International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Stock Price Prediction: Evaluating the Efficacy of CNN, LSTM, CNN-LSTM, and CNN-BILSTM Models

Ebiesuwa Seun[1], Adebanjo Olawunmi Asake[2], Adesoji Adedeji Adegbola[3], Adegbenjo Aderonke A[4], Okesola, Kikelomo Ibiwumi[5], Falana Taye Oluwaseun [6], Awoniyi Tolulope Amos[7]

Submitted:17/03/2024 **Revised**: 28/04/2024 **Accepted**: 08/05/2024

Abstract: The stock market's dynamic nature predicts accurate prices which is a daunting task for analysts and investors. Conventional 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 forecast 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 prediction accuracy and error minimization. Analysis 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: *Stock price, CNN, LSTM, BiLSTM*

1. Introduction

Financial markets, particularly the stock market, serve as the backbone of modern economies, acting as a gauge for economic health and a crucial mechanism for capital allocation. Understanding the stock market's movements is crucial since they reflect the economic strength and financial health of a nation. The market's behaviour 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.

The challenge lies in the inherent complexity and volatility of the stock market. Stock prices are affected by various factors, ranging from company-specific events to macroeconomic shifts, geopolitical developments, and even psychological factors affecting investor sentiment. This multifaceted nature makes the stock market notoriously difficult to predict, presenting a formidable challenge to investors, analysts, and researchers alike.

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. However, it can overlook short-term market sentiment and rapid shifts in investor behaviour. On the other hand, technical analysis looks at historical price patterns and market activity to forecast future movements, may fail to account for unexpected events or structural changes in the market. 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 algorithms like Linear Regression, Support Vector Machines, and ARIMA. Each of these has had varying successes, 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. The limitations of these traditional methods have spurred interest in more sophisticated approaches, particularly in the realm of machine learning and artificial intelligence. These technologies offer the potential to compute large volumes of data, identify complex patterns, and adjust to changing market conditions in ways that surpass human capabilities. However, even conventional machine learning algorithms like Linear Regression, Support Vector Machines, and ARIMA have struggled to fully capture the non-linear and often chaotic nature of stock price movements. The challenge lies not just in processing large volumes of data, but in understanding the intricate temporal dependencies

[1] S.D.M.College of Eng. &Tech, Fort Collins – 8023, USA
ORCID ID : 0000-3343-7165-777X
[2] KLE.Institute of Technology, Tsukuba – 80309, JAPAN
ORCID ID : 0000-3343-7165-777X
[3] Computer Eng., Selcuk University, Konya – 42002, TURKEY
ORCID ID : 0000-3343-7165-777X
** Corresponding Author Email: author@email.com*

and multifaceted influences that shape stock prices.

Deep learning, especially 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 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.

The significance of this research extends beyond academic interest. More accurate stock price predictions can have far-reaching implications, empowering investors with better decision-making, enabling companies to optimize capital allocation, and aiding policymakers in navigating market trends. Ultimately, this paper seeks to push the boundaries of present-day forecasting methodologies and contribute to a more stable, efficient, and prosperous financial market.

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 models. The Autoregressive Integrated Moving Average (ARIMA) model [1] is one of these models, though adept at handling linear relationships, falls 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 artificial neural networks (ANN) with random walk (RW) to forecast four financial time series datasets. Their findings indicated a noticeable enhancement in forecasting accuracy. [4] suggested an LM-BP neural network-based network architecture for forecasting stock prices. This innovation addressed the drawbacks of the traditional BP neural network, particularly its slow training speed and low precision, leading to improved forecasting outcomes.

An RNN, or recurrent Neural Network, 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 an accuracy rate of 57%, this accuracy could be improved upon. [7] in their study fed technical indicators to an LSTM network to predict stock market trends in Brazil, 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] combined CNN, MLP, and LSTM to predict the stock prices of four publicly traded U.S. companies. The results 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 predicting model for financial market time series using the LSTM deep neural network. They found that the LSTM deep neural network effectively predicted stock market time series and

achieved superior forecasting accuracy when compared to other traditional methods like traditional RNNs, BP neural networks, and an improved LSTM deep neural network.

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 different variants of neural network models including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), CNN-RNN, and CNN-LSTM. The results indicate that CNN-LSTM has the highest accuracy among the six forecasting models, with the predicted values nearly matching the real values.

[12] implemented a hybrid model for stock price forecasting that combines CNN, BiLSTM (Bidirectional Long Short-Term Memory) and an Attention mechanism. CNN-BiLSTM-Attention was evaluated by contrasting it with alternative models. The CNN-BiLSTM-Attention model fared better than the other models in forecasting the closing prices of these indices.

