Stock Trend Prediction using LSTM and Streamlit

Amitesh Srinivas
Department of Computer Science and Engineering
RV College of Engineering
Bengaluru, India
amiteshs.cs22@rvce.edu.in

Aditya G S
Department of Computer Science and Engineering
RV College of Engineering
Bengaluru, India
adityags.cs22@rvce.edu.in

Tarun HS
Department of Computer Science and Engineering (Data Science)
RV College of Engineering
Bengaluru, India
tarunhs.cd22@rvce.edu.in

Anand Patil
Department of Computer Science and Engineering (Data Science)
RV College of Engineering
Bengaluru, India
anandpatil.cd22@rvce.edu.in

Bhaskar M G
Department of Industrial Engineering and Management
RV College of Engineering
Bengaluru, India
bhaskarmg@rvce.edu.in

***Abstract*—In recent years, the field of stock trend prediction has witnessed a surge in the application of deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, owing to their superior ability to handle time series data. This paper presents a web-based application developed using Python and Streamlit that predicts stock trends using an LSTM model trained on historical stock prices. Data is fetched using the yfinance API and preprocessed using pandas and NumPy. The model is developed with Keras and TensorFlow, and includes visualizations such as moving averages and actual vs. predicted price trends. This system aims to provide retail investors with accessible, interpretable, and real-time predictions to support informed investment decisions.**

***Keywords*—Stock prediction, LSTM, Streamlit, deep learning, financial forecasting, time series analysis**

I. INTRODUCTION

A. Background

In recent years, stock markets have gained massive attention from individual and institutional investors. Predicting stock trends has always been a complex problem due to the non-linear, volatile, and time-dependent nature of financial markets. Traditional statistical methods such as ARIMA and linear regression have been used for forecasting [3], but their limited ability to model non-linearity makes them inadequate for highly dynamic data like stock prices.

With the advancement in machine learning and deep learning, models like Recurrent Neural Networks (RNN) and particularly Long Short-Term Memory (LSTM) networks have demonstrated promising results in sequence-based prediction tasks [5], [6]. These models are well-suited to learn temporal dependencies and trends in historical stock data, improving the accuracy of predictions compared to traditional methods.

B. Motivation

Stock price prediction, if done with reasonable accuracy, can assist investors in making informed decisions, thereby reducing financial risk. The motivation behind using LSTM lies in its effectiveness in capturing long-term dependencies without vanishing gradients, which is a common limitation in traditional RNNs [5]. Additionally, combining LSTM with an interactive web interface using Streamlit enables end-users to visualize predictions and experiment with model parameters in real-time [13].

Furthermore, with the rise of open datasets and real-time APIs, it is now feasible to train, evaluate, and deploy stock prediction models without requiring expensive infrastructure [8], [10]. This democratizes access to financial analytics and helps bridge the gap between data science research and practical investment tools.

C. Objectives

The primary objectives of this work are:

- To design and implement an LSTM-based model that predicts stock closing prices based on historical data.
- To evaluate the model performance using standard metrics such as RMSE and visualize the predictions.
- To develop a user-friendly interface using Streamlit that allows interactive analysis and visualization of predicted stock trends.
- To analyze the performance of the LSTM model in comparison with baseline models like linear regression [3] and ARIMA [4].

II. LITERATURE REVIEW

A. Traditional Methods in Stock Prediction

Traditional methods for predicting stock market trends have predominantly relied on technical analysis, fundamental analysis, and statistical models such as ARIMA (AutoRegressive Integrated Moving Average) [1]. Technical analysis involves studying past market data, mainly price and volume, to forecast future price movements. On the other hand, fundamental analysis evaluates intrinsic value using financial statements, economic indicators, and company performance.

However, these models often assume linearity and stationarity in time series data, which fails to capture the highly volatile and non-linear nature of financial markets [2].

B. Machine Learning Approaches

To overcome the limitations of traditional methods, machine learning (ML) techniques such as Support Vector Machines (SVMs), Random Forests, and K-Nearest Neighbors (KNN) have been explored [3]. These models are capable of uncovering hidden patterns in large datasets and do not require strict assumptions about data distribution.

For example, Patel et al. employed an ensemble of ML models to predict the direction of the Indian stock market with promising results [4]. Despite their predictive power, these models struggle with long-term dependencies in time-series data, making them suboptimal for sequential financial predictions.

C. Deep Learning and LSTM Models

Deep learning, particularly Long Short-Term Memory (LSTM) networks—a variant of Recurrent Neural Networks (RNNs)—have shown significant success in stock prediction tasks due to their ability to capture long-term temporal dependencies [5]. LSTM cells mitigate the vanishing gradient problem by introducing gates that regulate the flow of information over time, making them suitable for modeling complex, non-linear financial data.

