Chronos Trade:

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

Work Report

Pranav Pawar
24b2503
30 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

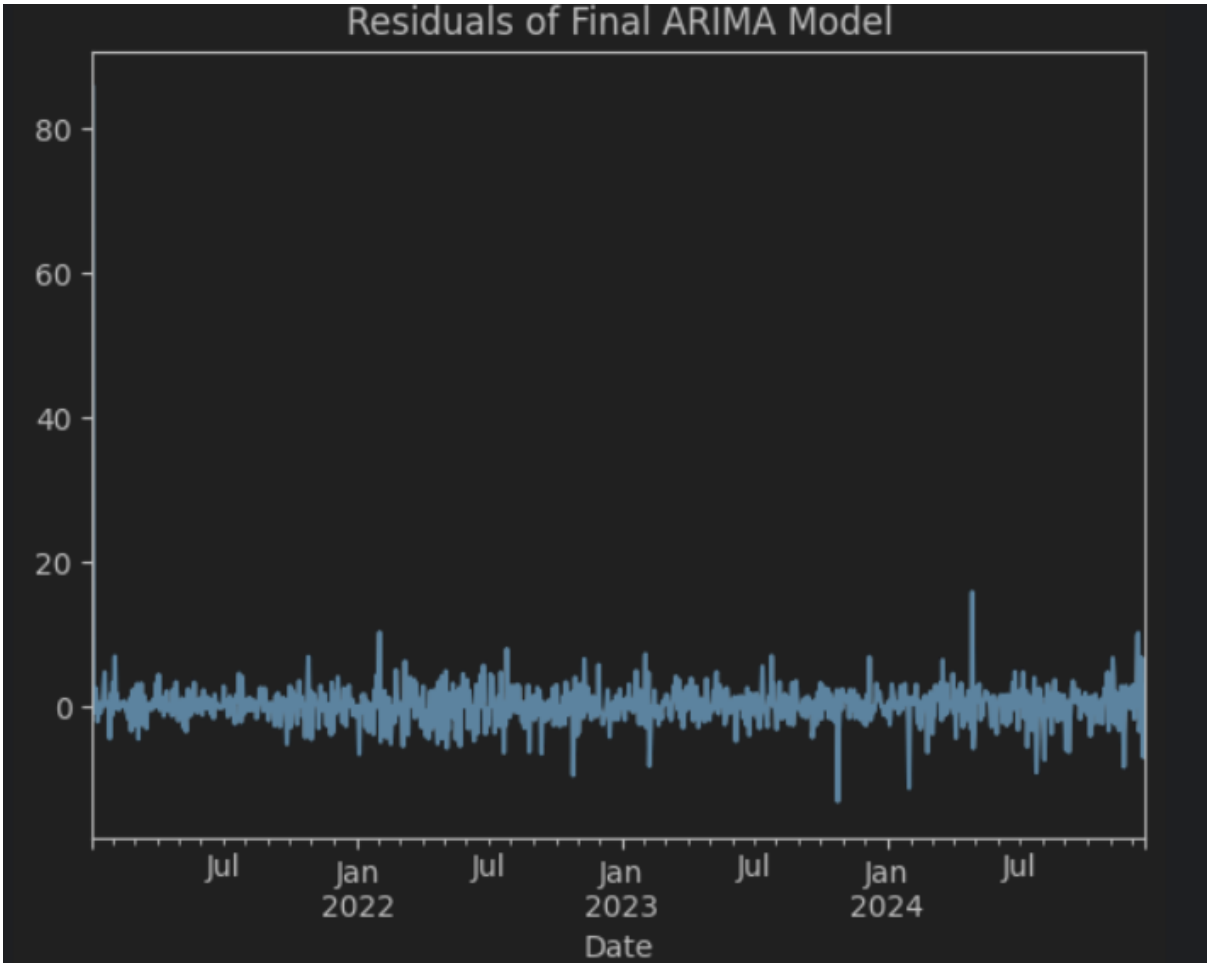
- Shows historical closing prices
- Highlights trend and volatility
- Confirms non-stationarity