3. CNN, LSTM AND BILSTM

Architectural breakdown of the Deep learning techniques are as follows;

3.1 Convolutional Neural Networks (CNN)

Convolutional Neural Network, commonly known as CNN, is a feedforward neural network that has gained prominence due to its effectiveness across different fields, such as 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:

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

The output feature map at time step t is Ocnn(t), I(t+i) represents stock market data at time t+i, F(i) is the filter's weights, and k is the kernel size. Post convolution, non-linearity is introduced by using an activation function, like the Rectified Linear Unit (ReLU), which helps the model learn intricate patterns:

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

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

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

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

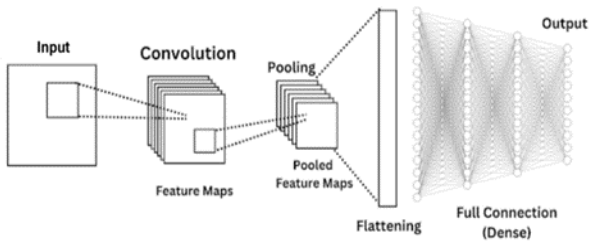

Fig 1 . *CNN Architecture*

3.2. LONG SHORT-TERM MEMORY (LSTM)

Convolutional Neural Network, commonly known as CNN, is a 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 control 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.

3.2.1. Forget Gate: decides which data is removed from the cell state. To determine which values in the cell state Ct−1 should be permitted to go through, it applies a sigmoid function σ to the former hidden state ht−1 and the current input $\mathcal{X}$t

$$ft = \delta(\mathrm{W}f.[h_{t-1,}\ X_{t,}\ ] + \mathrm{b}f) \tag{4}$$

3.2.2 Input Gate: which new information is to be added to the cell state is chosen by the input gate. The values to update are selected using a sigmoid function, and a vector of new candidate values C˜t that might be added to the state is created using the tanh function.

$$it = \delta(\mathrm{W}f.[h_{t-1,}\ X_{t,}\ ] + \mathrm{b}i) \tag{5}$$

$$C\tilde{}t = \tanh(\mathrm{W}C.[h_{t-1,}\ X_{t,}\ ] + \mathrm{b}c) \tag{6}$$

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

$$C\tilde{}t = \tanh(\mathrm{W}C.[h_{t-1,}\ X_{t,}\ ] + \mathrm{b}c) \tag{7}$$

3.2.4 Output Gate: Lastly, the output gate selects the value of the next hidden state *ht*. The hidden state has data about previous inputs. The sigmoid function selects the portions of the cell state that are sent to the output. The cell state is then passed through tanh (which pushes the value between -1 and 1) and multiplied by the output of the sigmoid gate, resulting in the output of only the selected parts.

$$Ot = \delta(\mathrm{W}o.[h_{t-1,}\ X_{t,}\ ] + \mathrm{b}o) \tag{8}$$

$$ht = O_t * \tanh(C_t) \tag{9}$$

The LSTM is a very good option for time series analysis because of its ability to handle data with long-range temporal variability, 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.

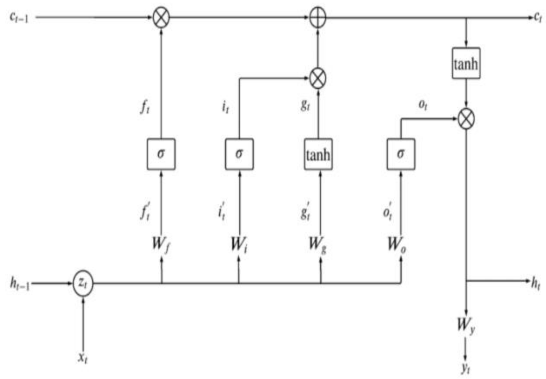

Fig 2. LSTM Architecture

3.3. 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 uses forward LSTM to process the sequence in its original order, while the other uses backward LSTM to process the sequence in reverse. 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.

