Comparative Study on Stock Market Prediction using Generic CNN-LSTM and Ensemble Learning

[1]Sanjay Kumar Raipitam
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar,India
sanjaykumar.raipitam@gmail.com

[2]Shekhar Kumar
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar,India
shekharseth.2609@gmail.com

[3]TusharDhanani
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar,India
tushardhanani002@gmail.com

[4]Saurabh Bilgaiyan
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar,India
saurabhbilgaiyan01@gmail.com

[5]Mahendra Kumar Gourisaria
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar,India
mkgourisaria2010@gmail.com

2023 International Conference on Network, Multimedia and Information Technology (NMITCON) | 979-8-3503-0082-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/NMITCON58196.2023.10275849

***Abstract*—The stock market is the platform where anyone can buy and sell or trade shares of public companies, and for that predicting the stock price helps us to forecast the future value of the company shares, derivatives, and mutual funds. The stock market is a composite and volatile system, and many factors can affect its performance. To evaluate a company's financial stability and performance, fundamental analysis is used. On the other hand, for reviewing historical price and bulk data, technical analysis has been carried out to recognize tendencies and patterns. Risk management, while Investing in the stock market carries inherent risks, and to mitigate those risks, it is crucial to spread out investments and establish stop-market orders, and other techniques. The aim of this paper is to suggest deep learning techniques in order to predict the stock prices of different companies such as AAPL(Apple), BAM(Brookfield Asset Management), and UBER and using two different models such CNN(Convolutional Neural Network) in CNN the paper uses One -Dimensional CNN(1D CNN) and LSTM(Long Short-Term Memory) uses Bidirectional LSTM(BLSTM). It implements the model on the static Apple dataset without an ensemble. While in the case of BAM, the ensemble model is on the static dataset. And, in the case of UBER, dynamic dataset for which we have fetched the dataset from Yahoo Finance(yfinance).**
Keywords— Deep learning, CNN, LSTM, ensemble, stock

I. Introduction

There has been significant research on predicting future stock prices, although supporters of the efficient market hypothesis argue that it is not feasible to predict them. Stock market prediction involves using various analytical techniques to forecast the future price movements of stocks, and it is a complex and challenging task. No can accurately predict the future movement of the stock market. It is necessary to understand that the stock market is a complex and volatile system, and many factors can affect its performance. Therefore, no one can accurately forecast the future stock market movement with complete accuracy [1] . Fundamental analysis involves analyzing the financial health and performance of a company, such as its earnings, revenue, assets, liabilities, and management [2] . It is possible to forecast future stock prices using this knowledge. In technical analysis, trends and patterns that may be predictive of future price movements are found by examining historical prices and bulk data [3]. Researchers have created complex yet sophisticated algorithms that can analyze enormous bulks of data to find patterns and forecast future prices, thanks to the advancement of machine learning and artificial intelligence. The literature for forecasting stock prices offers various time series decomposition methods, and technical analysis of stock price fluctuations has also been investigated. In this research, a detailed method has been offered to estimate stock price and price movement patterns using technical analysis, combining various statistical, machine learning, and deep learning methodologies. Using past stock price data, our goal is to create a reliable forecasting framework for stock price movement. The study further comprises an extensive evaluation of existing research on the prediction and estimation of stock price fluctuations, a discourse on the fundamental concepts, and an outline of the deep learning models employed in the analysis. Moreover, a comprehensive examination of the deep learning models' efficacy is presented, along with a contrast of their outcomes. The CNN-LSTM model utilized in this research utilizes CNN for isolating essential characteristics and LSTM for prognosticating the stock's closing price on the subsequent day [3] . This research contributes to the real world by suggesting deep learning techniques, such as 1D CNN and LSTM models, for stock market prediction. It improves forecasting accuracy, assists investors in making informed decisions, enhances risk management strategies, and promotes effective financial resource allocation and investment strategies. The main goal that the paper is driven by is to create a reliable and accurate predictive model that can help investors make wise judgements. It intends to utilize the temporal and spatial correlations within the data, capturing both short-term fluctuations and long-term patterns by utilizing the capabilities of CNN and LSTM.

