International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

STOCK PRICE PREDICTION: EVALUATING THE EFFICACY OF CNN, LSTM, CNN-LSTM, AND CNN-BILSTM MODELS

Ebiesuwa Seun[1] and Adebanjo Olawunmi Asake[2]

[1] Department of Computer science, Babcock university, Ilishan remo,Ogun state,Nigeria.
[2]Department of Software Engineering, Babcock university, Ilishan remo,Ogun state,Nigeria.

E-mail address: adebanjoo@babcock.edu.ng, ebiesuwao@babcock.edu.ng.

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: The stock market's dynamism and complexity make predicting accurate prices a daunting task for investors and analysts. Traditional statistical models struggle with this due to hidden non-linear relationships and time-dependent patterns in financial data. This sparks a rising interest in harnessing the power of machine learning, particularly neural networks, for improved stock price forecasting.This study uses four neural network models - CNN, LSTM, CNN-LSTM, and CNN-BILSTM to predict stock prices. Their performance is evaluated through four metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2), and Mean Absolute Percentage Error (MAPE). The US stock price dataset from 1998-2021 was used, the dataset was obtained from Kaggle and was preprocessed by normalizing and scaling. Python was used to train the models, the study then compares the hybrid models (CNN-LSTM and CNN-BILSTM) to their standalone counterparts, aiming to reveal their potential superiority in terms of prediction accuracy and error minimization.Analysis revealed that the hybrid models, particularly CNN-LSTM with its attention mechanism, outperformed their standalone counterparts in predicting stock prices and minimizing errors. CNN-BiLSTM followed closely, demonstrating strong performance as well. While CNN exhibited the lowest RMSE and MAE, its high MAPE suggests limited predictive power. This may be due to CNN's focus on feature extraction rather than temporal dependencies, highlighting the effectiveness of hybrid models in capturing complex market dynamics.

Keywords: Stockprice, CNN, LSTM, BiLSTM

1. INTRODUCTION

Understanding the stock market's movements is crucial since they reflect the economic strength and financial health of a nation. The market's behavior is influenced by a myriad of factors, from international trade dynamics to domestic economic performance, and from global events to government financial announcements and central bank policy shifts. It is a complex and volatile realm, where investments carry inherent unpredictability.

Traditionally, experts have used two main approaches to navigate this uncertainty and forecast future stock trends: Technical and fundamental analysis. Fundamental analysis delves deep into the financials of a company, market position, and economic indicators to predict its future performance. It builds a case for the investment based on how solid a company's business is. On the other hand, technical analysis looks at historical price patterns and market activity to forecast future movements. While insightful, the two methods fall short in capturing the full complexity and unpredictability of the stock market.

Machine learning steps up to address these limitations, using models like Linear Regression, Support Vector machines, and ARIMA. Each of these has had varying success, often limited by their inability to fully capture the random and complex patterns of stock prices which are influenced by numerous known and unknown variables.

Deep learning, and specifically Long Short-Term Memory (LSTM) networks, offers a breakthrough by accounting for long-term trends and dependencies in stock price data. LSTMs are adept at understanding the

E-mail:author's email

temporal sequences within the data, an essential aspect considering the time series nature of stock prices.

Conversely, Convolutional Neural Networks (CNNs), though traditionally associated with spatial data recognition in fields such as image processing, have shown promise in identifying intricate patterns within time-series data. Their ability to extract features from sequences makes them a valuable addition to stock price prediction models.

Recognizing the potential in these technologies, this research proposes a Hybrid LSTM-CNN model. The model aims to synergize LSTM's temporal data analysis capabilities with the pattern recognition strengths of CNNs, to provide a sophisticated tool for stock price forecasting. This hybrid approach seeks to enhance accuracy and reliability in predictions, offering valuable insights for investors, portfolio managers, and financial analysts. The ensuing study will explore this integrative model, with the intent to push the boundaries of current forecasting methodologies and offer a potent solution in the complex domain of stock market investments.

2. RELATED WORKS

Accurately forecasting stock prices holds substantial economic advantages, particularly for investors, portfolio managers, and policymakers. Over time, various models and methodologies have been created and applied to enhance predictive precision, progressing from basic statistical models to advanced machine learning algorithms. One instance of these models is the Autoregressive Integrated Moving Average (ARIMA) models [1], though adept at handling linear relationships, fall short in capturing the market's inherent nonlinearities, limiting their effectiveness in stock price forecasting.

Exploiting the potential of neural networks is now a leading focus in stock market forecasting research. This is due to their ability to identify important data characteristics from vast quantities of raw, high-frequency information without the need for pre-existing knowledge [2]. [3] introduced a novel approach that melded random walk (RW) with artificial neural networks (ANN) to forecast four financial time series datasets. Their findings indicated a noticeable enhancement in forecasting accuracy. [4] proposed a network architecture for stock price prediction based on the LM-BP neural network. This innovation addressed the shortcomings of the traditional BP neural network, particularly its slow training speed and low precision, leading to improved forecasting outcomes.

