MULTIVARIATE STOCK PRICE FORECASTING USING LONG SHORT-TERM MEMORY (LSTM)

Date of Submission: 06/03/2025

OVERVIEW

1. Abstract

Stock price prediction is challenging due to market volatility and complex dependencies. This project leverages LSTM networks to forecast stock prices using the Google stock price dataset. The model preprocesses stock data, transforms it into sequences, and optimizes performance using dropout regularization and learning rate scheduling. Evaluation is based on MSE metrics. Additionally, Hugging Face's FinBERT is fine-tuned for financial sentiment analysis, integrating stock-related news sentiment into predictions. Transfer learning with FinBERT enhances sentiment classification for better market trend forecasting. The expected outcome is a robust model combining time series analysis and sentiment insights for improved stock price prediction. Future work includes attention mechanisms and macroeconomic factor integration.

2. Introduction

Stock price prediction is a crucial financial task that involves forecasting future stock prices based on historical data. Traditional statistical methods, such as moving averages and autoregressive models, often fall short in capturing the complex, non-linear relationships present in financial time series data.
Direct application in financial decision-making, portfolio management, and algorithmic trading.

The project aligns with advancements in neural networks and deep learning by:

- *Utilizing LSTM to model complex dependencies in time series data.*
- *Incorporating NLP-based financial sentiment analysis to refine predictions.*

The primary challenge is to build a deep learning model that can:

- *Accurately forecast stock prices using multivariate analysis.*
- *Identify the most influential features affecting price movements.*
- *Leverage sentiment analysis to enhance predictions.*

3. Project Objectives

- *Develop a multivariate LSTM model for stock price prediction using historical Google stock data.*
- *Compare the performance of the deep learning model with traditional statistical methods.*
- *Evaluate the effectiveness of LSTM in capturing complex sequential dependencies in financial time series data.*

4. Methodology

The chosen Neural Network model is Long Short Term Memory(LSTM). Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is made to work well with data that has long-term patterns. Regular RNNs have a hard time remembering information from long sequences because of something called vanishing gradients. However, LSTMs solve this problem with a special memory cell and gates that help them keep track of information better.

An LSTM unit has three important gates: the forget gate decides what information to get rid of, the input gate decides what new information to keep, and the output gate controls what the unit will show as output. These gates control how information flows, helping the model remember important patterns over long periods while stopping unneeded data from piling up.

LSTMs are often used in predicting future events, understanding language, and recognizing speech because they can manage complicated patterns over time. They can be piled up in several layers to make deep LSTMs, which helps them learn more complex patterns. Also, bidirectional LSTMs (Bi-LSTMs) look at sequences in both forward and backward ways, which helps them understand the context better.

The data was gathered from Yahoo Finance for Google stock prices. After loading the dataset, we changed the Date column into the right date format so we can analyze the time correctly. Data cleansing was done to fix missing information and get rid of errors. The data was divided into three parts: training, validation, and testing sets to make sure the model was evaluated correctly. Usually, most of the data was used for training, and a smaller part was kept for checking and testing. Sequences can also be made using a sliding window method to get the data ready for LSTM input, helping the model learn the time patterns better.

Data visualization was performed to analyze stock price trends over the past five years. Older data was discarded, and only the most recent five years of data were retained in a new CSV file for improved model relevance. Trend analysis involved plotting the Date vs. Open Price and Date vs. Closing Price graphs to observe price fluctuations and long-term patterns. These visualizations helped identify trends, volatility, and potential seasonality in stock movements. The *insights gained from this analysis were crucial for feature selection and model training, ensuring that only meaningful data was used for LSTM-based forecasting.*

The LSTM model was created using TensorFlow-Keras and has four hidden LSTM layers. This helps it understand long-term patterns in stock price changes. The Adam optimizer was used to quickly adjust the model's weights and Mean Squared Error (MSE) was selected as the way to measure and reduce prediction mistakes. A Model Checkpoint feature was added to save the best model by looking at validation loss, making sure it works well. The model was trained for 200 epochs, helping it understand complicated patterns in stock prices. To avoid overfitting, we looked at methods like dropout and adjusting the learning rate.

