International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Stock Price Prediction using combination of LSTM Neural Networks, ARIMA and Sentiment Analysis

Omkar S. Deorukhkar[1], Shrutika H. Lokhande[2], Vanishree R. Nayak[3], Amit A. Chougule[4]

[1,2,3]Student, Department of Computer Engineering, L.E.S. G.V. Acharya Institute of Engineering and Technology, Shelu, Maharashtra

[4]Asst. Professor, Department of Computer Engineering, L.E.S. G.V. Acharya Institute of Engineering and Technology, Shelu, Maharashtra

---

Abstract – *Financial markets being highly volatile, there is a huge amount of uncertainty and risk associated with them. This paper presents an innovative method to predict next day closing prices of stocks using combination of deep learning approach using Long Short-Term Memory (LSTM), architecture of Recurrent Neural Networks (RNN), Auto Regressive Integrated Moving Average (ARIMA) time series model and Sentiment analysis model to predict next day closing prices of stocks. These models have been combined in a Feedforward Neural Network to give the final prediction. This approach of combining different methods is called as Ensemble Learning, which in majority of cases gives higher accuracy than using individual models.*

Key Words: **Deep Learning, LSTM, Recurrent Neural Networks, ARIMA, Sentiment Analysis, Feedforward Neural Network, Artificial Neural Networks, Ensemble Learning, Finance, Stock Market**

1. INTRODUCTION

Finance and Investment are the sectors, which are supposed to have exponential growth in coming era due to rise in consumerism, materialism and capitalist ideology worldwide. Investors have become quite cautious about their investments and expect high amount of returns in less time period. As conventional investment schemes, fail to give high returns in less time, investors have turned their attention towards financial markets. A financial market is a place where people trade financial securities and derivatives, like stocks, bonds, commodities and options, futures respectively. Financial markets give high rate of returns, conflated with equal amount of risk. Investors are classified into retail investors, the common man and institutional investors, professional trading institutions. The retail investor mostly invests in stocks, as trading in options and futures require in-depth knowledge and experience due to their complex and risky nature. Due to high volatility of the stock market, there is a high chance of losing money for the retail investors, pertaining to uncertainty of the price in future. Thus, this paper presents an innovative method, to predict the next day stock prices using a combination of deep learning, time series analysis and natural language processing for gaining maximum possible accuracy in the prediction.

Deep Learning is a subdomain of Machine Learning, which relies upon the use of Artificial Neural Networks (ANN) for mapping intuitions between features and labels. ANNs are further classified into various architectures with respect of application, of which prominent are Feedforward Neural Networks, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). This paper presents a methodology of using Long Short-Term Memory (LSTM) cells, a type of RNN, in combination with a time series model, called as Auto Regressive Integrated Moving Average (ARIMA), and a Sentiment Analysis model. The output of these three models are combined in a Feed forward Neural Network for predicting the final value of next day price. Thus, the system uses historic intuitions from ARIMA and LSTM, and current state of the market from Sentiment Analysis, bringing dynamicity and considering technical as well as fundamental aspects of the stock. The combination of these three approaches is called as Ensemble Learning and reduces the error of the prediction with respect to original prediction. The system will be of great use to retail investors who do inter-day trading as they would have a more precise estimate to next day prices. The individual models are described in their respective following sections in detail.

2. RELATED WORK

This section will throw some light on related works and research published in history in respect to concepts mentioned in this paper. A comprehensive study on effects of number of LSTM layer on prediction of the Chinese Stock Market Index, CSI 603899 was studied by Siyuan Liu, Guangzhong Liao, and Yifan Ding. It showed that more the layers, more is the accuracy of the model. A single layer of LSTM gave a sample accuracy rate of 0.66 and three layered LSTM gave a sample accuracy of 0.78[1]. ARIMA model has been applied by Debadrita Banerjee, in order to predict Sensex index of Indian stock market. The ARIMA model uses a configuration of (p=1, d=0, q=1), and gives a Root Mean Squared Error (RMSE) of 691.399 Sensex points[2]. A study of sentiment analysis of Facebook data by Troussas, C., Virvou, M., Espinosa, K. J.,

