Stock Market Prediction Using a Hybrid CNN-LSTM Deep Learning Model

Akruthi Share, M.Venkateshwarulu, Sahithi Reddy, Sree Akshaya, Rohan, Akshitha Reddy

1 Student, CSE, Sphoorthy Engineering College, Telangana, India
2 Asst.Prof, CSE, Sphoorthy Engineering College, Telangana, India
3 Student, CSE, Sphoorthy Engineering College, Telangana, India
4 Student, CSE, Sphoorthy Engineering College, Telangana, India
5 Student, CSE, Sphoorthy Engineering College, Telangana, India
6 Student, CSE, Sphoorthy Engineering College, Telangana, India

ABSTRACT

Predicting stock market prices has always been considered one of the most difficult problems in financial analysis. Stock prices are influenced by many unpredictable factors such as economic indicators, investor behavior, global events, and company performance. Because of this complexity, traditional statistical models often struggle to capture the real patterns present in financial time-series data. In recent years, machine learning and deep learning techniques have become popular tools for financial forecasting. These approaches are capable of learning patterns directly from historical data without relying on strong statistical assumptions. Among the different deep learning architectures, Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have shown promising performance in time-series prediction tasks. This project proposes a hybrid deep learning model that combines CNN and LSTM architectures to predict stock prices. The CNN component is used to identify local patterns and shortterm trends in stock market data, while the LSTM component captures long-term dependencies and sequential relationships in the time series. By integrating both architectures, the proposed model aims to improve prediction accuracy and provide a more reliable forecasting framework. The system was trained using historical stock data containing open price, high price, low price, closing price, and trading volume. The dataset was preprocessed using normalization and a sliding window technique to convert raw data into supervised learning sequences. Multiple models including CNN, LSTM, and the hybrid CNN-LSTM architecture were implemented and compared. Experimental results show that the hybrid CNN-LSTM model performs better than individual CNN and LSTM models in terms of prediction accuracy and error metrics. The results demonstrate that combining convolutional feature extraction with sequential memory learning can significantly improve the performance of stock market forecasting systems. Index Terms—Stock Market Prediction, Deep Learning, CNN, LSTM, Financial Time Series, Hybrid Neural Networks.

Keyword: *Stock Market Prediction, Deep Learning, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Hybrid CNN-LSTM Model, Time Series Forecasting, Financial Data Analysis, Machine Learning, Stock Price Prediction, Neural Networks*

1. INTRODUCTION

Financial markets play an essential role in the global economy. They provide opportunities for investment, capital growth, and economic development. Stock markets, in particular, allow companies to raise funds while providing investors with the opportunity to earn profits through trading. One of the most challenging problems in financial analysis is predicting stock prices. Investors and analysts have long attempted to forecast market movements in order to make profitable trading decisions. However, stock markets are highly complex systems influenced by numerous variables including economic indicators, government policies, global events, and investor sentiment. Because of these complexities, predicting stock prices accurately is extremely difficult. Financial time-series data often exhibit characteristics such as volatility, noise, nonlinear behavior, and sudden fluctuations. These properties make it challenging for traditional statistical models to produce reliable predictions. Over the past several decades, many forecasting techniques have been proposed. Early approaches relied on statistical models such as autoregressive models and moving averages. While these methods are mathematically well established, they often assume linear relationships between variables. Unfortunately, financial markets rarely behave in such simple