Recurrent Neural Network (RNN) is an advancement to neural networks equipped with internal memory, enabling it to make predictions by leveraging historical data features. This capability renders RNNs particularly adept for applications in Stock market forecasting. LSTM stands out as one of the most prominent variants of RNNs [5]. [6] developed an LSTM-based technique for gleaning insights and forecasting stock trends on Shanghai A-share financial markets, LSTM was seen to perform better than the other model with accuracy rate of 57%, this accuracy could be improved upon. [7] in their study fed technical indicators to an LSTM network to forecast the direction of stocks in the Brazilian stock market, demonstrating that LSTM outperformed the Multilayer Perceptron (MLP) with an accuracy of 55.9% with a high variance.

CNN gained widespread popularity in the domain of image recognition due to its remarkable ability to recognize complex patterns. This capability prompted its application in economic forecasting as well. Like traditional neural networks, CNN consists of multiple interconnected neurons organized hierarchically, with trainable weights and biases between layers [8]. [9] used convolutional neural networks (CNN) in time series prediction. They emphasized that deep learning, particularly CNN, was well-suited for addressing time series challenges. However, it was noted that using CNN in isolation led to relatively lower forecasting accuracy, likely due to its common application in image recognition and feature extraction.

[10] employed a combination of CNN, MLP and LSTM to forecast the stock prices of four U.S. public companies. The findings indicated that the three models surpassed comparable research in predicting price direction. However, it was noted that while LSTM is a computationally intensive algorithm requiring extended training periods, the MLP (Multilayer Perceptron) offers a more time-efficient alternative. Despite its faster processing, MLP still delivers competitive results akin to those achieved by LSTM. It was advised to weigh the balance between speed and accuracy when selecting the optimal forecasting model. Although LSTM was highly recommended for its accuracy, the consideration of its slower processing speed compared to other models is crucial in making an informed decision.

[11] also showed a very accurate short-term forecasting model for financial market time series using the LSTM deep neural network. Comparing this approach with traditional methods like traditional RNNs, BP neural networks and an improved LSTM deep neural network, they found that the LSTM deep neural network achieved superior forecasting accuracy and effectively predicted

stock market time series. Additionally, they noted that while LSTMs solve the vanishing gradient problem common in traditional RNNs, their research also pointed to room for further improvement in prediction accuracy. Notably, the inherent noise associated with stock market time series data was identified as a potential factor affecting the accuracy of LSTM predictions.

[2] compares the effectiveness of various neural network models including MLP (Multilayer Perceptron), CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), CNN-RNN, and CNN-LSTM, it showed that among the six forecasting models, CNN-LSTM showed the best accuracy, with the predicted values almost coinciding with the real values.

[12] implemented a hybrid model combining Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and an Attention mechanism for stock price forecasting. The evaluation process involved comparing the CNN-BiLSTM-Attention model with other models. Their results showed that the CNN-BiLSTM-Attention model outperformed the other models in forecasting the closing prices of these indices. For instance, the RMSE value of the CNN-BiLSTM-Attention model was 7.13% lower than that of the CNN-LSTM model.

A. *Convolutional Neural Networks (CNNa)*

Convolutional Neural Network, commonly known as a CNN, is a type of feedforward neural network that has gained prominence for its effectiveness across various domains, including image and natural language processing. Its application in time series forecasting is equally noteworthy, as it aptly captures temporal dependencies and patterns within sequential data [13].

The CNN architecture employs a clever design to reduce the model's complexity and prevent overfitting. This is achieved through local receptive fields, shared weights, and pooling layers, which collectively enhance the efficiency of the learning process [14].

At the heart of the CNN are its convolution layers, where multiple convolution kernels are applied to the input data. The convolution operation, which is pivotal for feature extraction, is mathematically formulated as follows:

$$Ocnn(t) = \sum_{i=0}^{k-1} \mathrm{I(t+i).F(i)} \qquad (1)$$

Ocnn(t) is the output feature map at time step t, I(t+i) represents the stock market data at time t+i, F(i) are the filter's weights, and k is the kernel size. Post convolution, an activation function such as the Rectified Linear Unit (ReLU) is applied to introduce non-linearity, enabling the model to learn complex patterns:

$$\mathrm{ReLU}(x)=\max(0,x) \qquad (2)$$

-

Following the convolution and activation, the pooling layer, typically max pooling, is employed to reduce the dimensionality of the feature maps. This operation simplifies the network by downsampling the output of the convolutional layers, retaining only the most significant features:

$$P(t) = max_{i=t}^{t+p-1} Ocnn(i) \qquad (3)$$

where P(t) is the pooled output at time step t, and p is the pooling size.