Llaguno, K., & Caro, J. , uses the Naive Bayes Classifier for classifying negative and positive posts on Facebook, by using posterior probability of sentiment given a sentence P(sentiment | sentence) . The model feed to a learning application with and has a precision of 0.77[3]. An application of LSTM Neural Networks in prediction of next day closing price of S&P 500 index is illustrated in the paper by Tingwei Gao, Yueting Chai, and Yi Liu. It uses a timestamp of 20 days and six trading features like Open, Close, High, Low etc., to predict next day's closing price. The activations used by the model are Rectified Linear Unit (ReLU) and Hard Sigmoid for outer and inner activations respectively. The model gives a Mean Absolute Error (MAE) of 11.409% for 100 days of test predictions of S&P 500[4]. The paper by Orlando De JesGs and Martin T. Hagan specifies the Backpropagation through Time (BPTT) Algorithm used to train Recurrent Neural Networks and their architectures like LSTM. It species the change in gradient and updating of synapses in with respect to current states as well as previous states. The algorithm actually reverse propagates the error at the output node and calculates the gradient with respect to each node[5]. A Feedforward Neural Network is modelled for prediction of wind speed in the paper presented by Akshay Kumar H, Dr. Yeresime Suresh. The network has 3 layers, uses sigmoid as its activation function, trains for 1000 epochs and has 94.28% as its training accuracy[6]. A comprehensive view of Backpropagation Algorithm used for training Feed forward Neural Networks is presented in a paper by Peng Wang, Gang Zhao and Xingren Yao. The algorithm is studied for 3 hidden layers and different learning rate and hidden layer nodes. Error is minimum when nodes in hidden layer are 5[7]. The work by Liang Kai, Zhou Zhiping evaluates an Ensemble Learning model using Naïve Bayes, Decision Trees and Neural Networks, giving an accuracy of 0.8534 for test set, for an E-Learning System [8].

3. METHODOLOGY

In existing systems, mostly only either RNN or LSTMs are used for mapping historic intuition and prediction of mostly market indices and not individual stock prices. Index of any market conveys the overall market performance of the market. But knowing only the index does not fulfill our purpose of risk mitigation and decrease of uncertainty in our investment decisions, as investments are made in individual stocks. Thus, the system presented here predicts the closing prices of next day, of any individual stock listed on the National Stock Exchange (NSE). Predicting individual stock prices is really a challenging and difficult work, because each single stock is a different time series, which has different seasonality, trends, moving averages and deviations. This task requires a prediction model, which can map these parameters and their behavior with respect to time, and use the learning obtained to predict future prices. LSTM networks are best fitted for this task as they can map very long term memory dependencies of sequential time series data like stocks [1].

Although market indices do not estimate individual stock prices, there exists correlation between individual stocks and the market index, as indices represent performance of entire market. To make the system more efficient, it must take in inclusion the index of the market. Hence, as index is a non-stationary time series, an ARIMA model with configuration (2, 1, 2), is used for prediction of next day NIFTY-100 index [2]. The predicted index from ARIMA and the predicted stock price are mapped into the Feed forward model along with their past 15 day correlation coefficient, precisely Pearson's correlation coefficient. ARIMA and LSTM do the task of mapping historic intuitions of the market index and stock price, as the system predicts immediate next day price, there is a need of having an idea of the state of market at the current time and also because it is the people who trade in the market, it is important to know in general people's sentiment about the prescribed stock and the overall market. This task is done by the sentiment analysis module, which is designed as a generic model, rather than having any individual platform [5].