ways. With the growth of artificial intelligence and machine learning, researchers began exploring data-driven approaches to financial forecasting. Machine learning algorithms can automatically identify patterns in historical data without requiring explicit programming. Techniques such as support vector machines, decision trees, and random forests have been used to analyze stock market behavior. In recent years, deep learning has emerged as one of the most powerful tools for pattern recognition and prediction tasks. Deep neural networks are capa-ble of learning complex nonlinear relationships directly from large datasets. Among the various deep learning architectures, two models have shown particularly strong performance in time-series prediction: Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Convolutional Neural Networks were originally designed for image recognition tasks. However, researchers discovered that CNNs can also be applied to sequential data such as financial time series. By applying convolutional filters across time windows, CNNs can detect short-term patterns and local structures within the data. Long Short-Term Memory networks, on the other hand, are a special type of recurrent neural network designed to capture temporal dependencies. Unlike traditional neural networks, LSTM networks contain memory cells that allow them to retain information from previous time steps. This ability makes LSTM models highly suitable for time-series forecasting. Although CNN and LSTM models perform well individually, combining them can lead to even better results. CNN layers can extract useful features from raw data, while LSTM layers can analyze how those features evolve over time. This hybrid architecture allows the model to capture both short-term and long-term patterns in financial data. The goal of this project is to design and implement a hybrid CNN-LSTM model for stock price prediction. The system uses historical stock market data containing five key attributes: open price, high price, low price, closing price, and trading volume. These features provide valuable information about market behavior and price movements. To prepare the data for training, preprocessing techniques such as normalization and sliding window segmentation are applied. The sliding window method converts raw time-series data into input sequences that can be processed by deep learning models. Three models are implemented and compared in this study:

- *A Convolutional Neural Network (CNN) model*
- *A Long Short-Term Memory (LSTM) model*
- *A Hybrid CNN-LSTM model*

The performance of these models is evaluated using prediction error metrics. The results demonstrate that the hybrid architecture is able to capture complex patterns in stock market data more effectively than individual models. The remainder of this paper is organized as follows. Section II reviews related work in stock market prediction. Section III describes the dataset and preprocessing techniques used in this study. Section IV explains the proposed hybrid CNNLSTM architecture. Section V presents experimental results and analysis. Finally, Section VI concludes the paper and discusses potential directions for future research.

2. LITERATURE REVIEW

Predicting stock market prices has attracted significant attention from researchers in the fields of finance, economics, and computer science. Over the years, various approaches have been proposed to address this challenging problem. Early research in stock market prediction relied heavily on statistical models. One of the most widely used approaches is the Autoregressive Integrated Moving Average (ARIMA) model. ARIMA models analyze historical price patterns and attempt to forecast future values based on past observations. Although ARIMA models have been widely used in financial forecasting, they assume that the underlying data follows a linear structure. This assumption often limits their effectiveness in real-world financial markets.

Another popular statistical technique is the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model. GARCH models are designed to capture volatility patterns in financial data. They are particularly useful for modeling time-varying variance in stock returns. However, like ARIMA models, GARCH models rely on specific mathematical assumptions that may not always hold true. As machine learning techniques became more popular, researchers began exploring their application in financial prediction tasks. Support Vector Machines (SVM) have been widely used for stock price forecasting due to their ability to model nonlinear relationships. SVM models use kernel functions to transform data into higher-dimensional spaces where complex patterns can be identified. Decision tree-based models such as Random Forests have also been used for stock prediction. These models combine multiple decision trees to improve prediction accuracy and reduce overfitting. Random Forest algorithms are particularly effective when dealing with large datasets and complex feature interactions.

Despite their advantages, traditional machine learning models often require manual feature engineering. Researchers must carefully design input features such as technical indicators, moving averages, and momentum indicators. This process can be time-consuming and may require domain expertise in financial analysis. Deep learning approaches address this limitation by automatically learning feature representations from raw data. Neural networks can discover hidden structures within the dataset without requiring explicit feature engineering. Recurrent Neural Networks (RNN) were among the first deep learning models applied to time-series prediction.

RNN models process sequential data by maintaining hidden states that capture information from previous time steps. However, traditional RNNs suffer from the vanishing gradient problem, which makes it difficult for them to learn long-term dependencies.