After the training, we can look at the model's loss and validation loss to see if it was improving and working well with new data. Also, TensorBoard was used to watch how training was going and to see important numbers. To make predictions more accurate, we used Hugging Face's FinBERT to improve how we analyze feelings in financial news. This helps us understand market sentiment better and use it to predict stock trends. FinBERT's ability to learn from other tasks made it better at understanding opinions about stocks, giving us more useful information about the market. These improvements made the LSTM model stronger and better at predicting stock price trends.

The trained LSTM model was used to predict future stock prices, and its performance was assessed using various evaluation metrics. Mean Squared Error (MSE) and Mean Absolute Error (MAE) were used to measure the accuracy of predictions by comparing them with actual stock prices. Root Mean Squared Error (RMSE) was also calculated to better understand deviations from real values. Additionally, SHAP (SHapley Additive exPlanations) was explored to interpret model predictions by identifying which input features had the most impact on price forecasting. This helped in understanding how different factors influenced stock price movements. The results provided insights into the model's effectiveness in capturing trends and minimizing errors.

The LSTM model can be deployed using FastAPI or TensorFlow Serving to serve predictions as an API. FastAPI enables lightweight and efficient RESTful endpoints for real-time stock price forecasting. TensorFlow Serving allows seamless deployment with built-in model versioning and scalability. The model can be hosted on cloud platforms such as Google Cloud, AWS, or Azure, ensuring high availability and performance. Additionally, containerization tools like Docker can be used for easy deployment and scaling. By integrating the model into a web or mobile application, users can retrieve real-time predictions and analyze market trends efficiently.

Tools Used:

Python, TensorFlow/Keras, NumPy & Pandas, TensorBoard, Matplotlib, FinBERT - Hugging Face

Fig. 1 presents the conceptual diagram of the model used.