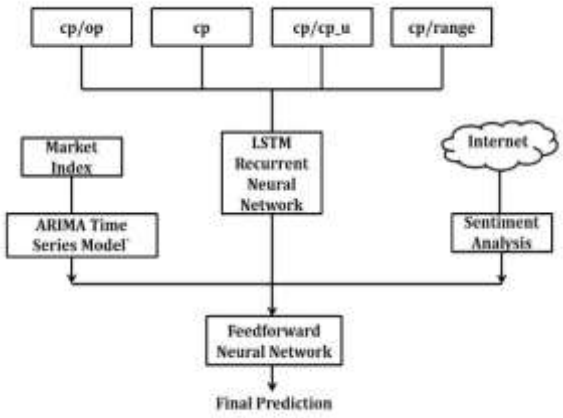

Figure 3.1 System Architecture

It scrapes news headlines about the stock and the market from the internet and makes a corpus, which is then analyzed using Naive Bayes Classifier to output a sentiment score between -1 to 1, the more positive score, the more positive the sentiment. This sentiment will be used to adjust the prediction in the Feedforward Neural Network model in order to tune the final prediction with respect to the current market trend [7].

3.1 Data Preprocessing and Feature Engineering

The data required for the neural networks and ARIMA is fetched using nsepy package available in python. The LSTM and ARIMA features are mentioned in Section 3.2 and 3.3 respectively. The features are scaled in range of 0 to 1. Scaling is done as follows:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3.2 Long Short-Term Memory (LSTM) Networks

The simple Feedforward Neural Network is not so efficient when it comes to time series or problems which involve dealing with data that is sequential. This is because at fundamental level in a feedforward neural network, the neuron does not consider learning from past data. Hence, Recurrent Neural Networks were designed to overcome this problem. RNNs consider previous time step output along with the current input state. This makes them better than Feedforward neural networks in learning sequential data. However, there are two major setbacks in RNNs called as vanishing gradient problem and exploding gradient problem. This means that the gradient in Backpropagation Through Time (BPTT) algorithm that is used to update synapses in the network either decrease to zero, or keep getting larger, which results in either no training or increase in loss, both of which cause the training to fail [6]. This cause the RNN to become redundant in case of long sequences. To overcome this problem, Long Short-Term Memory (LSTM) Cells, architecture of RNN was proposed. LSTM consists of an extra cell state that works as a memory unit. The decision of updating of cell state and the content of updating is decided by gates in LSTM cell.

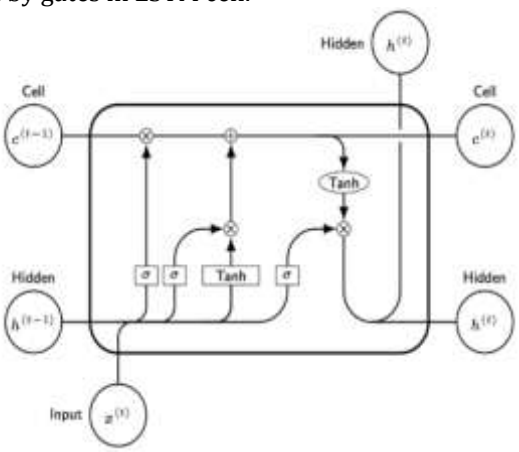

Figure 3.2.1 LSTM cell

- x represents feature vectors
- h represents hypothesis
- c represents cell state
- σ, tanh are activation functions

There are three gates forget gate, input gate and output gate. Due to this LSTM does not suffer from vanishing or exploding gradient problem and is very useful for long term sequential data [1][3].

$$\begin{aligned} i_t &= \sigma_1(W_i x_t + U_i c_{t-1} + b_i) \\ f_t &= \sigma_1(W_f x_t + U_f c_{t-1} + b_f) \\ o_t &= \sigma_1(W_o x_t + U_o c_{t-1} + b_o) \\ c_t &= f_t \bullet c_{t-1} + i_t \bullet \sigma_1(W_c x_t + b_c) \\ h_t &= o_t \bullet \sigma_2(c_t) \end{aligned}$$

