Design of Incremental Bidirectional LSTM Networks Approach for Stock Price Prediction Based on Fundamental and Technical Features and its Performance Analysis

Dara Rajesh Babu[1], Prof. Bachala Sathyanarayana[2]

[1] Research Scholar, Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Pradesh515003, India.

[2] Professor, Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Pradesh-515003, India.

Abstract- Stock price prediction plays a pivotal role in financial decision-making and investment strategies. This work aims to enhance stock price prediction and decision-making accuracy using Incremental Bidirectional Long Short-Term Memory (BiLSTM) networks *M. Jia et al. [2]*. The work compares the predictive capabilities of traditional LSTM models *Y. Bing et al. [1]* with those of the proposed Incremental BiLSTM models, incorporating both fundamental and technical analysis parameters for improved accuracy and decision support. *Gao P et al. [4]* have primarily focused on utilizing fundamental parameters such as open, close, high, low, and volume alongside technical parameters including Moving Average, Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Bollinger Bands, and Momentum *Dara Rajesh Babu, Bachala Sathyanarayana [5]*. These features have shown promise in predicting stock prices using LSTM models. However, the limitations of LSTM models in capturing complex temporal dependencies which has led to the exploration of more advanced architecture.In this paper we considered a novel approach for creating Incremental Bidirectional Long Short-Term Memory (BiLSTM) models are implemented for better performance *Hum Nath et al.* [7] using the selected input variables. The proposed method performances are evaluated in terms of Directional Accuracy, Hit Rate and Quantile Loss. From the experimental findings it has been observed that the multi-layer BiLSTM model exhibits a better fit to the data when compared to single layer BiLSTM model and achieved higher prediction accuracy.

Keywords Stock price prediction, Incremental Bidirectional LSTM, Financial decision-making, Investment strategies.

1. Introduction

The accurate prediction of stock prices has been a long-standing challenge in the field of finance and investment. Over the years, researchers have explored various methodologies to enhance prediction accuracy, leveraging both fundamental *Archit Kapoor et al. [8]* and technical indicators. In recent times, deep learning techniques, particularly Long Short-Term Memory (LSTM) models, have shown promising results in capturing intricate patterns and relationships within financial time series data *Ayan Maiti et al. [9]*.

This research aims to build upon the existing work of predicting stock prices using LSTM models by incorporating additional complexity and context through the utilization of Bidirectional LSTM

(BiLSTM) models *M. Jia et al. [2]*. The integration of both fundamental parameters (such as open, close, high, low, and volume) and technical parameters (including Moving Average, Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Bollinger Bands, and Momentum) *Hum Nath et al.* [7] offers a comprehensive approach to capturing the dynamics of stock price movements.

In this work, we propose a novel approach that employs Incremental BiLSTM networks to further enhance prediction accuracy. The bidirectional nature of the model enables the utilization of both past and future data points, effectively capturing temporal dependencies that may impact stock price behavior. This research extends the predictive capabilities beyond mere price forecasting by incorporating the prediction of stock volatility, as well as generating buy and sell decision signals.

The subsequent sections of this paper provide a related work, detailed methodology, experimental setup, results and analysis and conclusion and future works. Through this research, we seek to offer a comprehensive and robust solution for stock price prediction that integrates various dimensions of financial data analysis, ultimately aiding investors in making well-informed investment decisions.

2 Related Work

Artificial Neural Networks (ANNs) offer a promising avenue for stock price prediction, leveraging adaptive weights for effective forecasting. As emphasized by *Y. Bing et al. [1]*, predicting stock market prices is crucial for investors, directly impacting their returns. Traditionally, experts relied on historical data, including prices, volumes, patterns, and trends. However, modern stock prediction considers a wider range of factors, such as financial health, socio-economic conditions, politics, and even natural disasters. The inherent uncertainty and complexity associated with stock market returns demand more sophisticated prediction techniques. Conventional methodologies often fall short in providing accurate forecasts due to the intrinsic volatility of stock markets. To tackle this challenge, researchers have delved into advanced analytical techniques, ranging from purely mathematical models to expert systems and even neural networks. These endeavors have given rise to sophisticated financial trading systems aimed at enhancing stock price prediction accuracy.

Bidirectional Long Short-Term Memory networks (Bi-LSTMs) within Recurrent Neural Networks (RNNs) are invaluable for analyzing sequential data. *M. Jia et al. [2]* they consider data in both forward and reverse directions, capturing past trends and potential future trajectories. In response to stock market complexity, this study introduces a novel approach using Bidirectional LSTM (BLSTM) neural networks. Unlike traditional models, BLSTM enhances GREE stock price predictions by reducing RMSE and MAE by approximately 24.2% and 19.4%, respectively, while improving deviation accuracy by 0.13%. Despite its simplicity, this approach offers valuable insights for short-term investors.

Predicting stock price indices is a vital area in machine learning, given the market's complexity. *J. Eapen et al. [3]* this study introduces an innovative deep learning model merging Convolutional Neural Networks (CNNs) and Bi-directional Long Short-Term Memory (Bi-LSTM) units. It enhances prediction accuracy by 9% compared to single-pipeline deep learning and significantly outperforms Support Vector Machine regression on the S&P 500 challenge dataset. The study explores various model variations, including CNN kernel sizes and Bi-LSTM unit counts, to ensure improved performance while addressing overfitting concerns.

In their study, *Dara Rajesh Babu and Bachala Sathyanarayana [4]* developed an LSTM model for stock price prediction, using technical analysis indicators derived from historical stock data. These indicators were used as input features for training the LSTM model, which was evaluated using various metrics, including mean absolute error and root mean squared error. This study introduces a stock price prediction system, specifically for APPLE shares, employing LSTM-based neural networks and carefully chosen input variables. Performance assessment includes RMSE, MAPE, and R squared error to aid investment decisions.

Jaiwin Shah et al. [5] tackled the challenge of predicting the volatile and complex stock market. They explored various prediction models and concluded that the Bi-directional Long Short-Term Memory (Bi-LSTM) neural network stands out. This model effectively analyzes historical data from multiple stocks, leveraging time series information to forecast stock price trends. The Bi-LSTM's ability to capture temporal patterns enhances its performance, promising improved stock market predictions.