II. Related Work

Currently, using conventional econometric approaches, it can be challenging to assess the financial market today because it is a dynamic and complicated system. The two main categories of stock price forecasting approaches are traditional analysis method and machine learning methods. Neural networks have gained popularity as a research topic for stock forecasting in recent years. This is due to the fact that they are able to independently extract data features from an enormous amount of high-frequency raw data. White employed neural networks to forecast IBM shares in 1988. However, the outcomes weren't promising [4] . Zhang forecasted stocks in 2003 using neural networks and ARIMA which stands for Autoregressive Integrated Moving Average models. Although neural networks have clear advantages in nonlinear data forecasting, experimental findings indicated that their accuracy still needs to be increased [5] . A time series forecasting technique based on a fusion of the optimum partition algorithm (OPA) and a radial basis function (RBF) neural network was put out by Sun et al. in 2005 [6]. In order to predict four financial time series data, Adhikari et al. devised a method in 2014 that combines Random Walk (RW) and Artificial Neural Network (ANN). The findings indicated that there had been some improvement in forecasting accuracy [7] . In 2018, Zhang and his team suggested a network topology based on the LM-BP neural network for stock price forecasting. This was an improvement over the conventional BP neural network training algorithm's low precision and slow training speed [8]. CNNs can be used to solve the problem of time series prediction, according to experimental findings from the same year by HU and his team. Because CNN is more frequently used for feature extraction and image recognition, it had relatively low forecasting accuracy [9] . Kamalov forecasted the stock prices of four significant US public firms for the year 2019 using MLP, CNN, and LSTM. According to experimental findings, these three styles outperformed comparable studies that read the direction of price change [10]. In 2020, Xue and his team used the LSTM deep neural network to develop a short-term financial market forecasting model. The BP neural network, a conventional RNN, and an enhanced LSTM deep neural network were all put up against each other in this comparison. The outcomes depicted that the LSTM deep neural network was capable of accurately forecasting time series for the stock market and had the highest forecasting accuracy [11].

This paper aims to emphasize on the following objectives:

- By examining time series data and the correlation between stock prices, a CNN-LSTM based deep learning technique has been developed to forecast stock prices. This technique employs LSTM for data forecasting and CNN to extract time features from the data.
- By comparing the R^2 scores and other accuracy metrics of the 1D CNN and BLSTM as individual units and both the models in an ensemble, the paper suggests which setup provides the better results.

III. BASIC CONCEPT

A. Convolutional Neural Network

In 1998 Lecun et al. proposed a kind of deep neural network known as Convolutional Neural Networks(CNNs) [12] .It is effective in forecasting of time series applications. CNNs typically have a sequence of convolutional layers, then pooling layers, and finally fully linked layers. The input image is subjected to filters (kernels) by convolutional layers, which generate feature maps emphasizing certain characteristics of the image. The feature maps are then downsampled by the pooling layers, which reduces their size while keeping the crucial data [13] . The output of the final pooling layer is then fed into a typical neural network architecture for classification by the fully connected layers.CNN has 2 kinds of layers- the convolution and the pooling layer. Feature extraction on the data is done by the convolution layer of the 1D-CNN. To tackle the issue of increased dimensionality and reduction in cost of intensive training, the pooling layer is used, represented by (1):

$$L_t = tanh(x_t * k_t + b_t) \quad (1)$$

where, L_t = the output value after convolution, tanh = the activation function, x_t = the input vector, k_t = the weight of the convolution kernel, b_t = the bias of the convolution kernel.

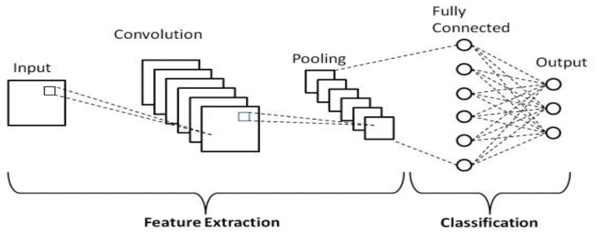

Fig. 1. Convolutional Neural Networks(CNNs) Architecture. [14]