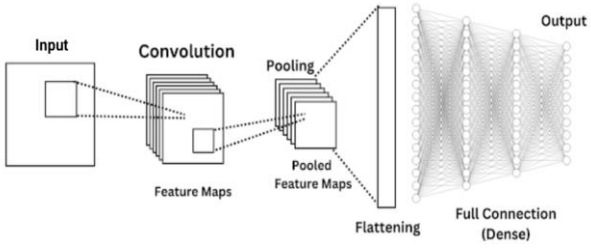


Fig 1 : ***CNN Architecture***

B. *LONG SHORT-TERM MEMORY (LSTM)*

Long Short-Term Memory (LSTM) networks, a specialized sub-branch of Recurrent Neural Networks (RNNs) first proposed by Hochreiter and Schmidhuber in 1997, address the long-term dependency issue inherent in RNNs by incorporating internal memory gates. LSTMs have proven effective in various sequential data applications, including speech recognition, language translation, and time series forecasting.

The primary innovation of LSTMs is its ability to address the vanishing gradient issue common in traditional RNNs. This challenge arises during the backpropagation process, where calculated gradients are passed backward through the network. In deep networks or with lengthy sequences, these gradients can diminish to negligible levels, hindering the

network's capacity to learn from extended data dependencies.

LSTMs tackle the vanishing gradient problem with a distinctive structure featuring memory cells and a series of gates that regulate information flow. These gates, specifically the input, forget, and output gates, enable LSTMs to selectively retain or discard information over extended periods. This capability is particularly beneficial for time series modeling, where significant intervals may separate relevant data points.

The internal mechanisms of an LSTM cell are governed by a set of equations that orchestrate these gates' functions, ensuring the effective management of information throughout the learning process.

- **Forget Gate:** Chooses what information is discarded from the cell state. It looks at the previous hidden state ht−1 and the current input $\mathcal{X}$t and applies a sigmoid function σ to decide which values in the cell state Ct−1 should be allowed through**.**

$$ft = \sigma(Wf \cdot [h_{t-1}, \mathcal{X}t] + bf) \quad (4)$$

- **Input Gate:** The input gate chooses which new information is added to the cell state. It uses a sigmoid function to determine which values to update, and a tanh function is use to create a vector of a new candidate values C˜t that could be added to the state.

$$it = \sigma(Wf \cdot [h_{t-1}, X_t] + bi) \quad (5)$$

$$\tilde{C}t = tanh(WC \cdot [h_{t-1}, \mathcal{X}t] + bC \quad (6)$$

Cell State Update: The cell state is updated by multiplying the old state by *f*t, the information that is no longer needed is discarded, and i_t* C˜t is added which are the new candidate values scaled by how much we decided to update each state value.

$$\tilde{C}t = ft * C_{t-1} + i_t * \tilde{C}t \quad (7)$$

Output Gate: Finally, the output gate chooses what the next hidden state *ht* should be. The hidden state contains information about previous inputs. The sigmoid function decides which parts of the cell state make it to the output, and then the cell state is put through tanh (so the value is pushed between -1 and 1) and then multiplied by the sigmoid gate output, so that we only output only the selected parts.

$$Ot = \sigma(Wo \cdot [h_{t-1}, \mathcal{X}_t] + bo) \quad (8)$$

$$ht = O_t * \tanh(C_t) \quad (9)$$

The LSTM's proficiency in handling data with long range temporal makes it a particularly suitable choice for time series analysis, where understanding the context and history is crucial for accurate forecasting. By leveraging its sophisticated gating mechanisms, dependencies LSTMs can maintain a memory of past information, using it to inform predictions and adapt to new data trends over time.

C. *BIDIRECTIONAL LSTM (BiLSTM)*

The Bidirectional Long Short-Term Memory (BiLSTM) network enhances traditional LSTM framework by incorporating both future and past context in its analysis. This advanced architecture is achieved by integrating two LSTM layers: one processes the sequence in its original order (forward LSTM), and the other processes the sequence in reverse (backward LSTM). This dual-layer setup ensures that at each point in the sequence, the network has comprehensive insights from both preceding and subsequent data points.

BiLSTMs are particularly adept in scenarios where the understanding of an entire sequence, including both historical and upcoming data, is crucial for accurate predictions or interpretations. This attribute renders BiLSTMs exceptionally suitable for time series forecasting tasks. In such applications, they excel by identifying patterns and dependencies that might be overlooked if the analysis were confined to historical data alone. By considering both past and future, the bidirectional approach allows the model to make significantly accurate predictions.

In the BiLSTM the forward layer processes the input sequence in the usual manner, generating an output sequence $h\rightarrow$. This sequence is obtained by moving through the input data from the beginning to the end, capturing forward temporal dependencies. The two sequences capture temporal dependencies from both past and future. The outputs from both layers are then combined, using a sigmoid function (σ), to form a unified output vector y_t. The final output of the BiLSTM layer is a vector $Y_t = [y_{t-n},....,y_{t-1}]$ where y_{t-1} represents the combined prediction for the next time step, leveraging insights from the entire sequence.

