International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

Optimizing Hyperparameters for Enhanced LSTM-Based Prediction System Performance

[1]Dr. Priyanka Paygude, [2]Prashant Chavan, [3]Dr. Milind Gayakwad, [4]Khushi Gupta, [5]Satyam Joshi, [6]Gopika, [7]Rahul Joshi, [8]Sudhanshu Gonge, [9]Ketan Kotecha
[3]Associate Professor, [2,3] Assistant Professor, [4,5,6]Student
Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
[7]Associate Professor, [8]Assistant Professor and [9]Professor
Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India
pspaygude@bvucoep.edu.in, pschavan@bvucoep.edu.in, mdgayakwad@bvucoep.edu.in,
g.khushi0410@gmail.com,satyam.joshi1203@gmail.com , gopikanathil@gmail.com, rahulj@sitpune.edu.in,
sudhanshu.gonge@sitpune.edu.in, head@scaai.siu.edu.in
*pschavan@bvucoep.edu.in (only corresponding author email)

Abstract: This research paper explores the application of deep learning and supervised machine learning algorithms, specifically Long Short-Term Memory (LSTM), for stock market prediction. The study focuses on the closing prices of three companies - Tata Steel, Apple, and Powergrid - using a dataset sourced from Yahoo Finance. Performance evaluation of the LSTM model employed RMSE, MAPE, and accuracy metrics, along with hyperparameter calibration to determine the optimal model parameters. The findings indicate that a single-layer LSTM model outperformed a multilayer LSTM model across all companies and evaluation metrics. Furthermore, a comparison with existing research demonstrated the superiority of the proposed model. The study emphasizes the effectiveness of LSTM models for stock price prediction, underscores the significance of proper hyperparameter tuning for optimal performance, and concludes that a single-layer LSTM model can yield superior results compared to a multilayer model.

Keywords-deep learning, hyperparameter calibration, LSTM, stock price prediction, yahoo finance.

I. INTRODUCTION

Stocks that are also called shares or equities, represent ownership in a company. Compared to other asset classes, stocks produce higher investment returns. Stock market is known for being volatile, dynamic and nonlinear [1] as it depends on various factors such as performance of the company, economic conditions, investor's sentiment, political events and global trends which can result in huge losses, so an accurate prediction of stocks is needed for desirable profits. The act of attempting to anticipate the future stock value is called stock price prediction. Stock price prediction, however, is a very complex and challenging task. It is observed that the change in stock values has some pattern and isn't random therefore it can be anticipated by carefully studying the historical stock data. The advancements in deep learning and machine learning techniques have made stock price prediction possible with high accuracy [2].

Many works have been performed in this field and the algorithms used include linear regression [3], ARIMA [4], LSTM [5], RNN [6], random forest [7], CNN [8], etc. Previous works have worked in datasets from yahoo finance [9], NSE (National Stock Exchange) [10], kaggle[11], google finance, quandl etc.

In this study, we use deep learning and machine learning techniques, particularly LSTM, to forecast stock values based on historical stock data. We utilized the dataset fetched from Yahoo Finance, consisting of 10 years of daily closing prices for three prominent companies: Tata Steel, Apple, and PowerGrid. The dataset was split into an 80% training set and 20% testing set, ensuring a robust evaluation process. We focused our analysis on the closing price as the target variable, as it is a crucial indicator. The time step selected for our study was [5]100 days. To assess the performance of our LSTM models, we employed three commonly used evaluation metrics: Mean Absolute Percentage Error (MAPE), Root Mean Squared Error, and accuracy [22]. Hyperparameter calibration was a crucial step in optimizing the LSTM neural network. We experimented with different architectures by altering the quantity of layers in the network, including one, two, and four layers [23]. Our results consistently demonstrated that the single-layer LSTM model outperformed the multi-layer model for all three companies across all three-evaluation metrics. This finding suggests that a[24][25] simpler architecture can effectively capture the underlying patterns in stock price data, resulting in more accurate predictions. Furthermore, we compared our proposed single-layer LSTM [51] model with

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

other research works and the comparative analysis revealed that our model outperformed previous approaches [26][27][50].

The remaining section of this research work is structured as followed: section 2 discuss the study of earlier works in the field of stock price prediction. Section 3 is focused on the proposed methodology using LSTM to construct a stock price prediction model with high accuracy. In section 4, implementation of the work is discussed by calibrating the hyperparameters. The subsections in section 4 discuss the comparative analysis of different layers of LSTM and result analysis of various evaluation metrics and comparative analysis of previous research works. Section 5 concludes the work and future scope.

II. LITERATURE SURVEY

In this literature survey, we provide an overview of existing research works that used machine learning and deep learning techniques for stock market prediction. [12] aims at building a brand-new stock price prediction model. For estimating the closing value of the following stock trading day, the LSTM model is used to first compute the investor sentiment prior [28] to the stock opening by fine-tuning the BERT model, then aggregate the generated investor sentiment and the fundamental stock quotation data. [13]aspires to offer a unique strategy that combines the LSTM deep neural network for time series [29] prediction with great regard for past stock data and the similarities between those stocks and their closest competitors [30]. Their model performed better than the competing approaches (vanilla LSTM, linear regression, convolutional neural network and random forest), according to results obtained from experiments on 4 active stocks from US stock market and three from Vietnamese stock market. [14]Demonstrate the model which applies a CNN [31] [32] [55] [56] framework with arrays as the input map. They were able to demonstrate that the intended approach produces potent outcomes through in-depth experimental findings employing the Taiwanese stock market. [15] desire to review new research on SVMs, neural networks and hidden Markov models used to forecast stock market fluctuations [33][34]. Also claims that the two most effective machine learning techniques in the field of stock price prediction are neural networks and SVM [35][36]. Additionally, a model for HMM-based stock market forecasting is proposed. In [16]they seek a time series data from PT. Ramayana Lestari SentosaTbk. from 02/03/2020 to 15/09/2021. Their model used an LSTM [53] [54] with same no. of epochs but varied numbers of activation functions, optimizers, and the nodes, which produced different outcomes and levels of accuracy and lately proposed the superior function which is far better compared to the activation functions of Sigmoid, Tanh, and Relu. [17]uses 10 stocks' daily closing prices which were utilized to train and assess the effectiveness of the model. Their experimental findings show that the suggested technique STPA outperforms other methods in terms of precision,F1-Score and recall rate for forecasting change in stock trends, and focuses on the significance of combining sentiment analysis of news headlines with daily stock data to forecast trends and stock prices, and the LSTM (Long Short-Term Memory) [52] cells are utilized for price prediction time series, also suggests using the ensemble technique XGBoost [39] to predict market trends. [18]Applied an attention mechanism to direct the model's attention to important information[40] and use an LSTM neural network because of its benefits in time-series data analysis [41].The auto-regressive integrated moving average (ARIMA) [49], neural network (NN), and LSTM were utilized in [19] to forecast [42] the closing prices data for Bursa Malaysia from February 1, 2020 to January 19, 2021. Every one of the models will be assessed utilizing RMSE [43] and MAPE. [20] proposes a deep learning method (RNN) [44][45] [46] to forecast the stock market because of its benefits in processing time series data they use Apples stock data from August 2009 – August 2020. Their loss was close to 0.1% and their accuracy [47][48] for prediction is around 95% .

