Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Evaluation of Deep Learning Models for Time Series Forecasting in Financial Markets

Balaram Pal

balarampal23@outlook.com

Techno Bengal Institute of Technology

Munshi Mostafijur Rahaman

Techno Bengal Institute of Technology

Soumyajit Paul

Indian Institute of Science Bangalore

Sakshi Verma

Techno Bengal Institute of Technology

Rithika Nayak

Techno Bengal Institute of Technology

Priti Seal

Techno Bengal Institute of Technology

Ankita Kumari

Techno Bengal Institute of Technology

Srijani Bera

Techno Bengal Institute of Technology

Research Article

Keywords:

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11563766/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Evaluation of Deep Learning Models for Time Series Forecasting in Financial Markets

Balaram Pal[1*], Munshi Mostafijur Rahaman[2*], Soumyajit Paul[3†], Sakshi Verma[4†], Rithika Nayak[5†], Priti Seal[6†], Ankita Kumari[7†], Srijani Bera[8†]

[1*]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[2*]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[3]Computer Science and Engineering Department, IISc Bangalore, Bengaluru, 560012, Karnataka, India.
[4]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[5]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[6]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[7]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.
[8]Computer Science and Engineering Department, Techno Bengal Institute of Technology, Kolkata, 700150, West Bengal, India.

*Corresponding author(s). E-mail(s): balarampal23@outlook.com; mostafijur.rahaman@gmail.com;
Contributing authors: soumyajit651@gmail.com; vermasakshi1976@gmail.com; rithikanayak312@gmail.com; wbpritiseal@gmail.com; ankitakumari2787@gmail.com; bsrijani896@gmail.com;
†These authors contributed equally to this work.

Abstract

Stock value prediction is a challenging task due to unstable and non-lectured nature of financial markets. This research examines the application of long- term short-term memory (LSTM) and Convolutional Neural Network (CNN) in forecasting stock price by taking advantage of historical market data. This study compares the effectiveness of these models in predicting stock price movements, using key matrix such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2) and accuracy. Integrating LSTM and CNN increase both temporary dependence and the future accuracy by capturing important market trends.

Keywords: Stock price prediction, financial forecasting, time-series analysis, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), market trends, sequential data, feature extraction, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2), accuracy, predictive modeling, investment decision-making.

1 Introduction

Stock Price prediction has been extensively studied as it plays a vital role in the financial industry. Accurate prediction of stock prices helps investors, traders, and financial analysts make better decisions and manage financial risks. However, predicting stock prices is difficult because financial markets are influenced by many changing factors. These include historical price movements, global economic conditions, political developments, investor sentiment, and overall market behavior. The combined effect of these factors makes stock prices difficult to predict accurately.

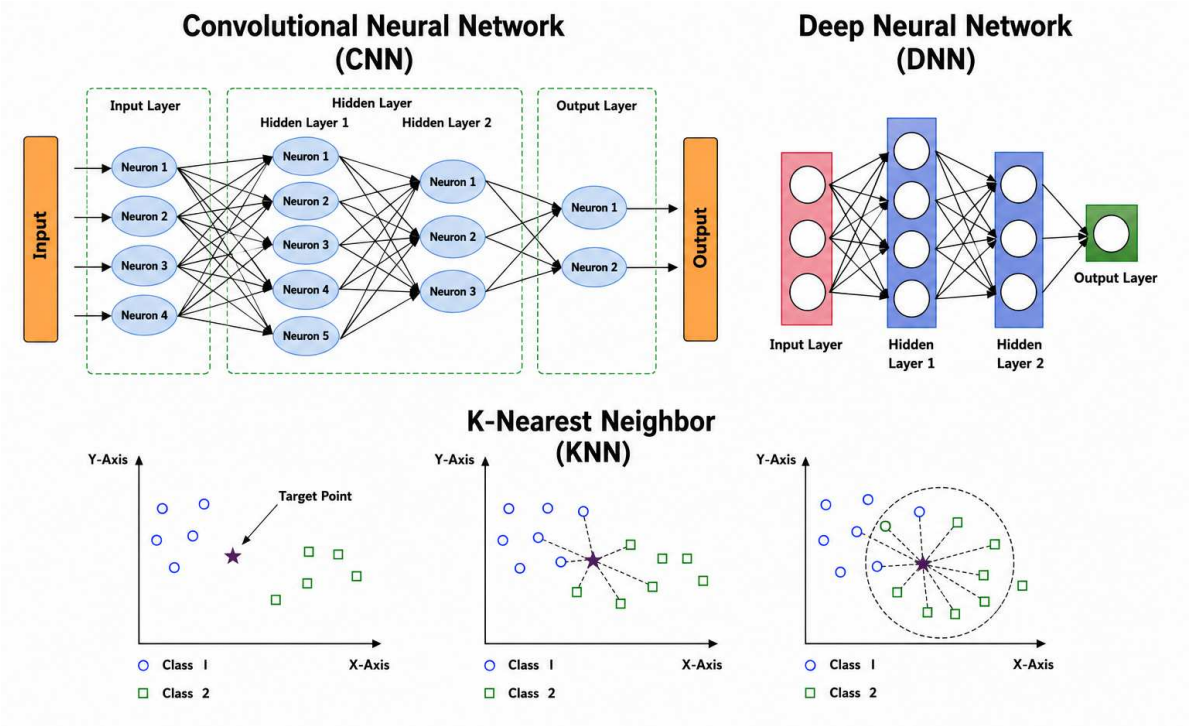


Fig. 1 CNN, RNN and KNN Model

Traditional statistical and econometric models (such as ARIMA and generalized autoregressive conditional heteroskedasticity (GARCH)) often fail to capture these nonlinear relationships and respond to rapid market movements. Thus, more sophisticated methods are needed to provide accurate and robust forecasts.

Data-driven analysis in finance has received significant attention in recent years, particularly with the advent of deep learning techniques to process a large amount of data and extract meaningful patterns from it. Some of these include Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, Deep Neural Networks (DNN) and k-Nearest Neighbors (KNN) which have shown potential in stock price prediction. Each of these uses different approaches to predict stock prices. CNNs can extract useful patterns from input data, helping identify features related to stock price movements. LSTM, or Long Short-Term Memory, is a type of Recurrent Neural Network (RNN) designed to process sequential data and learn long-term dependencies in historical stock prices. DNN, or Deep Neural Network, uses multiple hidden layers to learn complex relationships within large datasets, which may help improve prediction performance. On the other hand, KNN algorithm is non-parametric, which gives a simple but powerful way of making predictions based on past behavior of stock prices.The forecasting approaches discussed above differ in their capabilities and

limitations when applied to financial time-series data. Table 1 summarizes the main characteristics of these models.