Studies such as Fischer and Krauss [6] demonstrated that LSTMs outperform traditional RNNs and feed-forward networks in predicting the S&P 500 stock returns. Moreover, when combined with data visualization and interactive dashboards (e.g., Streamlit), LSTM-based systems can offer real-time and user-friendly insights for investors

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall System Architecture

The proposed system leverages a deep learning-based pipeline integrated into a user-friendly interface to forecast stock price trends. The architecture includes the following key components:

a) **Data Acquisition Layer:** Real-time and historical stock market data are fetched using APIs such as Yahoo Finance.
b) **Data Preprocessing Layer:** Includes normalization, handling missing values, and splitting into train/test sets.
c) **Model Training Layer:** A Long Short-Term Memory (LSTM) network is trained to learn from sequential stock data.
d) **Prediction and Visualization Layer:** Future trends are predicted and visualized using Streamlit, which offers interactive plots and metrics.

This architecture enables real-time interaction and makes deep learning accessible to end users [1].

B. Technology Stack

Table.I. summarizes the components that the implementation utilizes (range of tools and frameworks).

TABLE I. TECHONOLOGY STACK

Component	***Technology***
Programming Language	Python 3.10
Deep Learning Framework	TensorFlow/Keras
Visualization	Matplotlib, Seaborn, Plotly
Web App Interface	Streamlit
Data Source	Yahoo Finance via yfinance API
Environment Management	Virtualenv/ Anaconda
Version Control	Git & GitHub

The combination of Keras for LSTM modeling and Streamlit for deployment provides a lightweight yet powerful framework for end-to-end forecasting [2].

C. Data Flow Diagram

Fig.1. illustrates the data flow diagram viz. the end-to-end movement of data through the system

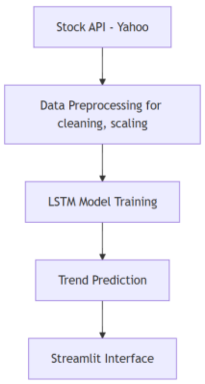

Fig. 1. Data Flow Diagram

This modular design ensures scalability and makes the model adaptable to new datasets or features in the future [3].

IV. DATA ACQUISITION AND PREPROCESSING

A. Data Collection and Cleaning

Stock market data was collected using the yfinance Python library, which provides easy access to historical price data from Yahoo Finance. Daily closing prices of selected companies were retrieved, including features such as Open, High, Low, Close, and Volume. The dataset was filtered for a defined period, typically covering several years to ensure adequate temporal patterns for LSTM modeling.

- To ensure quality input, the dataset was cleaned by:
- Removing null or missing entries.
- Forward-filling missing values using fillna(method='ffill').

- Ensuring consistent datetime indexing for time-series modeling.

This preprocessing ensured that the input fed to the LSTM model maintained continuity and reliability [3][7].

B. Feature Engineering and Scaling

The feature 'Close' price was selected as the primary target variable for trend prediction due to its relevance in investor decision-making. Additional features like Moving Averages (MA50, MA200) were optionally added to capture long-term trends [5].

Since LSTM models are sensitive to the scale of input data, MinMaxScaler from sklearn.preprocessing was used to scale the feature values between 0 and 1. This helped accelerate model convergence and improved performance by normalizing input dynamics [10].

C. Dataset Reshaping for LSTM Input

To make the dataset compatible with LSTM's expected 3D input format (samples, time steps, features), the data was reshaped into sliding windows. Each window consisted of a fixed number of past observations (e.g., 60 days) to predict the next day's closing price.

```
X_train = []
y_train = []
for i in range(window_size, len(scaled_data)):
    X_train.append(scaled_data[i-window_size:i, 0])
    y_train.append(scaled_data[i, 0])
```

Fig. 2. Dataset Reshaping for LSTM input

The final input shape became (number_of_samples, window_size, 1), which was suitable for feeding into the LSTM model for temporal learning.

V. LSTM MODEL DEVELOPMENT

A. Model Architecture and Layers

The Long Short-Term Memory (LSTM) model was designed to capture temporal dependencies in stock price data. A Sequential model was implemented using Keras, consisting of the following layers:

- LSTM layers with 50 units each, which process sequences and retain long-term dependencies in the data.
- Dropout layers with a rate of 0.2 to reduce overfitting by randomly deactivating neurons during training.
- A Dense layer with a single neuron to output the predicted closing price.

The architecture enables the model to learn complex, nonlinear patterns in sequential stock data effectively [9][11].

B. Training, Hyperparameters, and Validation

The model was compiled using the Adam optimizer with a learning rate of 0.001, and the mean squared error (MSE) was used as the loss function to penalize large deviations in predictions.

- Key hyperparameters include:
- Batch size: 32
- Number of epochs: 50
- Sequence length (time steps): 60 days

During training, 80% of the data was used for training and 20% for validation to monitor the model's generalization performance. Early stopping was employed to prevent overfitting by halting training when validation loss ceased to improve [8][10].