3. MATERIAL AND METHODS

D. *Data Sources*

This analysis utilizes the "US Historical Stock Prices with Earnings Data" dataset sourced from Kaggle. This dataset includes twenty-three years (23) daily stock prices and

earnings data from 1998-2021, with each piece of data containing the company symbols, date, market opening price (open), the highest point the stock got to (high), the lowest point (low) , the volume, the earnings per share(eps),the earnings estimate (eps_est) and the company close price(close) all this takes on a crucial role in the time series forecasting model.

E. Data Preprocessing

- The data was checked for missing values and Outliers.
- Pertinent features such as 'Open', 'Low', 'High', 'Close', and 'Volume' were selected and computed additional technical indicators to enrich the dataset.
- To ensure every of the feature received equal weight in the model's predictions, Min-Max scaling was used to scale the input data.

F. Data Preparation

- Sequence Formation: The time series data was transformed into sequences to facilitate the learning of temporal dependencies by the LSTM. This transformation involves creating a series of overlapping time windows, where each window is used to predict the subsequent value.

Training And Testing Split: Crucial for evaluating model generalizability, the dataset was partitioned into training and testing sets. The training data spanned up to 2016, while the testing data was from 2017 onwards, ensuring the model's predictions were based on truly "unseen" examples.

G. CNN Feature Extraction

The CNN part of the model processes the input features to extract spatial patterns. For time series data like stock prices, it identifies patterns across different technical indicators or across several days of price movements.

Rectified Linear Unit (ReLU) was used for Activation function and Max pooling for Pooling operation.

H. Flowchart

The output from the CNN layers, which is the condensed version of the stock market data, is flattened and passed through a time-distributed layer to maintain the temporal sequence for the LSTM. The LSTM and BiLSTM layers model the temporal relationships in the sequence of features extracted by the CNN. The LSTM operations include the gates and state updates. The BiLSTM extends this by processing the data bidirectionally.

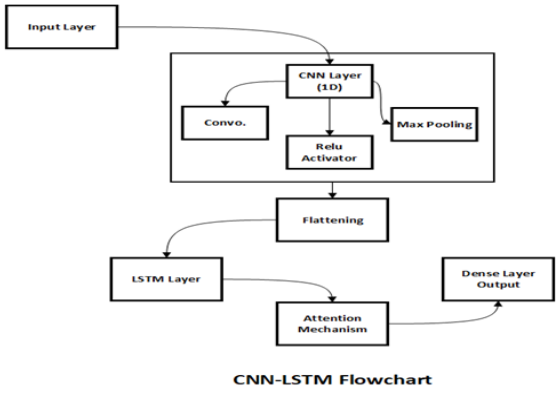


Fig 2 : ***CNN-LSTM Flowchart***

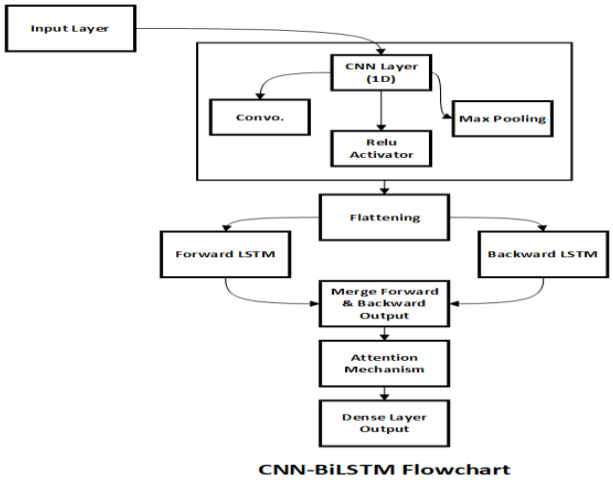


Fig 3 : ***CNN-BiLSTM Flowchart***

I. Attenton Mechanism and Hybrid Model output

Both models incorporate an attention mechanism after the LSTM/BiLSTM layers. This allows the model to focus selectively on certain parts of the input sequence, enhancing the model's ability to discern relevant information for making accurate predictions.

The final output of the hybrid models is a dense layer that takes the last output of the LSTM/BiLSTM sequence, refined by the attention mechanism, and produces the prediction for the next stock price or movement.

$$Yt = Wy \cdot ht + by \qquad (10)$$

Where Yt is the predicted stock price or movement at time t, ht is the last hidden state of the LSTM and Wy and by are the learned weights and biases of the output layer.