From the study of literature survey, it is noted that the focus of earlier research was mostly on the calibration of a small number of hyperparameters. It is noted that no research has ever been conducted in-depth on LSTM and how its layers function.

III. METHODOLOGY

Long Short-Term Memory (LSTM), is a form of recurrent neural network (RNN) which is frequently used in various natural language processing (NLP) applications and speech recognition. The main benefit of LSTMs over conventional RNNs is that they can overcome the vanishing gradient issue that occurs in deep neural networks while training on lengthy data sequences.

LSTM can capture long-term dependencies of data [37][38], by using a series of specialized memory cells that can selectively recall or forget information over time. The forget, input, and output gate are three primary gates found in each memory cell in an LSTM network. These gates manage the information flow into and out of the memory cell, deciding what data to retain and what data to discard [21]. Fig 1 shows the architecture of LSTM.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

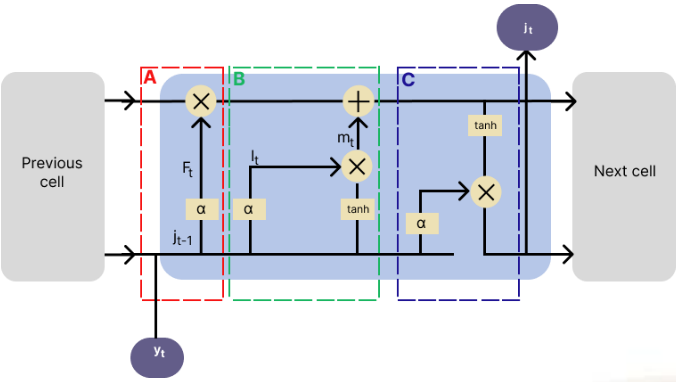

Figure 1. LSTM architecture

A. *Working*

Forget Gate: It is used to delete data that is no longer relevant to the cell state. The gate receives two inputs, y_t (current time input) and j_{t-1} (output of previous cell), which are multiplied with weight matrices and after that bias addition is performed. Then, through an activation function the result is passed which produces output in binary form. If output of the cell state is equal to 0, then the information is forgotten, however if it is equal to 1, then the information is saved for future use.

Input gate: The cell state is modified by adding relevant data by the input gate. In Fig 1, section B shows the working of input gate. In input gate data is regulated through sigmoid function and information to be remembered is filtered using inputs y_t and j_{t-1}. Then, using tanh function, a vector is generated that contains every possible value between y_t and j_{t-1}which produces an output that ranges from [-1,+1]. For extracting the necessary information, the values of vector and the manipulated values are multiplied.

Output gate: It is in charge of extracting relevant information from the current cell state and delivering it as output. In Fig1, section C shows the working of output gate. First, tanh function is utilized to generate a vector in the cell. Sigmoid function regulates the data. Then it is filtered by the remembered values using inputs (y_t and j_{t-1}). The values of the vector and manipulated values are multiplied and delivered as output which will act as the input for the next cell.

The working of gates in LSTM can be described by following equations[22] –

$$I_t = \alpha_1(W_i y_t + U_i m_{t-1} + \beta_i) \qquad (1)$$

$$F_t = \alpha_1(W_f y_t + U_f m_{t-1} + \beta_f) \qquad (2)$$

$$O_t = \alpha_1(W_o y_t + U_o m_{t-1} + \beta_o) \qquad (3)$$

$$m_t = f_t \cdot m_{t-1} + I_t \cdot \alpha_1(W_m y_t + \beta_m) \qquad (4)$$

$$j_t = o_t \cdot \alpha_2(m_t) \qquad (5)$$

Where I_t is vector of input gate, $O_t$is vector of output gate, at an instant of time t, $y_t$is input vector, $j_t$is output vector, $m_t$is memory cell state, F_t is the vector of forget gate, W_n& U_n are weight matrices, β_n is bias vector and $\alpha_n$is activation function.

IV. IMPLEMENTATION

A. *Dataset*

The dataset utilized is a key component in machine learning. Because even a small alteration or error in the data can have a significant impact on the conclusion, the dataset should be as precise as possible. So we fetched our dataset from yahoo finance [23], the most trusted site for stock data using the yfinance library. We worked on the dataset of Tata Steel. We took 10 years of daily data from 01/01/2010 to 31/12/2019.

The dataset has 7 parameters: date, open, close, high, low, volume and adjacent close. We worked on the closing parameter. Dataset is then divided into training and testing sections. Because this ratio produced the best results, we used 80% of data for training and the 20% for testing the model. Dataset details are shown in Table I.