Figure 3.2.2 LSTM gates equations

i_t , f_t, o_t, c_t, h_t, σ represents input gate, forget gate, output gate, cell state, hypothesis and activation function respectively. This paper presents a single layered LSTM architecture with 500 nodes. The number of nodes was decided by experimentation on various stocks with varying volatility. Generally multiple stacks of LSTM layers, with lesser number of nodes, but this architecture resulted in poor performance when stocks with high volatility were used. The loss function used is Mean Squared Error (MSE) and epochs are 100.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Figure 3.2.3 Mean Squared Error Equation

- n is number of data points
- Y_i is actual value
- $\hat{Y}_i$ is predicted value

The features given as (cp, cp/op, cp/cp_u, cp/range); where cp is 'Close Price', cp/op is ratio of 'Close Price' to 'Open Price', cp/cp_u is the ratio of Close Price to absolute mean deviation from 5 day rolling mean, and cp/range is the ratio of close price to the range that is absolute difference between High Price and Low Price of that day. These all prices are historic prices and the features are converted into a three dimensional tensor as required by a LSTM in the format of (length, timestamps, number of features). We use 15 time steps. LSTM outputs 4 values which are fed to a 4 layered feedforward network with 2 hidden layers, the number of nodes in the feedforward network are (4,5,6,1).

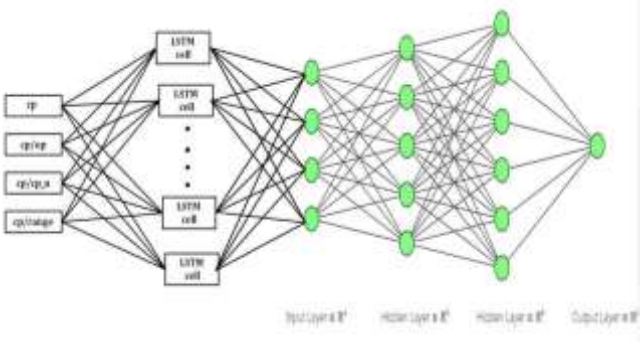

Figure 3.2.4 LSTM Neural Network Architecture

The reason behind using a deep network for output is that the 4 values that LSTM gives as output is mapped to a single final closing price and these layers map deep intuitions between the LSTM outputs and the final closing price. Backpropagation through Time (BPTT) Algorithm is

used for training the LSTM network and Backpropagation Algorithm is used for training the Feedforward network [5]. In both networks, linear activation function is used. The LSTM Network has 200 cells and undergoes 150 epochs.

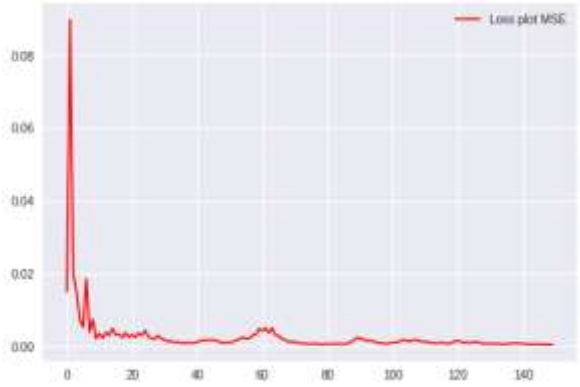

Figure 3.2.4 Loss plot (MSE)

3.3 Auto Regressive Integrated Moving Average (ARIMA)

The data of event which occurs after a fixed time interval is termed as a time series. Time series are an important class of data, which are regularly dealt in real time applications. Time series analysis deals with the fetching inferences and conclusions by analyzing a time series with help of statistical and mathematical methods. This paper uses a time series model called as Auto Regressive Integrated Moving Average (ARIMA), which is mainly used in non-stationary time series analysis [2]. Here, non-stationary time series mean those series which do not have constant mean and variance with respect to time. The system uses ARIMA to forecast next day price of NIFTY 100. But the index is not a stationary series, it shows a trend in upward sense, for last 10 years. Hence, to ma
ke the series stationary, we must difference it, a differenced series is given by following equation

$$X_{td} = X_t - X_{t-1}$$

ARIMA takes three parameters as input, those are (p,d,q)