4. MODEL TRAINING AND RESULTS

The selected model was built using the Python programming language of version 3.11, Pytorch. The evaluation of the hybrid CNN-LSTM and CNN-BiLSTM models for stock price forecasting was conducted using four key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE),R squared (R^2) and Mean Absolute Percentage Error (MAPE). To assess accuracy and effectiveness of these hybrid models, their performance was benchmarked against the standalone CNN and LSTM models using the same metrics. This comparative analysis aimed to establish the superiority of the hybrid models in terms of how accurate its prediction and error minimization are when contrasted with the individual performances of the standalone models

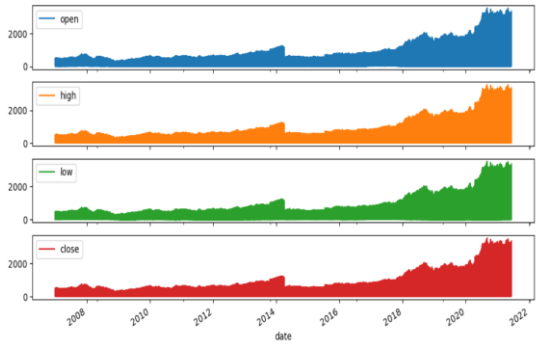
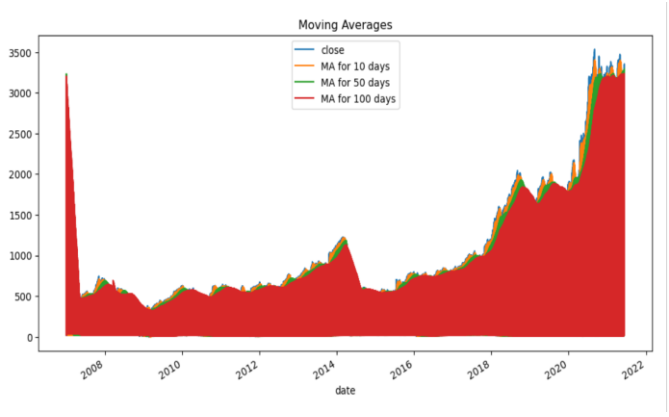


Fig 4 : ***Visual representation of the data***

The stock price data was preprocessed, and features like



Moving Average (MA10) and Relative Strength Index (RSI) were calculated. The data was then split into training and testing sets.

A. Model architecture

Both models used a combination of convolutional layers (for feature extraction) and LSTM or BiLSTM layers (for capturing temporal dependencies). An attention mechanism was also integrated to enhance the model's focus on relevant features.

B. Model Evaluation Metrics

Model performance was assessed using these key metrics:

- **MEAN ABSOLUTE ERROR (MAE):** MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's calculated as the average of the absolute differences between the predicted values and the actual values. The MAE is calculated using the formula:

$$MAE = (1/n)\sum^{n}_{i=1} |y_i\hat{} - ty_i| \qquad (11)$$

Where $y_i\hat{}$ is the true value and y_i is the predicted value. A lower MAE indicates superior forecasting accuracy.

- **ROOT MEAN SQUARE ERROR (RMSE):** RMSE is a quadratic scoring rule that measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation. The RMSE is calculated using the formula:

$$\textbf{RMSE} = \sqrt{(1/n)\sum^{n}_{i=1} (y_i\hat{} - ty_i)} \qquad (12)$$

Like MAE, a smaller RMSE signifies better forecasting performance.

- **R-SQUARED (R^2):** R^2 represents the proportion of the variance for the dependent variable that's explained by the independent variables in the model. R^2 values range from 0 to 1, with higher values indicating a better fit of the model to the data. The R^2 is calculated using the formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\widehat{y_i} - y_i)^2/n}{\sum_{i=1}^{n} (\widehat{y_i} - ty_i)^2/n} \qquad (13)$$

Where $y_i\hat{}$ is the average true value and y_i is the average predicted value. The R^2 value ranges from 0 to 1, with higher values indicating a better fit. A smaller MAE and RMSE suggest reduced error between predicted and true values, signifying higher forecasting accuracy. Additionally, a higher R^2 indicates better model fitting.

- **MEAN ABSOLUTE PERCENTAGE ERROR (MAPE):** MAPE expresses accuracy as a

percentage, which makes it easy to interpret. Unlike metrics like RMSE (Root Mean Squared Error) or MAE (Mean Absolute Error), which are scale dependent, MAPE gives a quick, intuitive understanding of the model's accuracy in terms of percentage error.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{y_i - \hat{y_i}}{y_i} \quad (14)$$

Where y_i is the actual value, $y_i\hat{}$ is the predicted value and n is the number of observations. A lower MAPE value indicates better predictive accuracy.

C. *CNN-LSTM Model*

The CNN-LSTM model is implemented with the following parameter settings. The architecture involves one-dimensional convolutional layers, LSTM layers, and an attention mechanism.