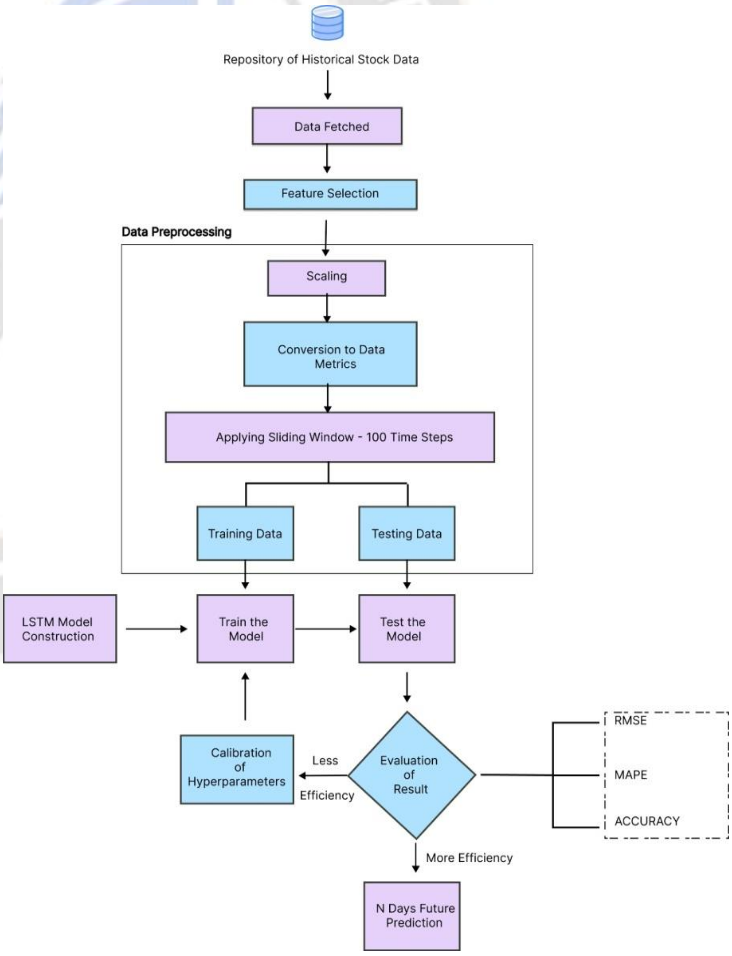

Figure 2. Proposed architecture of stock price prediction

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

TABLE I. DATASET DETAILS

Dataset	10 years dataset (1/1/2010 - 31/12/2019)		Parameters	Feature selection
	Training data-80%	Testing data-20%		
Tata Steel	1/1/2010 - 31/12/2017	1/1/2018 - 31/12/2019	Date, Open, Close, High, Low, Volume, Adjacent close	Close
Apple				
Powergrid				

B. Proposed Model

The training data needs to be scaled down using the minmaxscaler() from sklearn.preprocessing library. In this step, data normalization is done i.e., each value will lie in a particular range (0 to 1) to increase training stability and effectiveness of the model.

The x_train and y_train must now be defined. Let's look at an illustration to help you grasp their idea. Here, we're using a set of 100 days. Assume that the first 100 days' worth of a stock is represented as v1, v2, v3, v4, and so on. Now, rather than being a random number, the following value, v101, will depend on the prior 100 values (v1-v100). Similarly, v102's value will depend on prior 100 days value (v2-v101), and so forth. As a result, the 101th value will be appended in y_train and the 100 days value, on which the 101th day value depends, will be appended in x_train. After that, they are transformed into numpy arrays so that they can be fed to the model.

We made keras sequential model that allows addition of layers in neural network in sequence. Our proposed LSTM model has 1 LSTM layer with 100 memory units (cells) and uses the hyperbolic tangent (tanh) activation function. With a dropout rate of 0.1, a dropout layer is added which means that 10% of the input units will be randomly set to 0 during each training epoch. Dropout is a regularization strategy that, in order to avoid overfitting, randomly removes some neurons during training. At the end the model has 2 dense layers. One fully connected layer with 50 neurons and tanh activation function and other a final dense layer with a single neuron which will output the predicted value. Proposed LSTM model architecture is shown in Fig 3.

Model is then compiled using Adam optimizer which has default learning rate of 0.001 and MSE (mean squared error) loss. Adam is a optimization technique which is a hybrid of 2 gradient descent (RMSP and momentum).

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta \omega_t}\right] v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\omega_t}\right]^2 \quad (6)$$

We fit the y_train and x_train into the model for training with 100 epochs and batch size 32 where epochs are the number of iterations that needs to be made of training dataset for the training of the LSTM model and is used when all the training data is used all at once.

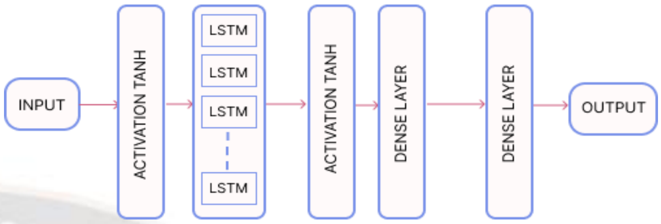

Figure 3. Architecture of proposed LSTM model

For testing, since we are considering the 100 days set, for making the first prediction we need the previous 100 days' data from the training part therefore we are going to append the testing data and the last 100 days of training data to make the final data frame for testing which is then scaled down. Now we'll consider x_test and y_test in the testing dataset just like x_train and y_train in the training data which are then converted into numpy arrays.

x_test is then provided to the model to make the predictions (y_predicted). So, y_test is the original data and y_predicted is the forecasted data. Both are then scaled up and are plotted.

C. Evaluation Metrics

RMSE (Root Mean Squared Error): It computes the average difference between the actual values in the test dataset and the predicted values. The RMSE formula is:

$$RMSE = \sqrt{\sum_{n}^{i=1} \frac{(\hat{y_i} - y_i)^2}{n}} \quad (7)$$

In above equation, $\hat{y}_i$ is the forecasted value, y_i is the original value and n is the number of samples.