B. Long Short-Term Memory

The traditional RNNs have some limitations in handling the long-term dependencies in sequences. To overcome this, Hochreiter and Schmidhuber introduced Long Short-Term Memory (LSTMs), in 1997 [15] . Depending upon the input and the internal state the LSTMs selectively forget or remember information that is store into their memory cell. The gates control the memory cell, which are composed of sigmoid neural network layers that determine how much of the input should be remembered, how much should be forgotten, and how much should be outputted. These gates provide LSTMs the ability to selectively recall or forget information depending on how relevant the past events are to the current task.In a standard LSTM, the information flows only in the forward direction, i.e., from the past to the future. But in a BLSTM, the input sequence is processed in both directions simultaneously. This enables the model to capture not only the past information but also the future information of the input sequence. The LSTM architecture is made up of memory cells, which consists of 3 parts, namely, input layer, output layer and forget layer. The calculation of LSTM model follows the following steps:

- The output at time ‘t-1’ and input at time ‘t’ are the inputs to the forget gate. The forget gate follows (2) to produce its output:

$$F_t = \sigma (W_F \times [h_{t-1}, x_t] + b_F \quad (2)$$

where,W_F =forget gate weight, b_F =forget gate bias, x_t=Input at present time ‘t’, h_{t-1} = Output at time ‘t-1’.The range of F_t falls under 0 to 1 exclusive.

The output at time ‘t-1’ and the input at time ‘t’ acts as the input for the input gate which is used to calculate the output value and candidate cell state for the gate using (3) and (4)

$$I_t = \sigma(W_I \,.\, [h_{t-1}, x_t] + b_I) \quad (3)$$

$$\breve{C}_t = tanh\,(W_C \,.\, [h_{t-1}, x_t] + b_C) \quad (4)$$

where, W_I = Input gate weight, b_I = Input gate bias, W_C = Candidate input gate weight, b_C = Candidate input gate bias. The value of I_t ranges between 0 to 1 exclusive.

- The current cell state is updated according to (5):

$$C_t = F_t \times C_{t-1} + I_t \times \breve{C}_t \quad (5)$$

- The output gate receives the output h_{t-1} and x_t as the inputs at the time ‘t’ and the output follows (6):

$$O_t = \sigma\,(W_O \times [h_{t-1}, x_t] + b_O) \quad (6)$$

where, W_O = Output gate weight, b_O = Output gate bias
The values of O_t ranges between 0 to 1 exclusive.

- The LSTM’s final output value is obtained using (7):

$$h_t = {}_{Ot} \times tanh\,(C_t) \quad (7)$$

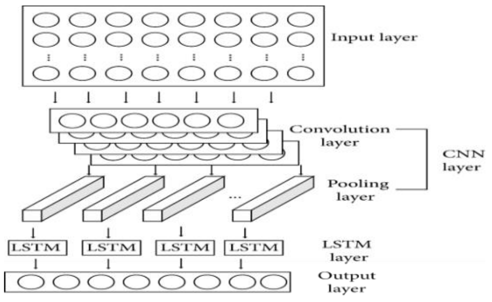

Fig 2. LSTM Memory Cell Architecture [3]

C. Data

In this paper, the different datasets that have been taken are AAPL, BAM and UBER. AAPL and BAM are static datasets, while for UBER dynamic data has been fetched from Yahoo Finance [16]. Table-1 shows some of the data, which is the trading data of UBER of 5 days starting from 24th April,2023 to 28th April,2023, starting from when the market opens that is 09:30 am to 15:59 pm. All the data is of 5 days within an interval of 1m. The first 1476 data is taken to be the training dataset and the last 369 data is taken as the test dataset. Table-2 displays trading data for BAM, covering the period from December 30, 1983, to December 12, 2022, with daily intervals. The first 7773 trading days are used for training, while the last 1944 trading days are reserved for testing. The BAM dataset is sourced from Kaggle, similar to the AAPL dataset [17]. Both datasets lack a time column and consist of static historical data recorded on a daily basis.