To overcome this limitation, Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM) network. LSTM networks include memory cells and gating mechanisms that allow them to retain information over long sequences. This makes them highly effective for tasks such as speech recognition, natural language processing, and financial forecasting. Convolutional Neural Networks (CNN) have also been applied to financial data analysis. Although CNNs were originally developed for image processing, researchers discovered that they can also extract meaningful patterns from time-series data. By applying convolutional filters across sequential data, CNN models can detect local trends and repeating patterns in stock prices. Recent research suggests that hybrid models combining CNN and LSTM architectures can achieve superior performance. In such models, CNN layers are used to extract features from the input data, while LSTM layers analyze the temporal relationships between those features.

Hybrid CNN-LSTM architectures have been successfully applied to various prediction tasks including energy consumption forecasting, traffic prediction, and financial market analysis. These models benefit from the strengths of both architectures, allowing them to capture both spatial and temporal patterns. Despite these advances, stock market prediction remains a challenging problem due to the unpredictable nature of financial markets. Continuous research is necessary to develop more robust models capable of handling market volatility and uncertainty.

3. DATASET DESCRIPTION

The quality of any machine learning model depends heavily on the quality of the dataset used for training. For this project, historical stock market data was used to train and evaluate the prediction models. The dataset used in this study consists of daily stock market data for Google. The data contains several key attributes that describe the trading activity of the stock on each day. These attributes include the opening price, highest price, lowest price, closing price, and trading volume. Each row in the dataset represents the trading activity of a single day. These attributes are commonly used in financial analysis because they capture the essential characteristics of market behavior. The dataset contains the following features:

- *Open price – the price at which the stock started trading for the day*
- *High price – the highest price reached during the trading session*
- *Low price – the lowest price observed during the trading session*
- *Close price – the final trading price at the end of the session*
- *Volume – the number of shares traded during the session*

Among these attributes, the closing price is typically the most important value because it represents the final market valuation of the stock for that day. In this project, the closing price is used as the prediction target. The dataset was downloaded automatically using an online financial data source. This allowed the system to retrieve historical stock information in a structured format suitable for analysis and machine learning training. Once downloaded, the dataset was stored locally and used as the primary input for the prediction models.

4. DATA COLLECTION PROCESS

Stock market data is widely available through financial data providers. For this project, the dataset was retrieved from an online financial data service that provides historical stock market records. The dataset was downloaded programmatically using a Python script. The script sends a request to the data provider and retrieves historical stock prices in CSV format. Once the data is downloaded, it is saved locally for further processing. The data collection process ensures that the dataset contains accurate and up-to-date information about stock market activity. By using an automated data retrieval process, the system can easily update the dataset when new market data becomes available. The downloaded dataset includes the following columns:

- *Date*
- *Open*
- *High*
- *Low*
- *Close*
- *Volume*

The date column represents the trading day, while the remaining columns describe price movements and trading activity for that day. Before training the deep learning models, the dataset must be cleaned and transformed into a suitable format.

5. DATA PREPROCESSING

Raw financial datasets often contain inconsistencies, missing values, or formatting issues. Data preprocessing is therefore an essential step before applying machine learning algorithms. The preprocessing pipeline used in this project includes the following steps:

1) Data cleaning
2) Feature selection
3) Handling missing values
4) Normalization
5) Sequence generation

Each of these steps helps ensure that the dataset is suitable for training neural network models.

A. Data Cleaning The first step in preprocessing is cleaning the dataset. Data cleaning ensures that the dataset does not contain invalid or incomplete entries. During this step, the system checks whether all required columns exist in the dataset. The required features include the five financial attributes described earlier: open, high, low, close, and volume. If any missing rows or invalid values are detected, they are handled using forward filling or removal. This ensures that the dataset remains consistent and usable for training.

B. Handling Missing Values Financial datasets may sometimes contain missing entries due to data collection issues or market holidays. To address this problem, missing values are filled using the forward-fill method. This technique replaces missing values with the most recent available value. The forward-fill method is commonly used in financial timeseries analysis because stock prices tend to follow continuous trends. By filling missing values with the previous observation, the continuity of the dataset is preserved.