Table 1 Comparison of Forecasting Models for Financial Time-Series Data

Model	**Main Approach**	**Key Limitation**
ARIMA	Models statistical patterns and temporal dependencies in time-series data	May struggle with complex non-linear relationships
GARCH	Models changing volatility in financial time series	Primarily focuses on volatility rather than directly forecasting price movements
CNN	Extracts local patterns and features from input data	Does not inherently model long-term sequential dependencies
LSTM	Learns temporal dependencies in sequential data	May require substantial training time and computational resources
DNN	Learns complex representations through multiple layers	Does not inherently capture temporal dependencies in sequential data
KNN	Makes predictions based on similarities between observations	Sensitive to feature scaling and the choice of **K**
CNN–LSTM	Combines CNN-based feature extraction with LSTM-based temporal learning	Performance depends on architecture, data quality, and parameter selection

Approaches show good performance on their own, but there is an increasing interest in applying hybrid methods, combining different techniques to improve forecast performance. The CNC-LSTM model is a powerful combination for predicting stock prices in a more efficient approach where we are looking for same information in a sequence. Although CNN makes it possible to extract important features of price movement, LSTM processes temporal patterns and thus enables more profound analysis of market trends.

For example, the use of DNN in predictive models may lead to increased accuracy through multiple levels of abstraction. KNN tends to be a standalone method and also improves the performance of the final classification layer of an installed deep learning model.
Many researchers, including Yang et al. (2016), have studied stock price prediction using statistical methods and individual machine learning models. However, predicting stock prices is still difficult because market conditions change frequently and prices are affected by many uncertain factors. Although CNN-LSTM models have been explored in previous studies, comparisons with other methods, such as DNN, KNN, and traditional models, are still important to understand which approaches perform better. Another limitation is that many studies focus mainly on prediction accuracy without discussing how their findings could be useful to investors. Therefore, this study compares different prediction models to understand their performance, identify their strengths and weaknesses, and examine their potential usefulness in stock market analysis. This work attempts to address the difficulties through implementation and

evaluation of a hybridized CNN-LSTM-based stock price prediction model.
The rest of the paper is organized as follows: The next section discusses the related work on stock price prediction and the research gap. Section 3 provides our methodology, which encompasses data selection, model architecture, and evaluation measures. Section 4 highlights results and analysis, and discusses main findings. Section 6 contains conclusion and future research avenues.

2 RELATED WORK

Researchers have explored different ways to predict trends in stock market using traditional methods and deep learning models. S. Mehtab and J. Sen (2019) developed a stock price prediction approach that used deep learning and natural language processing. Their method also considered market sentiment and other factors beyond historical stock prices to improve predictions. This suggested the potential value that non-numerical data can bring to financial forecasting and how it, as we have seen, can boost the model performance in challenging market conditions. A related study was conducted by M. Agrawal et al. (2022) [22], in which a deep learning model was implemented with dynamic indicators to predict stock indices, while targeting high-dimensional models trained with the integration of multiple variables to maximize the precision of predictions. Their findings helped to gain insights into the ability of the model to learn complex relationships in the financial specific datasets, particularly suggesting a strong performance even for short settlement cycles often occurring in modern markets.

R. Singh and S. Srivastava (2017) used Convolutional Neural Networks (CNNs) for stock price prediction. Their study explored how CNNs could identify patterns in time-series data, similar to the way they identify patterns in images. This showed that CNNs could also be applied to stock market forecasting, rather than being limited to image-related tasks. Later, Mukherjee et al. (2023) studied different algorithms to predict stock market trends. Their work examined CNN and Long Short-Term Memory (LSTM) networks for identifying patterns in financial data and dealing with the challenges caused by market volatility. This research highlights the importance of recurrent neural networks, especially in scenarios where understanding the sequences is crucial.

M. Agrawal, A.U. Khan, and P.K. Shukla (2019) performed promising stock index predictions using deep learning on technical indicators in earlier work. Specifically, they concentrated on using convolutional layers to detect complex trends and patterns in stock market datasets, showcasing the advantages offered by deep learning relative to traditional machine learning approaches in the finance domain.

A significant study by Chan et al. (2024) proposed a model which uses LSTM-CNN to predict stock trends while tuning parameters using Bayesian Optimization. The research emphasized the benefits of leveraging the ability of LSTMs to capture

long-term relationships in sequence data and the prowess of CNNs in feature extraction. Through the application of Bayesian Optimization, the proposed approach sought to improve the predictive ability of the model by fine-tuning hyperparameters, thus achieving superior predictive performance. The results obtained showed that the LSTM- CNN hybrid model produced more accurate forecasts than both conventional statistical approaches and standalone deep learning models when it comes to trending stock market predictions.

Out of the models that try sentiment analysis, Jayanth Balaji used 14 different Deep Learning approaches to test stock price prediction, where CNN and LSTM model performed to a good extent because they process social media data. By incorporating the social market sentiment data into the time series models (ARIMA), they showed that this was an effective way to model time dependent data and improve predictive accuracy.

2.1 RESEARCH OBJECTIVES

This study aims to introduce a new stock market price prediction model using a hybrid framework of CNN-LSTM and other models. Price motions in stock markets are complex decisions that are difficult to accurately predict despite having many data to work with. In stock price prediction, CNNs can be used to identify useful patterns from historical stock data, while Long Short-Term Memory (LSTM) networks help predict future prices by learning from past price movements. CNNs extract important features from the data, while LSTMs are designed to handle sequential information and learn long-term dependencies. Combining these two models can help analyze stock price patterns over time. Incorporating both methods together, this CNN-LSTM model was built to obtain advanced accuracy and reliability for stock price prediction.

Additionally, this research aims to investigate the predictive capabilities of CNN-LSTM in relation to conventional models such as Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), and individual CNN or LSTM models. MLP and RNN are traditional models that have been widely used in financial forecasting and often fail in adapting to the volatility of stock market. We will analyze through this study whether a specific model class for time series works better in one scenario than the others and will benchmark on the accuracy metrics by using MAE, RMSE and R^2. This comparative analysis is expected to be useful underlining the benefits of applying the CNN-LSTM approach instead of classical forecasting methods.

This study also aims to compare the prediction performance of the CNN-LSTM model with that of traditional predictive models, including Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN) and models only for CNN or LSTM. Even though traditional models like MLP and RNN had been widely used in financial forecasting, they lack skills in dealing with complex stock market changes. This reduces the time and chance helped us to analyze performance overview summary based on accuracy metrics MAE, RMSE and R^2 of various Machine Learning models that we turning to get through less time with high accuracy which help a lot to analyze the

project. Through this comparative analysis, the research aims to demonstrate the benefits of accepting CNN-LSTM over traditional forecasting techniques.