- p is the number of auto regressive factors
- q represents times the series is differenced
- q represents number of lagged forecast error

ARIMA(p,d,q) is given by the following equation

$$X_t = \theta_0 + \phi 1 X_{t-1} + \phi 2 X_{t-2} + \phi p X_{t-p} + et - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \ldots - \theta_q e_{t-q}$$

Here e_t and X_t are random error and original value at t, ϕ is model parameter. We use Augmented Dickey Fuller Test for testing stationarity of time series, we check whether the time series is stationary or not, by testing Null Hypothesis. We also compute the Auto Correlation Function and Partial Auto Correlation Function of the time series. ACF denotes the correlation between current value X_t and value before n periods X_{t-n}, and is between-1 and 1. PACF states the association degree between X_{t-1} and X_{t-n} that means partial correlation of its own lag and itself.
We can infer from the following graphs, that p=2 and q=2, from the above ACF and PACF graphs respectively.

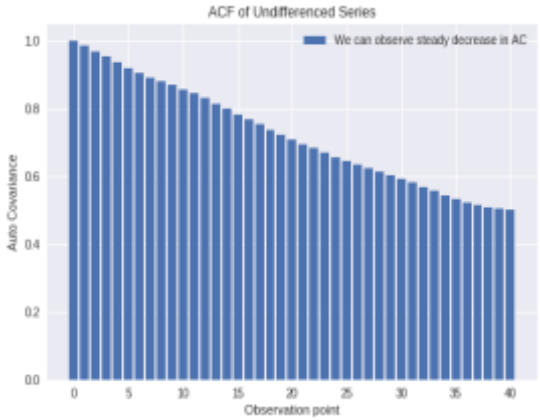

Figure 3.3.1 ACF of non-differenced series

Figure 3.3.2 ACF of differenced series

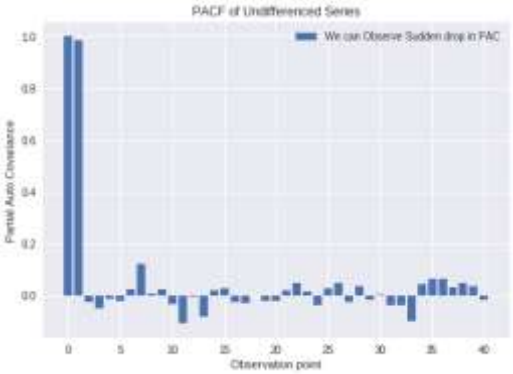

Figure 3.3.3 PACF of non-differenced series

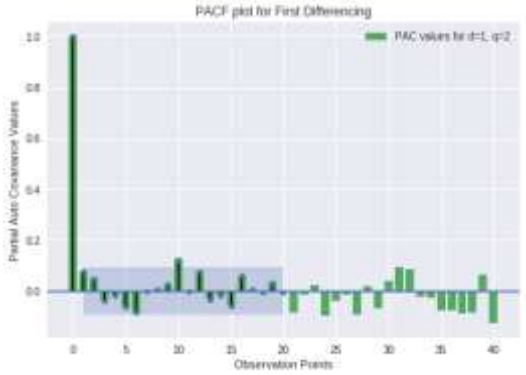

Figure 3.3.4 PACF of differenced series

The model gives forecast on 100 days test data as follows:

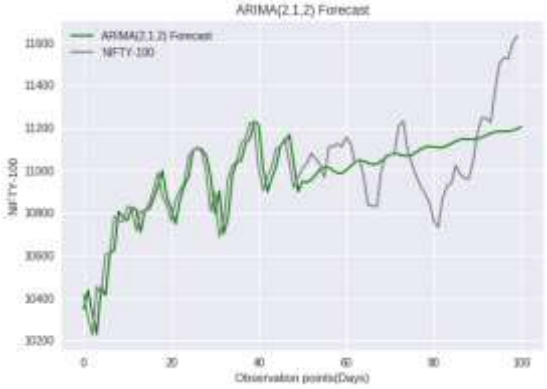

Figure 3.3.5 ARIMA forecast

3.4 Sentiment Analysis

The stock market is a place where trading is done by human beings, and thus emotions and sentiments play a marginal role in behavior of stock market. So, in combination with historic intuitions, current market sentiments must also be taken into consideration. To capture these sentiments, this system uses a generic model instead of depending on some kind of platform. It uses a scraper to scrape news about the stock, from the internet. It does a simple Google search with and scrapes only the news headlines from the search page and makes a collection from them. This collection is cleaned by removing the HTML tags using and a cleaned corpus is made, which contain only pure headline text. A pre-trained analyzer, which is trained on IMDB Movie data set, using Gaussian Naive Bayes Algorithm [4].

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

The Gaussian Naive Bayes formulation is used for classification problems and works on the concept of posterior probability. Posterior probability is the other name of Conditional probability, which conveys the probability of an event happening given an another event or set of events has already happened. Marginal probabilities of an event are calculated and used to calculate conditional probabilities with respect to the class in the training set. The class can be positive or negative, and thus each word is assigned a probability, according to which classification is done, after word extraction from given sentences and processes of stemming and lemmatization. In the feature sentence, the sentiment will be judged on the basis on these word probabilities. In the system, the corpus made from scraping headlines, undergoes this process and final sentiment is given as output between -1 to 1. If the number is more inclined towards 1, it means more positive the sentiment, and vice versa.

3.5 Feed forward Neural Network

The prior section explains that the LSTM RNN maps the historic intuitions from training data, which does the job of technical analysis and finds out trends intrinsic to the equity. The ARIMA model uses time series analysis to forecast the market index, as market indices do have some sort of correlation with respect to the equity prices. However, these two values must be accounted in combination to predict a more accurate future price. This task is done by a simple Feed forward neural network in the system, which takes into account the prediction given by the LSTM RNN, the ARIMA index forecast and the rolling correlation of 15 days between both the series. The rolling correlation is taken into account as to, it will guide the network of what influence the index has on the stock price. The Feed forward neural network simply maps the relation between these three values to give a refined and non-biased final output [6]. The network has 4 layers with 4, 6, 8 and 1 nodes respectively. The reason behind keeping 3 hidden layers is to map highly volatile stocks efficiently. It uses linear activation function and trains on 800 epochs of shuffled training data. The data is shuffled in order to avoid errors due to autocorrelation and map only the relationship between given features. Backpropagation algorithm is used for training [7].

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

Figure 3.5.1 Backpropagation Algorithm (Chain Rule).

Equation of a single neuron is given as,

$$\hat{y} = \phi(x_i w_i + \beta)$$

Here, ϕ is the activation function, x is feature vector, w is synapse weight vector and β is the bias. Following is the architecture of the network

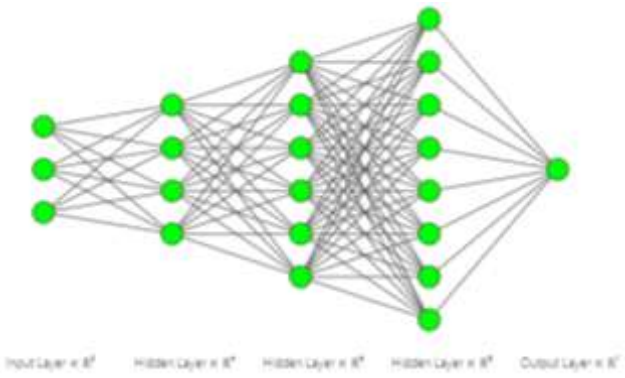

Figure 3.5.2 FFNN Architecture

The output of sentiment analysis is added to the output of the Feedforward network in the following manner