3. Methodology

3.1. Data Collection and Preprocessing:

The data for the APPLE stock was obtained from Yahoo Finance in a .csv format and saved in a MS Excel document with 2517 days of data. The dataset comprises fundamental data, including open, high, low, and close prices, as well as technical indicators like Moving Average, Exponential Moving Averages (EMA), Moving Average Convergence Divergence (MACD), Bollinger Bands and Momentum. Technical indicators are mathematical calculations performed on data such as price or other technical indicators, and are widely used by active traders for analyzing short-term price movements. The dataset covers the period from January 2011 to December 2020. Perform data preprocessing, including missing value imputation, normalization, and feature scaling to ensure consistent data quality.

S. No.	Feature	Description
1	Open Price	Opening price of the stock on a particular day
2	Close Price	The closing price of the stock at the end of a trading day
3	High Price	The peak price on a particular day
4	Low Price	The lowest price of the stock on a particular day
5	Volume	Represents the total number of shares traded in a given period
6	Moving Average (MA)	Smooth’s out price data over a specified time period
7	Exponential Moving Average (EMA)	Smooth’s out price data as it assigns more weight to recent prices.
8	Moving Average Convergence Divergence (MACD)	Calculates the difference between short-term and long-term moving averages
9	Bollinger Bands	Price's standard deviation from the middle line
10	Momentum	Rate of change in price over a specific time period

Table 1: List of potential features for the model

3.2. Bi-Directional Long Short Term Memory (BI-LSTM)

The Bidirectional LSTM (Bi-LSTM) presents an improved version of the standard LSTM [3]. It enhances the analysis by processing input data in both directions—unidirectional from past to future and vice versa. Consequently, the Bi-LSTM simultaneously preserves information from both historical and future contexts, facilitating a more comprehensive understanding of temporal patterns and dependencies within the data. This enhanced architecture enables the model to capture intricate relationships and dependencies in both temporal directions, contributing to more accurate and robust predictions.

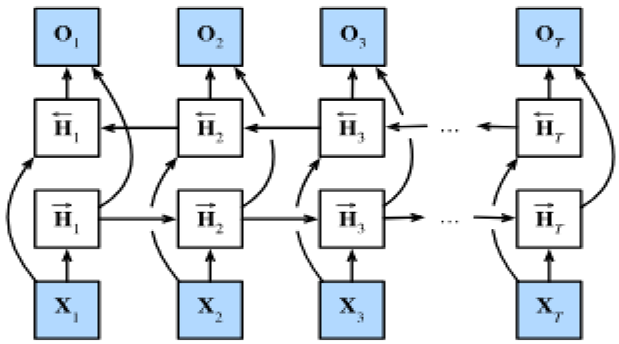


Fig 1: Architecture of a bidirectional RNN

Formally for any time step t, we consider a input $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ (number of examples =n; number of inputs in each example = d) and let the hidden layer activation function be Φ. In the bidirectional architecture, the forward and backward hidden states for this time step are $\overrightarrow{\mathbf{H}}_t \in \mathbb{R}^{n\times h}$ and $\overleftarrow{\mathbf{H}}_t \in \mathbb{R}^{n\times h}$ respectively, where *h* is the number of hidden units. The forward and backward hidden state updates are as follows:

$$\begin{aligned}\overrightarrow{\mathbf{H}}_t &= \phi(\mathbf{X}_t\mathbf{W}_{\mathrm{xh}}^{(f)} + \overrightarrow{\mathbf{H}}_{t-1}\mathbf{W}_{\mathrm{hh}}^{(f)} + \mathbf{b}_{\mathrm{h}}^{(f)}),\\ \overleftarrow{\mathbf{H}}_t &= \phi(\mathbf{X}_t\mathbf{W}_{\mathrm{xh}}^{(b)} + \overleftarrow{\mathbf{H}}_{t+1}\mathbf{W}_{\mathrm{hh}}^{(b)} + \mathbf{b}_{\mathrm{h}}^{(b)}),\end{aligned}$$

where the weights $\mathbf{W}_{\mathrm{xh}}^{(f)} \in \mathbb{R}^{d\times h}, \mathbf{W}_{\mathrm{hh}}^{(f)} \in \mathbb{R}^{h\times h}, \mathbf{W}_{\mathrm{xh}}^{(b)} \in \mathbb{R}^{d\times h}$, and $\mathbf{W}_{\mathrm{hh}}^{(b)} \in \mathbb{R}^{h\times h}$ and the biases $\mathbf{b}_{\mathrm{h}}^{(f)} \in \mathbb{R}^{1\times h}$ and $\mathbf{b}_{\mathrm{h}}^{(b)} \in \mathbb{R}^{1\times h}$ are all the model parameters.

Next, we concatenate the forward and backward hidden states $\overrightarrow{\mathbf{H}}_t$ and $\overleftarrow{\mathbf{H}}_t$ to obtain the hidden state $\mathbf{H}_t \in \mathbb{R}^{n\times 2h}$ for feeding into the output layer. In deep bidirectional RNNs with multiple hidden layers, such information is passed on as *input* to the next bidirectional layer. Last, the output layer computes the output $\mathbf{O}_t \in \mathbb{R}^{n\times q}$ (number of outputs = q):

$$\mathbf{O}_t = \mathbf{H}_t\mathbf{W}_{\mathrm{hq}} + \mathbf{b}_{\mathrm{q}}.$$

Here, the weight matrix $\mathbf{W}_{\mathrm{hq}} \in \mathbb{R}^{2h\times q}$ and the bias $\mathbf{b}_{\mathrm{q}} \in \mathbb{R}^{1\times q}$ are the model parameters of the output layer.