TABLE I. CNN-LSTM PARAMTERS

Parameters	Values
Convolution layer filters	64
Convolution layer kernel size	3
Convolution layer activation function	Relu
MaxPooling 1D layer	Pool size = 2
Droupout rate	0.2
LSTM units	100
Activation	tanh
Return Sequences	True
Learning rate	0.0005
Optimizer	Adam
Loss function	MSE
Time_step	10
Epochs	50

D. *CNN-BiLSTM Model*

The CNN-BiLSTM model is implemented with the same parameter settings as the CNN-LSTM model, with the addition of bidirectional LSTM layers. The bidirectional layers enhance the model's ability to capture temporal dependencies in both forward and backward directions.

E. *Result*

TABLE II. MODEL PERFORMANCE COMPARISON

MODEL	RMSE	MAE	R^2	MAPE
CNN-BiLSTM	**116.74**	**64.88**	**0.940**	**88.12%**
CNN-LSTM	**76.79**	**25.02**	**0.974**	**25.66%**
CNN	**39.23**	**8.80**	**0.993**	**390.47%**
LSTM	**194.09**	**40.42**	**0.835**	**332.58%**

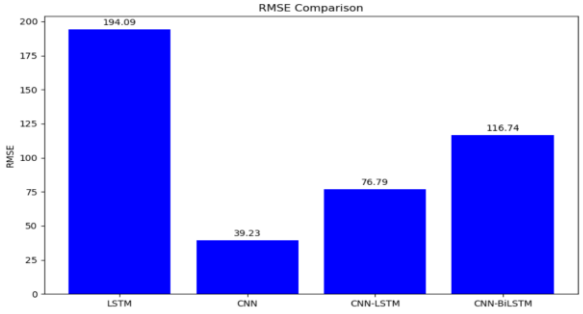


Fig 4 : ***RMSE comparison visualization***

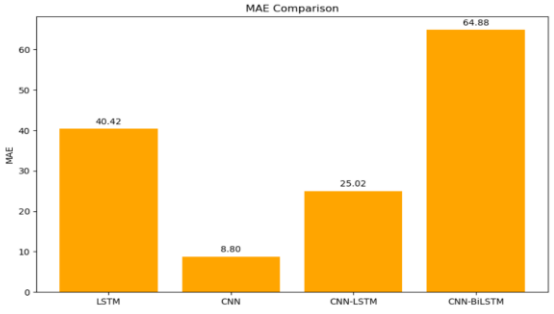


Fig 5 : ***MAE comparison visualization***

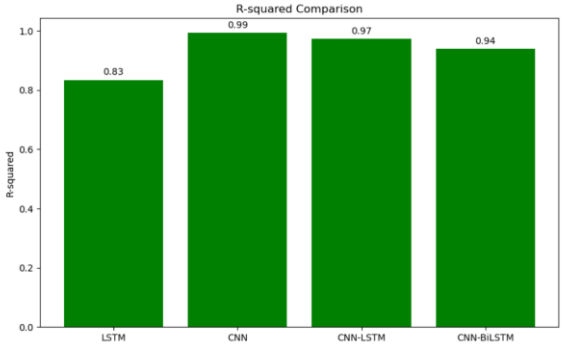


Fig 6: ***R^2 comparison visualization***

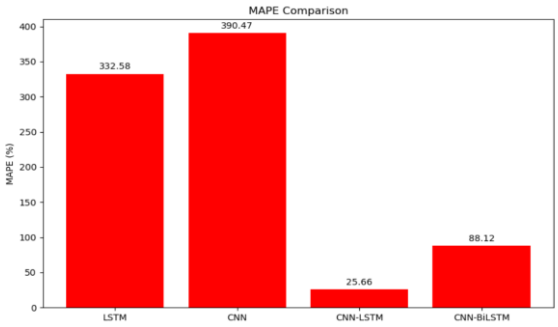


Fig 7 : ***R² comparison visualization***

F. Comparative evaluation

The comparative evaluation of the different models provides valuable insights into their individual strengths and weaknesses in time-series forecasting. The CNN model stands out with its superior performance in terms of RMSE and MAE, showcasing its strong predictive accuracy and consistency. However, the notably high MAPE of the CNN model raises concerns about its reliability, especially in scenarios where actual values are low, leading to disproportionately high MAPE values.

CNN's proficiency in feature extraction significantly contributes to its high RMSE and MAE performance. CNNs excel at identifying complex patterns in data, a critical aspect of accurate outcome prediction. This capability to discern and learn intricate features underpins the model's precision and consistency, as reflected in the low RMSE and MAE. Nevertheless, the elevated MAPE suggests that while the model generally predicts accurately, it may exhibit substantial relative errors when inaccuracies occur. This could be attributed to potential overfitting or a lack of generalization across diverse data points.

In contrast, the CNN-LSTM model, which merges CNN's feature extraction prowess with LSTM's sequential data handling, demonstrates a more balanced performance. It outperforms the standalone LSTM model in RMSE and MAE metrics and most importantly, it achieves the lowest MAPE among all models. This indicates that beyond high accuracy, the CNN-LSTM model maintains consistent relative error across diverse data points. The inclusion of the attention mechanism in this model likely contributes to its enhanced focus on relevant features and temporal contexts, leading to more accurate and reliable predictions. This balance positions it as a potentially more dependable model for this specific dataset and task.