TABLE 1. PARTIAL SAMPLE DATA OF UBER FETCHED FROM YAHOO FINANCE AND PRE-PROCESSED

	Open	High	Low	Close	Adj. Close	Volume	Date
0	30.680000	30.719999	30.600100	30.635000	30.635000	328570.0	2023-04-24
1	30.770000	30.799999	30.730000	30.790001	30.790001	77499.0	2023-04-24

TABLE II PARTIAL SAMPLE DATA OF BAM TAKEN FROM KAGGLE AND PRE-PROCESSED.

	Date	Low	Open	Volume	High	Close	Adj-Close
0	30-12-1983	1.149471	1.154125	69836	1.154125	1.149471	0.2992
1	04-01-1984	1.154125	1.154125	53720	1.158779	1.158779	0.3016

IV. PROPOSED WORK

The paper proposes the use of 1D CNN because by modifying CNN to 1D CNN, the model's capacity to extract significant features and produce precise predictions is enhanced by the convolutional operations along the temporal axis, which allow the model to capture local patterns and relationships in the sequential data. To capture local patterns or features in the time series, filters (sometimes referred to as kernels) are applied along the time axis in place of the 2D filters often employed in image processing. Convolutions are performed by the filters as they move through the sequence to extract pertinent features. After the convolutional layers, activation and pooling layers are active in which in order to introduce non-linearity, activation functions like ReLU are used after the convolutional layers. The feature maps can be down sampled along the time axis while still preserving crucial features by using pooling layers, such MaxPooling [18] . After that fully connected layers can be added to the architecture followed by convolutional and pooling layers. The model may learn higher-level representations of the data due to these layers, which capture complicated correlations between the retrieved features. Then the target variable, which can be the future stock price, is predicted by the 1D CNN's final layer. Mathematically, the 1D convolutional operation at time t is represented in (8):

$$h_t = ReLU(\textstyle\sum_{i=0}^{K-1} W_i + X_{t+i} + b) \quad (8)$$

where, h_t =Output feature map at time ‘t’, K= filter size, W_i =filter parameter, X_{t+i} =Input data, b =Bias. On the basis of labelled data, the model is trained, and optimization methods like gradient descent are used to update the model's parameters and reduce prediction error.The paper proposes the use of BLSTM because it gives the model the ability to use context from the past as well as the future, leading to a more thorough comprehension of the input sequence dynamics. In cases where capturing long-term dependencies and utilizing future information are essential for making correct predictions in the stock market, this can result in increased prediction accuracy. It maintains two separate hidden states, one for processing the sequence from the start to the end, and the other for processing the sequence in reverse. This gives the model a deeper comprehension of the temporal dynamics in the data and may help with prediction accuracy. After that concatenation of outputs was done at each time step in BLSTM, the outputs from the forward and backward LSTM layers are concatenated. This combined representation offers a more thorough picture of the input sequence that incorporates information from both the past and the future then enhanced modelling of long-term dependencies were done. Dependencies that develop forward can be captured by the forward LSTM layer, whereas dependencies that develop backward can be captured by the backward LSTM layer. BLSTM can more effectively model complex patterns and interactions that span a wider time horizon by integrating these two orientations. And finally training and optimization occurred similar to LSTM, BLSTM is trained using labelled historical data via gradient descent optimization and backpropagation. To reduce the prediction error, the BLSTM units' biases and weights are changed during training. Mathematically, the forward and backward LSTM operations at time 't' as (9) and (10):

$$h_t^{(f)} = LSTM_{forward}(h_{t-1}^{(f)}, X_t; W_f, b_f) \tag{9}$$

$$h_t^{(b)} = LSTM_{backward}(h_{t+1}^{(b)}, X_t; W_b, b_b) \tag{10}$$

where, $h_t^{(f)}$, $h_t^{(b)}$=hidden states of the forward and backward LSTMs at time 't', W_f,W_b=weight matrices, b_f,b_b=bias vectors. From the above discussion, the paper introduces an ensemble model combining 1-D CNN and LSTM.