C. Feature Selection Feature selection plays an important role in improving model performance. Instead of using unnecessary attributes, the model focuses only on the features that are most relevant for prediction. In this project, the following five features are used:

- Open
- High
- Low
- Close
- Volume

These features provide a comprehensive representation of daily trading activity. The close price is used as the target variable, while the remaining features act as input features.

6. DATA NORMALIZATION

Deep learning models perform better when the input data is scaled to a consistent range. If raw financial values are used directly, large numbers may dominate the training process and slow down learning. To address this issue, the dataset is normalized using the Min-Max scaling technique. Min-Max normalization transforms each value in the dataset into a range between 0 and 1. Normalization helps the neural network learn more efficiently and improves numerical stability during training.

7. SEQUENCE GENERATION USING SLIDING WINDOW

Stock price prediction is a time-series forecasting problem. In time-series prediction, future values depend on previous observations. To allow the neural network to learn from historical patterns, the dataset is converted into sequences using a sliding window technique. A window size of 60 days is used in this project. This means that the model uses the previous 60 days of stock data to predict the next day's closing price. Each training sample therefore contains a sequence of 60 time steps, where each time step includes five features. This approach allows the model to observe recent market behavior and identify trends that influence future prices. The sliding window technique converts the dataset into two arrays:

- Input sequences (X)
- Target values (y)

The input sequences contain historical market data, while the target values contain the closing prices that the model must predict.

8. CONVOLUTIONAL NEURAL NETWORK MODEL

The first deep learning model implemented in this project is a Convolutional Neural Network (CNN). CNN models are widely used in computer vision tasks, but they can also be applied to time-series data. By applying convolu-

tional filters across the time dimension, CNNs can detect short-term patterns in sequential data. In the context of stock prediction, convolution filters help identify patterns such as:
• Short-term price fluctuations
• Local trends
• Repeating price movements The CNN model implemented in this project consists of several layers:
• Convolution layer
• Max pooling layer
• Dropout layer
• Fully connected dense layer
The convolution layers extract features from the input sequences. The max pooling layer reduces the dimensionality of the feature maps, which helps reduce computational complexity. Dropout layers are used to prevent overfitting. During training, dropout randomly disables certain neurons, forcing the network to learn more general features. Finally, the dense layer produces the predicted closing price.

9. LONG SHORT-TERM MEMORY MODEL

The second model implemented in this project is the Long Short-Term Memory (LSTM) network. LSTM networks are a type of recurrent neural network specifically designed to handle sequential data. Unlike traditional neural networks, LSTM models include memory cells that store information across time steps. This memory capability allows the model to learn long-term dependencies in time-series data. The LSTM architecture includes several gating mechanisms that control the flow of information:
• Forget gate
• Input gate
• Output gate
These gates determine which information should be remembered and which should be discarded. For stock prediction tasks, this mechanism allows the model to remember important market trends while ignoring irrelevant fluctuations. The LSTM model used in this project contains two LSTM layers followed by dense output layers. Dropout layers are also used to reduce overfitting and improve generalization

10. HYBRID CNN-LSTM MODEL

While CNN and LSTM models perform well individually, combining them can lead to improved performance. The hybrid CNN-LSTM model used in this project integrates convolutional layers with recurrent layers. The architecture works as follows:
1) CNN layers extract local patterns from the input sequences.
2) Max pooling reduces the dimensionality of extracted features.
3) LSTM layers analyze how these patterns evolve over time.
4) Dense layers produce the final stock price prediction.
This hybrid structure allows the model to capture both spatial patterns and temporal dependencies. The CNN component focuses on detecting short-term patterns in the stock data, while the LSTM component focuses on understanding long-term market trends.