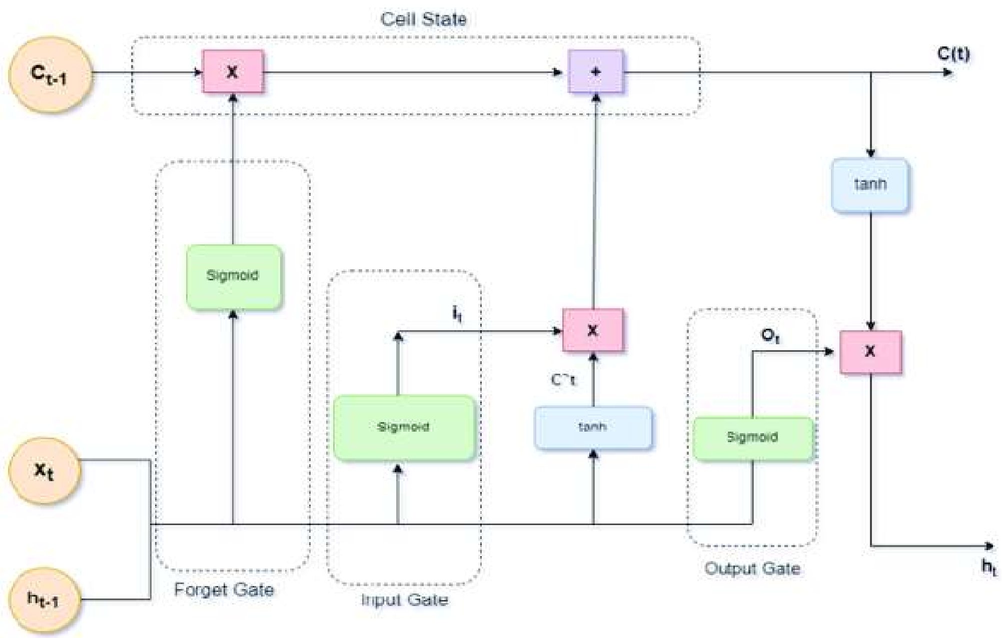

Fig. 1 Architectural Diagram

5. Project Plan

Phase	Tasks	Milestone	Duration
Phase 1: Data Collection & Preprocessing	*- Collect stock data (Yahoo Finance) - Clean, normalize & split data - Perform feature engineering*	*Dataset is preprocessed and ready for training*	*2 Day*
Phase 2: Model Development (LSTM)	*- Implement LSTM model - Fine-Tuning hyperparameters - Train on historical stock data*	*Trained LSTM model with initial performance metrics*	*2-3 Days*
Phase 3: Inference, Explainability & Visualization	*- Model Inference - Apply SHAP for feature importance - Integrate TensorBoard for monitoring*	*Insights from SHAP and TensorBoard*	*2 Day*
Phase 4: Deployment & Optimization	*- Deploy model using TensorFlow Serving - Optimize for latency (quantization, batching)*	*Hosting model with API endpoints*	*2 Days*
Phase 5: NLP Integration (FinBERT)	*- Process financial news data - Use FinBERT to add sentiment scores - Retrain model with sentiment-enhanced features*	*Improved predictions with sentiment analysis*	*2-3 Days*

6. Expected Outcomes

Anticipated Results:

- ***Accurate Stock Price Predictions****: The LSTM model that we built will forecast stock prices with high precision using multivariate time series data.*
- ***Feature Importance Insights****: SHAP analysis will reveal the impact of each stock feature (Open, High, Low, Close, Volume) on predictions.*
- ***Sentiment-Aware Predictions****: FinBERT integration will improve forecasting by incorporating financial news sentiment.*
- ***Improved Model Interpretability****: TensorBoard visualizations will provide transparency into model performance and training dynamics.*

Potential Impact and Applications:

- ***Investment Strategy Optimization****: Assists investors and financial analysts in making data-driven stock market decisions.*
- ***Risk Management in Trading****: Enables proactive identification of market trends and potential fluctuations, minimizing financial risks.*
- ***Scalable and Production-Ready****: Deployable as an API for real-time stock prediction applications.*

7. Evaluation Metrics

Quantitative Error Metrics *:* ***Measures how close predicted stock prices are to actual values.***	
Mean Absolute Error (MAE)	*Average absolute difference between predicted and actual stock prices. Lower MAE = better accuracy.*
Root Mean Squared Error (RMSE)	*Square root of the average squared differences between predicted and actual values. Penalizes large errors more.*
Mean Absolute Percentage Error (MAPE)	*Error as a percentage of actual stock prices, useful for comparing across different price ranges. Lower MAPE = better performance.*
Model Performance Evaluation *:* ***Assesses the effectiveness of the trained LSTM model in stock forecasting.***	
Validation	*Tests the model on unseen stock price data (validation set) to measure generalization.*
Trend & Pattern Analysis	*Plots* ***predicted vs. actual stock prices*** *over time to identify trends, deviations, and anomalies.*
Confidence Intervals	*Use* ***bootstrapping*** *to estimate uncertainty in stock price predictions, ensuring reliability in financial decision-making.*

8. Conclusion

Summary of Key Points:

- *Develop an LSTM model to predict stock prices using historical data and sentiment analysis.*
- *Evaluate model performance using error metrics like MAE, RMSE, and MAPE.*

Project Significance and Contributions:

- *Contribute a reliable model for accurate stock price prediction using deep learning techniques.*
- *Improves financial forecasting by using sentiment analysis to understand market trends better.*

TEAM DETAILS

Roll Number	Name	Section
CB.EN.U4CSE22301	Aakaash M S	D
CB.EN.U4CSE22309	Aniketha Prasad	D
CB.EN.U4CSE22348	S Karthik Ram	D
CB.EN.U4CSE22435	Riya Rajesh	E