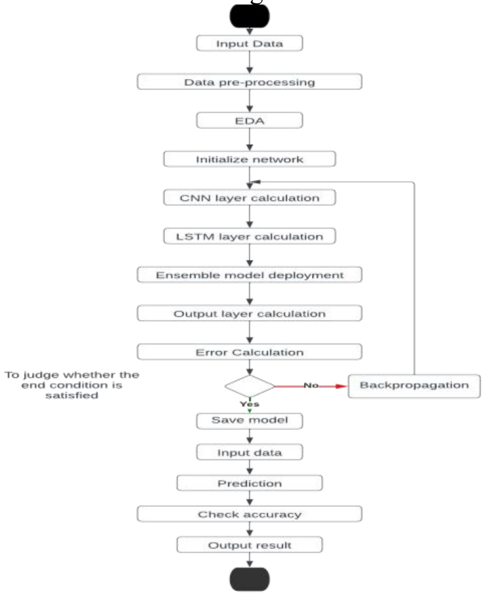

Fig. 3. Activity Flowchart of the CNN-LSTM model implementation.

V. Model Implementation

A. Implementation of CNN-LSTM

The paper deals with time series data and the problem under consideration is a regression analysis problem. The input data is assumed to be a 3D tensor. This 3D tensor is defined using 3 parameters namely, batch size (the number of samples), timesteps (the number of time steps in each individual sample) and Num features (the number of features in every time step). The CNN layers are used for feature extraction from the time series stock data. The LSTM layers function in order to collect and investigate the temporal trends in the data. The function of Time Distributes layer, can be explained as, it is used for the application of the same Convolutional operation to each time step independently. MaxPooling1D is applied after each convolutional layer to reduce the dimensionality of the output. After the CNN layers, the Bidirectional LSTM layers are used to capture the forward and backward temporal dependencies in the data. Dropout is applied to avoid overfitting, and for the regression analysis, a single output is produced by the final Denser layer with linear activation function.The working model uses the Mean Squared Error as the loss function and the Adam optimizer for compilation. While training the data, it tries to reduce the MSE between the observed and predicted prices. The model gets trained for 40 epochs where batch_size is set to 40 and evaluation is done on a separate validation dataset.

Convolution layer filters:	64, 128, 64
Convolution layer kernel_size:	3
Convolution layer activation function:	'relu'
Convolution layer padding:	'valid'
Pooling layer pool_size:	2
Pooling layer padding:	'valid'
Pooling layer activation function:	no activation function used.
Number of hidden units in LSTM layer:	100
LSTM layer activation function:	'tanh'
Time_step:	depends on the input data.
Batch_size:	40
Learning rate:	0.001
Optimizer:	'adam'
Loss function:	mean squared error.
Epochs:	40

B. Implementation of Ensemble Model

In this paper, a function is first created to implement a single instance of the model architecture. The ensemble model architecture framework comprises the CNN and LSTM layers. After the data is passed through the above layers in order to produce a final single continuous value, a final dense layer is employed which uses a "tanh" linear activation function in this project.Then, a list of num_models instances of the ensemble is created. Each instance is trained separately on the same training data, with same hyperparameters. After this, the fit method is used for each instance for training, using the training data, in order to predict by calling the method for each instance, for predicting the results. The predictions generated by each model instance are combined using the average of the predicted values across all the models. This ensemble prediction is used as the final output of the model.

C. Evaluation Metrics

In order to comprehend the accuracy of the forecasts provided by the model used, we have 3 metrics, namely: MAE, MSE and R^2 score.

- The MAE (mean absolute error) is calculated according to (11):

$$Mean\ Absolute\ Error = \frac{1}{N}\sum_{i=1}^{N} |Y_{true} - \widehat{Y}_{predicted}| \tag{11}$$

- The MSE (mean squared error) is calculated according to (12):

$$Mean\ Squared\ Error = \frac{1}{N}\sum_{i=1}^{N} (Y_{true} - \widehat{Y}_{predicted})^2 \tag{12}$$

- The R^2 score is calculated according to (13):

$$R^2 = 1 - \frac{(\sum_{i=1}^{N}(Y_{true} - \widehat{Y}_{predicted})^2)/N}{(\sum_{i=1}^{N}(Y_{avg} - \widehat{Y}_{predicted})^2)/N} \tag{13}$$

The closer the value of R^2 score to 1, the better the model.