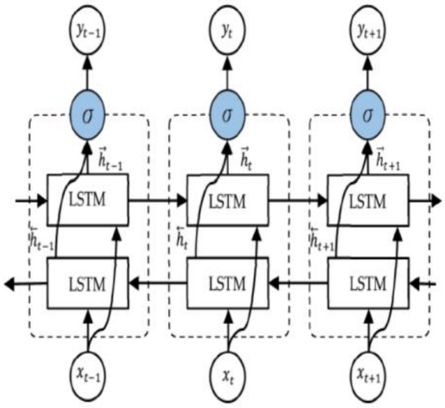

Fig 3. BiLSTM Architecture

4. MATERIAL AND METHODS

4.1. Data Sources

This analysis utilizes the "US Historical Stock Prices with Earnings Data" dataset sourced from Kaggle. This dataset includes twenty-three years (1998-2021) of daily stock prices and earnings data, with each entry containing the company symbol, date, market opening price (open), the highest point the stock reached (high), the lowest point (low), trading volume (volume), earnings per share (eps), earnings estimate (eps_est), and the company closing price (close). All these features play a crucial role in the time series forecasting model. The chosen (1998-2021) is relevant for several reasons. Firstly, it captures significant market events like the dot-com bubble burst (2000), the Great Recession (2008), and the COVID-19 pandemic (2020). These events significantly impacted stock prices and provided valuable data for training the model to recognize diverse market conditions. Secondly, with 108,100 data points, this timeframe offers sufficient historical depth to learn long-term trends and seasonal patterns in stock price movements.

4.2. Data Preprcocessing

- The data was checked for missing values and Outliers, there were no missing values
- Pertinent features such as 'Open', 'Low', 'High', 'Close', and 'Volume' were selected and computed additional technical indicators to enrich the dataset.

abbreviation “i.e.,” means “that is,” and the abbreviation “e.g.,” means “for example” (these abbreviations are not italicized).

4.3. Normalization/Standardization

- Feature Scaling: To ensure all features contribute evenly to the model's predictions, feature scaling techniques were employed to normalize the input data. This is particularly important for models like CNNs and LSTMs, which are sensitive to the scale of input data. One specific technique used for this purpose was Min-Max Scaling. This method transforms each feature value to a range between 0 and 1, as it ensures that all features contribute evenly to the model's learning process.

4.4. 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

4.5. 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.

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

4.6. Preparation for LSTM

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 tate updates. The BiLSTM extends this by processing the data bidirectionally.

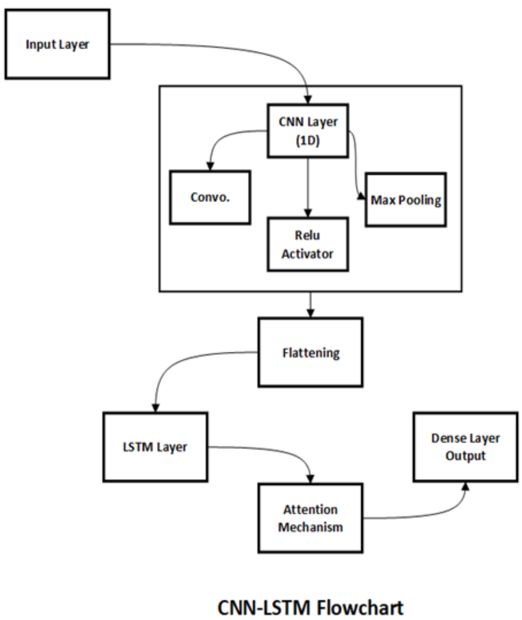

Fig 4. CNN-LSTM Flowchart

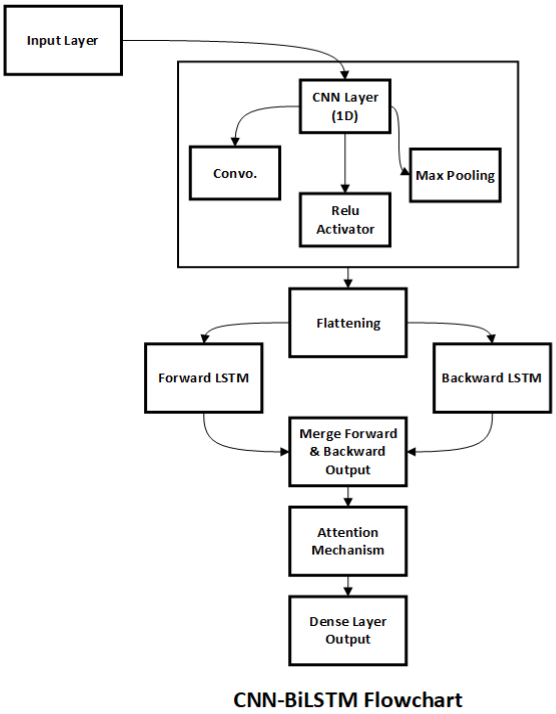

Fig 5. CNN-BiLSTM Flowchart

4.7 Attention Mechanism and Hybrid Model Output

Both models incorporate an attention mechanism after the LSTM/BiLSTM layers. This enables the model to concentrate 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.

$$Yt = Wy.ht + by \tag{10}$$

Where Yt is the predicted stock price or movement at time t, ht is the last hidden state of the LSTM and the output layer's learnt weights and biases are denoted by Wy and by.