This study also aims to compare the prediction execution of the CNN-LSTM model with that of traditional predictive models, including Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), and CNN or LSTM only models. Even though traditional models like MLP and RNN have been widely used in financial forecasting, they lack skills in dealing with complex stock market changes. This reduces the time and chance helped us to analyze performance overview summary based on accuracy metrics MAE, RMSE, R^2 and accuracy of various Deep Learning models that we turning to get through less time with high accuracy which helps a lot to analyze the project. Through this comparative analysis, the research aims to demonstrate the benefits of accepting CNN-LSTM over traditional forecasting techniques.

In addition, the study specified an objective to improve the accuracy of stock price prediction by using historical market data to predict the closing stock price of each day for the next six months after 30 June 2020. Financial markets generate large amounts of historical data that can be analyzed to understand price movements and identify possible future trends. In this study, the CNN-LSTM model will be trained and tested using historical stock data to evaluate its prediction performance and reduce forecasting errors. The main aim is to develop a reliable prediction approach that can help investors and financial analysts understand stock market trends and make informed decisions.

This study also aims to contribute to stock price prediction by examining how the proposed model performs. Accurate predictions can support investment planning and risk management, although stock prices remain uncertain. The findings may help traders, investors, and financial analysts better understand market movements and assess the usefulness of the model for financial decision-making. This study aims to contribute to the body of knowledge in the field of financial market analysis by introducing new structured approaches to improve stock price prediction using advanced data-driven methodologies.

2.2 RESEARCH GAP

The stock market is a complex machine, multi-dimensional with tallies the number of domains; thus, stock market prediction is a difficult task. Noise in stock market data poses challenges that are not addressed by traditional models such as statistical modeling, time-series forecasting, and single neural network model.CNN and LSTM networks are widely used in deep learning, but their performance in stock price prediction still needs to be studied through proper comparisons. In particular, comparing hybrid models with individual models under the same dataset and experimental conditions can help us understand their strengths and limitations. This study focuses on evaluating these approaches to identify differences in their prediction performance.

Some previous studies have also explored deep learning methods for stock market forecasting. For example, Mehtab and Sen (2019) considered market sentiment and other non-traditional information along with deep learning techniques to predict stock prices. Their work highlighted the potential value of using additional information in forecasting. However, further comparison is needed to understand how combining CNN and LSTM performs against individual models and other approaches under the same conditions.

Similarly, Agrawal et al. (2022) proposed a deep learning model for stock index prediction by using dynamic indicators. They worked on high-dimensional models that are trained on many variables to make an accurate prediction. Indeed, their work focused on hybrid CNN-LSTM models without comparing their performance with other deep learning architectures. Although their results showed the potential of deep learning for financial forecasting, they performed a broad comparison of methods, not a fully controlled comparison of hybrid methods under the same experimental conditions.

In addition, previous stock prediction models focus mainly on being accurate while overlooking the pragmatic side of stock investment decisions. Furthermore, the prediction results in many studies are rarely interpretable, and most measures are agnostic to the investor, significantly limiting the value of the findings to finance researchers and practitioners. Moreover, the vast majority of models are based purely on past price data, ignoring external factors like economic policies, financial news, and market sentiment, which play a critical role in stock movements.

The limitations of these methodologies are addressed using the proposed model which uses the capabilities of CNNs to extract unique features and combines those with LSTMs for better sequential learning. Previous studies have explored CNNs for identifying patterns in stock market data (Singh & Srivastava, 2017) and LSTM networks for learning relationships between past and future values in sequential data (Mukherjee et al., 2023). However, further evaluation is needed to compare standalone and hybrid models using the same dataset and experimental conditions. This study also aims to present the results in a way that is useful for investors, rather than focusing only on prediction scores.

The proposed CNN-LSTM model combines the feature extraction ability of CNNs with the sequence-learning capability of LSTMs. The aim is to evaluate how effectively this combination predicts stock prices and whether it offers any advantage over the other models included in the study. External factors may also be considered in future work to explore their influence on prediction performance. The study presented in this paper not only enriches the literature on hybrid deep learning models, but also serves as a reference for the inclusion of market sentiment and macroeconomic factors in future studies.

3 MATERIALS AND METHOD

3.1 CNN-LSTM MODEL

CNNs are useful for identifying important patterns in input data, which makes them suitable for analysing large and complex datasets. In stock price prediction, CNNs can help extract patterns from historical price movements. LSTMs, on the other hand, are designed to work with sequential data. They can learn relationships between values over time, making them useful for time-series forecasting.

The proposed model combines CNN and LSTM to use the strengths of both networks. First, historical stock price data is given as input to a one-dimensional convolutional layer. This layer extracts useful features and identifies local patterns in the data. A pooling layer is then used to reduce the size of the extracted features, which helps make the computation more efficient.

The extracted features are passed to an LSTM layer, which learns how these patterns change over time. This helps the model understand the sequence of stock price movements. A dropout layer is added after the LSTM layer to reduce overfitting and help the model perform better on data it has not seen during training.

Finally, a dense layer uses the information learned by the previous layers to generate the predicted stock price. In this study, the CNN processes historical stock data up to June 30, 2020, and the LSTM learns the temporal relationships needed to predict stock prices from July 1 onward. By combining feature extraction with sequence learning, the model aims to capture patterns in historical data and improve prediction performance. The overall architecture is shown in Figure 2.