The standalone LSTM model, while not matching the CNN or CNN-LSTM in RMSE and MAE, still exhibits commendable predictive capabilities. LSTMs are adept at capturing protracted temporal dependencies in time-series data, essential for accurate forecasting. However, its higher RMSE and MAE suggest a less effective capture of data nuances compared to the CNN-based models.

The CNN-BiLSTM model, integrating bidirectional LSTMs with CNNs, delivers moderate performance. It doesn't achieve the accuracy levels of the CNN model but surpasses the standalone LSTM. The bidirectional approach of this model, processing information from both past and future data points, aims to enhance predictive accuracy. However, in this specific scenario, the added complexity of the bidirectional LSTM doesn't proportionately improve predictive accuracy.

In summary, while the CNN model excels in RMSE and MAE, its high MAPE indicates potential reliability issues in certain contexts. The CNN-LSTM model, with its balanced performance and the lowest MAPE, emerges as a potentially more reliable choice for this dataset and task, likely benefiting from the attention mechanism's ability to enhance feature focus and contextual understanding.

G. Predictive vs actual data visualization

The Predicted ‘close’ value for all the model was plotted against the actual ‘close’ values of the dataset.

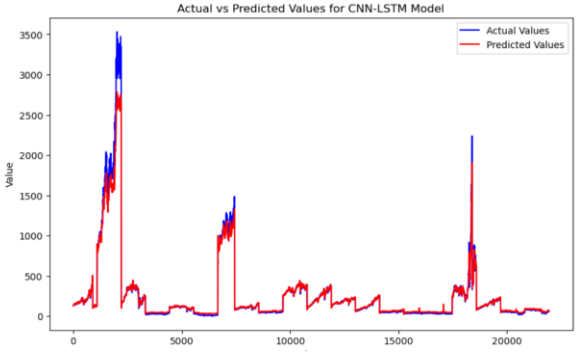


Fig 8 : ***Actual vs CNN-LSTM model***

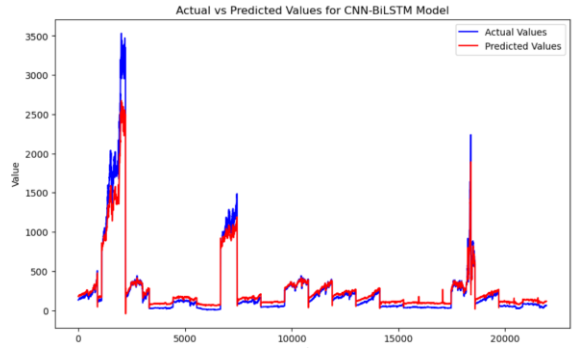


Fig 9 : ***Actual vs CNN-BiLSTM model***

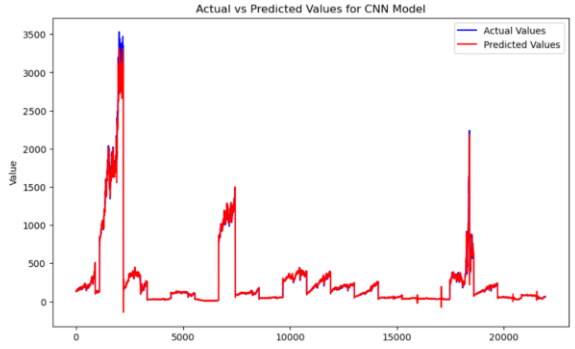


Fig 10 : ***Actual vs CNN model***

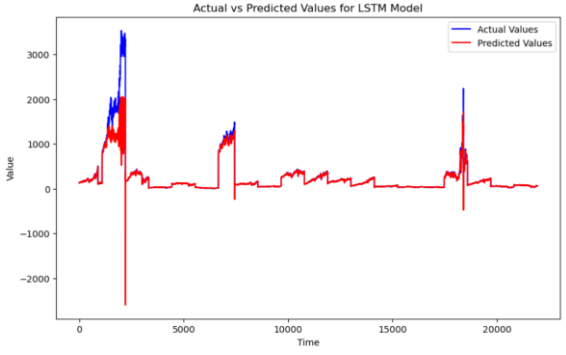


Fig 11 : ***Actual vs LSTM model***

5. CONCLUSION

This study presents a comprehensive evaluation of four distinct models ; CNN, LSTM, CNN-LSTM, and CNN-BiLSTM in the context of stock price forecasting. The results highlight the strengths and limitations of each approach. The CNN model, renowned for its feature extraction capabilities, excels in terms of RMSE and MAE, indicating high accuracy and consistency in predictions. However, its significantly high MAPE suggests potential overfitting or lack of generalization in certain scenarios. The LSTM model, while not as accurate as CNN in RMSE and MAE, demonstrates its effectiveness in capturing long-term dependencies in time-series data.