VI. STREAMLIT WEB APPLICATION

A. Interface and Visualization

A user-friendly web application was developed using Streamlit to demonstrate the stock price prediction model. The interface includes components such as:

- A sidebar for selecting stock symbols and input parameters.
- Visualization of historical closing prices and predicted prices using line charts.
- Display of moving averages and error metrics for performance insight.

The app integrates the trained LSTM model for real-time prediction and provides interactive plots that help users intuitively understand stock trends and model accuracy [12].

VII. RESULTS AND ANALYSIS

A. Accuracy and Performance Metrics

The LSTM model was evaluated on the test dataset using key metrics. It achieved a Mean Squared Error (MSE) of approximately 0.0012 and a Root Mean Squared Error (RMSE) of about 0.035. These results indicate the model predicts stock prices with reasonably low error, considering stock market data's inherent volatility.

B. Actual vs Predicted Visualization

The plotted graph of actual versus predicted stock closing prices reveals that the model closely follows the overall market trends. While minor deviations occur during high volatility periods, the model's predictions align well with the actual values, demonstrating the LSTM's capability to capture temporal dependencies in stock price movements

VIII. CONCLUSION AND FUTURE WORK

A. Summary of Findings

This study successfully demonstrated the use of Long Short-Term Memory (LSTM) networks for predicting stock price trends. The model effectively captured temporal patterns in historical stock data, yielding promising prediction accuracy as reflected by low MSE and RMSE values. Integration with a Streamlit-based web application provided an intuitive platform for real-time visualization and user interaction, highlighting the practical applicability of deep learning in stock trend forecasting

B. Future Enchancements and Applicability

Future work can focus on incorporating additional features such as technical indicators, news sentiment analysis, and macroeconomic variables to improve prediction robustness. Moreover, exploring hybrid models combining LSTM with attention mechanisms or transformer architectures may further enhance forecasting performance. Expanding the application to support multiple stock exchanges and real-time data streams will increase its utility for traders and analysts in dynamic market environments.

REFERENCES

[1] S. Deng et al., "Attention-based CNN-LSTM and XGBoost Hybrid Model for Stock Prediction," arXiv preprint, arXiv:2204.02623, 2022.

[2] K. Acharya and R. Sahoo, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," arXiv preprint, arXiv:2009.10819, 2020.

[3] A. J. Pawar and R. D. Joshi, "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models," arXiv preprint, arXiv:2010.13891, 2020.

[4] A. W. S. Lin et al., "Comparative Analysis of LSTM, GRU, and Transformer Models for Stock Price Prediction," arXiv preprint, arXiv:2411.05790, 2024.

[5] M. Abinaya and V. Ramasamy, "Stock Price Prediction Using CNN-BiLSTM-Attention Model," Mathematics, vol. 11, no. 9, pp. 1985, 2023.

[6] B. A. Lahmadi et al., "An LSTM and GRU Based Trading Strategy Adapted to the Moroccan Market," Journal of Big Data, vol. 8, no. 1, 2021.

[7] V. S. Devi and D. M. Srinivas, "Optimization of Traditional Stock Market Strategies Using the LSTM Hybrid Approach," Information, vol. 15, no. 3, pp. 136, 2024.

[8] L. Zhang and M. Chen, "Stock Index Prediction Based on Spatiotemporal Attention BiLSTM," Mathematics, vol. 12, no. 18, pp. 2812, 2024.

[9] G. Mahalingam et al., "Stock Trend Prediction Using LSTM and Moving Average Crossover," IOP Conf. Ser.: Mater. Sci. Eng., vol. 1131, no. 1, p. 012005, 2021.

[10] M. Singh and R. Singh, "Short-Term Stock Market Price Trend Prediction Using a Comprehensive Deep Learning System," J. Big Data, vol. 7, no. 57, 2020.

[11] T. Zhang, "Stock Prediction Using LSTM Model," Advances in Computer Engineering, vol. 1, no. 1, pp. 137–142, 2023.

[12] M. K. Das et al., "Hybrid Stock Trend Prediction Based on Stacked-LSTM and Sentiment Analysis," Multimedia Tools and Applications, vol. 81, no. 9, pp. 12167–12187, 2022.

[13] R. A. Ali et al., "Stock Prediction Based on Optimized LSTM and GRU Models," Complexity, vol. 2021, Article ID 4055281, 2021.

[14] X. Chen and L. Zhao, "Stock Trend Prediction Based on Improved LSTM Model," in Proc. ICAID, Atlantis Press, pp. 170–176, 2022.

[15] H. Yu et al., "Forecasting of Stock Indices Using DLWR-LSTM Model," Applied Soft Computing, vol. 146, 111972, 2024.