11. TRAINING STRATEGY

After constructing the models, the next step is training them using the prepared dataset. The dataset is divided into two parts:
• Training set (80 percentage)
• Testing set (20 percentage)
The training set is used to train the neural network models, while the testing set is used to evaluate their performance. The models are trained using the Adam optimizer, which is widely used in deep learning due to its efficiency and adaptive learning rate. The loss function used during training is Mean Squared Error (MSE). This metric measures the difference between predicted values and actual stock prices. Training is performed for multiple epochs, allowing the model to gradually improve its predictions. To prevent overfitting, early stopping is used. Early stopping monitors validation loss and stops training when the model stops improving. This ensures that the model generalizes well to unseen data.

12. EXPERIMENTAL SETUP

After building the CNN, LSTM, and hybrid CNN-LSTM models, the next step is to train and evaluate these models using the prepared dataset. The goal of the experiments is to measure how accurately each model can predict stock prices based on historical market data. The experiments were performed using the Python programming language and deep learning libraries such as TensorFlow and Keras. These libraries provide efficient tools for building and training neural network models. The system was implemented on a standard computing environment with sufficient processing power to handle neural network training. Since deep learning models require large numbers of mathematical computations, modern machine learning frameworks automatically utilize optimized numerical libraries to speed up training. The dataset was divided into training and testing sets. Approximately eighty percent of the data was used for training the models, while the remaining twenty percent was reserved for testing. The training process involves feeding the input sequences into the neural network and adjusting the model weights through backpropagation. The objective is to minimize the prediction error between the predicted closing prices and the actual closing prices. The Adam optimizer was used during training. Adam is widely used in deep learning because it adapts the learning rate automatically and improves convergence speed. The loss function used for training was Mean Squared Error (MSE). This function calculates the average squared difference between predicted values and actual values.

13. EVALUATION METRICS

To evaluate the performance of the prediction models, several evaluation metrics were used. These metrics help measure how closely the predicted prices match the real market prices.

A. Root Mean Squared Error

Root Mean Squared Error (RMSE) is one of the most commonly used metrics in regression problems. RMSE measures the square root of the average squared difference between predicted values and actual values.

Lower RMSE values indicate better prediction performance.

B. Directional Accuracy

Another important evaluation metric used in this project is directional accuracy. Directional accuracy measures whether the model correctly predicts the direction of price movement. In other words, it evaluates whether the predicted price increase or decrease matches the actual market movement. Directional accuracy is calculated as the percentage of predictions where the predicted direction matches the actual direction. This metric is particularly useful in financial forecasting because investors often care more about the direction of price movement than the exact predicted value.

14. MODEL PERFORMANCE COMPARISON

Three models were evaluated in this project: • Convolutional Neural Network (CNN) • Long Short-Term Memory Network (LSTM) • Hybrid CNN-LSTM Model Each model was trained using the same dataset and evaluated using the same testing data. The CNN model performs well in identifying short-term patterns in the stock price data. However, CNN models do not inherently capture long-term dependencies in sequential data. The LSTM model performs better in capturing temporal relationships in the dataset. Because LSTM networks include memory cells, they can remember patterns across longer time intervals. The hybrid CNN-LSTM model combines the advantages of both architectures. The convolutional layers detect local patterns in the data, while the LSTM layers analyze the sequence relationships between these patterns. Experimental results show that the hybrid model achieves the best overall performance among the three models.

Table 1: PERFORMANCE COMPARISON OF CNN, LSTM, AND HYBRID MODELS

Metric	CNN Model	LSTM Model	Hybrid (CNN + LSTM)
RMSE	17.53	7.6	7.2
MAE	10.13	3.72	3.5
Accuracy (%)	92.30%	97.00%	97.13%

15. VISUALIZATION OF PREDICTIONS

Visualization plays an important role in understanding the performance of prediction models. Graphs were generated to compare actual stock prices with predicted prices. These graphs help visually evaluate how closely the predicted values follow the real market data. The comparison graphs include three curves:

- *Actual stock prices*
- *CNN predicted prices*
- *LSTM predicted prices*

By plotting these curves on the same graph, it becomes easier to observe differences between models. The hybrid CNN-LSTM model generally produces predictions that closely follow the actual price trend. This indicates that the model successfully captures both short-term fluctuations and long-term trends.