3.2.1 Algorithm for hyper parameter tuning procedure

Input preparation: split train and validation datasets and create the input of the form (#observations, timesteps, #features)
Input: (#observations, timesteps, #features); choice of optimizers, learning rates and batch sizes Initialize: set number of epochs sufficiently large.


```
1. Initialize
2. for each optimizer in 'optimizers' do
     3. for each learning_rate in 'learning_rates' do
         4. for each batch_size in 'batch_sizes' do
             Create a new sequential model.
             Add desired layers and architecture to the model.

              Compile the model using the current optimizer and learning_rate.

              Split the dataset into training and validation sets (e.g., 80% train, 20% validation).

              Set up early stopping.

              for each epoch from 1 to 'num_epochs' do
                  Train the model on the training data using the current batch_size.
                  Use the validation data for monitoring and early stopping.
                  if early stopping criteria met then
                     end do (Stop training)
                  end if
              end do (Epochs)

              Evaluate the trained model on the validation set.
              Compute the validation loss.

              if validation loss is better (lower) than 'best_score' then
                  Update 'best_score' with the current validation loss.
                  Set 'best_model' to the current model.
              end if

         5. end do (Batch Sizes)
     6. end do (Learning Rates)
 7. end do (Optimizers)

 8. Return 'best_model' as the final model selected based on hyper parameter tuning.
end
```

3.2.2 Algorithm for Incremental BiLSTM model

Input preparation split train and test datasets and create and input of the form (#observations, timesteps, #features)
Input: (#observations, timesteps, #features); chosen hyperparameter (optimizer, learning rate, batch size) obtained from hyperparameter tuning algorithm for each model
Initialize: set number of epochs sufficiently large.

```
1. Create a BiLSTM model:
    Initialize an empty sequential model.
```

Add a Bidirectional LSTM layer with the desired number of units and an appropriate activation function.
Compile the model using the chosen optimizer, learning rate, and an appropriate loss function.

2. Split the training data into training and validation sets (e.g., 80% train, 20% validation).

3. Set up early stopping with a patience of 'patience'.

4. for each epoch from 1 to 'num_epochs' do
   Train the model on the training data with the chosen batch size.
   Use the validation data for monitoring and early stopping.
   if early stopping criteria met then
      end do (Stop training)
   end if
   continue

5. Evaluate the trained model on the test data to obtain performance metrics (e.g., accuracy, mean squared error, etc.).

6. Return 'trained_model' as the final BiLSTM model.
end

4 Experimental Setup

The implementation of the proposed models occurred on a Windows 10 system featuring an Intel i3 processor, 8 GB of RAM, and an NVidia GeForce GTX 512 graphics processor.

4.1 Preprocessing the Dataset

The collected dataset comprises OHLC (Open, High, Low, Close, and Volume) prices, containing a total of 2517 samples. This dataset was divided into training and testing sets using an 80:20 ratio. After removing any null values, the data underwent a cleaning process. Subsequently, the MinMaxScaler from the sklearn library was utilized to normalize the data. This normalization method involves subtracting the minimum value in the dataset from each data point and then dividing every data point by the maximum value in the dataset. This procedure was applied to multiple time series, each representing Opening Price, Closing Price, High Price, and Low Price and Volume.

4.2. Input Layer

- The input matrix is composed of ten distinct columns: Open, High, Low, Close, Volume, and a set of technical indicators including Moving Average Convergence Divergence (MACD), Moving Average (MA), Exponential Moving Average (EMA), Bollinger Bands, and Momentum. These columns collectively constitute a time series dataset.

- The output matrix consists of a solitary column named "Adjusted Close," which determines the market's opening price for the subsequent day.
- The data was divided into a training set and a test set with a ratio of 9:1.
- Following the normalization process, every element of the input data has been transformed to fall within the range of 0 to 1.
- Each entry within our dataset incorporates the preceding one hundred days as its input, resulting in our input dataset being structured as a 3-D matrix with dimensions (N, 100, 9).

4.3. Implementation of the Incremental BI-LSTM model

The central purpose of the Incremental Bidirectional Long Short-Term Memory (Bi-LSTM) is to discern the information that should be retained or discarded within the Bi-LSTM cell, as illustrated in Figure 1. This determination is facilitated by the sigmoid layer. When the output is '0', the information is disregarded, while an output of '1' signifies information retention.

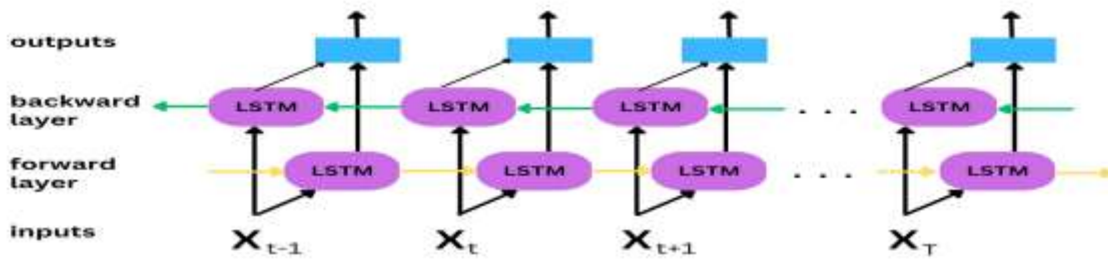


Fig 2: Single Layer BiLSTM architecture

The subsequent stage involves determining the specific information to be stored within the Bi-LSTM cell. This task is carried out through a set of two layers incorporated within the Bi-LSTM cell. The initial layer features a sigmoid activation, responsible for discerning the values to be updated in order to attain the intended output.