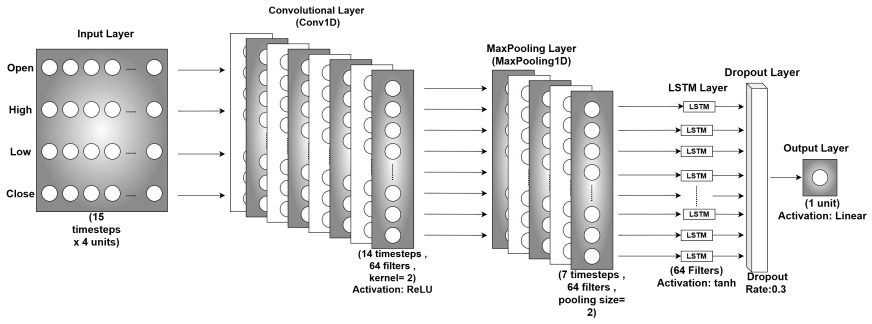


Fig. 2 Model Architecture Of CNN-LSTM

3.2 CONVOLUTIONAL NEURAL NETWORK (CNN)

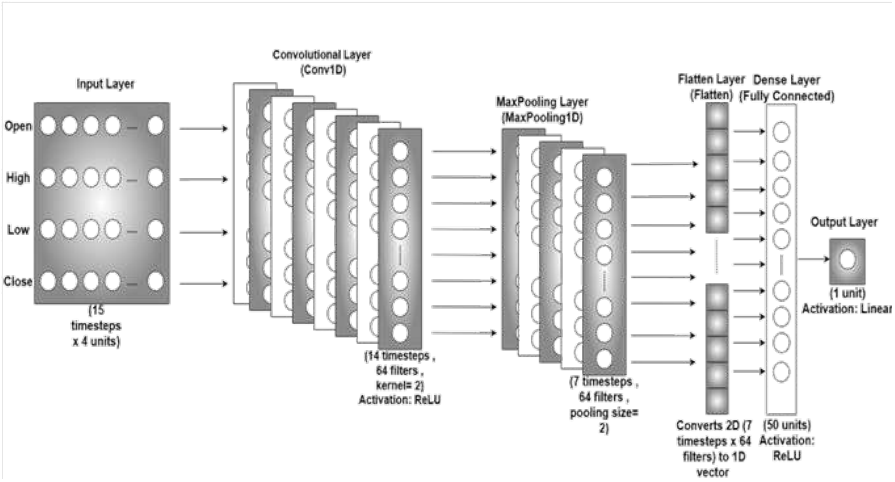


Fig. 3 Convolutional Neural Network (CNN)

CNN (Convolutional Neural Network), first defined in Lecun et al. which was designed in 1998, is a specific type of feed forward neural network often used in image processing and natural language processing. This makes it a good fit for time-series forecasting due to its performance on structured data. In addition, CNN has local perceptions and weight-sharing characteristics, the number of parameters has reduced dramatically, enabling the model to learn more efficiently.

CNN Architecture (Linear and pooling layer) the CNN architecture mainly has two components that are convolution layer and pooling layer where the convolution layer extracts feature from the input data by applying multiple convolution kernels. But high dimensional features are extracted which increases the computational complexity. To solve this problem and maximize training efficiency a pooling layer follows the convolution layer limiting feature dimension:

$$l_t = \tanh(x_t * k_t + b_t) \tag{1}$$

where l_t represents the output value after convolution, tanh is the activation function, x_t is the input vector, k_t is the weight of the convolutional kernel and b_t is the convolutional kernel bias.

3.3 LONG SHORT TERM METHOD (LSTM)

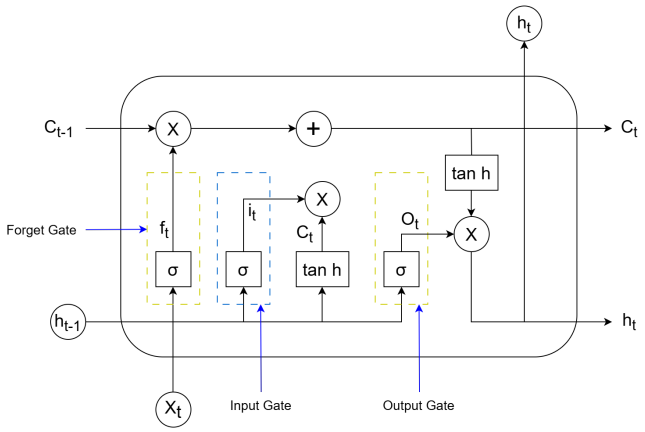


Fig. 4 Long Short Term Memory Cell (LSTM)

LSTM solve problems such as exploding gradient and vanishing gradient of traditional RNN. Unlike conventional RNN layers that can often fail to pull long-range dependencies from sequential data, LSTM incorporates a gating mechanism, enabling it to overshadow retain or filter information, thereby making it incredibly efficient for time-series forecasting.

Originally used for speech recognition and sentiment analysis, LSTM has recently become popular for financial forecasting, particularly stock prediction. Unlike a regular RNN, its architecture contains multiple memory cells, each consists of three critical gates: the forget gate, which decides what information to forget; the input gate, which manages the flow of new information entering the memory; and the output gate, which determines the impact that the stored information will have on the prediction.

The controlled flow of information in LSTM enables the model to learn complex temporal dependencies, which is critical for problems such as stock price prediction. LSTM is inclusive with CNN for feature extraction and forecast accuracy in this study. LSTM: Long Short Term Memory network along with its internal memory cell processing its different operations for learning from previous stock information.

The LSTM calculation process is as follows:

1. The output of the previous time step and the current input are fed into the forget gate, which processes them to determine the retention of past information. The resulting output is calculated as follows:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2}$$

   Where the value range of f_t is $(0, 1)$, W_f is the weight of the forget gate, x_t is the input value in the current time step, and h_{t-1} is the output value from the previous time step.
2. The output value from the previous time step and the input value of the current time step are fed into the input gate. After calculation, the output value of the input gate and the candidate cell state are obtained, as shown in the following formulas:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{4}$$

where the value ranges of i_t is $(0, 1)$, W_i is the weight of input gate, b_i is the bias input gate, W_c is the weight of candidate input gate, and b_c is the bias of the candidate input gate.

3. Update the current cell state:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$

where the value range of C_t is $(0, 1)$.

4. The output h_{t-1} and input x_t are received as input values of the output gate at time t, and the output O_t of the output is obtained as follows:

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{6}$$

where the value range of O_t is $(0, 1)$, W_o is the weight of the output gate, and b_o is the bias of the output gate.

5. the output value of LSTM is obtained by calculating the output of the output gate and the state of the cell, shown in the following formula.

$$h_t = O_t \odot \tanh(C_t) \tag{7}$$

3.4 CNN-LSTM MODEL TRAINING AND TESTING

The CNN-LSTM process of training and prediction is shown in Figure **??**. The main steps are as follows:

1. Input data: Import the stock market data, including multiple relevant features such as opening price, closing price, highest price, and lowest price.
2. Data standardization: Normalize the data using the Min-Max method to ensure uniform scale across features, which helps stabilize and accelerate the training process. This scales all numerical price values to a uniform range between 0 and 1 using the formula:

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{8}$$

3. Data segmentation: Divide the standardized data into two sets: one for training the model and the other for testing its prediction performance.
4. Sequence construction: Generate time-series input sequences using a sliding window approach. Each sequence consists of a fixed number of previous time steps as input and the next step as the target output.
5. Initialize network: Build the CNN-LSTM model architecture and compile the model with Mean Squared Error as loss function and Adam as optimizer.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

where

- n = Total number of observations (data points)
- y_i = Actual (true) value
- $\hat{y}_i$ = Predicted value by the model
- $(y_i - \hat{y}_i)^2$ = Squared error for each prediction

