for ARIMA was 24.53, and RMSE (root mean squared error) for LSTM and ARIMA are 5.43 and 4.95. The overall accuracy of both of the algorithm were widely applied for real-world prediction among the stock market analysis.

KEYWORDS

ARIMA, LSTM, Prediction Model, Stock Market

INTRODUCTION

Stock market tends to be a strong financial infrastructure of a nation where it can be reliable quoted as backbone or economic indicator of a country (Idrees, 2019). In similar, context the stock market is proportionally linked with the prices of credible companies which are linked with the country setup. Moreover, growth rate of nation is escalated when the companies associated stock price goes higher in the market. So, predicting the stock price plays a very crucial role for overall development of a country and for the citizens of that country, who willingly tries to invest their money in share market. Although, still there is no full proof system existing which can guarantee exhaustively correct prediction for share market. We can say that, share market tends to be very volatile in nature, where the prediction of share price can extensively go up and down instantaneously in short duration of

DOI: 10.4018/IJBDCN.339890

*Corresponding Author

time. Undoubtedly, if you overview any particular stock, you may perceive a lot of fluctuation in the data moment of time (Di Persio & Honchar, 2006). Certainly, investors try to take the advantage of volatile nature of share market and makes lot of money but, if any investor has less knowledge about the stock market data, then the chances are higher to risk the amount he has embarked in the market.

Moreover, share market data is complex in nature where the data is gathered in time series format. The data is usually a real world where it is gathered by varied high end potential companies for disseminating the knowledge to the end users. These potential companies are involved in deciding the stock market index of any country just like in India we have two stock exchanges BSE(Bombay stock exchange) & NSE(National stock exchange) and we also have two separate indexes for these stock exchanges. Sensex is the index used for BSE and Nifty is the index used by the NSE. BSE is the oldest stock exchange as compare to NSE, it started its operation in 1875 (Zhou et al., 2019) and NSE comes into the picture around 1994. There are approximately 5000 companies registered in BSE (Chauhan & Kaur, 2017) on the other end in NSE there are approximately 1600 companies registered (Chauhan et al., 2010). Top 30 companies of BSE decide Sensex and top 50 companies of NSE decides Nifty (Idrees, 2019).

In similar, a stock market index can help an investor to pick the right stock for investment. Comparably, we can say that, that there are many factors which can decide the fate or price of stocks. In, current scenario of share market, an investor must be aware about these facts before investing. In this paper we are going to discuss few fundamental factors which can decide upon the future the price of any stock or predict the price of new stock. In general share prices are affected with economical fundamental factors of an organization. Few of them are as follow such as: plans and policies of any company, technical structure of company, staffing policies, change in government policies for certain industries, positive or negative news in market, customer satisfaction with the products and may more similar things. So, an investor must keep a vigilant mind to apply these fundamental factors while investing in stock market. In current study of approach, the emphasis is to predict the future stock price of organization while implementing machine learning techniques so investors can get an appropriate knowledge of stock price before investing. By using these machine learning techniques an investor can increase its profitability and reduce the chances to lose money (Alzaman, 2023; Dai et al., 2020; Dai & Zhu, 2020; Weng et al., 2018).

In past there exists several machine learning algorithms which are implemented with time series data. Moreover, time series data has certain complexity where all machine learning algorithms are unable to handle this complex nature of data. Certainly, few algorithms such as Support vector machine, ARCH (Autoregressive Conditional Heteroskedasticity), GARCH (Generalized Autoregressive Conditional Heteroskedasticity), Random-Forest tends to be appropriate for time series model.

Fortunately, the machine learning algorithms has changed the scenario from past decade while evolving itself from large to big data in wide application domain. In the current study of approach, we have designed and implemented a framework for time series data, where the focus of study is to retrieve patterns for future prediction. The study is based on a regression-based technique which is an ARIMA model and LSTM model applies deep learning technique, we can say that it is an enhanced version of Recurrent Neural Network with inbuilt memory buffer (Jain et al., 2018). Moreover, ARIMA model can be discussed as a combination of two different timeseries forecasting models AR & MA (Li & Chiang, 2013), where AR stands for Autoregressive and MA stands for Moving Average. ‘I’ stands for Integrated and it is used as a differencing term in this model (Li & Chiang, 2013).

To implement the current study of approach we have utilized python language on Jupyter lan. The hardware chosen for prototype development was on operating system windows 10 with 8GB RAM. Several packages were installed to instantiate the development of framework such as scikitlearn, Plotly, matplotlib, pandas and Jupyter dash for visualization of data. Further, the dataset was captured from State Bank of India (SBI) from January 2013 to December 2018, where each data has significant opening and closing price of stock market (Finance, n.d.). The result predicts the overall accuracy of the algorithm which can be widely applied for real world prediction among the stock market analysis

The remainder of the paper can be discussed as Section II imparts the knowledge of stock market prediction in the past, also numerous literature review are cited. Section III discuss about the framework designed to access prediction of stock market where the entire schema is elaborated about the study. Section IV deliberated the overall implementation of the study with retrospective results. Lastly, paper is concluded and discussed about the future prospects of the study.

RELATED WORK

Forecasting in financial research is quiet anticipated area of research from past decades, a lot of research work has been proposed by the researchers for future forecasting. To predict the knowledge, ARIMA model has been applied to forecast Indian stock market. An ARIMA model is discussed as a univariate model for time series forecasting (Idrees, 2019). In context, to same several other techniques are available for forecasting which include Artificial Neural Network i.e. MLP, CNN and LSTM were used to predict the stock market price and found LSTM outperform other two techniques (Di Persio & Honchar, 2006). A complex neuro fuzzy system (CNFS) has been integrated with ARIMA for prediction of stock price movement and depicted positive results (Li & Chiang, 2013). An FGL (Financial loss or gain) model has been proposed for the prediction of future electricity price of GenCo company, an integrated approach using Silhouette criterion and k-means clustering technique were applied to improve the prediction results (Doostmohammadi & Zareipour, 2017).

A framework has been proposed using "Two-Stream Gated Recurrent Unit" and "Sentiment Analysis" for prediction of short-term trend prediction of stock market (Minh, Sadeghi-Niaraki, Huy, Min, & Moon, 2018). Over- fitting of data is very common problem of Deep Neural Network, a novel technique Dropout is introduced to deal with overfitting problem (Srivastava, Hinton, Krizhevsky, & Sutskever, 2014). A Fully Convolutional Network (FCN)is augmented with LSTM and a model is introduced FCN- LSTM for prediction of time series data and found the results better than some state of art algorithms (Karim et al., 2016).