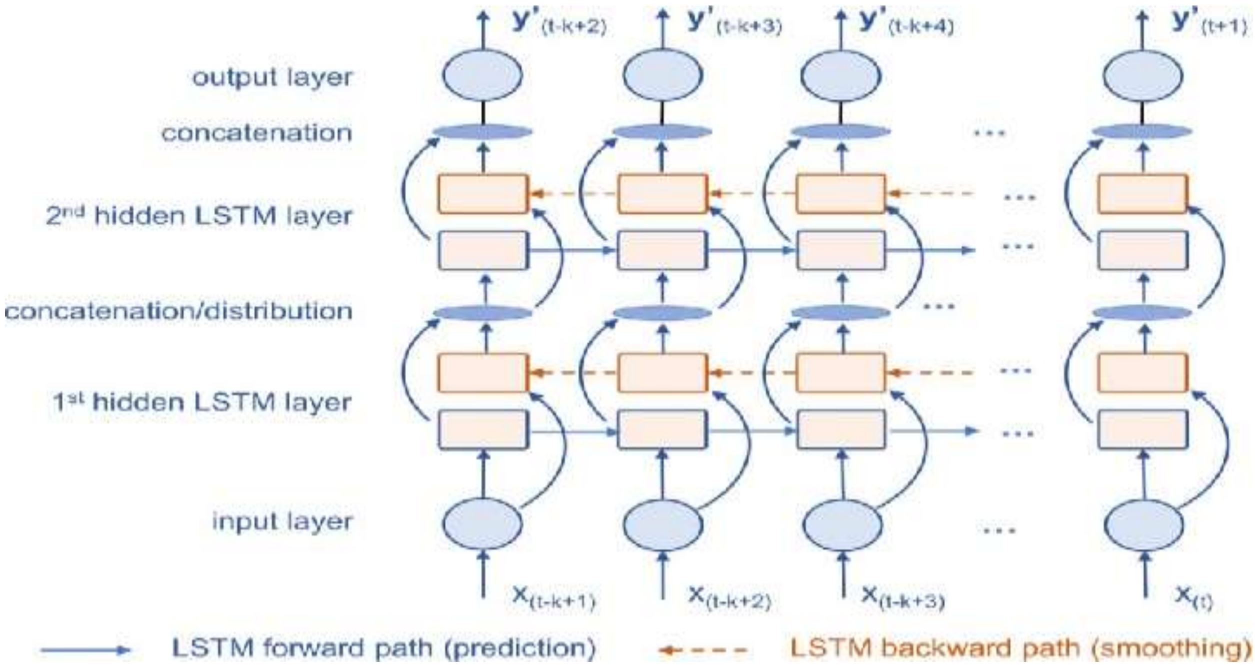


Fig 3: Multi-Layer BiLSTM architecture

The subsequent layer is implemented using the tanh activation function. This layer assumes the responsibility of generating fresh candidate values, which can be utilized within our Bi-LSTM cell at a later point. Following this, the two layers are merged to formulate a novel Bi-LSTM cell state, housing solely the essential information.

The outcome of the Bi-LSTM yields a refined iteration of our cell state. We subject this output to a tanh function to constrain its value range between -1 and 1. Following this transformation, we merge the output with the sigmoid layer, ensuring that the throughput is tailored to meet our specific criteria. As a result, only the pertinent information is presented as the final output.

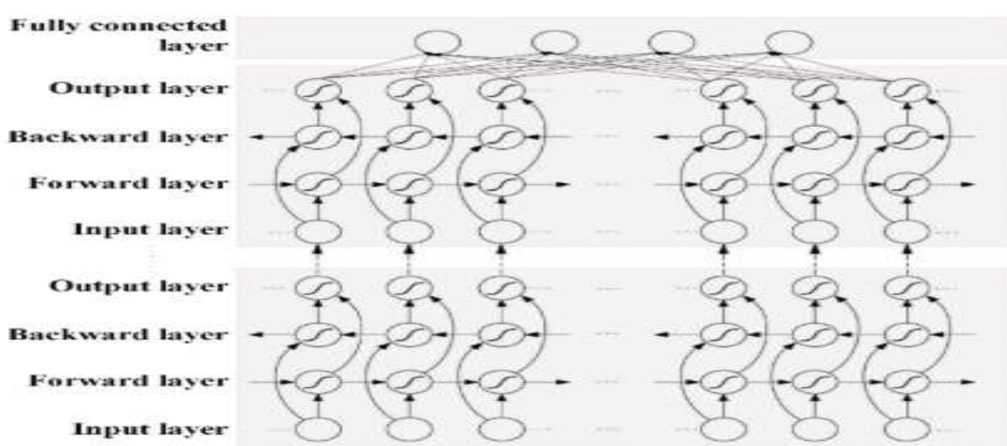


Fig 4: The proposed Multi-Layer BiLSTM architecture for stock price prediction.

4.4. Hyperparameter Tuning

The process of selecting the optimal model architecture involves thorough exploration across two main categories: (a) single-layer BiLSTM and (b) multi-layer BiLSTM. In the single-layer BiLSTM model, there exists only a single LSTM layer within the model. Conversely, in the multi-layer BiLSTM model, there are multiple LSTM layers integrated into the model. In the context of the single-layer BiLSTM architecture, six distinct models are experimented with, employing 10, 30, 50, 100, 150, and 200 neurons, all optimized using the Adam optimizer learning rate 0.001 and batch size is 64. Each model undergoes 10 executions for every combination of hyperparameters, and the RMSE score is computed. The selection of the most optimal hyperparameter combination for a given model based on achieving the lowest RMSE score when evaluated on the validation data. Likewise, the most suitable hyperparameters are determined for six distinct multi-layer BiLSTM models, characterized by configurations such as (10,5), (20,10), (50,20), (100,50), and (100,50,20) neurons. In this context, the notation (n1, n2) represents the configuration of the first and second hidden layers.

4.5 Model Performance Metrics

4.5.1 Direction accuracy - Evaluates the directional correctness of a predictive model it measures the percentage of correct directional predictions made by the model, it assesses whether the model correctly predicts whether the target variable will go up or down. Direction accuracy is typically used when the goal is not only to predict the exact value of a target variable but also to capture the overall trend or direction of the variable.

Direction Accuracy = (Number of Correct Directional Predictions) / (Total Number of Predictions) * 100

Where:

"Number of Correct Directional Predictions" is the count of predictions where the model correctly predicted the direction (e.g., the model predicted an increase, and the actual value increased).

"Total Number of Predictions" is the total number of predictions made by the model.

4.5.2 Hit rate - It is also known as the true positive rate (TPR) or sensitivity, is a performance metric used in binary classification to measure the proportion of true positive predictions (correctly predicted positives) out of all actual positive instances. It quantifies the model's ability to correctly identify positive cases.

TPR = True Positives (TP) / True Positives (TP) + False Negatives (FN)

True Positives (TP) are the instances that are actually positive and correctly predicted as positive by the model.
False Negatives (FN) are the instances that are actually positive but incorrectly predicted as negative by the model.

4.5.3 Quantile loss - To evaluate the accuracy of predicted quantiles or percentiles of the target variable. It measures the discrepancy between the predicted quantiles and the actual quantiles of the target variable.

$$\ell(y,\hat{y}) = \begin{cases} \alpha \cdot (y - \hat{y}) & , \hat{y} \leq y \\ (1-\alpha) \cdot (\hat{y} - y) & , \hat{y} > y \end{cases}$$

5. Results and Analysis

5.1 Single Layer BiLSTM Results

Single-layer model is trained using the training dataset, which constitutes 80% of the total dataset. The selection of the best-performing model is based on the performance score computed on the test dataset. The results in Table 1 indicate that the model with 150 neurons surpasses its counterparts in terms of Directional Accuracy, Hit Rate and Quantile Loss. Therefore, it is safe to declare that the single-layer BiLSTM model with 150 neurons stands out as the frontrunner among the different models.