VI. RESULTS

In terms of forecasting accuracy, firstly talking about the static historical stock data, from Fig.10 the maximum residual error is found to be 0.615561, from Fig. 9 the R^2 score is 0.950553 as calculated from equation (10), from Fig. 11 the mean absolute error is found 0.05690 from equation (8) and from Fig. 12 the mean squared error can be seen as 0.00576 from equation (9), without the ensemble learning whereas with ensemble learning, from Fig. 9 the R^2 score is found to be 0.94176, from Fig. 10 the maximum residual error is reduced to 0.23614, from Fig.11 mean absolute error is found to be 0.03176 and from Fig. 12 the mean squared error is 0.00176. Talking about the dynamic dataset, again from Fig.10 the R^2 score is 0.940492, from Fig. 9 the maximum residual error is 0.010727, mean absolute error is 0.002011 as shown in Fig.11 and from Fig. 12 the mean squared error is 7.5072e-06 without ensemble learning models whereas for ensemble learning, the R^2 score is 0.923374 as can be seen from Fig.9, from Fig.10 the maximum residual error is 0.011855, from Fig. 11 the mean absolute error is found to be 0.002036 and from Fig. 12 the mean squared error is 7.8143e-06.The CNN-LSTM ensemble model outperforms traditional econometric methods, ARIMA, MLP, CNN, LSTM, and hybrid models used in previous studies. The model's R2 score for both static and dynamic data demonstrates superior predictive capability. The ensemble approach shows scope in enhancing accuracy, particularly in dynamic market conditions. The outcomes demonstrate how well the CNN-LSTM model performs in case of static data with ensemble and without ensemble and dynamic data with ensemble and without ensemble. Fig. 9–12 showcase the comparative analysis of the different methods and data. In terms of forecasting accuracy, in case of static data, the R^2 score, 0.950553, is 0.8% higher in comparison from with ensemble to that of without ensemble. In case of dynamic data, the R^2 score, 0.940492, is 1.71% higher in comparison from ensemble to that of without ensemble. Therefore, the CNN-LSTM model without the ensemble model works better in forecasting accuracy.

The research required a system with minimum specifications, including an Intel Core i7, 10th Generation Processor, 16GB DDR4 RAM, 512GB SSD storage, and a dedicated Nvidia graphics card.It also required Jupyter Notebook with Python 3.8, utilizing essential libraries like NumPy, pandas, scikit-learn, TensorFlow. Data visualization was achieved with matplotlib, seaborn.

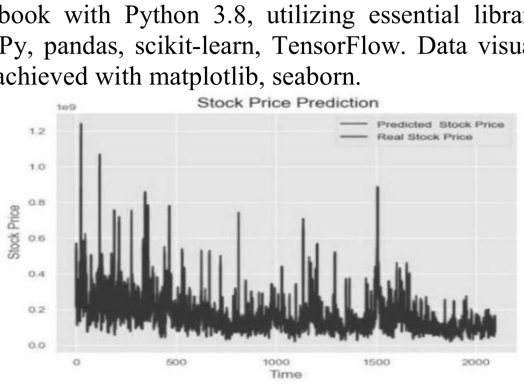

Fig. 4. Static stock data for Apple (without using ensemble model)

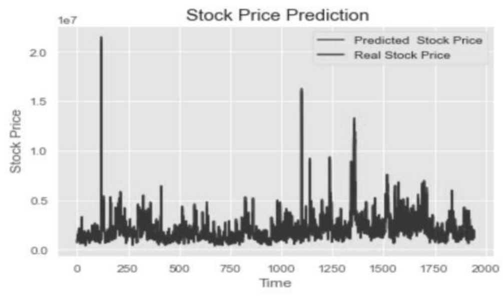

Fig. 5. Static stock data for BAM (without using ensemble model)

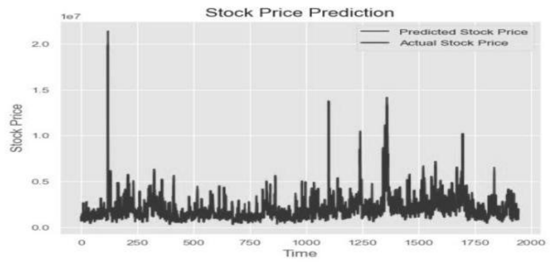

Fig. 6. Static stock dataset for BAM (using ensemble model)