6. CNN layer calculation: The input sequences are passed through the convolutional and pooling layers to extract short-term temporal features from the data.
7. LSTM layer calculation: The feature maps from the CNN layers are processed by the LSTM layer to capture long-term dependencies and temporal patterns.
8. Output layer calculation: The output of the LSTM layer is passed through a dense layer to produce the final prediction for the next time step.
9. Error Correction and Finding Accuracy: The predicted output is compared to the actual target output, and the error is evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared (R^2) and Accuracy, based on Mean Absolute Percentage Error (MAPE).
10. To judge whether the end condition is satisfied: Check whether training has reached the maximum number of epochs or if the error is below a defined threshold. If the condition is not satisfied, proceed to the next step.
11. Error backpropagation: Propagate the error backward through the network to update the weights of all layers, and repeat steps 6 to 11 until the end condition is met.
12. Save the model: Once training is complete, save the trained CNN-LSTM model for use in forecasting.
13. Input data: Load the new input data that needs to be predicted.
14. Data standardization: Normalize the input data using the same scaler used during training to maintain consistency.
15. Prediction: Feed the standardized input data into the trained CNN-LSTM model to obtain the predicted output.
16. Data standardized restore: Transform the predicted values back to their original scale using the inverse transformation of the scaler.
17. Output Results: Output the restored results to complete the forecasting process.

“‘latex

4 EXPERIMENT

To evaluate the effectiveness of the CNN-LSTM model, we compared its performance with five other models: MLP, CNN, RNN, LSTM, and CNN-RNN. All models were trained and tested using the same datasets under identical conditions. The experiments were conducted on a system with an Intel AMD Ryzen 5 7520U with 2.80GHz processor, 16 GB and Windows 11 operating system. The input features included opening price, highest price, lowest price and closing price. Based on these factors, the models were used to predict the each day's closing price after 30 june 2020 for next six months.

Table 2 Raw Data Sample

Date	Opening	High	Low	Close
1/1/2020	12,202.15	12,222.20	12,165.30	12,182.50
1/2/2020	12,198.55	12,289.90	12,195.25	12,282.20
1/3/2020	12,261.10	12,265.60	12,191.35	12,226.50
1/6/2020	12,170.60	12,179.10	11,974.20	11,993.05
1/7/2020	12,079.10	12,152.15	12,005.35	12,052.95
1/8/2020	11,939.10	12,044.95	11,929.60	12,025.35

4.1 DATA

In this experiment, the NIFTY 50 index is used as the dataset. Daily trading data from January 1, 2020, to June 30, 2020 are collected, comprising four key features: opening price, highest price, lowest price, and closing price. The data is divided into two parts: the training set consists of data up to June 30, 2020, while the remaining data is used as the test set.

4.2 MODEL IMPLEMENTATION

To evaluate the forecasting performance of various deep learning models four evaluation metrics are employed: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared (R^2), and Accuracy (derived from Mean Absolute Percentage Error, MAPE). These metrics are computed after training each model on the NIFTY 50 stock index dataset, which includes the features: Open, High, Low, and Close prices.

The metrics are defined as follows:

1. Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

2. Root Mean Square Error (RMSE): Quantifies the square root of the average of squared differences:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

The closer the value of MAE and RMSE to 0, the smaller the error between the predicted value and the higher forecast accuracy.

3. R-squared: Represents the proportion of variance in the actual values that is predictable from the predicted values. Values closer to 1 indicate a better fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{12}$$

where $\bar{y}$ represents the mean of the actual values.

4. Accuracy: This is derived from Mean Absolute Percentage Error and expresses how close predictions are to actual values.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{13}$$

$$Accuracy = 100 - MAPE \tag{14}$$

4.3 IMPLEMINTATION OF CNN-LSTM

The CNN-LSTM model used in this experiment is configured as described in Table **??**. According to the parameter settings of the CNN-LSTM network, the structure is designed as follows: the input training dataset is a three-dimensional shape data vector $(None, 15, 4)$, where 15 represents the window size of the time step and 4 represents the number of input features (Open, Close, High, Low).

Table 3 CNN-LSTM Model Hyperparameters

Parameters	**Values**
Convolutional Layer Filter	64
Convolutional Layer Kernel Size	2
Convolutional Layer Activation Function	ReLU
Convolutional Layer Padding	valid
Pooling Layer Pooling Size	2
Pooling Layer Padding	valid
Number of Hidden Units in LSTM Layer	64
LSTM Layer Activation Function	tanh
Time Step	15
Input Features	4
Batch Size	8
Learning Rate	0.001
Optimizer	Adam
Loss Function	Mean Squared Error
Epochs	50

First, the input passes through a 1D convolutional layer with 64 filters and a kernel size of 2, activated using the ReLU function. This layer extracts local temporal features from the sequential data and produces an output shape of $(None, 14, 64)$ due to the convolution operation.

Next, the output of the convolutional layer is passed through a MaxPooling1D layer with a pool size of 2, reducing the temporal dimension and resulting in an output shape of $(None, 7, 64)$. This pooling operation helps reduce sample and retain the most important features.

This is followed by an LSTM layer with 64 hidden units, which captures long-term dependencies and sequential relationships in the data. The LSTM layer outputs a vector of shape $(None, 64)$.

To avoid overfitting, a Dropout layer with a rate of 0.3 is applied after the LSTM layer.

Finally, a Dense layer with 1 output unit is used to predict the one target value, corresponding to the normalized price components of the stock for the next time step.

The CNN-LSTM model is trained using the Adam optimizer and the mean squared error loss function.

The model structure is conceptually illustrated in Figure **??**.