A fuzzy method algorithm was applied for prediction of time series data; it is basically inherited the features of Japanese Candlestick theory used for assisting financial prediction (Lee et al., 2006). A comparative study of Discreet Wavelet Transform (DWT), ARIMA & RNN for predicting the traffic over computer network (Madan & Mangipudi, 2018). A novel Prophet algorithm is introduced to work with time series data for future prediction, this algorithm considers various type of seasonality in future prediction (Taylor & Letham, 2017). A new method based on convolutional neural network to simplify noisy-filled financial temporal series via sequence reconstruction by leveraging motifs is used for prediction of stock price and found that the results are 4% to 7% better than the traditional signal processing methods (Wen et al., 2019). A new integrated approach which combines the CNN and LSTM and makes a new technique Conv1D- LSTM, it is used to make prediction on the stock price two Indian origin companies TCS and MRF and found the good results (Jain et al., 2018). Another research that uses a generalized regression neural network (GRNN), which was applied to automate the time series prediction process (Yan, 2012). Again, different deep learning techniques, autoencoder and restricted Boltzmann machine are applied over Chinese stock market and compare the results with other machine learning algorithms in the same area (Chen et al., 2018). A similar approach related to the Prophet algorithm for prediction of future sales in retail sector was concluded to represent that Prophet algorithm works well with this kind of data (Žunić et al., 2020). The series of deep learning techniques, in this paper author compares two deep learning techniques CNN and LSTM over Indian stock market data and found that LSTM outperforms CNN with time series data (Kumar et al., 2020). An optimized heterogeneous structure of LSTM model has been proposed by the author for prediction of electricity prices, where SMBO technique has been utilized for optimization of hyperparameters (Zhou et al., 2019). A CNN-LSTM based model has been proposed by the authors, it is a hybrid technique where firstly CNN is applied for feature extraction and then LSTM applied for prediction (Lu et al., 2020). A CNN-LSTM based model has been proposed for time series prediction, where Grey Wolf Optimizer (GWO) is used to optimize learning hyperparameters (Xie et al., 2020). A hybrid model is built using LSTM and Genetic Algorithm to predict Korean stock prices, this research mainly focused on the temporal nature of stock market data and also fix a time window size (Chung & Shin, 2018). An LSTM neural network has been applied for emotion classification and at the same time Differential Evolution (DE) is also applied to optimize the LSTM hyperparameter, they have also compared the result of proposed method with existing optimizing techniques like PSO and Simulated Annealing (Nakisa et al., 2018). A feature engineering and deep learning-based model for predicting stock price trend was proposed by the author's, developed model has been compared with existing machine learning techniques and found better then all of them (Shen & Omair Shafq, 2020). Paper shows the study of seasonal effect on share price, they have identified monthly, yearly and weekly seasonal effect on share prices (Kushwah & Munshi, 2018; Wheeb, 2017).

METHODOLOGY

In current study of approach, we have implicated the proposed model and detect the information while adopting the architecture design for prediction.

Architecture for Prediction of Stock Market

The proposed architecture is designed and implemented into varied sections to assure the prediction is guaranteed for real world databases, and can be utilized by forecasting specialists for future decision making. First section of proposed model is focused on pre-processing of data where the focus is to extract the information while removing the missing and the redundant values from the data. Secondly, we adopted the algorithmic power of the varied machine learning models using LSTM and ARIMA to detect the hidden patterns from time series databases and lastly, we configure to detect the decision making. Figure 1depicts the overall architecture of the same.

Raw Database

In current study of approach, we have applied machine learning algorithms ARIMA and LSTM for prediction of future stock prices of SBI (State Bank of India). The real-world data was gathered for 6 years out of which maximum data is used for training purpose and only 60 days data is used for testing purpose. Based on training data and correctness of model we have forecast the share price of SBI for next 60 days. The consideration of predicting data for 6 years time spam was formulated as the accurate time series prediction analysis for each forecast time can be measured with at least 5 years' time. Also, utilizing the most recent data will be able to achieve the most recent trends.

Encoding of Data

As, we know the data is getting accumulated with high speed, we require high end technology which can handle inconsistent, missing values and redundant nature of the data. Likely, data preprocessing play the major role where the focus is to assure the datasets are free from any inconsistencies in the data. Moreover, inconsistencies among the data can detect the fraudulent results, which further can prove to be detrimental for future decision making. In, current study of approach we have implemented the pre-processing of data to assure that data gathered from 6 years should not have redundant and missing values. The data gets encoded to assure that we get the quality of data which can widely implemented using the machine learning algorithms for detection of information.

Implemented Technology

As, we have applied the data preprocessing techniques on the databases, we further adopted varied machine algorithms to discover the ground knowledge of data. While, predicting the actual stock price for future is a trivial task we discuss two different optimized techniques such as the first algorithm is

Table 1. Previous Literature Work

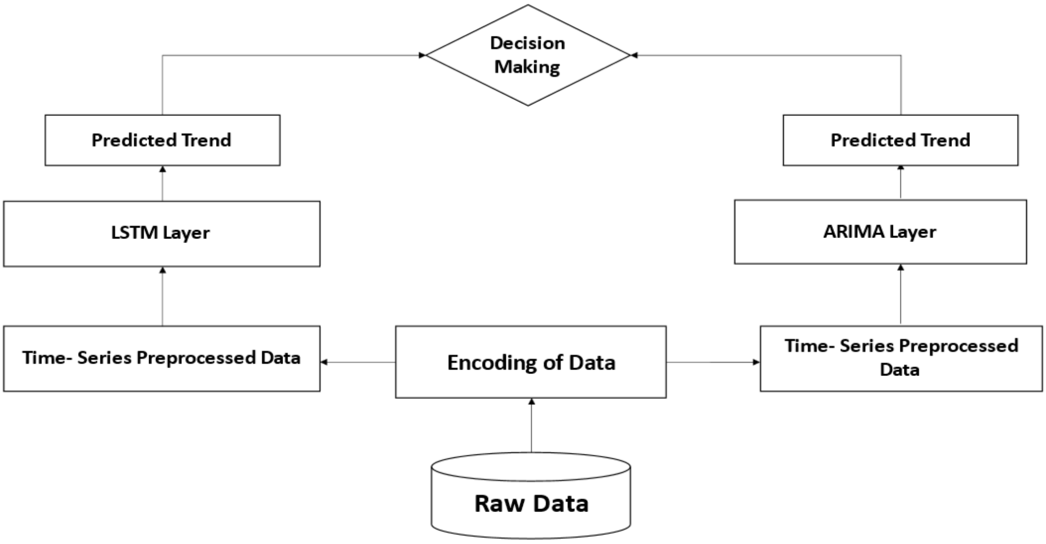
S.No.	Author	Work	Approach of Study	Proposed Approach
1.	Di Persio and Honchar (2006)	Artificial neural networks approach to the forecast of stock market price movements	It mentions the application of different artificial neural network approaches but lacks a detailed comparison between them. It states that convolutional neural networks (CNN) can model financial time series better than other architectures, but it does not provide specific performance metrics or comparisons with alternative models.	Our research fills this gap by presenting a detailed comparison between two specific models, ARIMA and LSTM, based on various performance parameters such as MAE, MSE, RMSE, and Bias. This allows for a clearer understanding of the strengths and weaknesses of each model, which is lacking in the previous work.
2.	Idrees et al. (2019)	A Prediction Approach for Stock Market Volatility Based on Time Series Data	It suggests that the ARIMA approach is good enough for handling time series data but does not discuss any identified limitations or specific areas for improvement based on the current study's findings.	Our study acknowledges the superior performance of the ARIMA model over LSTM in this particular study, implying the identification of limitations and potential areas for improvement. This acknowledgment could lead to further investigation into improving LSTM models or exploring other forecasting techniques.
3.	Minh et al. (2018)	Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network	This study mentions the overall accuracy of the proposed TGRU model and its outperformance compared to other models, it lacks specific performance metrics such as MAE, MSE, or RMSE. It also does not provide a detailed empirical evaluation of the model's performance.	Our research provides empirical evidence to support the comparison between ARIMA and LSTM models, offering specific values for various performance metrics. This detailed information enhances the clarity and reliability of the reported findings.
4.	Srivastava et al. (2014)	Dropout: A Simple Way to Prevent Neural Networks from Overfitting	This research focuses on dropout technique and its application in improving neural networks across various domains but does not specifically address time series data analysis or forecasting.	Our study specifically discusses the design and implementation of a framework for time series data analysis, with a focus on predicting future patterns. This fills the gap in addressing the specific challenges and techniques relevant to time series forecasting.
5.	Karim et al. (2018)	LSTM Fully Convolutional Networks for Time Series Classification	It discusses the wide-ranging applicability of LSTM-FCNs on various sequence modeling tasks such as text analysis, music recognition, and voice detection. However, it does not specifically address the application of these models in the context of financial time series forecasting or stock market prediction.	Our study focuses specifically on designing and implementing a framework for time series data, with the goal of predicting stock price movements. It addresses the practical implications of using different models for financial forecasting, providing insights into their potential applications in real-world scenarios.