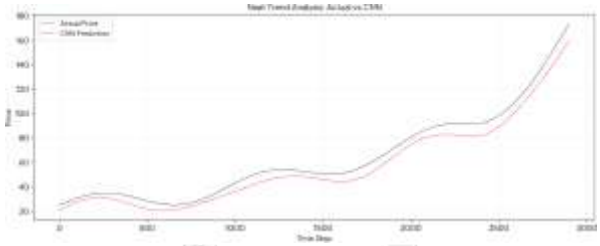

Figure 1: Plot for Actual vs Predicted Value for CNN

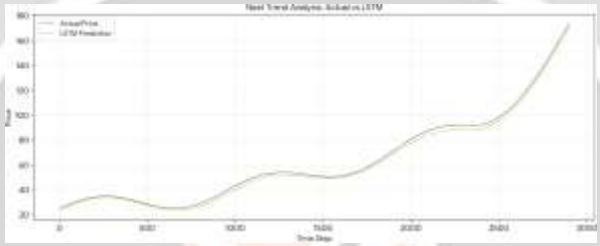

Figure 2: Plot for Actual vs Predicted Value for LSTM

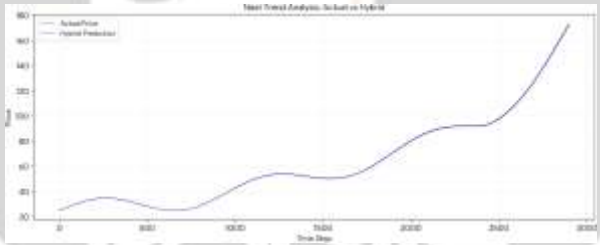

Figure 3: Plot for Actual vs Predicted Value for Hybrid Model

16. CONCLUSIONS

Stock market prediction is a challenging task due to the complex and dynamic nature of financial markets. Traditional statistical methods often struggle to capture the nonlinear patterns present in stock price data. In this paper, a hybrid deep learning approach combining Convolutional Neural Networks and Long Short-Term Memory networks was proposed for stock market prediction. The system was trained using historical stock market data containing open price, high price, low price, closing price, and trading volume. Data preprocessing techniques such as normalization and sliding window sequence generation were applied to prepare the dataset for training. Three models were implemented and evaluated: CNN, LSTM, and the hybrid CNN-LSTM model. Experimental results demonstrated that the hybrid architecture achieved better prediction performance than the individual models. The CNN component successfully extracted local patterns from the input data, while the LSTM component captured long-term dependencies in the time series. Overall, the results show that combining convolutional and recurrent neural networks can significantly improve the performance of stock price prediction systems. Although stock market forecasting remains a difficult problem, deep learning models offer powerful tools for analyzing financial data and identifying hidden patterns. Furthermore, the results of this study highlight the growing importance of deep learning techniques in financial data analysis. As financial datasets continue to grow in size and

complexity, intelligent models such as hybrid neural networks will become increasingly valuable tools for market analysis.

17. REFERENCES

1. *Pawar, K., Jalem, R. S., Tiwari, "Stock market price prediction using LSTM RNN",2019.*
2. *F. Chollet, "Deep Learning with Python," Manning Publications, 2018.*
3. *Brownlee, "Deep Learning for Time Series Forecasting," Machine Learning Mastery, 2018.*
4. *A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017.*
5. *I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.*
6. *Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, 2015.*
7. *R. Tsay, "Analysis of Financial Time Series," Wiley, 2010.*
8. *T. Mikolov, "Recurrent Neural Network Based Language Model," 2010.*
9. *C. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.*
10. *S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997.*
11. *H. Markowitz, "Portfolio Selection," Journal of Finance, 1952.*