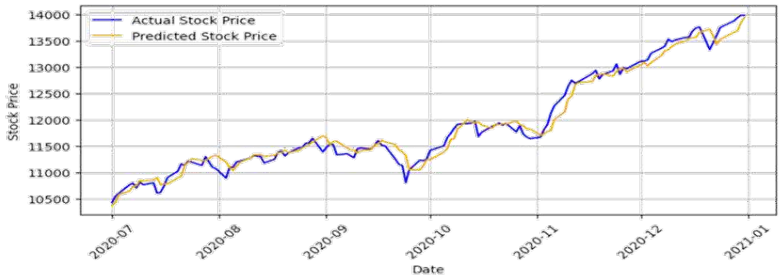


Fig. 5 Comparison of the predicted stock price and actual stock price for MLP

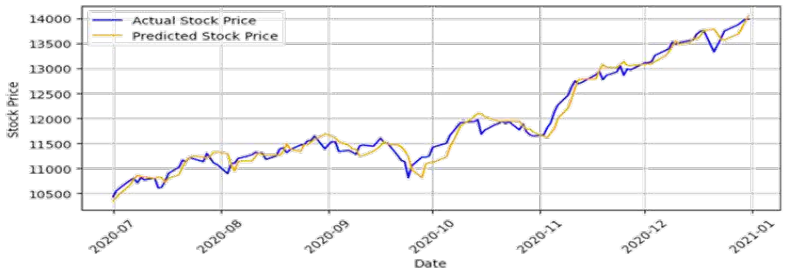


Fig. 6 Comparison of the predicted stock price and actual stock price for CNN

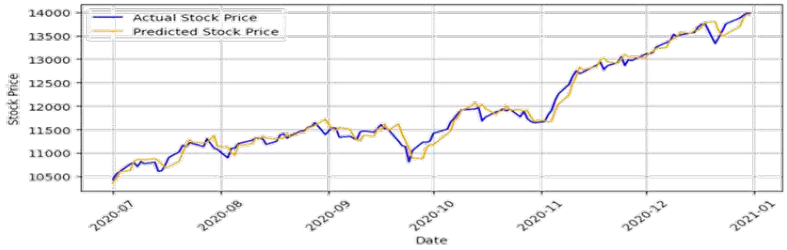


Fig. 7 Comparison of the predicted stock price and actual stock price for RNN

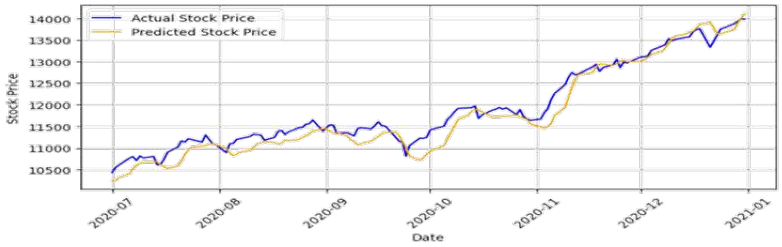


Fig. 8 Comparison of the predicted stock price and actual stock price for LSTM

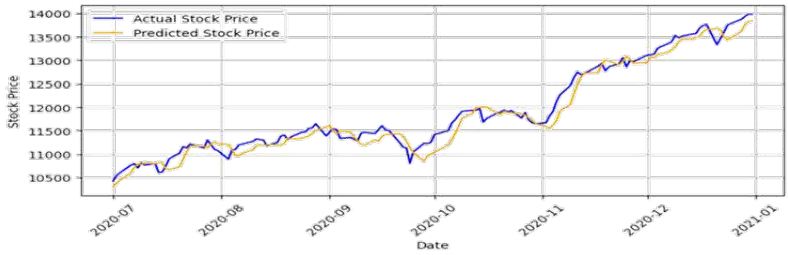


Fig. 9 Comparison of the predicted stock price and actual stock price for CNN-RNN

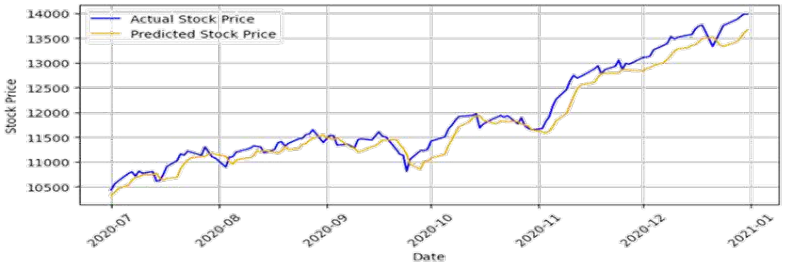


Fig. 10 Comparison of the predicted stock price and actual stock price for CNN-LSTM

Table 4 Comparison Table

Model	MAE	RMSE	R^2	Accuracy (%)
MLP	113.99	148.91	0.98	99.04
CNN	124.40	159.67	0.97	98.94
RNN	105.98	138.91	0.98	99.10
LSTM	196.58	242.77	0.93	98.32
CNN-RNN	148.34	180.88	0.96	98.75
CNN-LSTM	184.18	219.20	0.95	98.48