ARIMA and second algorithm is LSTM, to detect patterns which can be further utilized for meaningful analysis of data for future decision making.

Algorithm 1

ARIMA (Autoregressive Integrated Moving Average): ARIMA is the one of the best model to work with time series data. It is basically the combination of two different models AR & MA, and these are quite powerful models, once you combine both of them that makes the ARIMA model.

Figure 1. Architecture for Prediction of Stock Market

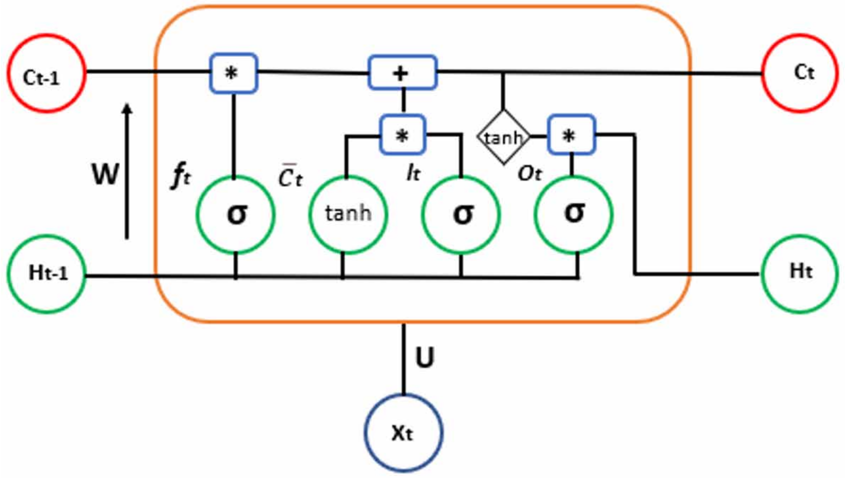


AR stands for Autoregressive part and MA stands for moving average. So "AR" is a separate model and "MA" is a separate model and what's binds it together is the integration part that is indicated by "I". AR is nothing but the correlation between previous time period to the current. Let say there is a current time period t and t-1 and t-2 are previous time periods, now if you find any correlation between t-2 and t that is Autoregressive part. There may be some kind of noise or irregularity attached in time series, so we need to ascertain that noise in fact we need to average that out. So, whenever we averaged it out the cross and drops set of prison in that noise that can be smoothen out and we can have average focused of that noise. Before applying the ARIMA model over data, the data should be stationary means constant mean and constant variance. There are two popular tests to check whether data is stationary or not Rolling Statistics & Augmented Dickey Fuller Test (ADCF). In this paper we have used ADCF test of stationarity. If the data is not stationary, we must make it stationary before applying ARIMA over it. ARIMA model has three parameters, it has p it has q and it has d. So, p basically refers to your Autoregressive lags and q stands for Moving Average and d is the order of differentiation (Li & Chiang, 2013). So we have each parameter for each of the model, so if we take the integration of order one so the value of d would be one if we differentiate in order of two so the value of d would be two so that is how we can predict these values p, d & q and each of them has a different method to it. So if we want to predict the value of p so we have to use PACF (Partial Auto Correlation) Graph, then to predict q value we need to use ACF (Auto Correlation) Graph and d is use to make data stationary using some kind of differentiation. So, the order of differentiation defines the value of d. After converting the data to stationary the formula of ARIMA will looks like this:

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} \ldots\ldots \varphi_p y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} \ldots\ldots - \theta_q e_{t-q} \quad (1)$$

In the above formula p is the order of AR model, q is the order of MA model and e is the white noise, φ and θ are model parameters.

Figure 2. Architecture for LSTM



Algorithm 2

LSTM (Long Short-Term Memory): LSTM is a type of RNN (Recurrent Neural Network) RNN's requires sequential data that depends on the past values, what makes the RNN models capable enough to work with the sequential data is number of states used in this model (Kumar et al., 2020). This is nothing but the temporary memory that stores the previous data. You can see in the fig.1. general RNN structure, in the next few lines we are trying to understand how this RNN works.

Traditional RNN's are not good at capturing long range dependencies means when we tend to work with very huge data set with RNN we are at the risk of vanishing gradient problem. What is vanishing gradient problem? When you train a deep neural network then gradient or the derivative decreases exponentially as it propagates down the layer this is known as vanishing gradient problem. These gradients are actually used to update the weights of neural network but when the gradient vanish these weights will not be updated in the worst case it may completely stop the neural network from training it is very common issue in very deep neural networks. So, to overcome this vanishing gradient problem in RNN's, LSTM was introduced (Chen et al., 2018). It is basically a modification to the RNN's hidden layers, LSTM can remember RNN's weights and their inputs over a very long period of times. In LSTM in addition to the hidden state cell state is passed down to the next block, LSTM can capture long term dependencies. It can have memory of previous inputs for very extended time duration the way LSTM cell does this is by using three main gates, first one is the forget gate. Forget gate removes the information that is no longer useful in cell state then we have input gate additional information to the cell state is added by input gate and finally we have something called as output gate, additional useful information to the cell state is also added by an output gate this gating mechanism has allowed network to learn the condition for when to forget ignore or keep information in the memory cell. Fig. 1 shows a typical LSTM network with three gates and *tanh* activation function, where:

Xt = Input vector
Ht-1 = Previous cell output
Ct-1 = Previous cell memory
Ht = Current cell output
Ct = Current cell memory
* = Element wise multiplication

+ = Element wise addition
W,U = Weight vectors for input(i), output(o) & forget(f) gates

$$f_t = \sigma\left(X_t * U_f + H_{t-1} * W_f\right) \tag{2}$$

$$I_t = \sigma\left(X_t * U_i + H_{t-1} * W_i\right) \tag{3}$$

$$O_t = \sigma\left(X_t * U_o + H_{t-1} * W_o\right) \tag{4}$$

$$\overline{c}_t = tanh\left(X_t * U_c + H_{t-1} * W_c\right) \tag{5}$$

$$C_t = f_t * C_{t-1} + I_t * \overline{c}_t \tag{6}$$

$$H_t = O_t * \tanh\left(C_t\right) \tag{7}$$

Eq. 2,3 & 4 represents forget, input and output gates respectively, apart from these we have another vector shows in eq.5 that is use to modify the cell state it has the *tanh* activation function, tanh is used to avoid gradient vanishing problem. Eq 6 is applied over output gate to obtain the hidden vector, eq. 7 gives us the final output of the LSTM network.

Prototype of Algorithm

To implement the current study of approach we have utilized python language on Jupyter lan. The hardware chosen for prototype development was on operating system windows 10 with 8GB RAM. Several packages were installed to instantiate the development of framework such as scikitlearn, Plotly, matplotlib, pandas and Jupyter dash for visualization of data (Mckinney, 2010; Plotly, n.d.-a; Plotly, n.d.-b; PyPI, n.d.). Further, the dataset was captured from State Bank of India (SBI) from January 2013 to December 2018, where each data has significant opening and closing price of stock market. In Table 2, a representative psudo codefor stock market prediction with annotated technology is discussed.

RESULTS AND CONCLUSION

In, current study of approach we have applied both LSTM and ARIMA model for specific period of 6 years where a comparison study is obtained to predict future stock prices. The result obtained reveals the overall performance of each model in representative time series data.

Table 2. Pseudo Code of Implemented study

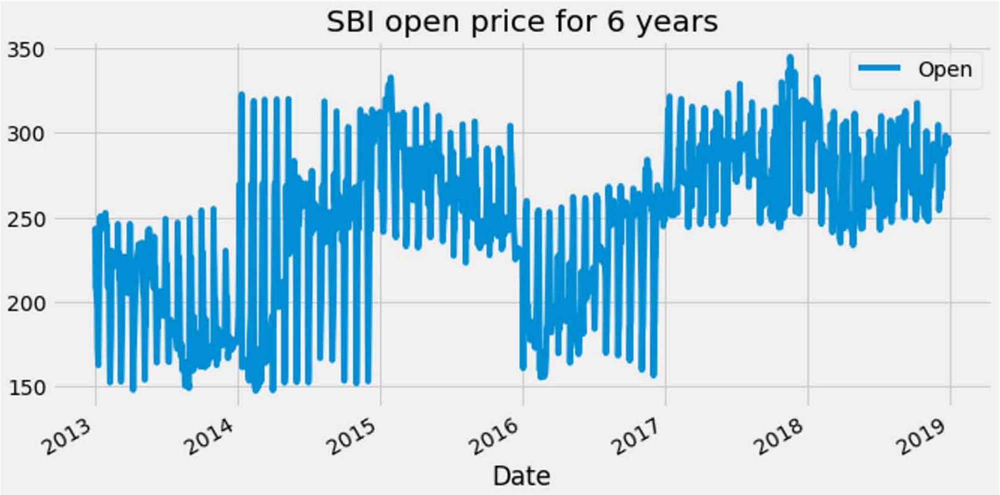
```
for e in stock_ list_training // The list is read for input of training data
    Construct the data frame for the input training data
end for
for t in stock_list_testing // The list is read for input of testing data
    Construct the data frame for the input testing data
end for
Created a list empty list for prediction
for k in range length of stock_list_training
    Create ARIMA model with order (1,1,1)
    Create Sequential LSTM model with RELU activation and 100 neurons in each layer
    Predict the stock prices for next 60 days
    Compare the predicted stock price with test data for ARIMA & LSTM
    Calculate the error rate in prediction
    Save the extracted information in CSV files
end for
```

Implementation of LSTM model: The LSTM model has been measured with varied epochs parameter, to obtain the results which are relevant for future prediction modelling. The data has been modulated for 6 years data which is then gathered from SBI from Jan 2013 to Dec 2018. The data has been distributed among the training and testing purpose, out of which records 1-1416 are used for training purpose and 1417 – 1477 are used for testing purpose.

```
Algorithm for LSTM:
dataset→SBI data for 6 years
dataset→Dataset['Date','Open' ]
len(dataset)
x= len(dataset)-60
train=dataset.iloc[:x]
test= dataset.iloc[x:]
train_scaled = scaler.transform(train)
test_scaled = scaler.transform(test)
seq_size = 60
n_features = 1 .
model = Sequential()
model.add(LSTM(100,activation='relu',return_sequences=True, input_
shape=(seq_size, n_features)))
model.add(dropout(0.2))
model.fit_generator(train_generator,
                              validation_data=test_generator,
                              epochs=150, steps_per_epoch=10)
future = 60
predicted_stock_price = Sequential.predict(test)
predicted_stock_price_inverse = sc.inverse_transform(predicted_
stock_price)
```

Further, a sequential LSTM model has been trained with 1 input and output layer with 3 hidden layers. In each layer 100 neurons are used, and relu activation function used. While, each layer has a dropout ratio which is set to be 20 percent, also lookback sequence size is set to 60 and number of features extracted is set to 1. Number of epochs used are 150 and steps in each epoch are set to 10.

Figure 3. The Overall SBI Open Price



Further, Fig.4 represents the overall validation loss in each epoch. The graph clearly validates the lower tend in the loss when the number of epochs is increased. We can say that, validation loss is inversely proportion to the predicting capability of a model. Further, If the loss is more prediction will be less accurate and vice versa.

Fig.5 represents the overall prediction of share price along with the data reserved for testing purpose. Whereas, Green line predicts the 60 days data usage for testing purpose and red line represent the prediction of 60 days with the designed model.

Further, the study was implemented using ARIMA model where the similar data from Jan 2013 to Dec 2018 is applied to retrieve the prediction for another forthcoming years.

