Chronos Trade:

Multi-Modal Stock Market prediction using LTSM, ARIMA and Sentiment Analysis.

Work Report

Pranav Pawar
24b2503
6 Jan,2026

1. Introduction

Time series forecasting plays a crucial role in financial analysis, particularly in predicting stock prices. In this project, I implemented and compared ARIMA (AutoRegressive Integrated Moving Average) and LSTM (Long Short-Term Memory) models to forecast the closing price of Google (GOOGL) stock using historical data obtained from Yahoo Finance.

The objective of this work was:

- To understand and implement both **statistical** and **deep learning–based** time series models
- To compare their assumptions, behavior, and predictive performance
- To gain practical insights into real-world challenges of financial time series forecasting

2. Understanding the ARIMA model

The ARIMA model is a widely used statistical method for time series forecasting. It combines three key components:

- **AR (AutoRegressive):** Uses the dependency between an observation and a number of lagged observations.
- **I (Integrated)**: Applies differencing to make the time series stationary.
- **MA (Moving Average)**: Models the dependency between an observation and past error terms.

An ARIMA model is represented as ARIMA**(p, d, q),** where:

- **p** = number of autoregressive terms
- **d** = number of differences needed to make the series stationary
- **q** = number of moving average terms

In this project, the optimal parameters obtained were ARIMA(0,1,0), which essentially represents a random walk model with differencing.

3. LSTM Model

LSTM is a type of **Recurrent Neural Network (RNN)** designed to handle long-term dependencies in sequential data. Unlike ARIMA, LSTM:

- Does not require stationarity
- Learns patterns directly from data
- Handles non-linear relationships

Key characteristics:

- Uses memory cells and gates (forget, input, output)
- Requires large datasets and careful preprocessing
- Computationally intensive

- Powerful for complex time series

4.Statistical vs Deep Learning Approaches

Aspect	ARIMA	LSTM
Model type	Statistical	Deep learning
Stationarity required	Yes	No
Handles non-linearity	Poorly	Very well
Interpretability	High	Low
Data requirement	Small	Large
Computational cost	Low	High

5.Important concepts Explained

5.1 Stationarity and Differencing

A time series is **stationary** if its:

- Mean
- Variance
- Autocovariance

  remain constant over time.

  Stock prices are typically **non-stationary**, so **differencing** is applied:

$$Y_t' = Y_t - Y_{t-1}$$

Differencing removes trends and stabilizes the mean, which is essential for ARIMA modeling.

5.2 ACF and PACF Plots

- **ACF (Autocorrelation Function)**
  Measures correlation between the series and its lagged values
  → Used to identify **MA(q)**
- **PACF (Partial Autocorrelation Function)**
  Measures direct correlation excluding intermediate lags
  → Used to identify **AR(p)**
  These plots guided the selection of optimal ARIMA parameters.

5.3 Sliding Window Technique (LSTM)

LSTM requires supervised learning format.
A **sliding window** converts time series into input–output pairs:

- Input: past n time steps
- Output: next time step

This allows LSTM to learn temporal dependencies.

5.4 Data Normalization

Neural networks are sensitive to scale.
Min-Max Normalization was used:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Why normalization is important:

- Faster convergence
- Stable gradients
- Prevents dominance of large values

Normalization was applied **only using training data statistics** to avoid data leakage.

6.Implementation Insights and Challenges

6.1 Challenges with ARIMA

- Achieving stationarity required careful differencing
- Parameter selection using ACF/PACF was not always clear
- Performance degraded during volatile market periods

6.2 Challenges with LSTM

- Long training time
- Sensitive to hyperparameters (look-back window, batch size, learning rate)
- Overfitting risk without proper validation
- Requires reshaping data into 3D format

7.Observations from Model Performance

- The ARIMA(0,1,0) model was suitable for the given dataset.
- First-order differencing was sufficient to achieve stationarity.
- The model produced reasonable short-term forecasts but showed limitations for long-term prediction.