8.2 ARIMA vs LSTM Forecasts

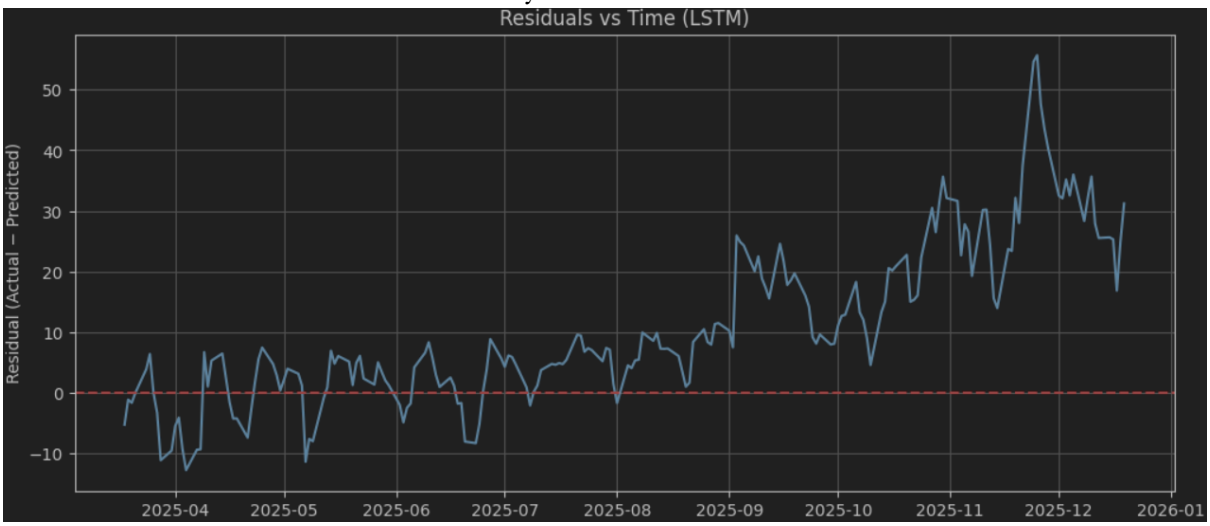
- ARIMA predictions closely follow recent trends
- LSTM produces smoother forecasts
- LSTM adapts better to long-term dependencies