Figure 4. Validation Loss With Training Loss

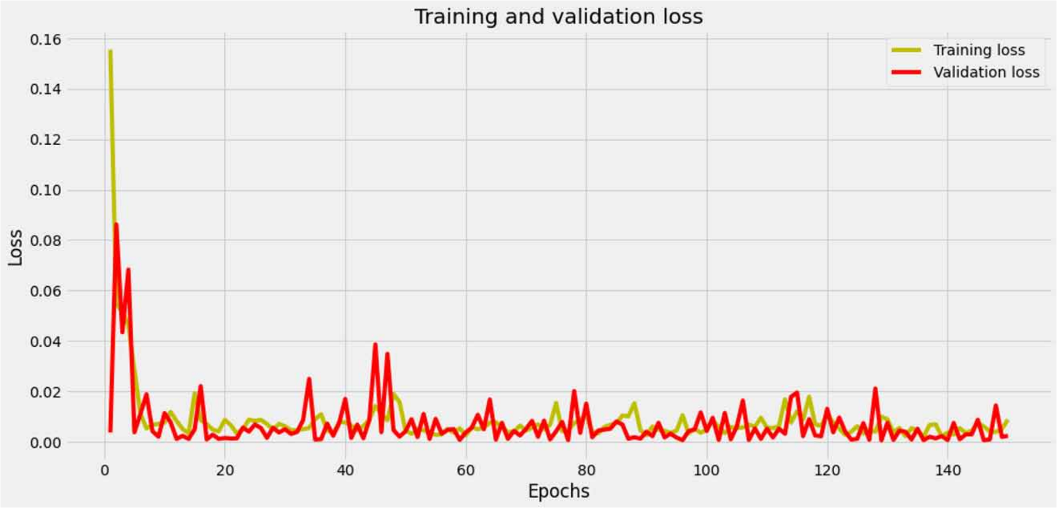
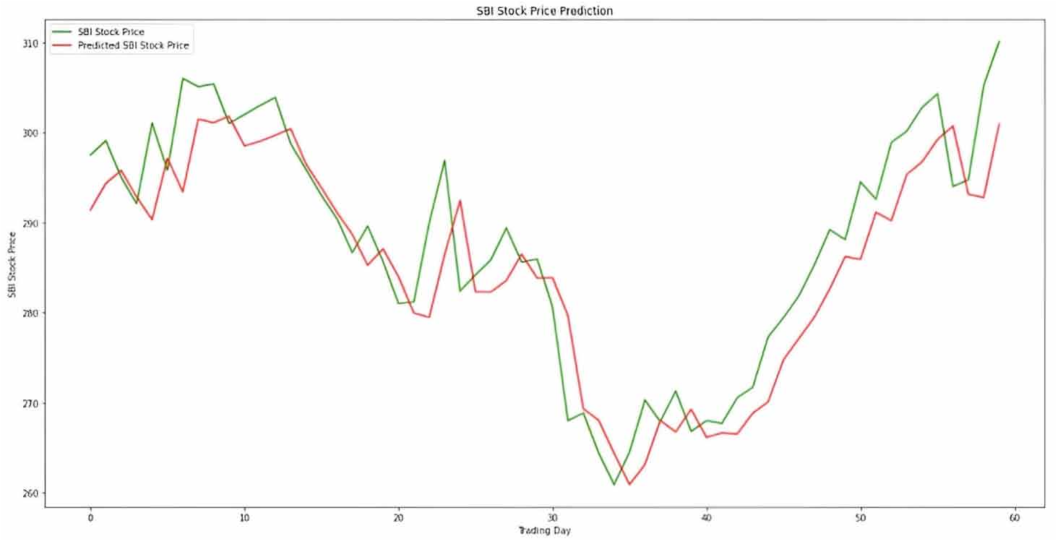


Figure 5. SBI Stock Price Prediction



```
ARIMA Algorithm:
series →SBI 6 years training data
test → SBI 6 years test data
X → series.values
Y → test.values
size_train → int(len(X))
size_test → int(len(Y))
train → X[0:size_train]
test → Y[0:size_test]
history → [x for x in train]
predictions → list()
for t in range(len(test)):
 model → ARIMA(history, order=(1,1,1))
  model_fit → model.fit(disp=0)
  output → model_fit.forecast()
  yhat → output[0]
  predictions.append(yhat)
  obs → test[t]
  history.append(obs)
  print → (yhat, obs, (yhat-obs))
```

To identify whether data is stationary or not there are two ways one is rolling mean and other is standard deviation, we can also implicate a dickey fuller test to measure the stationarity of the data. In, current study of approach we have applied both the techniques to check the stationarity of data. We have chosen 1 year rolling mean and one year rolling standard deviation. In the fig. 6, data vibrant distinguishes the seasonality and trend in the data as the mean value of data goes shows up and down trends. We have also implicated the stationarity with dickey fuller test, in dickey fuller test we obtained the p value that is 0.25 which is higher than 0.05, which proves that data is stationary while the p value must be $p<0.05$. So, we can say our data is nonstationary, we must make that data stationary before passing it through the ARIMA model. There are various ways to make data stationary but, in this paper, we have applied differencing method by shifting of one value down. After getting the differencing, again we have checked the stationarity of differenced data and its value is 0.0, which is less than 0.05 now our data is stationary and we can pass it through the ARIMA model for making predictions over it.

Further, for the time series analysis, we have to separate the trend and seasonal component from the series. In the fig 6 a representative graph visualize a separated trend and seasonal component among the data. Further, we have calculated the ACF (Autocorrelation Function), which measures the correlation of a variable with its lagged values, also it can be applied to decide the MA or q value of ARIMA model. PACF (Partial Autocorrelation Function) graphs show the auto correlation after removing the relationship with the previous lags and it is also used to identify AR or p value in ARIMA model.

In, Fig 7 an implicated graph represents the 1 year rolling mean and 1-year standard deviation, it is used to check whether series is stationary or not. In general, for calculation of rolling mean we have engaged the window size of 230 to check yearly seasonality in the database. Further, we have assumed that in one year the share market may open at most 230 days.

In, Fig 8 a representative series was implicated where it was broken down into 3 components: trend, seasonality, and residual. Trend is upward and downward of the series over time, whereas seasonality is seasonal variance of in series, residual is the sudden spikes and throughs at random intervals.

Further, we have applied the one order differencing in the series and check the seasonality with differenced data with augmented dickey fuller test and found that there is no more seasonality in the data. We have shown the results in fig.9, which clearly indicates that the graph is almost stationary.