	L.R	S.V.R	R.F	LSTM	S.L BiLSTM
Directional Accuracy	0.43	0.45	0.41	0.53	0.58
Hit Rate	0.61	0.63	0.59	0.67	0.69
Quantile Loss	5.56	3.87	1.77	1.35	1.15

Table 2: The performance score achieved by Single Layer BiLSTM compared with existing models

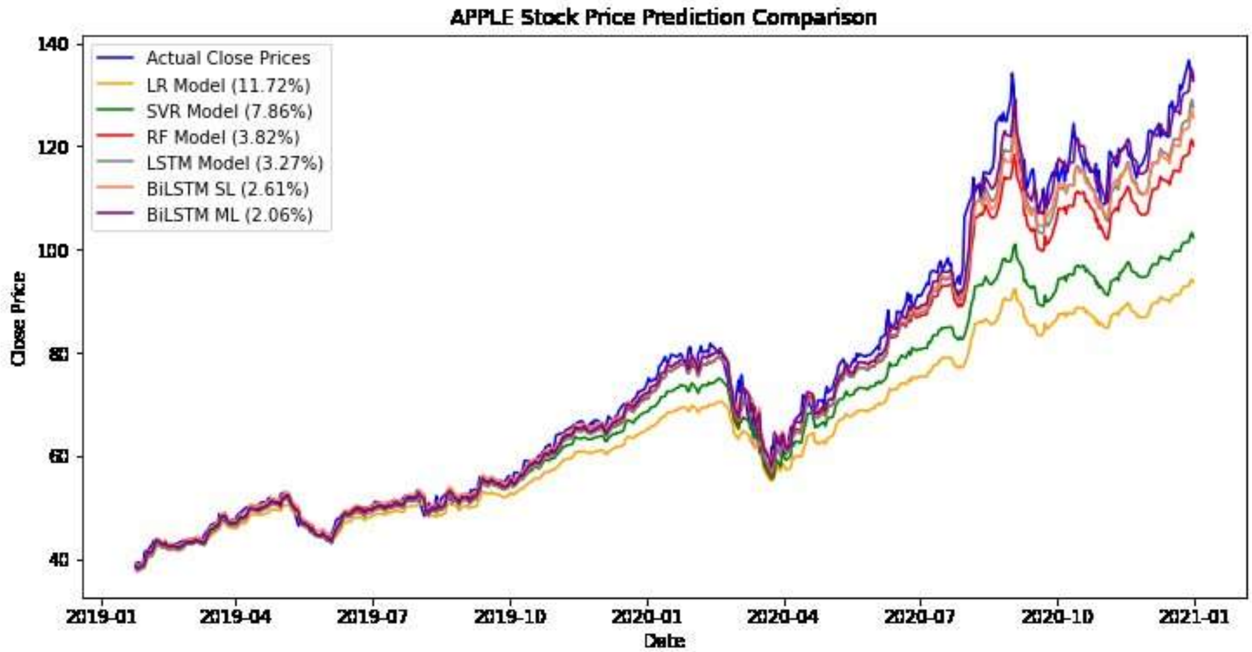
5.2 Multi-layer BiLSTM Results

Multi-layer LSTM model is trained using the training dataset, and the selection of the optimal model is based on performance scores computed using the test dataset. The data entries displayed in Table 2 represent the performance metrics of multi-layer BiLSTM models. The table illustrates that the model featuring (150, 100) neurons outperforms single-layer BiLSTM. Furthermore, it highlights that multi-layer BiLSTM models exhibit improved performance compared to their single-layer counterpart.

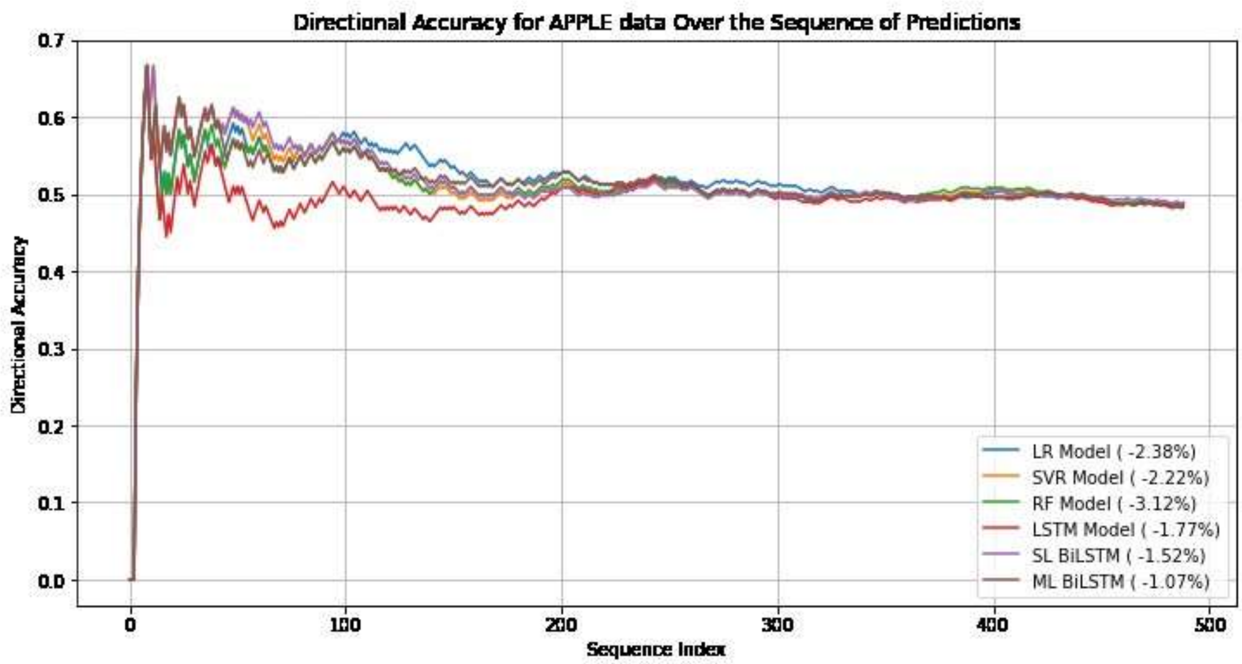
	S.L BiLSTM	M.L BiLSTM
Directional Accuracy	0.58	0.63
Hit Rate	0.69	0.74
Quantile Loss	1.15	0.81

Table 3: Comparision of performance scores between Single Layer BiLSTM vs Multi Layer BiLSTM model

Fig 5(a) The resulting plot provides a visual comparison of the actual Apple stock prices and the predictions made by various models along with performance (percentage deviation) relative to the actual prices. This allows for an easy visual assessment of how well each model performs in predicting stock prices. **Fig 5(b)** The resulting plot shows the directional accuracy scores for each of the six models over the sequence of predictions. It allows for a visual comparison of how well each model performs in terms of directional accuracy as the sequence progresses.