MAPE (Mean Absolute Percentage Error):It calculates the difference between the test dataset's actual values and the values that were anticipated in percentage form. The MAPE equation is:

$$MAPE = \frac{\sum \frac{|A-F|}{A} \times 100}{N} \quad (8)$$

Where N is the number of samples, F is forecasted value and A is the original value.

Accuracy: It is measured through RMSE and MAPE using the following formula -

Accuracy = 100% * (1 - (RMSE/mean (actual values))) * (1 - MAPE/100) (9)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

D. Results and Analysis

On the LSTM neural network, hyperparameter calibration is done to determine the optimum model hyperparameter values that minimize the difference between expected and actual values on a given dataset. Mean squared error loss is used throughout. For calibration the hyperparameters considered are:

1. The number of LSTM layers: This number describes how many LSTM layers are stacked on top of one another to create the entire LSTM design. A group of memory cells that process the input sequence are present in each LSTM layer.

2. Units: The quantity of memory cells or neurons found in each LSTM layer is referred to as units. The model becomes more complex as the number of units increases, but there is also a greater chance of overfitting.

3. Activation function: This is a non-linear function that is applied to the output of every LSTM cell to introduce non-linearity and aid the model's ability to learn intricate patterns. Sigmoid, tanh, and ReLU are frequently used activation functions in LSTM models.

4. Dropout: This describes a regularization strategy that randomly drops certain neurons during training to prevent the model from overfitting. Dropout encourages the model to learn more robust features by lowering the correlation between the neurons.

5. Dense layer: This term describes a completely connected layer that uses linear transformation to make the final prediction from the output of the LSTM layer. The intricacy of the problem and the desired result determine how many units should be in the dense layer.

6. Optimizer (Default learning rate): The algorithm used to adjust the model's weights during training is referred to as the optimizer. To enhance the effectiveness and efficiency of the model, the optimizer reduces the loss function and modifies the weights. LSTM models frequently employ optimizers like Adam, RMSprop, and SGD.

7. Epochs: This term describes the number of times the complete training dataset is run through the model. One epoch is considered complete when all the training samples have been processed once through the model.

8. Batch size: This term describes the quantity of samples that are processed in a single iteration during training. Large batch sizes can accelerate training but also increase memory usage and increase the risk of overfitting. A small batch size can help the model to generalize better but it can slow down the training process.

TABLE II. CALIBRATION OF HYPERPARAMETERS IN 1 LSTM LAYER MODEL

Units	Activation function	Dropout	Dense layer	Optimizer	Epochs	Batch size	RMSE	MAPE
50	relu	0.1	1	adam	50	32	1.4392	2.2473
100							1.2931	1.9939
150							1.3344	2.0484
100	linear						1.3176	2.0131
	tanh						1.2887	1.9663
	sigmoid						1.8397	2.9498
	tanh	0.2					1.3039	1.9970
		0.1	2(50,1)				1.1942	1.8079
			2(100,1)				1.2412	1.8915
			2(50,1)	sgd			1.8792	2.9900
				adagrad			2.1028	3.3758
				rmsprop			1.3980	2.1066
				adadelta			3.1326	4.9957
				adam	100		1.1508	1.7391
					150		1.1573	1.7832
					100	64	1.1708	1.7756
Suitable parameters								
100	tanh	0.1	2(50,1)	adam	100	32	1.1508	1.7391

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

TABLE III. CALIBRATION OF HYPERPARAMETERS IN 2 LSTM LAYER MODEL

Units	Activation function	Dropout	Dense layer	Optimizer	Epochs	Batch size	RMSE	MAPE
50, 50	relu	0.1, 0.1	1	adam	50	32	1.6103	2.5556
50, 100							1.5609	2.4728
100, 100							1.4376	2.2162
100, 150							1.4036	2.1814
150, 150							1.3288	2.0436
150, 200							1.5570	2.3791
150, 150	linear						1.3637	2.0897
	tanh						1.2415	1.8819
	sigmoid						1.9661	3.0380
	tanh	0.1, 0.2					1.2513	1.9013
		0.1, 0.1	2(50,1)				1.1662	1.7813
			2(100,1)				1.2237	1.8703
			2(50,1)	sgd			2.5185	4.0792
				adagrad			2.8006	4.5550
				rmsprop			1.7996	2.8340
				adadelta			3.2235	5.1859
				adam	100		1.1876	1.8030
					50	64	1.3210	2.0079
Suitable parameters								
150, 150	tanh	0.1, 0.1	2(50,1)	adam	50	32	1.1662	1.7813

TABLE IV. CALIBRATION OF HYPERPARAMETERS IN 4 LSTM LAYER MODEL

Activation function	Dropout	Dense layer	Optimizer	Batch size	Epochs	Units	RMSE	MAPE
relu	0.1	1(1)	adam	32	50	50	1.7670	2.8350
linear							1.4574	2.3042
tanh							1.4455	2.2778
sigmoid							2.2567	3.5705
tanh	0.2						1.5897	2.5140
	0.1	2(50,1)					1.4095	2.2039
		2(100,1)					1.3251	2.0644
		2(150,1)					1.3219	2.0329
		2(200,1)					1.2883	1.9635
		2(250,1)					1.2849	1.9721
		3(50,50,1)					1.3149	2.0330
		3(100,100,1)					1.2522	1.9117
		3(150,150,1)					1.4033	2.2184
		4(50,50,50,1)					1.2717	1.9603
		4(100,100,100,1)					1.3817	2.1262
		3(100,100,1)	sgd				3.6012	5.9270
			adagrad				3.7913	6.2719
			rmsprop				1.9010	2.9869
			adadelta				4.5954	7.2104
			adam	64			1.5403	2.3962
				32	100		1.4020	2.1190
					50	100	1.2408	1.8767
						150	1.5249	2.2980
Suitable parameters								
tanh	0.1	3(100,100,1)	adam	32	50	100	1.2408	1.8767

In Table II, III and IV calibrations of hyperparameters is performed on 1, 2 and 4 LSTM layer model respectively to find the most optimized or suitable hyperparameters for that particular number of LSTM layers. Comparison of the 3 models (1, 2 and 4 LSTM layer) are shown in Fig 4, 5 and 6 from which we can conclude that for Tata Steel single layer LSTM model outperforms multilayer LSTM model.