8.3 Residual Plots

- ARIMA residuals showed autocorrelation → model limitations

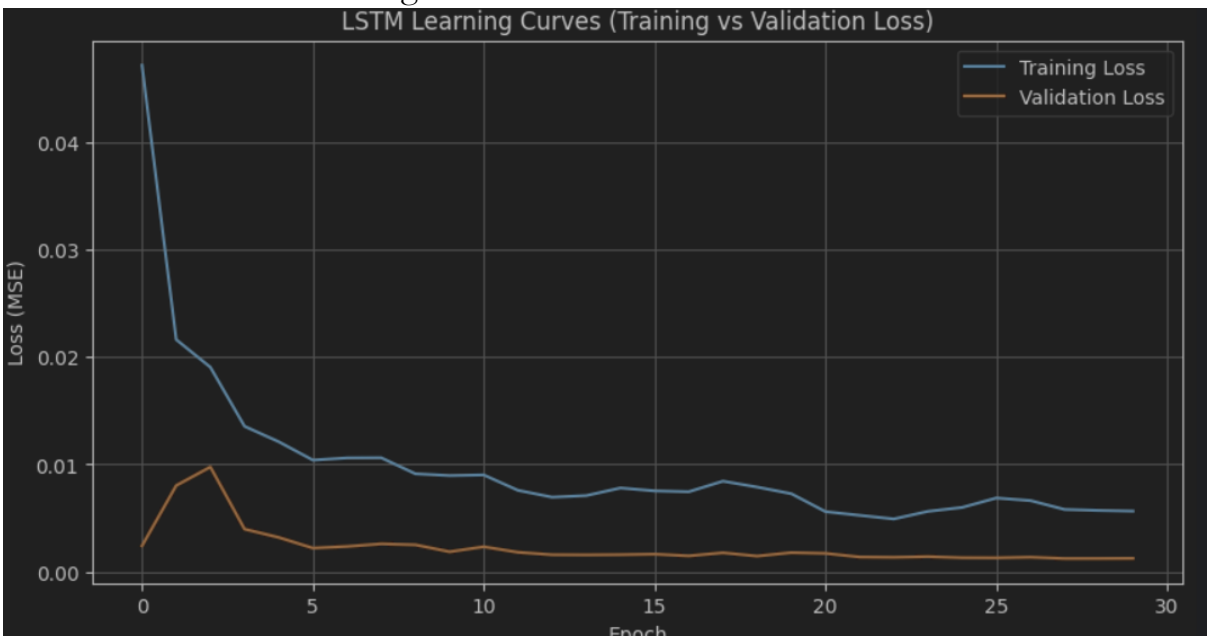


- LSTM residuals were more randomly distributed → better fit



8.4 LSTM Learning Curves

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



9. Final Project: Multi-Modal Stock Price Prediction Using NLP, Sentiment Analysis, and LSTM

The final stage of this project extends traditional time-series forecasting by incorporating **Natural Language Processing (NLP)–based sentiment analysis** along with historical stock price data. This creates a **multi-modal prediction system**, where both numerical market data and textual sentiment information contribute to the final stock price prediction.

9.1 Motivation for Using Sentiment Analysis

Stock prices are influenced not only by historical trends but also by:

- News articles
- Market sentiment
- Public perception and reactions

Purely numerical models like ARIMA or standalone LSTM fail to capture this **external psychological factor**. Sentiment analysis helps bridge this gap by quantifying market emotions

9.2 NLP and Sentiment Analysis Pipeline

The sentiment analysis module follows these steps:

1. **Text Data Collection**
   Financial news headlines related to the selected stock were collected from online sources.
2. **Text Preprocessing**
   - Lowercasing
   - Removal of stop words and punctuation
   - Tokenization
   - Lemmatization
3. **Sentiment Scoring**
   NLP techniques were applied to classify each headline as:
   - Positive
   - Negative
   - Neutral

Each text sample was converted into a **numerical sentiment score**, representing overall market mood for that day.

4. **Daily Sentiment Aggregation**
   Multiple headlines per day were aggregated to generate a **single daily sentiment value** aligned with stock price timestamps.

9.3 Feature Fusion: Sentiment + Price Data

The final prediction model uses **feature-level fusion**, where:

- Historical closing prices
- Corresponding daily sentiment scores

are combined into a single input sequence.

This results in an input format such as:

- [Price(t−n), … , Price(t−1)]
- [Sentiment(t−n), … , Sentiment(t−1)]

These combined features allow the model to learn **relationships between market movement and public sentiment**.

9.4 LSTM-Based Final Prediction Model

An LSTM network was used as the final prediction model due to its ability to:

- Capture long-term temporal dependencies
- Handle non-linear relationships
- Learn interactions between numerical and textual-derived features

Key characteristics of the final LSTM model:

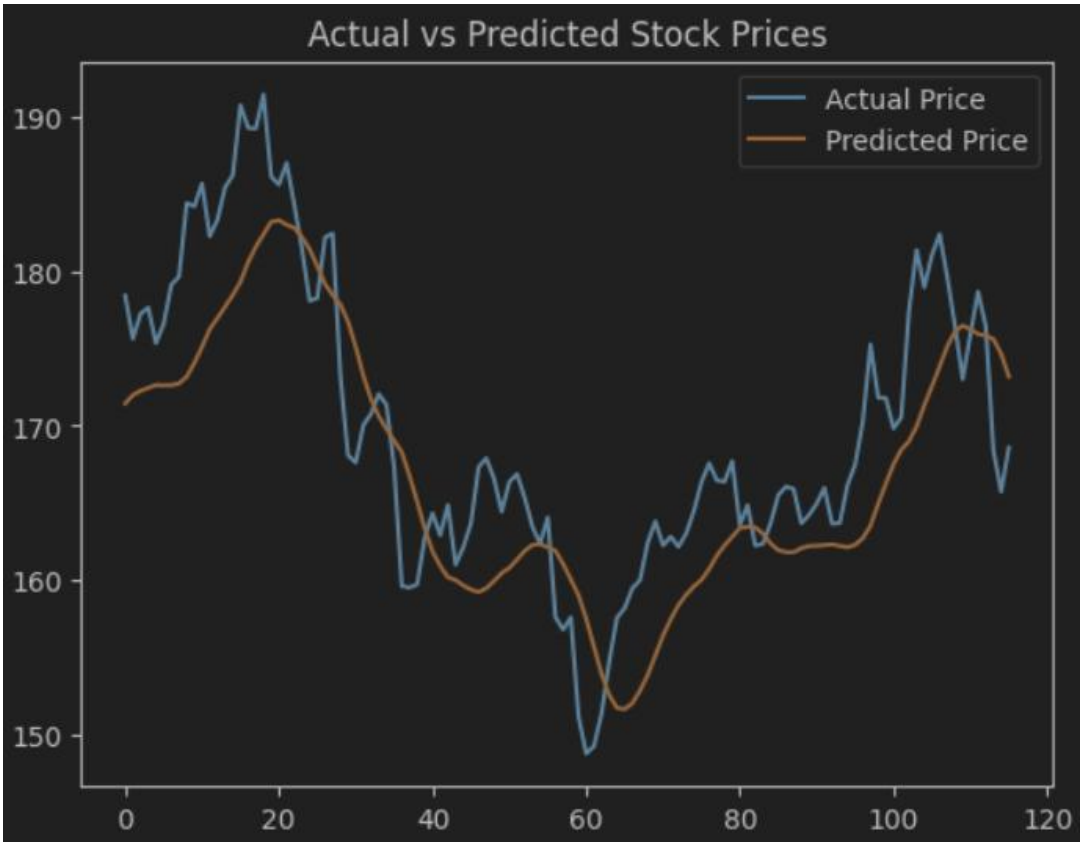
- Sliding window–based supervised learning
- Multi-feature input (price + sentiment)
- Mean Squared Error (MSE) as loss function
- Adam optimizer for training

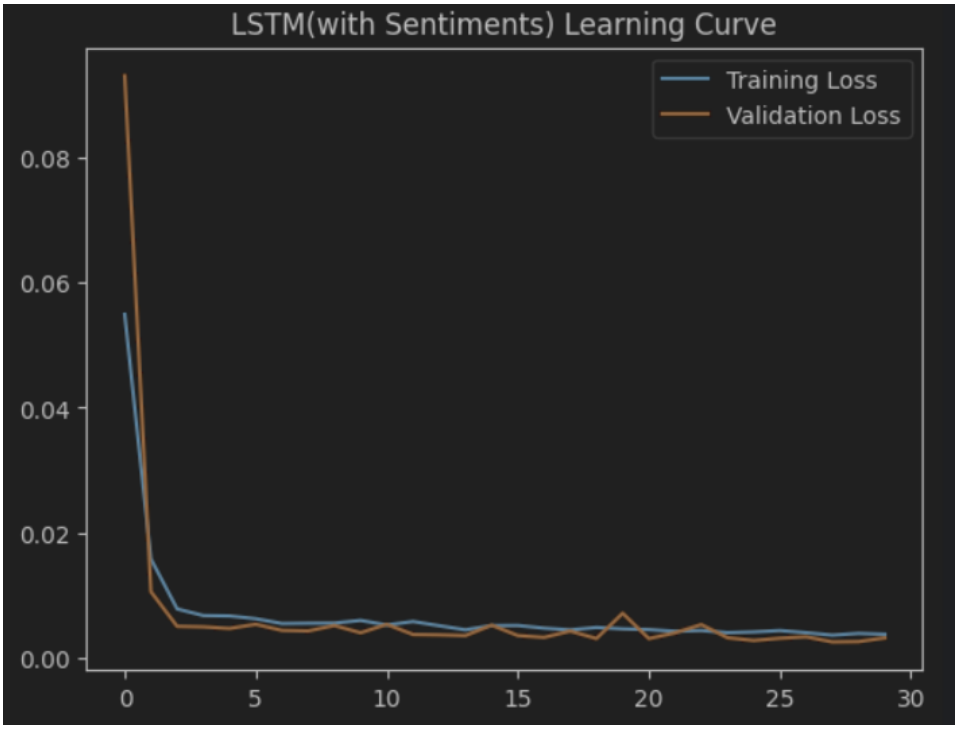
9.5 Observations from Sentiment-Augmented Prediction