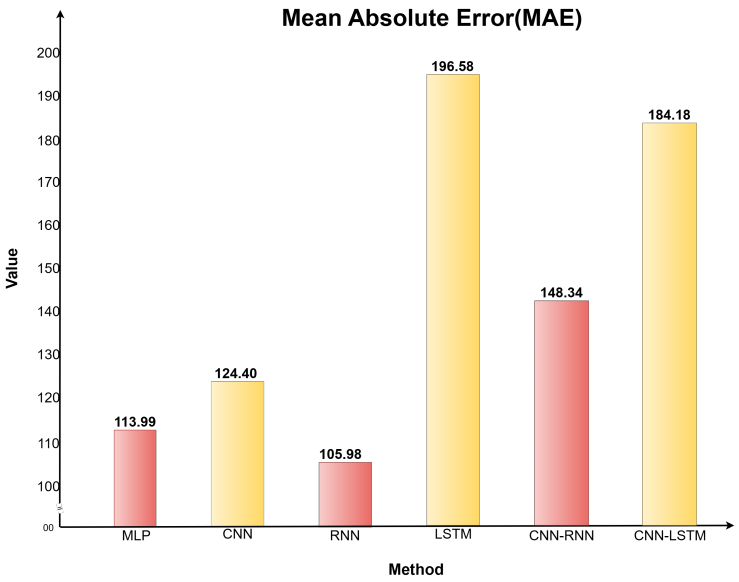


Fig. 11 The result of MAE comparison among different methods

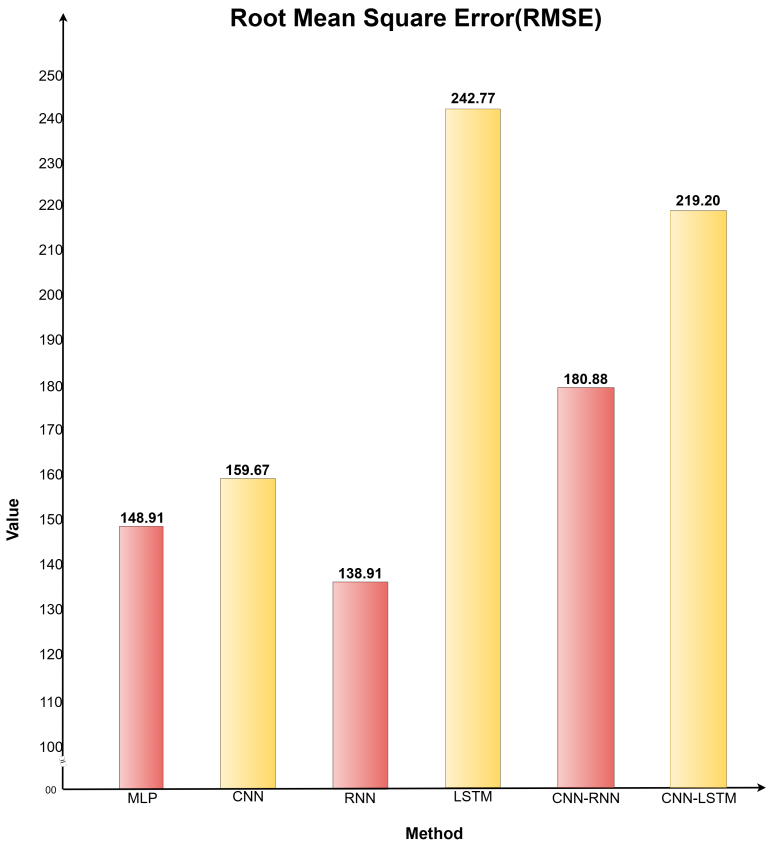


Fig. 12 The result of RMSE comparison among different methods

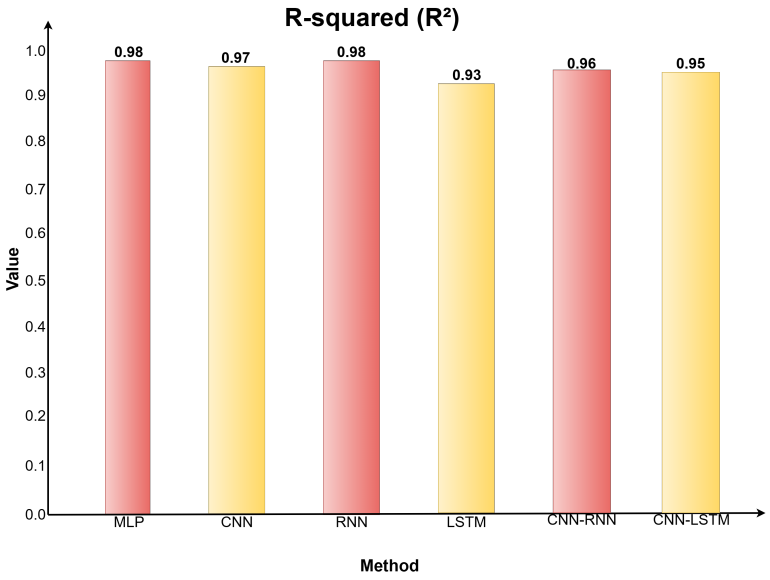


Fig. 13 The result of R^2 comparison among different methods

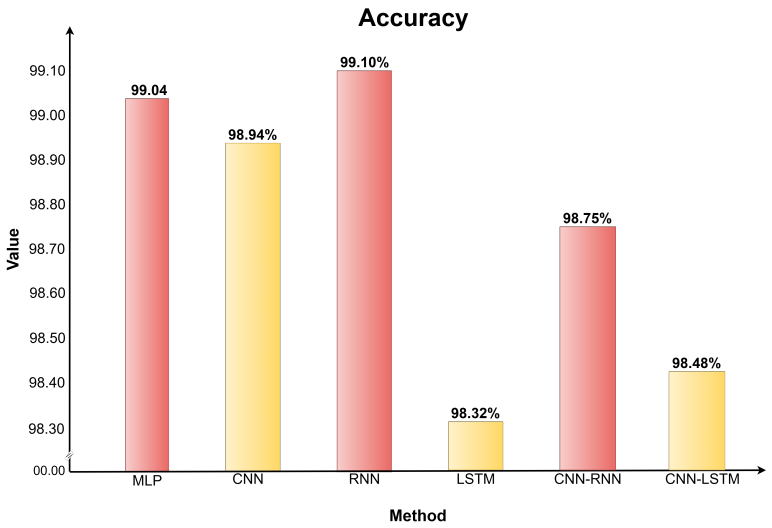


Fig. 14 The result of Accuracy comparison among different methods

5 Result

After using the processed training set data to train the MLP, CNN, RNN, LSTM, CNN-RNN, and CNN-LSTM models respectively, the trained models were used to predict the test set data. The real values were compared with the predicted values, as shown in Figures 5–10.

In Figures 5–10, among the six forecasting methods, the degree of broken-line alignment between the real and predicted values is highest for the RNN and MLP models, followed by CNN, while CNN-RNN, CNN-LSTM and LSTM exhibit weaker fitting accuracy. Although LSTM shows strong alignment in some segments, the reported error and accuracy metrics indicate that RNN outperforms all others. The CNN-LSTM and LSTM models demonstrate the weakest performance in this scenario.

Based on the predicted and actual values of each method, the evaluation indices were calculated and the comparison results are summarized in Table 3 and Figures 11–14.

From the updated Table, the LSTM model has the highest MAE (196.58) and RMSE (242.77), with an R^2 of 0.93 and an accuracy of 98.32%, indicating the weakest performance among all models. performance among all models. The CNN-LSTM model performs slightly better than LSTM with a MAE of 184.18, RMSE of 219.20, R^2 of 0.95, and accuracy of 98.48%. The CNN-RNN model achieves improved metrics with MAE of 148.34, RMSE of 180.88, R^2 of 0.96, and accuracy of 98.75

Among all models, the RNN model reports the best overall results, with the lowest MAE (105.98), lowest RMSE (138.91), highest R^2 (0.98), and highest accuracy (99.10%). The MLP model follows closely with a MAE of 113.99, RMSE of 148.91, R^2 of 0.98, and precision of 99.04%.

Comparing RNN with LSTM, RNN shows a significant performance advantage. MAE decreased from 196.58 to 105.98 (a reduction of 46.1%), RMSE dropped from 242.77 to 138.91 (a reduction of 42.8%), and R^2 improved from 0.93 to 0.98, indicating a much stronger ability to capture patterns in stock price series.

The CNN model yields a MAE of 124.40, RMSE of 159.67, R^2 of 0.97, and precision of 98.94%, performing better than hybrid models such as CNN-LSTM and LSTM but not exceeding MLP or RNN.

These results indicate that the RNN model outperforms all others in terms of error minimization and prediction accuracy, making it best suited for short-term stock price forecasting in this study. The MLP and CNN models also perform well, while LSTM-based models show relatively lower accuracy and higher error. Therefore, RNN and MLP architectures provide more accurate and reliable predictions for stock market closing prices, offering useful references for investor decision-making.

6 Conclusion

This study focuses on predicting the daily closing prices of the NIFTY 50 index for six months after June 30, 2020, using deep learning techniques. The study uses historical stock data, including opening, highest, lowest, and closing prices. These values are normalized and arranged into time-series sequences before being used for training. Six models, namely MLP, CNN, RNN, LSTM, CNN-RNN, and CNN-LSTM, are developed and tested under the same experimental conditions to compare their performance.