The hybrid CNN-LSTM with the attention mechanism model emerges as a balanced solution, leveraging CNN's feature extraction prowess and LSTM's sequential data handling to achieve lower RMSE and MAE, and most notably, the lowest MAPE among all models. This balance suggests a high degree of reliability and consistency across various data points. The CNN-BiLSTM model, integrating bidirectional LSTMs, shows moderate performance, indicating that its added complexity does not necessarily translate into significantly improved accuracy for this dataset.

A. Future work direction

Future research could focus on optimizing these models further, particularly in addressing the high MAPE observed in the CNN model.

External factors such as market sentiment, economic indicators, and geopolitical events can also be considered.

Advanced architectures can also be experimented with. More advanced neural network architectures, such as Transformer models, might offer improvements over the current models, especially in capturing complex patterns in stock price data.

Cross-Market Analysis: Extending the analysis to different stock markets and sectors might help in understanding the models' effectiveness across various market conditions and environments which will help in adoption of the model.

REFERENCES

[1] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," Omega, vol. 33, no. 5, pp. 497-505, 2005

[2] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," Hindawi Complexity, vol. 2020, Article ID 6622927, 2020. [Online]. Available: https://doi.org/10.1155/2020/6622927

[3] R. Adhikari and R. K. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," Neural Comput. & Appl., vol. 24, pp. 305-315, 2014. [Online]. Available: https://doi.org/10.1007/s00521-013-1386-y

[4] L. Zhang, F. Wang, B. Xu, W. Chi, Q. Wang, and T. Sun, "Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI," Neural Comput. & Appl., vol. 30, no. 5, pp. 1425-1444, 2018.

[5] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., pp. 6645-6649, 2013.

[6] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," J. Supercomput., vol. 76, no. 3, pp. 2098-2118, 2020.

[7] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in Proc. Int. Joint Conf. Neural Netw., pp. 1419-1426, 2017.

[8] B. Hu, Z. Lu, H. Li, and Q. Cheng, "Convolutional neural network architectures for matching natural language sentences," in Adv. Neural Inf. Process. Syst., vol. 27, pp. 2042-2050, 2014.

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

[10] F. Kamalov, "Forecasting significant stock price changes using neural networks," arXiv preprint arXiv:1912.08791, 2019. [Online]. Available: https://arxiv.org/pdf/1912.08791.pdf

[11] Y. Xue, C. Wang, and C. Miao, "Research on financial assets transaction prediction model based on LSTM neural network," Neural Comput. & Appl., 2020. [Online]. Available: https://doi.org/10.1007/s00521-020-04992-7

[12] J. Zhang, L. Ye, and Y. Lai, "Stock price prediction using CNN-BiLSTM-Attention model," Mathematics, vol. 11, no. 9, p. 1985, 2023. [Online]. Available: https://doi.org/10.3390/math11091985

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[14] L. Qin, N. Yu, and D. Zhao, "Applying the convolutional neural network deep learning technology to behavioral recognition in intelligent video," Tehnicki Vjesnik-Technical Gazette, vol. 25, no. 2, pp. 528-535, 2018.

[15] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Computer Science, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

[16] A. Salehpour, "Predicting Automobile Stock Prices Index in the Tehran Stock Exchange Using Machine Learning Models," I.J. Intelligent Systems and Applications, vol. 5, pp. 12-27, Oct. 2023. [Online]. Available: http://www.mecs-press.org/. DOI: 10.5815/ijisa.2023.05.02

[17] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," arXiv preprint arXiv:1801.02143, 2018.

[18] A. R. Gosthipaty, D. Chakraborty, and R. Raha, "Long Short-Term Memory Networks," PyImageSearch, 2022. [Online]. Available: https://pyimagesearch.com/2022/08/01/long-short-term-memory-networks

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," MIT Press, vol. 9, no. 8, pp. 1735-1780, 1997.

[20] Sidharth, "Convolutional Neural Network (CNN): Architecture Explained | Deep Learning," PyCodeMates, 2023. [Online]. Available: https://www.pycodemates.com/2023/06/introduction-to-convolutional-neural-networks.htmlF.

[21] E. Alibasic, B. Fazo, and I. Petrovic, "A new approach to calculating electrical energy losses on power lines with a new improved three-mode method," Tehnicki Vjesnik-Technical Gazette, vol. 26, no. 2, pp. 405-411, 2019.

Adebanjo Olawunmi Asake is an Assistant lecturer at the department of Software Engineering Babcock University, Ilishan-Remo, Nigeria. She holds an MSc in Computer Science from the University of Lagos and a BSc in Computer Science from the Federal University of Agriculture, Abeokuta. Her research interests include Machine Learning, Deep Learning, Data science and Human-Computer Interaction (HCI).