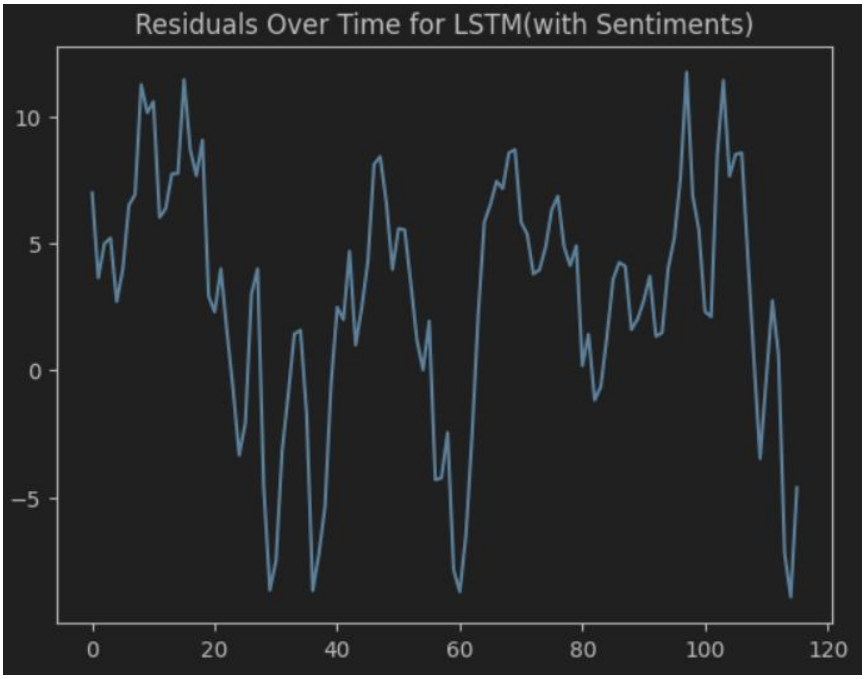
Including sentiment information resulted in:

- Improved responsiveness to sudden price movements
- Better prediction during news-driven volatility
- Reduced lag compared to price-only models

The model showed higher adaptability during periods of strong positive or negative sentiment, confirming the importance of textual data in financial forecasting.


Actual vs Predicted Stock Prices
Actual Price
Predicted Price
190
180
170
160
150
0
20
40
60
80
100
120


LSTM(with Sentiments) Learning Curve
Training Loss
Validation Loss
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
30



10. Conclusion

This project successfully demonstrates a **multi-modal stock market prediction system** by integrating NLP-based sentiment analysis with LSTM-based time-series forecasting. By combining historical price data with market sentiment extracted from textual information, the final model provides a more realistic and robust representation of stock price behavior.

The results highlight that **sentiment-aware deep learning models outperform price-only approaches**, especially during volatile and news-sensitive market periods. This work emphasizes the importance of combining numerical and textual data for modern financial prediction systems.

11. Future Improvements and Extensions

- Use transformer-based sentiment models (FinBERT)
- Assign different weights to news sources
- Include social media sentiment
- Build a real-time prediction dashboard
- Experiment with attention-based LSTM models