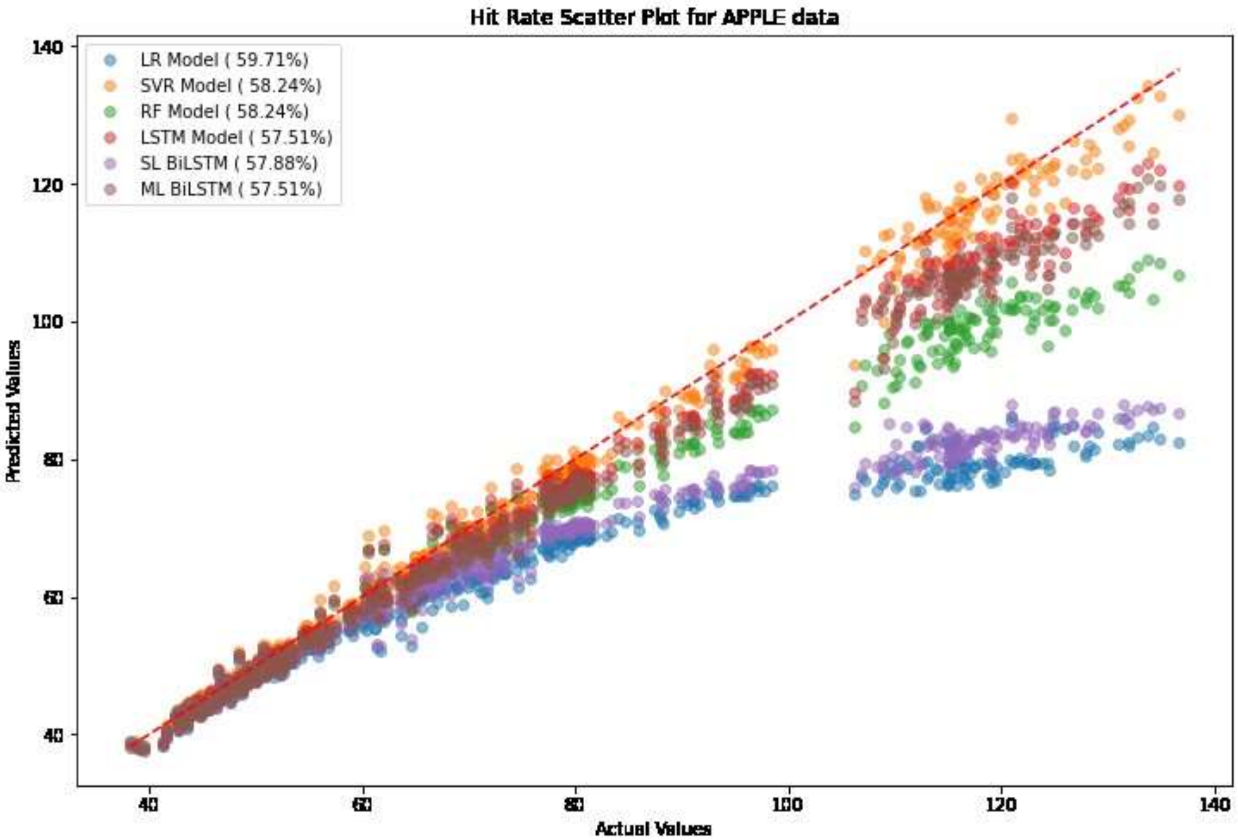


5(a)

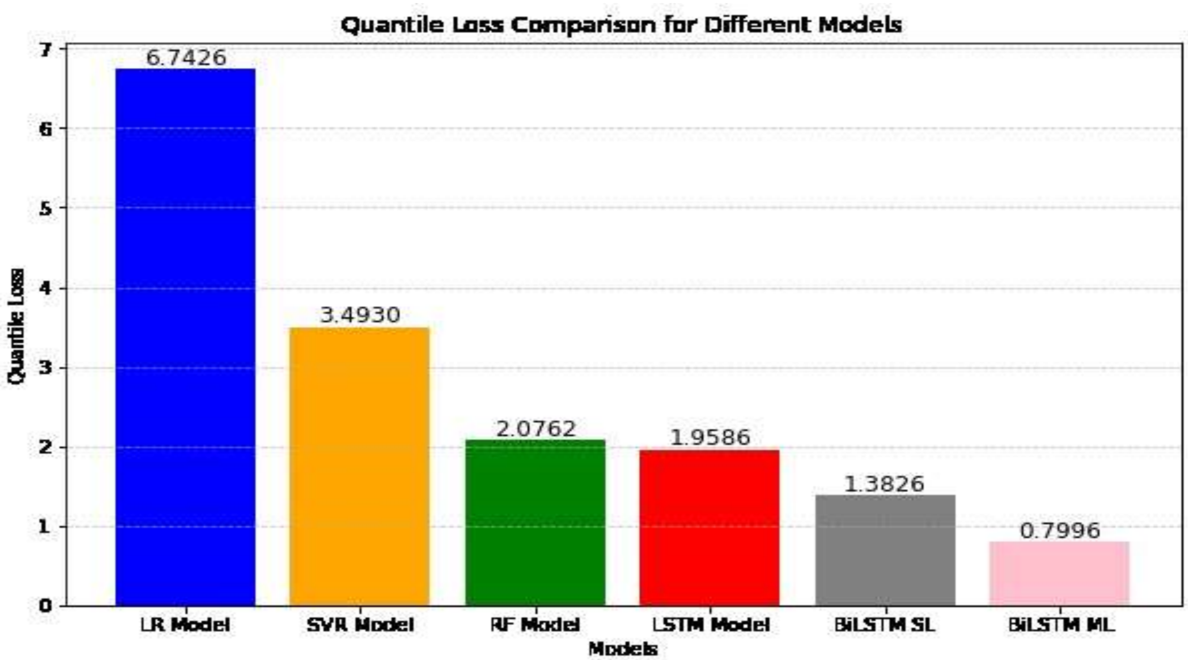


5(b)

Fig 5(c) The scatter plot represent the predicted values compared to the actual values, and the diagonal red dashed line represents perfect predictions. Points close to this line, indicating accurate predictions, points above the line indicate overpredictions, and points below the line indicate underpredictions. **Fig 5(d)** The resulting heatmap visually represents quantile loss values for each quantile level and time step. Darker areas indicate higher quantile loss values, while lighter areas represent lower losses. This visualization helps in understanding how different quantile levels and time steps impact the performance of the models, particularly in terms of predicting extreme quantiles.