We tested our models for 2 more datasets (Apple andPowergrid) and their results are shown in Table V.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7620
***Article Received**: 01 June 2023 **Revised**: 29 July 2023 **Accepted**: 10 August 2023*

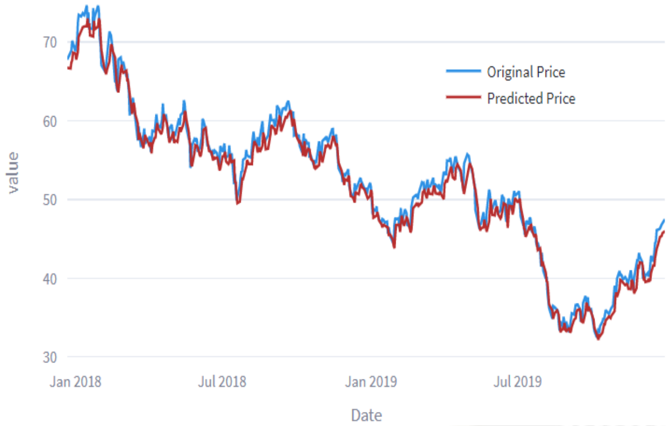

Figure 4. Original vs predicted price of Tata Steel for 4 LSTM layer model

Figure 5. Original vs predicted price of Tata Steel for 2 LSTM layer model

Figure 6. Original vs predicted price of Tata Steel for 1 LSTM layer model

TABLE V. COMPARISON OF DIFFERENT LSTM LAYER MODELS FOR DIFFERENT COMPANIES DATASETS

Dataset	LSTM layers	RMSE	MAPE	Accuracy%
Tata Steel	1	1.1508	1.7391	96.10
	2	1.1662	1.7813	96.03
	4	1.2408	1.8767	95.79
Apple	1	0.8664	1.2988	96.97
	2	1.0061	1.5771	96.42
	4	1.0981	1.7179	96.10
Powergrid	1	1.9128	1.0003	97.70
	2	2.0537	1.0800	97.52
	4	2.1988	1.1530	97.35

Graphical representation of how different companies datasets react to different LSTM layer models can be seen in Fig 7, 8 and 9.

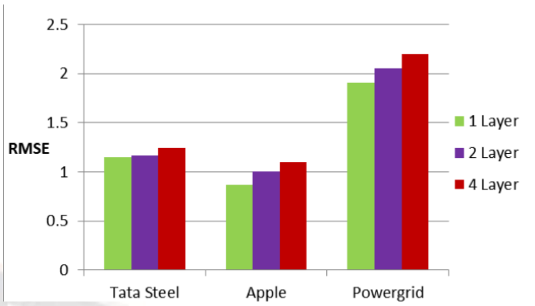

Figure 7. RMSE values for different companies datasets for different LSTM layer models

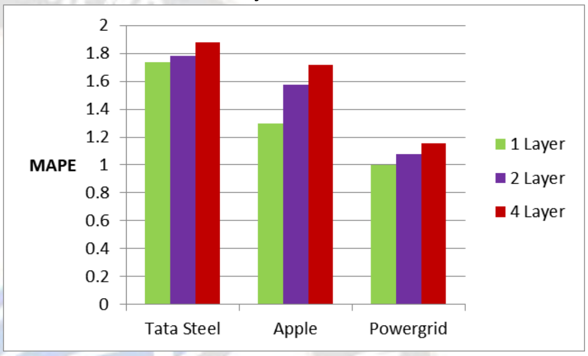

Figure 8. MAPE values for different companies datasets for different LSTM layer models

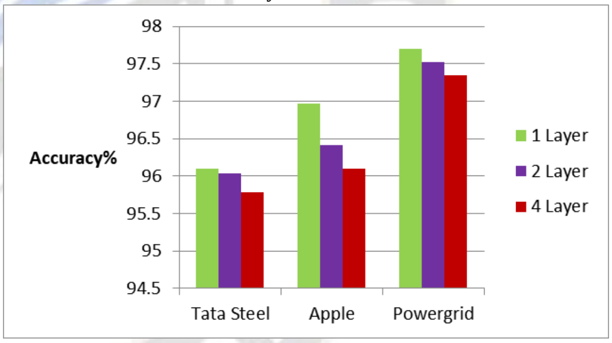

Figure 9. Accuracy values for different companies datasets for different LSTM layer models

From Table V and Fig. 7,8 and 9 we can conclude that single layer LSTM can perform better than multi-layer LSTM.

TABLE VI. COMPARISON OF PROPOSED MODEL WITH OTHER MODELS.

Dataset	Compared Models	RMSE	MAPE	Accuracy %
Apple	LSTMSAA [24]	3.364	2.184	95.08
	Proposed model	3.0921	2.0727	95.41
S&P 500	RCSNet[25]	46.471	-	-
	Proposed model	15.383 1	0.6355	98.51
Tata Consumer	Bi-LSTM Multitask Learning[26]	7.8734	-	-
	Proposed model	5.5036	1.6925	96.07

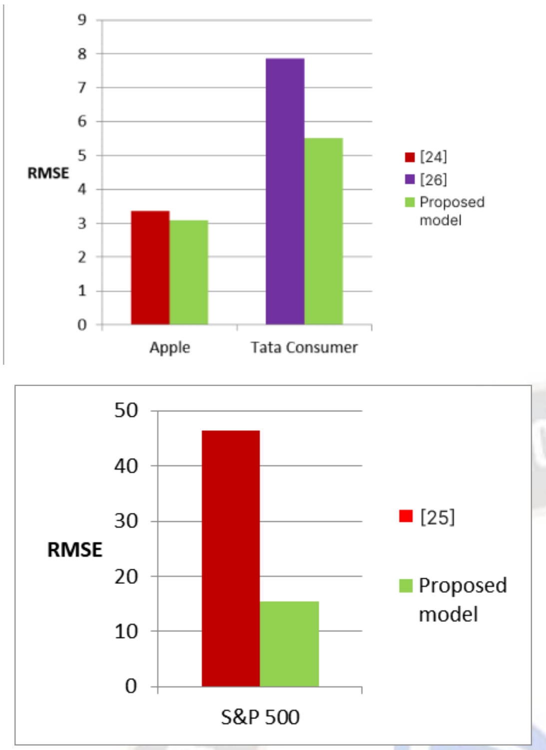

Figure 10. Comparison of proposed model with previous research works based on RMSE.