Figure 6. Autocorrelation Correlation in Dataset

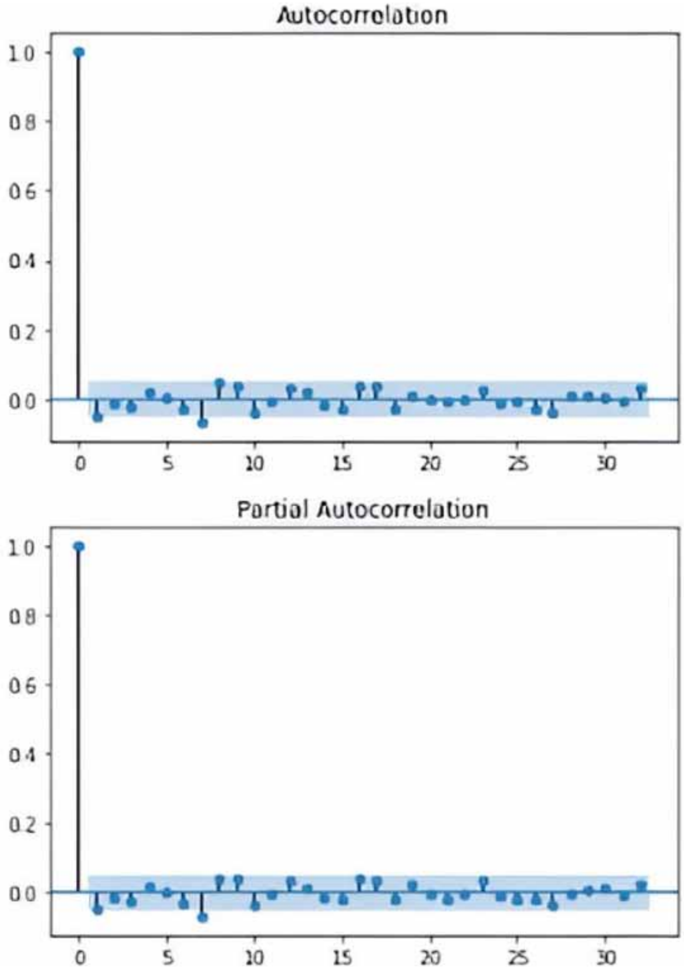


Figure 7. Prediction for 1 Year

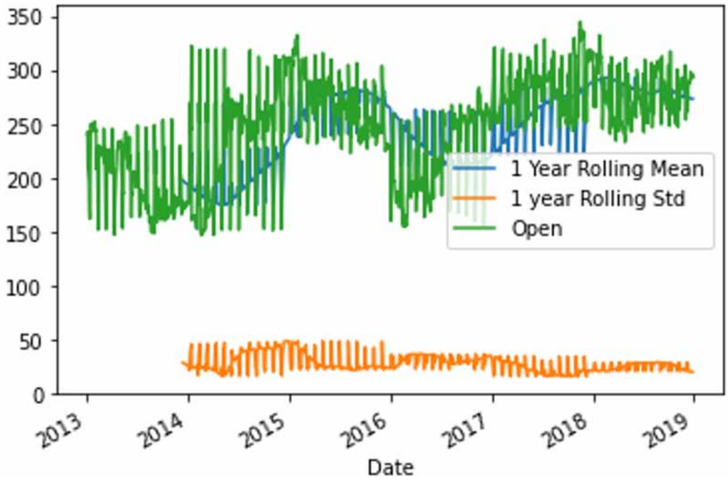


Figure 8. Series for Prediction

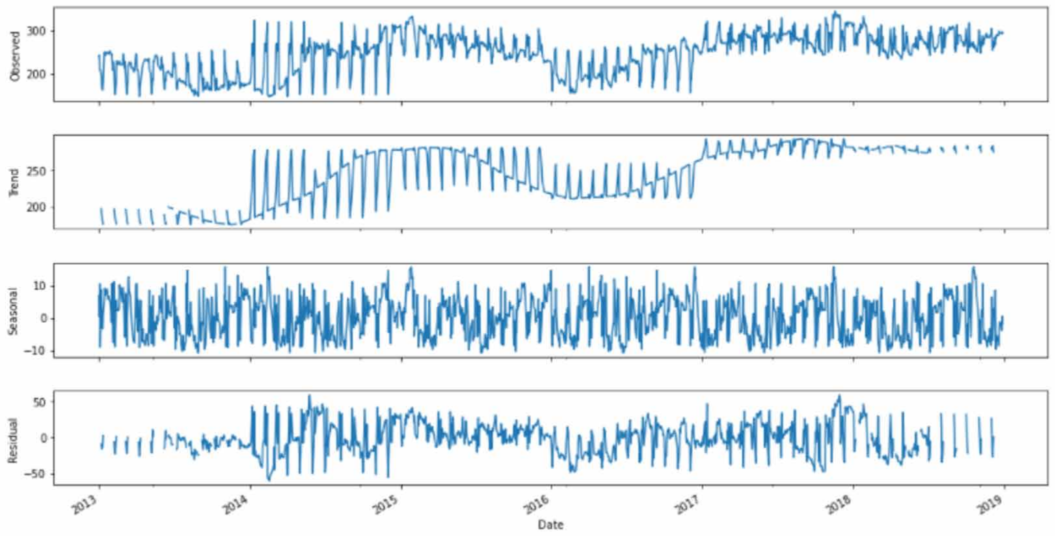


Figure 9. Stationary Graph

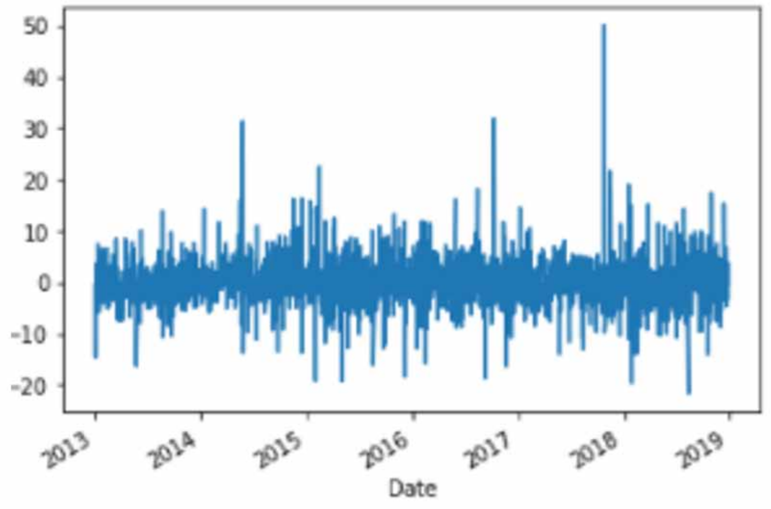
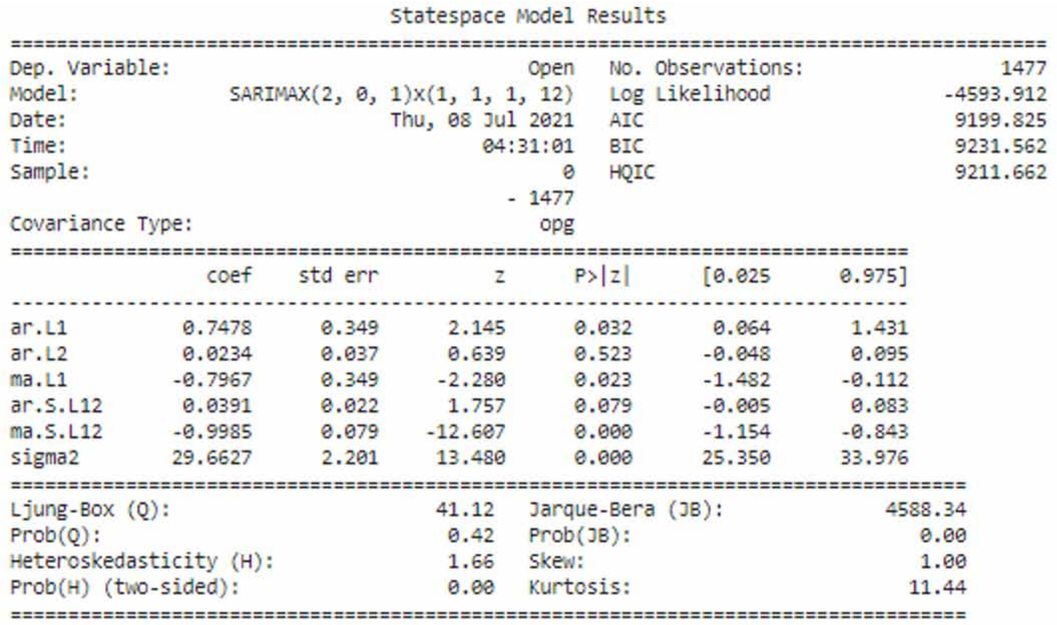