CNN layers are used to extract useful patterns from historical stock data, while recurrent layers learn how stock prices change over time. The models are evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), the coefficient of determination (R^2), and predictive accuracy. Among the tested models, RNN gives the best results, with an MAE of 105.98, RMSE of 138.91, R^2 of 0.98, and accuracy of 99.10%. These results indicate that RNN performs well on the dataset used in this study.

The MLP model also achieves strong results, with an accuracy of 99.04%, MAE of 113.99, RMSE of 148.91, and R^2 of 0.98. The CNN model records an MAE of 124.40, RMSE of 159.67, R^2 of 0.97, and accuracy of 98.94%. The hybrid models perform slightly less well, with CNN-RNN achieving 98.75% accuracy and CNN-LSTM achieving 98.48%. LSTM records the lowest performance among the tested models, with an MAE of 196.58, RMSE of 242.77, R^2 of 0.93, and accuracy of 98.32%.

This study compares different deep learning models for stock price prediction and provides information that may be useful for data-driven financial analysis. However, the models rely only on numerical stock market data and do not consider other

factors that can influence prices, such as financial news, market sentiment, or economic policies. Future research could include these factors to examine whether they improve prediction performance and make the models more useful in real-world market conditions.

7 REFERENCES

1. R. Singh, and S. Srivastava, "Stock prediction using deep learning", *Multimedia Tools and Applications*, vol. 76, pp. 18569–18584, 2017.
2. S. Mehtab, and J. Sen, "A robust predictive model for stock price prediction using deep learning and natural language processing", *Proceedings of the 7th International Conference on Business Analytics and Intelligence (BAICONF'19)*, Bangalore, India, 5–7 December, DOI: 10.2139/ssrn.3502624, 2019.
3. M. Agrawal, A. U. Khan and P. K. Shukla, "Stock indices price prediction based on technical indicators using deep learning model," *International Journal on Emerging Technologies*, vol. 10, no. 2, pp. 186–194, 2019.
4. S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy and S. De, "Stock market prediction using deep learning algorithms", *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 82–94, 2023.
5. M. Agrawal, P. K. Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock Prediction Based on Technical Indicators Using Deep Learning Model", *Computers, Materials & Continua*, vol. 70, no. 1, 2022.
6. B. K. Chan, O. V. Johnson, X. Chew, K. W. Khaw, M. H. Lee, and A. Alnoor, "Proposed Bayesian Optimization Based LSTM-CNN Model for Stock Trend Prediction," *Computing and Informatics*, vol. 43, pp. 38–63, 2024, doi: 10.31577/cai_2024_1_38.
7. W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-Based Model to Forecast Stock Prices," *Complexity*, vol. 2020, Article ID 6622927, pp. 1–10, Nov. 2020, doi: 10.1155/2020/6622927.
8. A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market." *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.
9. C. Jung and R. Boyd, "Forecasting UK stock prices," *Applied Financial Economics*, vol. 6, no. 3, pp. 279–286, 1996.
10. A. Adebiyi, A. Adewumi, and C. Ayo, "Stock price prediction using the ARIMA LA model." In *Proceedings of the 2014 UKSim AMSS 16th International Conference on Computer Modelling and Simulation*, IEEE, Cambridge, UK, March 2014.
11. J. Shah, D. Vaidya, and M. Shah, "A comprehensive review on multiple hybrid deep learning approaches for stock prediction," *Intelligent Systems Applications*, vol. 16, p. 200111, 2022.
12. K. Sudhakar and S. Naganjaneyulu, "An optimised CNN-stacked LSTM neural network model for predicting stock market time-series data," *International Journal of Computational Economics and Econometrics*, vol. 15, no. 1–2, pp. 196–224, 2025.
13. G. Singh, L. Rani, P. K. Sarangi, K. Sharma, and A. K. Sahoo, "CNN-RNN Based Hybrid Deep Learning Model for Predicting Fluctuations in the Stock Market," in

Proceedings of the 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 417–422, 2023.
14. J. B. Pandya, "Deep Learning Approach for Stock Market Trend Prediction and Pattern Finding," Gujarat Technological University, Ahmedabad, 2024.
15. S. H. Jafar, S. Akhtar, H. El-Chaarani, P. A. Khan, and R. Binsaddig, "Forecasting of NIFTY 50 index price by using backward elimination with an LSTM model," *Journal of Risk and Financial Management*, vol. 16, no. 10, p. 423, 2023.
16. A. Dhaliwal, N. Polatidis, and E. Pimenidis, "A novel LSTM-CNN architecture to forecast stock prices," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 466–477, 2022.
17. A. N. Sadon, S. Ismail, A. Khamis, and M. U. Tariq, "Heteroscedasticity effects as component to future stock market predictions using RNN-based models," *PLOS ONE*, vol. 19, no. 5, e0297641, 2024.
18. R. Sable, S. Goel, and P. Chatterjee, "Techniques for Stock Market Prediction: A Review," *International Journal of Recent Innovation Trends in Computing and Communication*, vol. 11, no. 5s, 2023.
19. W. Chen, W. Hussain, F. Cauterucci, and X. Zhang, "Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models," *CMES-Computer Modeling in Engineering & Sciences*, 2023.
20. K. Alkhatib, H. Khazaleh, H. A. Alkhazaleh, A. R. Alsoud, and L. Abualigah, "A new stock price forecasting method using active deep learning approach," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 8, no. 2, p. 96, 2022.
21. P. Chhajer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction," *Decision Analytics Journal*, vol. 2, p. 100015, 2022.
22. H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock decomposition," *Expert Systems with Applications*, vol. 169, p. 114332, 2021.
23. R. Kamble and A. Nayak, "A Hybrid Deep Learning Improved Method for Share Price Prediction," *AIDE-2023 and PCES-2023*, p. 360, 2023.
24. C. K.-S. Leung, R. K. MacKinnon and Y. Wang, "A machine learning approach for stock price prediction," *Proceedings of the 18th International Database Engineering & Applications Symposium*, 2014, pp. 274–277.
25. K. Khare, O. Darekar, P. Gupta and V. Z. Attar, "Short term stock price prediction using deep learning," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bengaluru, India, 2017, pp. 482–486.
26. M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, 2016, pp. 322–327.
27. C. F. Tsai and S. P. Wang, "Stock price forecasting by hybrid machine learning techniques," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, no. 755, pp. 60, 2009.
28. Shunrong Shen, Haomiao Jiang, and Tongda Zhang, "Stock market forecasting using machine learning algorithms," Department of Electrical Engineering, Stanford University, Stanford, CA, pp. 1–5, 2012.