We compared our proposed model with previous research works [24][25][26]which are shown in Table VI. Fig 10 shows graphical representation of comparison of proposed model with other models based on RMSE.

V. ACKNOWLEDGEMENT

This research was funded by “Research Support Fund of Symbiosis International (Deemed University), Pune, Maharashtra, India”

VI. CONCLUSION AND FUTURE WORK

The present research work aimed to predict the stock prices of 3 companies - Tata Steel, Apple & PowerGrid using the LSTM neural network. The study used 10 years of daily stock data obtained from yahoo finance and worked on closing price with a time step value of 100 days. The dataset was split into a training set (80%) and a testing set (20%). The study employed RMSE, MAPE and accuracy as evaluation metrics.

Calibration of hyperparameters is performed on LSTM to find the optimum values of model’s hyperparameters for achieving the best outcomes on a given dataset. The study compared the performance of multi-layer and single-layer LSTM models and found that the single-layer model outperformed the multi-layer model for all three companies in terms of RMSE, MAPE and accuracy. With single layer best results for Tata Steel, Apple and PowerGrid were achieved with RMSE – 1.1508, MAPE – 1.7391 & accuracy – 96.1%, RMSE – 0.8664, MAPE – 1.2988 & accuracy – 96.97% and RMSE – 1.9128, MAPE – 1.0003 & accuracy – 97.7% respectively. The study also compared its model with other research works. LSTMSAA’ s[24]RMSE and MAPE values are 3.364 & 2.184 and proposed model’ s values are 3.0921 & 2.0727 respectively. RCSNet[25] and Bi-LSTM Multitask Learning’ s[26] RMSE values are 46.471& 7.8734 and proposed model’ s values are 15.3831 & 5.5036 respectively.

For future work, the study suggested exploring varying length of datasets, different parameters, different time steps length, more evaluation metrics and additional companies datasets to support the findings that the single-layer LSTM can perform better than multi-layer LSTM.

ACKNOWLEDGEMENT

This research was funded by “Research Support Fund of Symbiosis International (Deemed University), Pune, Maharashtra, India”

REFERENCES

[1] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock Closing Price Prediction using Machine Learning Techniques,” Procedia ComputSci, vol. 167, pp. 599–606, 2020, doi: 10.1016/j.procs.2020.03.326.

[2] X. Ji, J. Wang, and Z. Yan, “A stock price prediction method based on deep learning technology,” International Journal of Crowd Science, vol. 5, no. 1, pp. 55–72, Apr. 2021, doi: 10.1108/IJCS-05-2020-0012.

[3] D. Bhuriya, G. Kaushal, A. Sharma, and U. Singh, “Stock market predication using a linear regression,” in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), IEEE, Apr. 2017, pp. 510–513. doi: 10.1109/ICECA.2017.8212716.

[4] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock Price Prediction Using the ARIMA Model,” in 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, IEEE, Mar. 2014, pp. 106–112. doi: 10.1109/UKSim.2014.67.

[5] T. Damrongsakmethee and V.-E. Neagoe, “Stock Market Prediction Using a Deep Learning Approach,” in 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, Jun. 2020, pp. 1–6. doi: 10.1109/ECAI50035.2020.9223142.

[6] Y. Zhu, “Stock price prediction using the RNN model,” J Phys ConfSer, vol. 1650, p. 032103, Oct. 2020, doi: 10.1088/1742-6596/1650/3/032103.

[7] T. Manojlovic and I. Stajduhar, “Predicting stock market trends using random forests: A sample of the Zagreb stock exchange,” in 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, May 2015, pp. 1189–1193. doi: 10.1109/MIPRO.2015.7160456.

[8] S. Sarvesh, R. V. Sidharth, V. Vaishnav, J. Thangakumar, and S. Sathyalakshmi, “A Hybrid Model for Stock Price Prediction

using Machine Learning Techniques with CNN," in 2021 5th International Conference on Information Systems and Computer Networks (ISCON), IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ISCON52037.2021.9702382.

[9] I. Parmaret al., "Stock Market Prediction Using Machine Learning," in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, Dec. 2018, pp. 574–576. doi: 10.1109/ICSCCC.2018.8703332.

[10] H. M, G. E.A., V. K. Menon, and S. K.P., "NSE Stock Market Prediction Using Deep-Learning Models," Procedia ComputSci, vol. 132, pp. 1351–1362, 2018, doi: 10.1016/j.procs.2018.05.050.

[11] K. Ullah and M. Qasim, "Google Stock Prices Prediction Using Deep Learning," in 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), IEEE, Nov. 2020, pp. 108–113. doi: 10.1109/ICSET51301.2020.9265146.

[12] X. Weng, X. Lin, and S. Zhao, "Stock Price Prediction Based On Lstm And Bert," in 2022 International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, Sep. 2022, pp. 12–17. doi: 10.1109/ICMLC56445.2022.9941293.

[13] H. G. T. Thu, T. N. Thanh, and T. Le Quy, "A Neighborhood Deep Neural Network Model using Sliding Window for Stock Price Prediction," in 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, Jan. 2021, pp. 69–74. doi: 10.1109/BigComp51126.2021.00022.