8. Visualization and Output Explanation

8.1 Stock Price vs Time Plot

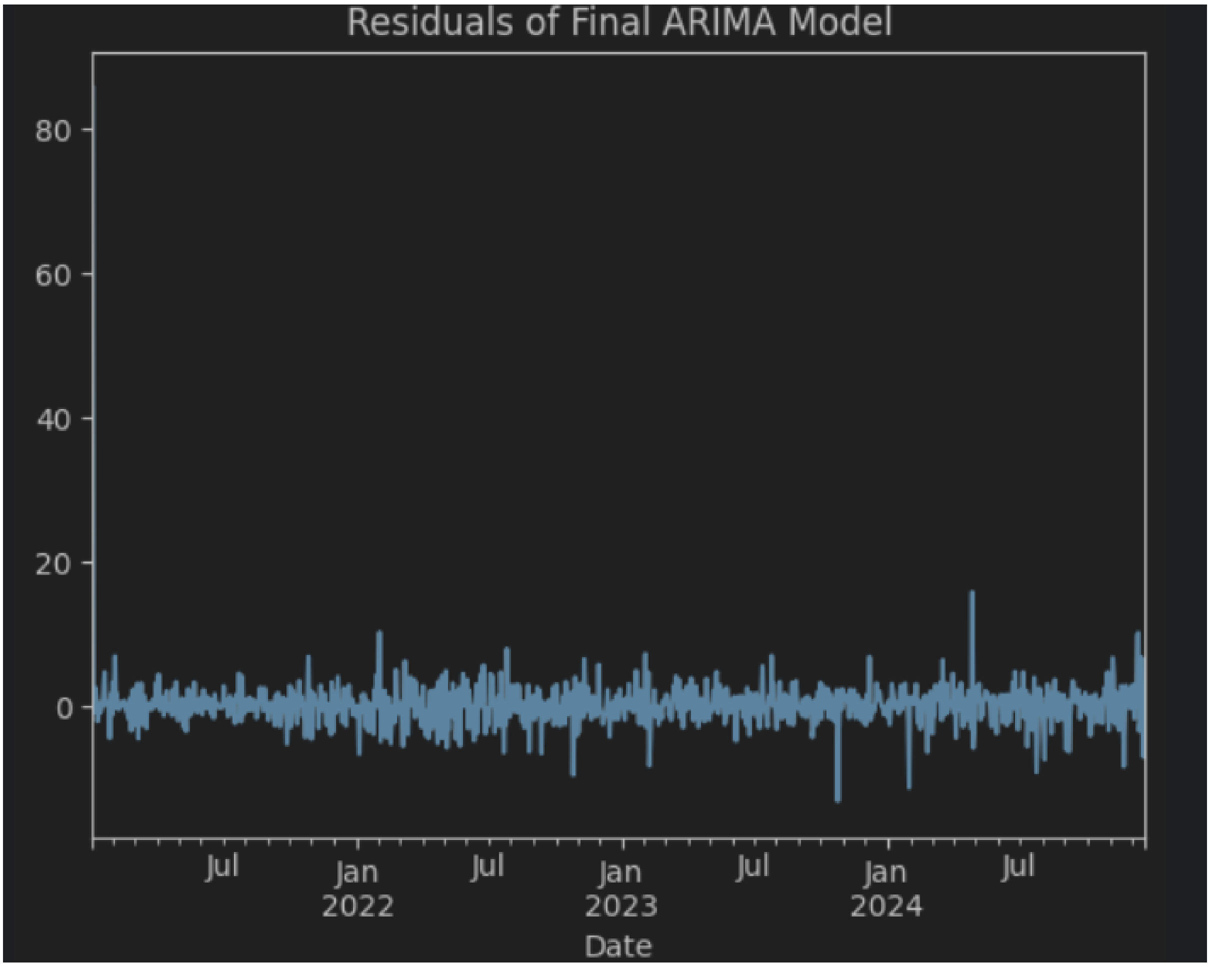
- Shows historical closing prices
- Highlights trend and volatility
- Confirms non-stationarity

8.2 ARIMA vs LSTM Forecasts

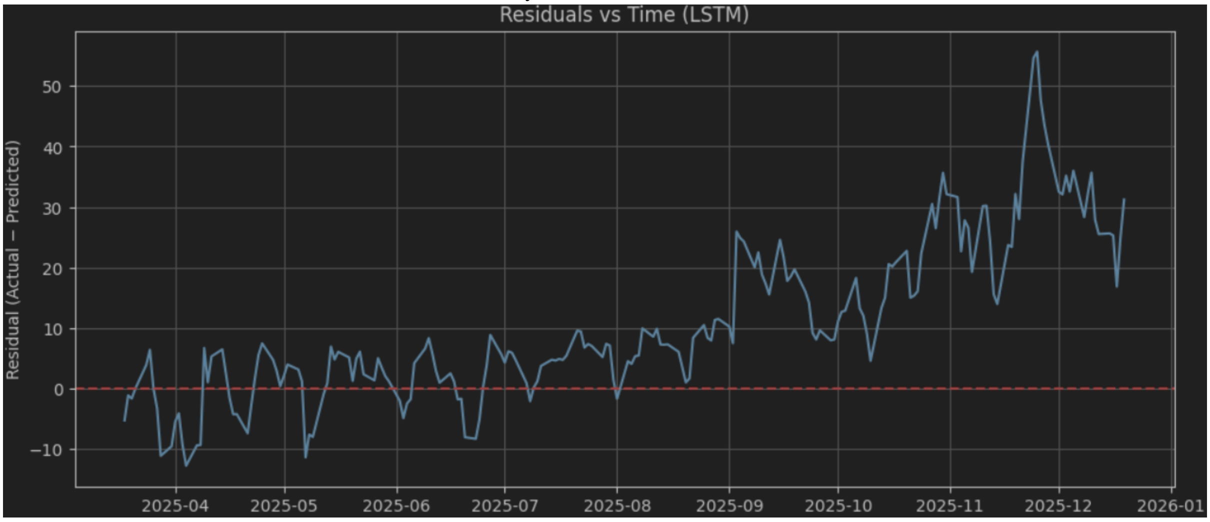
- ARIMA predictions closely follow recent trends
- LSTM produces smoother forecasts
- LSTM adapts better to long-term dependencies