5. 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: Mean Absolute Error (MAE), R squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). To appraise the 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

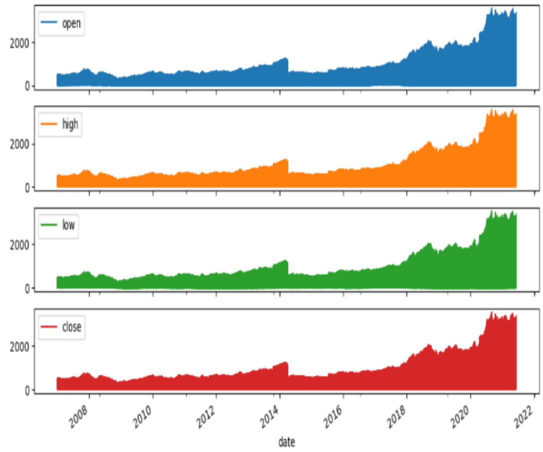

Fig 6. Visual Representation of the data

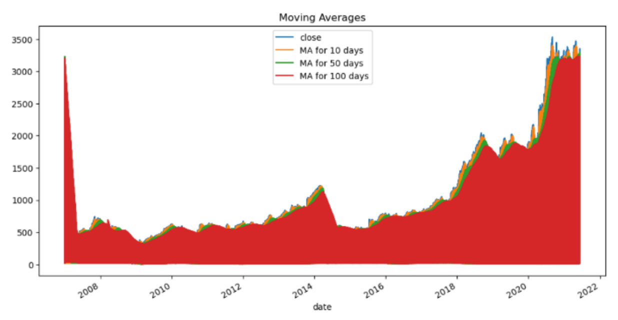

Fig 7. Moving Average visualization

The stock price data was preprocessed, and features like Moving Average (MA10) and Relative Strength Index (RSI) were calculated. The data was then divided into training and testing sets.

5.1. Model Architecture

Convolutional layers were utilized in both models for feature extraction, while LSTM or BiLSTM layers were used to capture temporal dependencies. An attention mechanism was also integrated to enhance the model's focus on relevant features.

5.2. Model Evaluation Metrics

Model performance was assessed using these key metrics: mean absolute error (MAE), root mean square error (RMSE),R-squared (R^2) and mean absolute percentage error (MAPE).

5.3. CNN-LSTM Model

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

Table 1. CNN-LSTM parameters and values

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
Timestep	10
Epochs	50

5.4. 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.

5.5. Result

Table 2. 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%

5.6. Interpretations

5.6.1. LSTM Model

RMSE: 194.09. This is relatively high, showing that, on average, the model's predictions and the actual values differ by 194.09 units.

MAE: 40.42. On average, the absolute error of the predictions is around 40.42 units.

R-squared: 0.835. This indicates that the independent factors account for about 83.5% of the variance in the dependent variable.

MAPE: 332.58%. This extremely high percentage indicates poor predictive accuracy, as the average error is more than three times the actual value.

5.6.2. CNN Model

RMSE: 39.23. This is significantly lower than the LSTM model, indicating better predictive accuracy.

MAE: 8.80. In comparison to the LSTM model, the CNN model appears to have a higher average accuracy per prediction, as indicated by its lower MAE.

R-squared: 0.993. This high number suggests that almost all the response data variability around its mean can be explained by the model.

MAPE: 390.47%. Despite the lower RMSE and MAE, the MAPE is extremely high, suggesting that the model may not be reliable for certain types of predictions or in certain conditions.

5.6.3. CNN-LSTM Model:

RMSE: 76.79. This indicates moderate prediction accuracy because it is lower than the LSTM model but greater than the CNN model.

MAE: 25.02. This suggests a reasonable average accuracy per prediction.

R-squared: 0.974. A high number, meaning a significant amount of data variation can be explained by the model.

MAPE: 25.66%. In terms of percentage error, this is the lowest of all the models, indicating that the CNN-LSTM model has the most reliable and accurate predictions.

5.6.4. CNN-BiLSTM Model

RMSE: 116.74. This shows that there is an average difference of 116.74 units between the model's predicted value and the actual value.

MAE: 64.88. With the largest MAE of all the models, it predicts less accurately on average.

R-squared: 0.940. This is a strong score, indicating that a sizable amount of data variance can be explained by the model.