[14] J. M.-T. Wu, Z. Li, G. Srivastava, J. Frnda, V. G. Diaz, and J. C.-W. Lin, "A CNN-based Stock Price Trend Prediction with Futures and Historical Price," in 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), IEEE, Dec. 2020, pp. 134–139. doi: 10.1109/ICPAI51961.2020.00032.

[15] R. Kumar, C. M. Sharma, V. M. Chariar, S. Hooda, and R. Beri, "Emotion Analysis of News and Social Media Text for Stock Price Prediction using SVM-LSTM-GRU Composite Model," in 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), IEEE, May 2022, pp. 329–333. doi: 10.1109/CISES54857.2022.9844375.

[16] L. Setiawan, N. F. Muntasir, S. Fahreza, and A. Sholahuddin, "Prediction of Stock Price Data of PT. Ramayana Lestari SentosaTbk. using Long Short Term Memory Model," in 2021 International Conference on Artificial Intelligence and Big Data Analytics, IEEE, Oct. 2021, pp. 226–230. doi: 10.1109/ICAIBDA53487.2021.9689702.

[17] J. Chen, J. Du, Z. Xue, and F. Kou, "Prediction of Financial Big Data Stock Trends Based on Attention Mechanism," in 2020 IEEE International Conference on Knowledge Graph (ICKG), IEEE, Aug. 2020, pp. 152–156. doi: 10.1109/ICBK50248.2020.00031.

[18] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, "Stock Price Prediction using Bi-Directional LSTM based Sequence to Sequence Modeling and Multitask Learning," in 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, Oct. 2020, pp. 0078–0086. doi: 10.1109/UEMCON51285.2020.9298066.

[19] M. K. Ho, H. Darman, and S. Musa, "Stock Price Prediction Using ARIMA, Neural Network and LSTM Models," J Phys ConfSer, vol. 1988, no. 1, p. 012041, Jul. 2021, doi: 10.1088/1742-6596/1988/1/012041.

[20] Y. Zhu, "Stock price prediction using the RNN model," J Phys ConfSer, vol. 1650, p. 032103, Oct. 2020, doi: 10.1088/1742-6596/1650/3/032103.

[21] Md. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), IEEE, Oct. 2020, pp. 87–92. doi: 10.1109/NILES50944.2020.9257950.

[22] T. Gao, Y. Chai, and Y. Liu, "Applying long short term momory neural networks for predicting stock closing price," in 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), IEEE, Nov. 2017, pp. 575–578. doi: 10.1109/ICSESS.2017.8342981.

[23] "'Yahoo's financial, "Financial Stock Prices," 2023. https://finance.yahoo.com/.'"

[24] C. Chou, J. Park, and E. Chou, "Predicting Stock Closing Price After COVID-19 Based on Sentiment Analysis and LSTM," in 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), IEEE, Mar. 2021, pp. 2752–2756. doi: 10.1109/IAEAC50856.2021.9390845.

[25] Y. Zhao and Z. Chen, "Forecasting stock price movement: new evidence from a novel hybrid deep learning model," Journal of Asian Business and Economic Studies, vol. 29, no. 2, pp. 91–104, May 2022, doi: 10.1108/JABES-05-2021-0061.

[26] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, "Stock Price Prediction using Bi-Directional LSTM based Sequence to Sequence Modeling and Multitask Learning," in 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, Oct. 2020, pp. 0078–0086. doi: 10.1109/UEMCON51285.2020.9298066.

[27] Beldar, Kavita K., M. D. Gayakwad, and M. K. Beldar. 2016. "Optimizing Analytical Queries on Probabilistic Databases with Unmerged Duplicates Using MapReduce." Int. J. Innov. Res. Comput. Commun. Eng 4: 9651–59.

[28] Beldar, Kavita K., M. D. Gayakwad, Debnath Bhattacharyya, and Hye-Jin Kim. 2016a. "Query Evaluation on Probabilistic Databases Using Indexing and MapReduce." International Journal of Database Theory and Application 9 (10): 363–78.

[29] Beldar, Kavita K., M. D. Gayakwad, Debnath Bhattacharyya, and Tai-Hoon Kim. 2016b. "A Comparative Analysis on Contingence Structured Data Methodologies." International Journal of Software Engineering and Its Applications 10 (5): 13–22.

[30] Beldar, Miss Menka K., M. D. Gayakwad, and Miss Kavita K. Beldar. 2018. "Altruistic Content Voting System Using Crowdsourcing." International Journal of Scientific Research and Review 7 (5): 477–86.

[31] Beldar, Miss Menka K., M. D. Gayakwad, Miss Kavita K. Beldar, and M. K. Beldar. 2018. "Survey on Classification of Online Reviews Based on Social Networking." IJFRCSCE 4 (3): 55.

[32] Boukhari, Mahamat Adam, Prof MilnidGayakwad, and Prof DrSuhasPatil. 2019. "Survey on Inappropriate Content Detection in Online Social Media." International Journal of Innovative Research in Science, Engineering and Technology 8 (9): 9297–9302.

[33] Gayakwad, M. D., and B. D. Phulpagar. 2013. "Research Article Review on Various Searching Methodologies and Comparative Analysis for Re-Ranking the Searched Results." International Journal of Recent Scientific Research 4: 1817–20.

[34] Gayakwad, Milind. 2011. "VLAN Implementation Using Ip over ATM." Journal of Engineering Research and Studies 2 (4): 186–92.

[35] V. Vamsi Krishna, G. Gopinath. (2023). Effectiveness of ICT-Based Intelligent System Using Cloud Computing for Research Programme During Covid-19 Pandemic. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 418–427. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2688

[36] Gayakwad, Milind, and SuhasPatil. 2020. "Content Modelling for Unbiased Information Analysis." Libr. Philos. Pract, 1–17.

[37] "Analysis of Methodologies to Model the Content for Conveying the Correct Information." In 2021 International Conference on Computing, Communication and Green Engineering (CCGE), 1–4. IEEE.