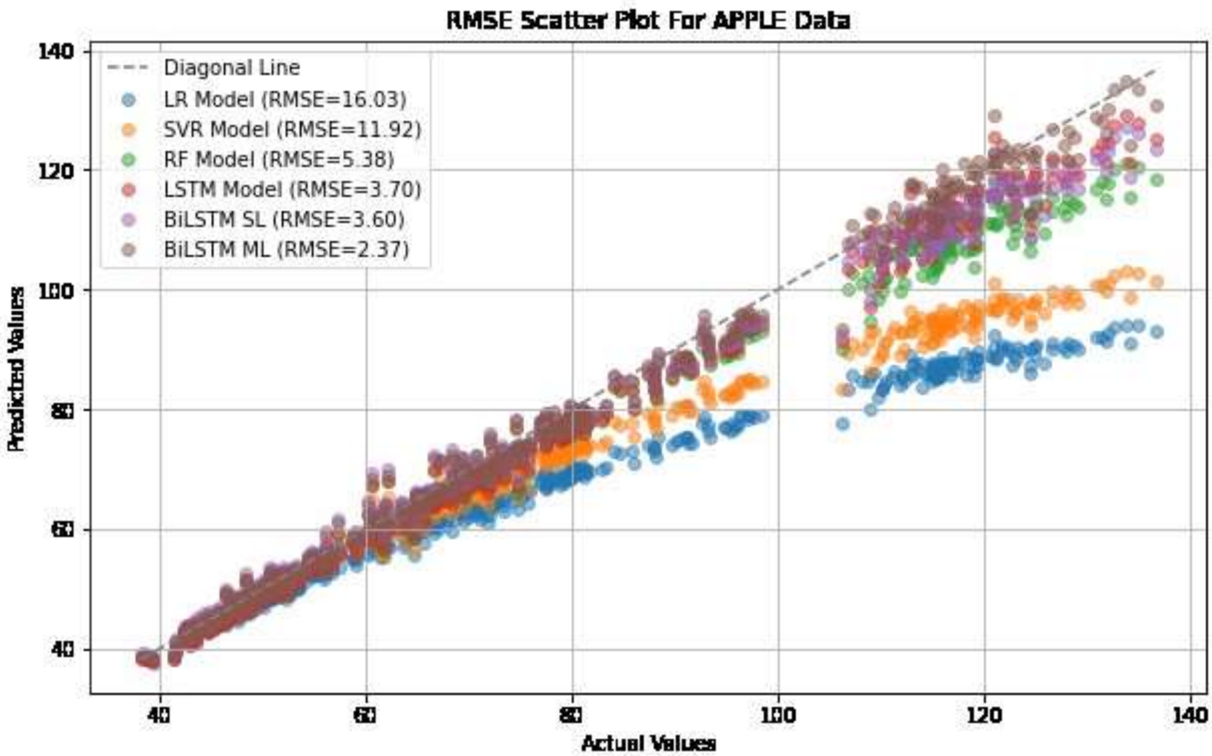


5(c)

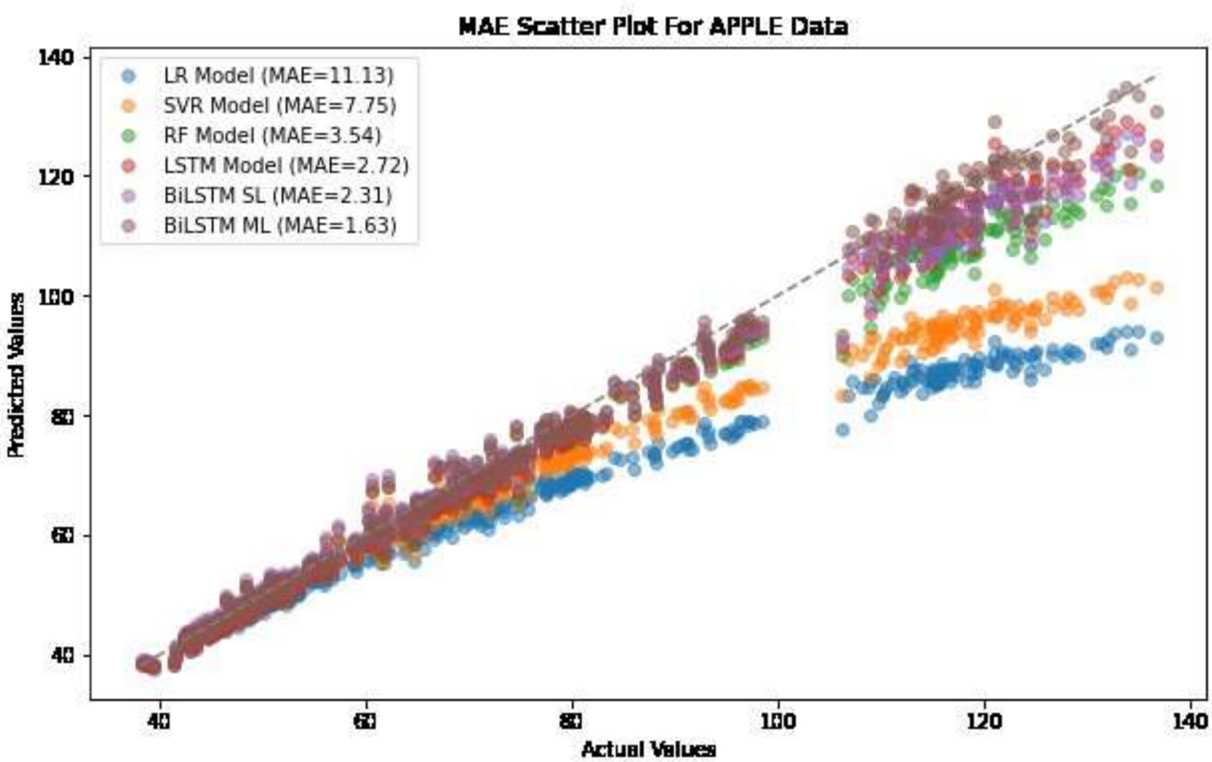


5(d)

Fig 5(e) & 5(f) represents the scatter plot to measure the performance obtained from different models using RMSE and MAE metrics.