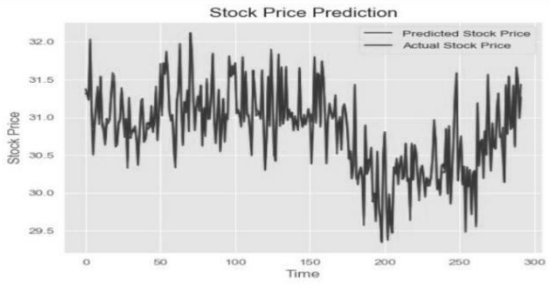

Fig. 7. Dynamic stock dataset for UBER (using ensemble model)

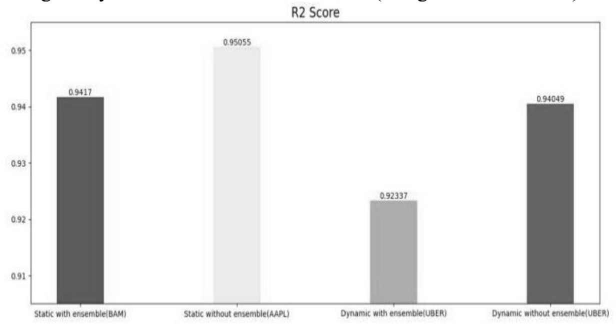

Fig. 8. Showcases the R^2 score on different methods and data.

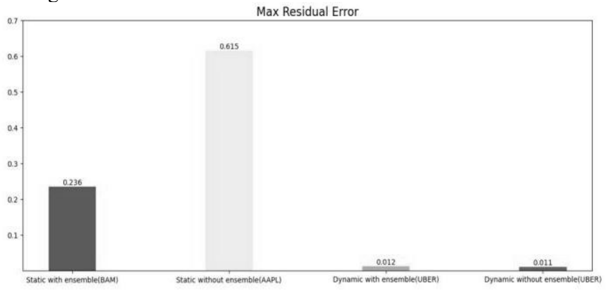

Fig. 9. Showcases the Max Residual Error on different methods and data.

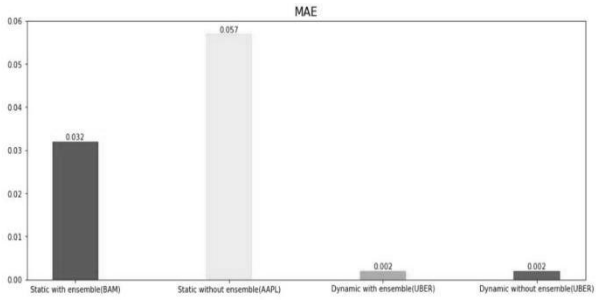

Fig. 10. Showcases the Mean Absolute Error(MAE) on different methods and data.

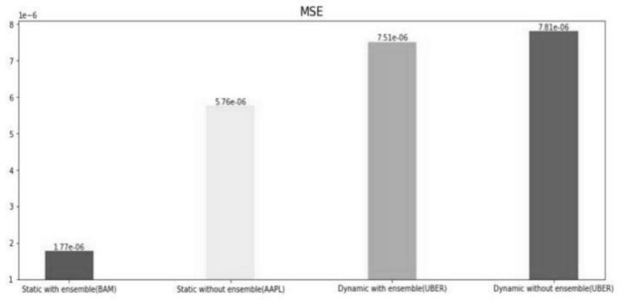

Fig. 11. Showcases the Mean Squared Error(MSE) on different methods and data.

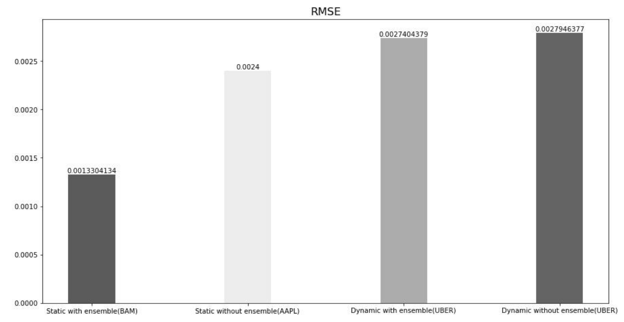

Fig. 12. Showcases the Root Mean Squared Error(RMSE) on different methods and data.