[38] 2021b. "Assessment of Source, Medium, and Intercommunication for Assessing the Credibility of Content." In 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 1–5. IEEE.

[39] Gayakwad, Milind, SuhasPatil, Rahul Joshi, SudhanshuGonge, and Sandeep DwarkanathPande. 2022a. "Credibility Evaluation of User-Generated Content Using Novel Multinomial 37]38. Classification Technique." International Journal on Recent and Innovation Trends in Computing and Communication 10 (2s): 151–57.

[40] Gayakwad, Milind, SuhasPatil, Amol Kadam, Shashank Joshi, KetanKotecha, Rahul Joshi, Sharnil Pandya, et al. 2022. "Credibility Analysis of User-Designed Content Using Machine Learning Techniques." Applied System Innovation 5 (2): 43.

[41] Harane, Swati T., GajananBhole, and Milind Gayakwad. 2017. "SECURE SEARCH OVER ENCRYPTED DATA TECHNIQUES: SURVEY." International Journal of Advanced Research in Computer Science 8 (7).

[42] KavitaShevale, GajananBhole, Milind Gayakwad. 2017. "Literature Review on Probabilistic Threshold Query on Uncertain Data." International Journal of Current Research and Review 9 (6): 52482–84.

[43] Mahamat Adam Boukhari, Milind Gayakwad. 2019. "An Experimental Technique on Fake News Detection in Online Social Media." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 8 (8S3): 526–30.

[44] Maurya, Maruti, and Milind Gayakwad. 2020. "People, Technologies, and Organizations Interactions in a Social Commerce Era." In Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI-2018), 836–49. Springer International Publishing.

[45] Milind Gayakwad, B. D. Phulpagar. 2013. "Requirement Specific Search." IJARCSSE 3 (11): 121.

[46] Panicker, Aishwarya, Milind Gayakwad, Sandeep Vanjale, PramodJadhav, Prakash Devale, and SuhasPatil. n.d. "Fake News Detection Using Machine Learning Framework."

[47] Sharma, Jitin, Prashant C. Chavan, T. B. Patil, Supriya C. Sawant, and Milind Gaykawad. 2022. "A Comparative Analysis of Brain Tumor Classification and Prediction Techniques by Applying MRI Images Encompassing SVM and CNN with Transfer Learning Method." Journal of Algebraic Statistics 13 (3): 393–405.

[48] Shevale, Kavita, GajananBhole, and Milind Gayakwad. 2017. "Probabilistic Threshold Query on Uncertain Data Using SVM." Int. J. Adv. Res. Comput. Sci 8: 1967–69.

[49] Singh, Mahendra Kumar, Amol K. Kadam, Milind Gayakwad, PramodJadhav, Vinayak N. Patil, Prasad Kadam, Vinod Patil, and SunitaDhotre. n.d. "No Title]." https://www.researchgate.net/profile/Amol-Kadam-3/publication/363210290_An_Impirical_Approach_for_Underwater_Image_Quality_Enhancement_and_Object_Detection_using_Deep_Learning_An_Impirical_Approach_for_Underwater_Image_Quality_Enhancement_and_Object_Detection_using_Deep_L/links/6311901cacd814437ff7a165/An-Impirical-Approach-for-Underwater-Image-Quality-Enhancement-and-Object-Detection-using-Deep-Learning-An-Impirical-Approach-for-Underwater-Image-Quality-Enhancement-and-Object-Detection-using-Deep-L.pdf.

[50] Mishra, S., P. Paygude, and S. Chaudhary. 2018. "Use of Data Mining in Crop Yield Prediction." 2018 2nd International. https://ieeexplore.ieee.org/abstract/document/8398908/.

[51] Paygude, Priyanka, Aatmic Tiwari, Bhavya Goel, and Akshat Kabra. 2023. "Comparative Analysis of Stock Prices by Regression Analysis and FB Prophet Models." In Data Intelligence and Cognitive Informatics, 295–307. Springer Nature Singapore.

[52] Tyagi, Shubham, Rishabh Solanki, Adarsh Tiwari, Rohit Ray, and Priyanka Paygude. 2021. "Analysis of Various Machine Learning Models for Detecting Depression in Twitter Tweets." Turkish Online Journal of Qualitative Inquiry 12 (7): 6616–25

[53] Paygude, Priyanka, and Aatmic Tiwari. 2021. "Comparative Analysis on Different Generations of Cryptocurrency." International Journal of Future Generation Communication and Networking 14 (1): 3016–26.

[54] Paygude, Priyanka, and Shashank Joshi. 2020. "Optimization of Test Case Prioritization through Calibration of Genetic Algorithm." Solid State Technology 63 (1s): 2377–86.

[55] Varshney, Abhishek, Saumya Asthana, Chetan Sharma, Swapnil Patil, Ankur Kashyap, and Priyanka Paygude. 2020. "Study of Solving Knapsack Problem Using Genetic Algorithm." Solid State Technology, February, 93–99.

[56] Malge, Shrikant, Prof Snehal D. Chaudhary, and Priyanka Paygude. 2017. "Secure and Policy Update Method on Big Data Access in Cloud Storage." IOSR Journal of Computer Engineering 19 (03): 48–52.

[57] Mishra, S., P. Paygude, and S. Chaudhary. 2018. "Use of Data Mining in Crop Yield Prediction." 2018 2nd International. https://ieeexplore.ieee.org/abstract/document/8398908/.

[58] Paygude, P., D. S. D. Joshi, and D. M. Joshi. 2020. "Fault Aware Test Case Prioritization in Regression Testing Using Genetic Algorithm." International Journal of Emerging Trends in. https://scholar.google.ca/scholar?cluster=11372764101768453854&hl=en&as_sdt=0,5&sciodt=0,5.

[59] Paygude, P., and S. D. Joshi. 2020. "Use of Evolutionary Algorithm in Regression Test Case Prioritization: A Review." Proceeding of the International Conference on. https://link.springer.com/chapter/10.1007/978-3-030-24643-3_6.