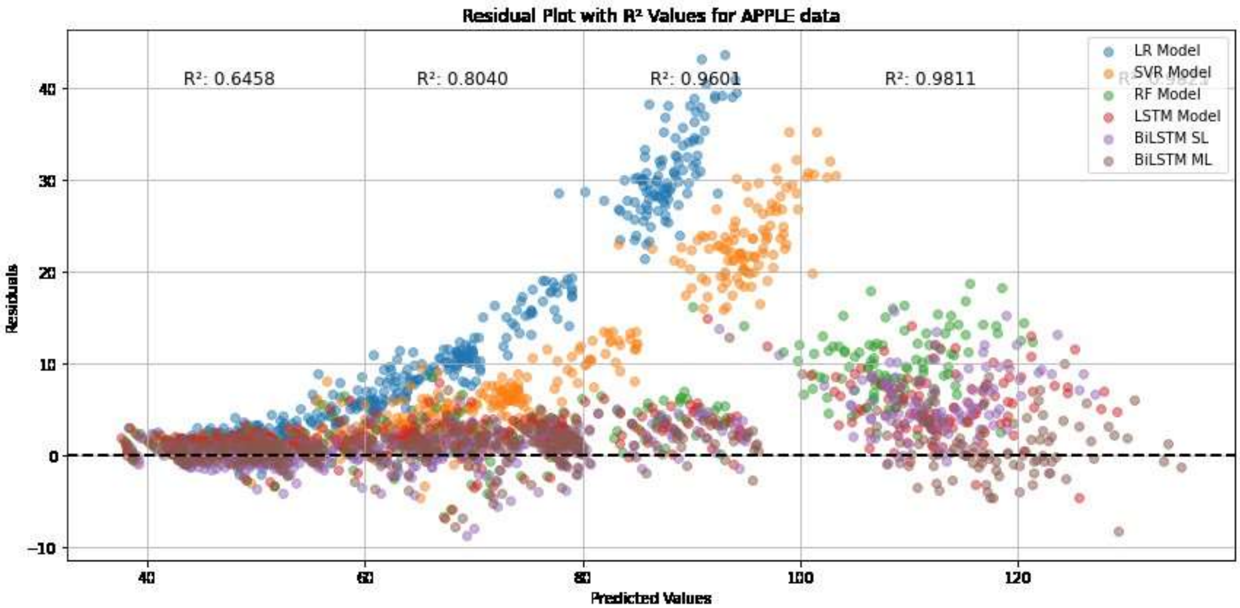


5(e)



5(f)

Fig 5(g) the residual plot helps visualize the relationship between predicted values and their corresponding residuals for each model. R^2 values provide a measure of how well each model fits the data, and the horizontal line at y = 0 helps identify patterns of overestimation or underestimation in the predictions.



5(g)

6. Conclusion and Future Work

The model introduced in this paper has demonstrated superior performance compared to standard model, with only a few exceptions. Its potential is especially promising when contrasted with previous methodologies. Multi-Layer BiLSTM model has captured **DA** of **0.63** which indicate us how well the model performs in terms of directional accuracy as the sequence progresses which is better result when compared to the Single-Layer BiLSTM model value is **0.58**. In terms of **quantile loss** also M.L BiLSTM has shown loss of **0.81** when compared with S.L BiLstm **1.15.** Leveraging extensive datasets has allowed our model to uncover more intricate patterns, enabling better calibration of layer weights. Bi-directional LSTMs prove advantageous by retaining information from both past and future states, facilitating tracking of short-term dependencies within stock prices over extended periods.

Comparing our model to other stock prediction methods reveals a noteworthy enhancement in accuracy. In the coming times, our research aims to delve into the integration of unstructured textual data into our predictive models. This includes incorporating data like investor sentiment gathered from social media, earnings reports from underlying companies, immediate policy-related news, and research reports provided by market analysts. Additionally, we are considering the exploration of hybrid predictive models that combine LSTM with other neural network architectures. To enhance prediction accuracy, our future endeavors involve the implementation of hybrid optimization algorithms. These algorithms will combine existing local optimizers with global optimizers such as genetic algorithms and particle swarm optimization algorithms to fine-tune model parameters effectively.

References

1. Y. Bing, J. K. Hao, and S. C. Zhang, “Stock Market Prediction Using Artificial Neural Networks,” Advanced Engineering Forum, vol. 6–7, pp. 1055–1060, Sep. 2012.

2. M. Jia, J. Huang, L. Pang, and Q. Zhao, “Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network,” in Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), pp.467-473, 2019.

3. J. Eapen, D. Bein, and A. Verma, “Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction,” in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0264-0270, 2019

4. Gao P., Zhang R., Yang X. The application of stock index price prediction with neural network Mathematical and Computational Applications, 25 (3) (2020), p. 53

5. Dara Rajesh Babu, Bachala Sathyanarayana “Design and Implementation of Technical Analysis Based LSTM Model for Stock Price Prediction” International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), Volume: 11, pp. 1143-1149, March 2023.

6. Jaiwin Shah, Rishabh Jain, etl. “Stock Market Prediction using Bi-Directional LSTM” in 2021 International Conference on Communication information and Computing Technology (ICCICT), pp. 581-586.

7. Hum Nath Bhandari, Binod Rimal, etl. “Predicting stock market index using LSTM” Machine Learning with Applications, volume 9, September 2022.

8. Archit Kapoor, Varun Rastogi, etl, “Forecasting Daily Close Prices of Stock Indices using LSTM”, 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp 978-982.

9. Ayan Maiti, Pushparaj Shetty D. “Indian Stock Market Prediction using Deep Learning”, IEEE REGION 10 CONFERENCE (TENCON) Osaka, Japan, November 16-19, 2020

10. Srinath Ravikumar, Prasad Saraf "Prediction of Stock Prices using Machine Learning (Regression,Classification) Algorithms”, 2020 International Conference for Emerging Technology (INCET) Belgaum, India. Jun 5-7, 2020.