VII. Conclusion

The models used were trained on historical stock market data. It used a dynamic dataset to make the predictions more accurate. The results obtained were promising, with the models achieving a good level of accuracy in predicting stock prices.

By looking at the graphs shown in the Fig. 9-12 we see that in case of static data, the R^2 score, 0.950553, is 0.8% higher in comparison from with ensemble to that of without ensemble. In case of dynamic data, the R^2 score, 0.940492, is 1.71% higher as compared to ensemble to that of without ensemble. We can say for sure that the combination of 1D CNN and BLSTM can be used for as an ensemble training model in future for forecasting of future of stock price predictions as it works well with the dynamic systems as well. However, there is still room for improvement in terms of refining the models and enhancing their accuracy. In the future, this study might be expanded to explore the use of other deep learning models, like transformers, or to incorporate sentimental data from social media or news articles. The current research has several limitations that should be acknowledged. Firstly, data limitations, such as data quality and availability, may impact the model's accuracy and reliability. The model's performance may be specific to certain market conditions, making its generalization uncertain.Lastly, the lack of interpretability in deep learning models like CNN-LSTM poses challenges in understanding feature impact.

References

[1] Fama, E.F., "Efficient Capital Markets: A Review of Theory and Empirical Work," *The Journal of Finance,* vol. 25, no. 2, pp. 383-417, 1970.

[2] Ovidiu Popescu, "Technical Analysis of the Financial Markets," 10 03 2023. [Online]. Available: https://www.morpher.com/blog/technical-analysis.

[3] Lu, W., Li, J., Li, Y., Sun, A., Wang, J., "A CNN-LSTM-Based Model to Forecast Stock Prices," *Complexity,* vol. 2020, p. 10, 2020.

[4] Halbert White, "Economic prediction using neural networks: the case of IBM daily stock returns," *IEEE 1988 International Conference on Neural Networks,* vol. 2, pp. 451-458, 1988.

[5] G. Peter Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing,* vol. 50, p. 159–175, 2003.

[6] Sun, Y.F., Liang, Y.C., Zhang, W.L. et al, "Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting," *Neural Computing and Applications,* vol. 14, pp. 36-44, 2005.

[7] Adhikari, R., Agrawal, R.K., "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Computing and Applications,* vol. 24, pp. 1441-1449, 2014.

[8] Zhang, L., Wang, F., Xu, B. et al., "Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI," *Neural Computing and Applications,* vol. 30, p. 425–1444, 2018.

[9] Hu, Y., "Stock market timing model based on convolutional neural network–a case study of Shanghai composite index," *Finance& Economy,* vol. 4, p. 71–74, 2018.

[10] Alibasic, E., Fazo, B., i Petrovic, I., "A new approach to calculating electrical energy losses on power lines with a new improved three-mode method," *Tehnicki Vjesnik,* vol. 26, p. 405–411, 2019.

[11] Yan, X., Weihan, W. & Chang, M., "Research on financial assets transaction prediction model based on LSTM neural network," *Neural Computing and Applications,* vol. 33, pp. 257-270, 2021.

[12] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[13] L. Qin, N. Yu i D. Zhao, "Applying the Convolutional Neural Network Deep Learning Technology to Behavioural Recognition in Intelligent Video," *Tehnicki Vjesnik - Technical Gazette,* vol. 25, no. 2, pp. 528-535, 2018.

[14] "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," *Applied Sciences,* vol. 9, p. 4500, 2019.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[16] "PyPI," [Online]. Available: https://pypi.org/project/yfinance/.

[17] "Kaggle Stock Market Data," [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/stock-market-data.

[18] Shen, J., Shafiq, M.O., "Short-term stock market price trend prediction using a comprehensive deep learning system," *Journal of Big Data,* vol. 7, no. 66, 2020.