$$Y_f = Y_o + sentiment\_score * \sigma_\mu$$

Here, Y_f is final prediction, Y_o is output of the network and σ_μ is the mean deviation with respect to last 5 days rolling mean. Below is the next day closing price prediction for MRF stocks for 100 test points.

Figure 3.5.3 Test Set Predictions

4. CONCLUSION

The reason behind using three different approaches and combining them in a single Feedforward network is using an approach of Ensemble Learning. The model gives an R^2 value of 0.9449 for above MRF stocks test set. Hence, it is observed that LSTM RNN is very beneficial in learning long term sequential dependencies, due to its properties and the combined model using LSTM, ARIMA and Sentiment Analysis is very efficient to predict next day closing price, as it considers historic intuitions as well as current market sentiment.

REFERENCES

[1] Siyuan Liu, Guangzhong Liao, Yifan Ding, " Stock Transaction Prediction Modeling and Analysis Based on LSTM ", 2018 13th IEEE Conference on Industrial Electronics and Applications(ICIEA), DOI:10.1109/ICIEA.2018.8398183

[2] Debadrita Banerjee, "Forecasting of Indian stock market using time-series ARIMA model", 2014 2nd International Conference on Business and Information Management (ICBIM), DOI: 10.1109/ICBIM.2014.6970973

[3] Tingwei Gao, Yueting Chai , Yi Liu, "Applying long short term memory neural networks for predicting stock closing price", 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), DOI: 10.1109/ICSESS.2017.8342981

[4] Christos Troussas, Maria Virvou, Kurt Junshean Espinosa, Kevin Llaguno , Jaime Caro, "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning", IISA 2013, DOI: 10.1109/IISA.2013.6623713

[5] O. De Jeses, M.T. Hagan, "Backpropagation through time for a general class of recurrent network", 2001 IJCNN'01. International Joint Conference on Neural Networks, DOI: 10.1109/IJCNN.2001.938786

[6] Akshay Kumar H, Yeresime Suresh, "Multilayer feed forward neural network to predict the speed of wind", 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), DOI: 10.1109/CSITSS.2016.7779372

[7] Peng Wang, Gang Zhao, Xingren Yao, "Applying back-propagation neural network to predict bus traffic", 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), DOI: 10.1109/FSKD.2016.7603269

[8] Liang Kai, Zhou Zhiping, "Using an Ensemble Classifier on Learning Evaluation for E-learning System", 2012 International Conference on Computer Science and Service System, DOI: 10.1109/CSSS.2012.140

BIOGRAPHIES

Mr. Omkar S. Deorukhkar, Final Year Student of B.E. (Computer Engineering), at G.V. Acharya Institute Of Engineering And Technology, Shelu, Maharashtra. Domain Of Interest – Machine Learning, Data Science, Artificial Neural Networks

Ms. Shrutika H. Lokhande, Final Year Student of B.E. (Computer Engineering), at G.V. Acharya Institute Of Engineering And Technology, Shelu, Maharashtra. Domain Of Interest – Web Technologies, Databases

Ms. Vanishree R. Nayak, Final Year Student of B.E. (Computer Engineering), at G.V. Acharya Institute Of Engineering And Technology, Shelu, Maharashtra. Domain Of Interest – Machine Learning, Databases.

Mr. Amit A. Chougule (M.Tech – Computer Engineering), Asst. Professor and Head Of Department (HOD)-Computer Engineering, G.V. Acharya Institute Of Engineering And Technology, Shelu, Maharashtra. Domain Of Interest – Computer Networks, Compilers, Databases