Fig 10 illustrates the comparison graph between actual share price and predicted share price for 60 days. LSTM model predict the share price for next 60 days based on training data, also we have reserved 60 days test data for comparison with the predicted data which we have plot in the fig 9.

Further, the Fig 10 represents the overall summary of trained model, where the order has been formulated for p, d & q is 2,0,1. We have suggested to choose the static values to have least AIC value, whereas several other random values was taken but the chosen values prove to have better predicted values.



```
                             Statespace Model Results
==========================================================================================
Dep. Variable:                               Open   No. Observations:                 1477
Model:             SARIMAX(2, 0, 1)x(1, 1, 1, 12)   Log Likelihood               -4593.912
Date:                            Thu, 08 Jul 2021   AIC                           9199.825
Time:                                    04:31:01   BIC                           9231.562
Sample:                                         0   HQIC                          9211.662
                                           - 1477
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.7478      0.349      2.145      0.032       0.064       1.431
ar.L2          0.0234      0.037      0.639      0.523      -0.048       0.095
ma.L1         -0.7967      0.349     -2.280      0.023      -1.482      -0.112
ar.S.L12       0.0391      0.022      1.757      0.079      -0.005       0.083
ma.S.L12      -0.9985      0.079    -12.607      0.000      -1.154      -0.843
sigma2        29.6627      2.201     13.480      0.000      25.350      33.976
===================================================================================
Ljung-Box (Q):                       41.12   Jarque-Bera (JB):              4588.34
Prob(Q):                              0.42   Prob(JB):                         0.00
Heteroskedasticity (H):               1.66   Skew:                             1.00
Prob(H) (two-sided):                  0.00   Kurtosis:                        11.44
===================================================================================
```

In, Table 3 we tried to retrieve the error rate of both LSTM as well as ARIMA model, to assure which model predicts the effective and efficient analysis of real time data. The data illustrates that ARIMA model which has better MAE (Mean Absolute Error) value than LSTM which is 4.32, whereas for ARIMA it is 3.83. Also, MSE (Mean squared Error) value for LSTM is 29.52 for ARIMA we found it 24.53 and RMSE (Root Mean Squared Error) for LSTM and ARIMA are 5.43 and 4.95 respectively, whereas Bias value for LSTM is 2.40 and for ARIMA it is 0.16. LSTM represents very less error values but ARIMA shows better result in every aspect as compare to the LSTM

CONCLUSION

Machine learning algorithms has changed the scenario from past decade while evolving itself from large to big data in wide application domain. In the current study of approach, we have designed and

Figure 10. Comparative Graph Between Actual and Predicted Price

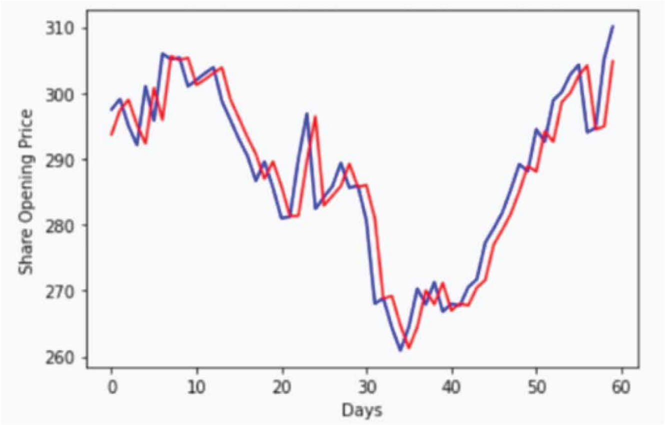


Table 3. Represent Error Matrix of LSTM and ARIMA

MODEL	LSTM	ARIMA
MAE	4.32	3.83
MSE	29.5	24.5
RMSE	5.43	4.95
Bias	2.4	0.16

implemented a framework for time series data, where the focus of study is to retrieve patterns for future prediction. The study is based on a regression-based technique which is an ARIMA model and LSTM model applies deep learning technique, we can say that it is an enhanced version of Recurrent Neural Network with inbuilt memory buffer Empirically we have found the various performance parameters such as MAE, MSE, RMSE and Bias for both LSTM as well as ARIMA. Table 1 represents the performance matrix of LSTM and ARIMA respectively, if we compare the ARIMA model which has better MAE (Mean Absolute Error) value than LSTM which is 4.32, whereas for ARIMA it is 3.83. Also, MSE (Mean squared Error) value for LSTM is 29.52 for ARIMA we found it 24.53 and RMSE (Root Mean Squared Error) for LSTM and ARIMA are 5.43 and 4.95 respectively, whereas Bias value for LSTM is 2.40 and for ARIMA it is 0.16. LSTM represents very less error values but ARIMA shows better result in every aspect as compare to the LSTM.

FUTURE WORK

The current scope of work is focused on ARIMA and LSTM model for prediction of stock. Moreover, the future of the proposed approach will be focused on developing an environment where deep learning algorithm will be applied to predict future of stock price. Hence, by combining the new area of technology such feature engineering, quantum computing and other diverse domain, there tends to be a high potential for a user-based system which can comprehend the diverse types of stock prediction.

REFERENCES

Alzaman, C. (2023). Forecasting and optimization stock predictions: Varying asset profile, time window, and hyperparameter factors. *Systems and Soft Computing, 5*. 10.1016/j.sasc.2023.200052

Chauhan, R., & Kaur, H. (2017). A Feature Based Reduction technique on Large Scale Databases. *International Journal of Data Analysis Techniques and Strategies.*, *9*(3), 207. doi:10.1504/IJDATS.2017.086630

Chauhan, R., Kaur, H., & Alam, A. M. (2010). Data Clustering Method for Discovering Clusters in Spatial Cancer Databases. *International Journal of Computer Applications, 10*(6), 9-14.