MAPE: 88.12%. This is high but lower than the LSTM and CNN models, indicating moderate predictive accuracy in terms of percentage error.

5.7. Evaluation Metrics Visualization

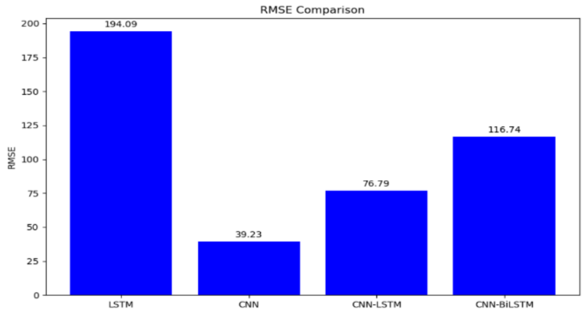

Fig 8. RMSE comparison visualization

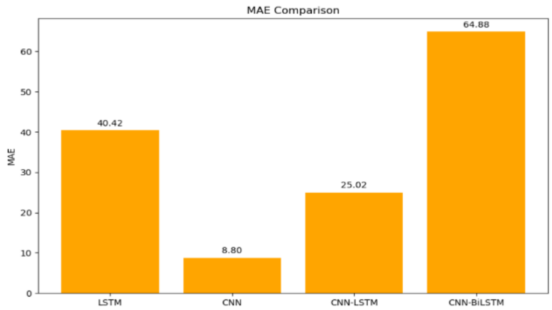

Fig 9. MAE comparison visualization

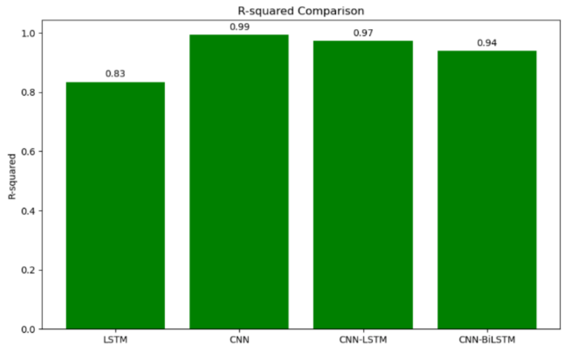

Fig 10. R^2 comparison visualization

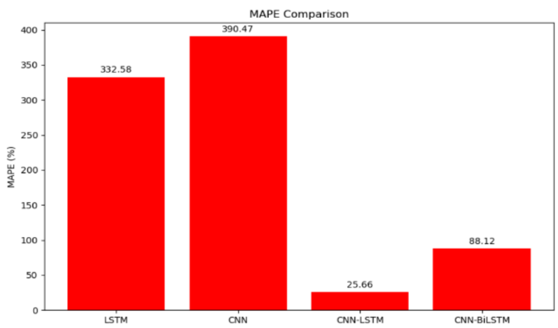

Fig 11. R^2 comparison visualization

5.8. Comparative evaluation

The comparative evaluation of the different models provides valuable insights into their strengths and weaknesses in forecasting time series. The CNN model stands out with higher 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 as opposed 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.

6. Conclusion

This study presents a comprehensive comparative analysis of four distinct models: CNN, LSTM, CNN-LSTM, and CNN-BiLSTM in the context of stock price forecasting. The results highlight the strengths and weaknesses of each approach. The CNN model, renowned for its feature extraction capabilities, excels in terms of RMSE (39.23) and MAE (8.80), indicating high accuracy and consistency in predictions. However, its significantly high MAPE (390.47%) 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 identifying long-term relationships in time-series data. The hybrid CNN-LSTM model emerges as a balanced solution, leveraging CNN's feature extraction prowess and LSTM's sequential data handling to achieve lower RMSE (76.79) and MAE (25.02), and most notably, the lowest MAPE (25.66%) 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.

6.1. Future work direction

Future research could focus on optimizing these models further, particularly in addressing the high MAPE observed in the CNN model. Optimizing hyperparameters, especially techniques like dropout or L1/L2 regularization, could potentially improve generalizability.

Incorporating additional features beyond historical prices, such as market sentiment, economic metrics, and geopolitical events, might enhance the models' ability to capture complex market dynamics. Advanced architecture 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 inform their potential adoption in real-world scenarios.

Conflicts of interest

The authors declare no conflicts of interest.

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.

[22] Sidharth, "Derivation of Backpropagation in Convolutional Neural Network (CNN)," PyCodeMates, Aug. 15, 2023. [Online]. Available: https://www.pycodemates.com/2023/07/backward-pass-in-convolutional-neural-network-explained.html. Accessed: Oct. 20, 2023.