8.3 Residual Plots

- ARIMA residuals showed autocorrelation → model limitations

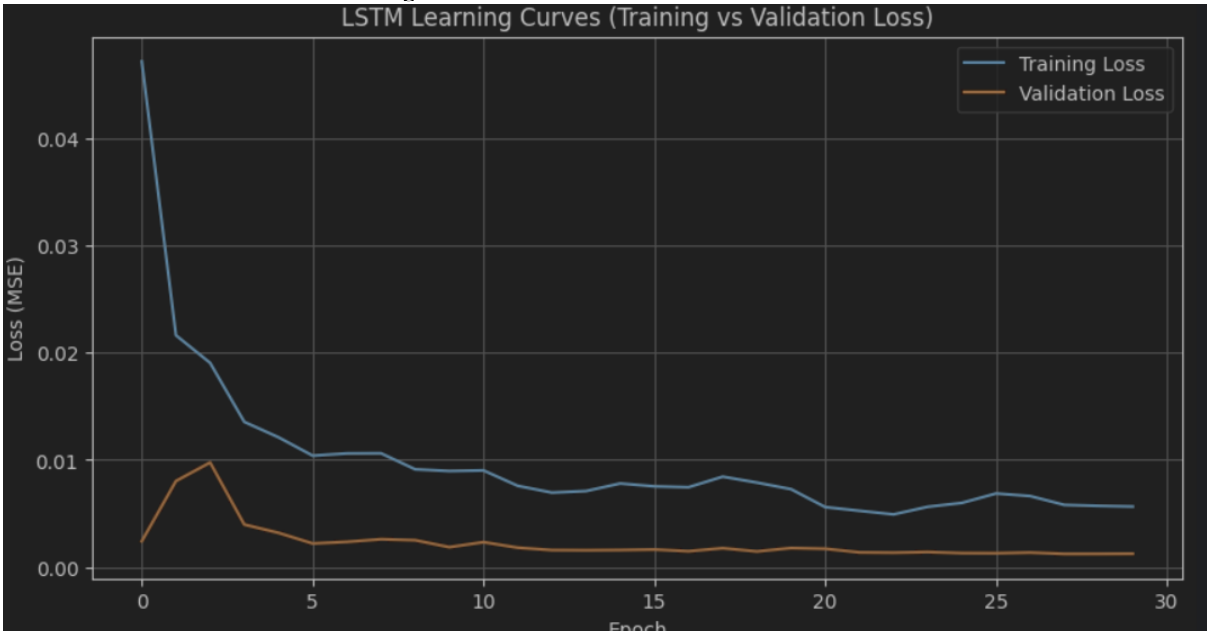


- LSTM residuals were more randomly distributed → better fit



8.4 LSTM Learning Curves

- Training and validation loss decreased steadily
- Gap between curves indicated controlled overfitting
- Confirmed effective learning



9.Conclusion

This **Key Findings**

- ARIMA is effective for linear, stationary data
- LSTM excels in capturing complex and non-linear patterns
- Deep learning models outperform statistical models for volatile stock data

Strengths and Limitations

ARIMA

- ✓ Interpretable
- ✓ Low computational cost
- ✕ Poor non-linear modeling

LSTM

- ✓ Handles non-linearity
- ✓ Flexible and powerful
- ✕ Computationally expensive
- ✕ Hard to interpret

10. Future Improvements and Extensions

- Use Bidirectional LSTM or GRU
- Incorporate technical indicators (RSI, MACD, volume)
- Add exogenous variables (news sentiment, market indices)
- Perform hyperparameter optimization
- Combine ARIMA + LSTM (hybrid model)