Chauhan, R., Kaur, H., & Chang, V. (2017). *Advancement and applicability of classifiers for variant exponential model to optimize the accuracy for deep learning. Journal Ambient Intelligent Human Computing*. doi:10.1007/s12652-017-0561-x

Chauhan, R., Kaur, H., & Chang, V. (2020). An Optimized integrated Framework of Big Data Analytics Managing Security and Privacy in Healthcare Data. *Wireless Personal Communication*.

Chen, L., Qiao, Z., Wang, M., Wang, C., & Du, R. (2018). *Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?* (Vol. 6). IEEE Access.

Chung & Shin. (2018). Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction. MDPI Journal, Sustainability.

Dai, Z., Zhou, H., Wen, F., & He, S. (2020). Efficient predictability of stock returnvolatility: The role of stock market implied volatility. *The North American Journal of Economics and Finance*, *52*, 101174. doi:10.1016/j.najef.2020.101174

Dai, Z., & Zhu, H. (2020). Stock returns predictability froma mixed model perspective. *Pacific-Basin Finance Journal*, *60*, 101267. doi:10.1016/j.pacfin.2020.101267

Di Persio, L., & Honchar, O. (2006). Artificial neural networks approach to the forecast of stock market price movements. *International Journal of Economics and Management Systems*, *1*.

Doostmohammadi, A., & Zareipour, H. (2017, September). Day-Ahead Financial Loss/Gain Model- ing and Prediction for a Generation Company. *IEEE Transactions on Power Systems*, *32*(5), 3360–3372. doi:10.1109/TPWRS.2017.2654363

Explorator.io. (n.d.). https://blog.exploratory.io/an-introduction-to-time-series-forecasting-with-prophet-package-in-exploratory-129ed0c12112

Finance. (n.d.). https://finance.yahoo.com/quote/SBIN.NS/

Github. (n.d.). https://github.com/facebook/prophet

Idrees, S. M. (2019, February). A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access : Practical Innovations, Open Solutions*. Advance online publication. doi:10.1109/ACCESS.2019.2895252

Idrees, S. M., Alam, M. A., & Agarwal, P. (2019, February). A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 17287–17298. doi:10.1109/ACCESS.2019.2895252

Jain, Gupta, & Moghe. (2018). Stock Price Prediction on Daily Stock Data using Deep Neural Networks. *International Journal of Neural Networks and Advanced Applications, 5*.

Karim, Majumdar, Darabi, & Chen. (2016). LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access, 6*.

Karim, F., Majumdar, S., Darabi, H., & Chen, A. S. (2018). LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 2018. doi:10.1109/ACCESS.2017.2779939

KDNugets. (2018). https://www.kdnuggets.com/2018/11/sales-forecasting-using-prophet.html

Kumar, N., Chauhan, R., & Dubey, G. (2020). *Applicability of Financial System Using Deep Learning Techniques. Ambient Communications and Computer Systems. Advances in Intelligent Systems and Computing* (Vol. 1097). Springer.

Kushwah, S. V., & Munshi, S. (2018, December). The Effect of Seasonality Over Stock Exchanges In India, Amity. *Journal of Management*, *6*.

Lee, C.-H. L., Liu, A., & Chen, W.-S. (2006, May). Pattern Discovery of Fuzzy Time Series for Financial Prediction. *IEEE Transactions on Knowledge and Data Engineering*, *18*(5).

Li, C., & Chiang, T.-W. (2013, June). Complex Neurofuzzy ARIMA Forecasting—A New Approach Using Complex Fuzzy Sets. *IEEE Transactions on Fuzzy Systems*, *21*(3), 567–584. doi:10.1109/TFUZZ.2012.2226890

Lu, W., Li, J., Li, Y., & Sun, A. (2020). *A CNN-LSTM-Based Model to Forecast Stock Prices. Wiley*.

Madan, R., & Mangipudi, P. S. (2018). Predicting Computer Network Traffic: A Time Series Forecasting Approach using DWT, ARIMA and RNN. *Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3).*

Mckinney, W. (2010). *Data Structures for Statistical Computing in Python*. Academic Press.

Minh, , Sadeghi-Niaraki, , Huy, , Min, , & Moon, . (2018). Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network. *IEEE Access : Practical Innovations, Open Solutions*, *6*.

Nakisa, Rastgoo, Rakotonirainy, Maire, & Chandran. (2018). Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework. *IEEE Access, 6*.

Plotly. (n.d.-a). *Dash Documentation & User Guide*. https://dash.plotly.com/

Plotly. (n.d.-b). *The front end for ML and data science models*. https://plotly.com/

PyPI. (n.d.). https://pypi.org/project/ares/

Shen, J., & Omair Shafq, M. (2020). *Short-term stock market price trend prediction using a comprehensive deep learning system*. Springer Open Journal of Big Data. doi:10.1186/s40537-020-00333-6

Srivastava, N., Hinton, G., Krizhevsky, A., & Sutskever, I. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Wen, Li, Zhang, & Chen. (2019). Stock Market Trend Prediction Using High-Order Information of Time Series. *IEEE Access, 7*.

Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, *112*, 258–273. doi:10.1016/j.eswa.2018.06.016

Wheeb, A. H. (2017). Performance Analysis of VoIP in Wireless Networks. *International Journal of Computer Networks and Wireless Communications*, *7*(4), 1–5.

Xie, Zhang, & Lim. (2020). CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer. *IEEE Access, 8*.

Yan, W. (2012, July). Toward Automatic Time-Series Forecasting Using Neural Net- works. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(7). PMID:24807130

Zhou, S., Zhou, L., Mao, M., Tai, H.-M., & Wan, Y. (2019). An Optimized Heterogeneous Structure LSTM Network for Electricity Price Forecasting. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 108161–108173. doi:10.1109/ACCESS.2019.2932999

Žunić, E., Korjenić, K., Hodžić, K., & Đonko, D. (2020, April). Application Of Fa- cebook's Prophet Algorithm For Successful Sales Forecasting Based On Real-World Data. *International Journal of Computer Science and Information Technologies*, *12*(2).

Neeraj Kumar received B.Tech (CSE) degree from Uttar Pradesh Technical University, India and M.Tech (CSE) degree from Amity University, Uttar Pradesh, India. Currently he is pursuing Ph.D in Information Technology from Amity University Uttar Pradesh, India. He is having total 12 years of Teaching and Research experience. His research work includes Time series analysis using various regression based and deep learning techniques such as RNN, LSTM, CNN, ARIMA, VAR and Prophet Algorithm. He has also worked with various bagging and boosting techniques for time series forecasting such as Random Forest, Ada-Boost and XG-Boost.

Ritu Chauhan earned her Ph.D. in Computer Science from Jamia Hamdard University, New Delhi, India on the topic of Applications of Data Mining techniques in Spatial Databases. She graduated from the University of Delhi, New Delhi. She has previously served as a Lecturer in Computer Science, IMI. Currently, she is an Assistant Professor at Amity Institute of Biotechnology, Amity University She has published numerous research articles in refereed international journals and conference proceedings and chapters in an edited book. She is a member of several international bodies. Her main research interests are in the fields of Data analysis with applications to medical databases.