Stock Price Forecasting using ARIMA, LSTM, and LSTM with Sentiment Analysis

1. Project Overview and Objective

Stock price forecasting is one of the most studied problems in financial data science because it lies at the intersection of economics, statistics, psychology, and machine learning. Unlike many engineering prediction problems where systems follow physical laws, financial markets are driven by human decisions, expectations, and reactions to information. This makes prediction inherently uncertain and complex.

The objective of this project is to explore how different modeling approaches behave when applied to stock price prediction. Instead of assuming one model is perfect, the project follows a progressive workflow: start with ARIMA, move to LSTM, and then enhance LSTM with textual sentiment information. This layered approach allows comparison between statistical modeling, deep learning on numbers, and deep learning with text signals.

The goal is not to build a trading system, but to understand modeling behavior and data challenges.

2. Nature of Financial Data

Financial data is time-dependent. Today's price depends on yesterday’s price and recent trends. Rows are not independent. Financial data is also noisy due to trading activity and investor behavior. Prices are non-stationary, meaning their average level changes over time. Markets are competitive and predictable patterns disappear quickly.

3. Dataset Description

The dataset combines news headlines and stock prices. Each headline has a date and ticker. Headlines represent information flow such as earnings reports or analyst opinions. Stock data contains daily closing prices downloaded from Yahoo Finance.

4. Data Cleaning and Preparation

Dates were aligned by converting timestamps to simple dates. Missing prices on non-trading days were forward-filled. Invalid or delisted tickers were removed. Frequently occurring tickers were prioritized to ensure enough data for modeling.

5. ARIMA Modeling

ARIMA stands for AutoRegressive Integrated Moving Average.

AutoRegressive uses past prices as hints. Integrated removes trends by differencing. Moving Average learns from past errors. ARIMA is interpretable but limited for complex patterns.

6. Transition to LSTM

ARIMA is linear while markets are nonlinear. LSTM is used to capture complex dependencies.

7. LSTM Workflow

LSTM remembers long-term information. It uses memory cells and gates.

Forget gate removes irrelevant info.

Input gate stores important info.

Output gate controls prediction output.

Data is normalized and converted into sliding windows. Chronological train-test split is used.

8. Adding Sentiment

TextBlob assigns polarity scores from -1 to +1. Daily sentiment is averaged. Sentiment is used as an additional feature.

9. Single Ticker Choice

Using many tickers adds complexity and imbalance. Therefore the most frequent ticker is chosen for sentiment modeling to isolate sentiment impact.

10. LSTM With vs Without Headlines

One model uses only price. Another uses price plus sentiment. Comparing them shows headline contribution.

Stock Price Forecasting using ARIMA, LSTM, and LSTM with Sentiment Analysis – Part 2

11. Detailed Modeling Pipeline

After preparing and cleaning the dataset, the next stage focuses on a structured modeling pipeline.

The pipeline ensures reproducibility and fair comparison. It includes normalization, sequence generation,

training, prediction, and evaluation.

12. Price Normalization

Prices vary widely, so Min-Max scaling converts values to a 0–1 range.

This stabilizes neural network training. Scaling is done per ticker.

13. Sequence Creation

A sliding window is used. For example, 30 past days predict the next day.

This transforms time series into supervised learning samples.

14. Chronological Train-Test Split

Random shuffling is avoided. The first 80% is training, last 20% testing.

This reflects real forecasting.

15. ARIMA Training Workflow

Steps include checking trends, differencing, selecting AR and MA orders, fitting, and forecasting.

16. LSTM Model Architecture

A simple architecture is used: input layer, one LSTM layer, and a dense layer.

Simple models generalize better on noisy data.

17. Training Process

Uses backpropagation through time, MSE loss, and Adam optimizer.

Validation helps avoid overfitting.

18. Prediction Generation

Predictions are inverse-scaled back to real prices for comparison.

19. Evaluation Metrics

RMSE and MAE measure prediction error.

20. Visualization

Plots compare actual vs predicted prices.

21. LSTM with Sentiment Workflow

Includes scoring headlines, aggregating daily sentiment, aligning with prices, and combining features.

22. Sentiment Behavior

Sentiment fluctuates quickly and may not immediately affect prices.

23. Comparing LSTM Models

One model uses price only, another uses price plus sentiment.

24. Observed Behavior

LSTM captures smoother trends. ARIMA is slower to react.

25. Feature Alignment Importance

Correct date alignment ensures realistic learning.

Stock Price Forecasting using ARIMA, LSTM, and LSTM with Sentiment Analysis – Part 3

26. Experimental Design

The experiments are designed to ensure fairness between models.

Each model is trained and tested on the same time periods.

This prevents bias and allows meaningful comparison.

27. Baseline Establishment

ARIMA acts as a baseline.

If a complex model cannot outperform ARIMA,

its usefulness is questionable.

28. Hyperparameter Choices

Window size, epochs, and learning rate are chosen carefully.

Very large networks are avoided to reduce overfitting.

29. Role of Window Size

Short windows capture recent momentum.

Long windows capture broader trends.

A balanced window is selected for stability.

30. Data Leakage Prevention

Future data is never used in training.

Scaling parameters are learned only from training data.

31. Stability of Training

Multiple runs can give slightly different results.

This happens due to random initialization.

32. Price Dynamics Observed

Some stocks show trending behavior.

Others show sideways movement.

Models adapt differently to each pattern.

33. Reaction to Sudden Changes

Sudden jumps are difficult to predict.

They often come from unexpected news.

34. Headline Timing Impact

Morning news may influence same-day prices.

Late news may reflect next-day movement.

35. Sentiment Lag Effect

Sometimes sentiment leads price.

Other times price moves first.

36. Model Sensitivity

LSTM reacts strongly to recent movements.

This can help or hurt depending on noise.

37. Overfitting Awareness

If training error is very low but test error high,

the model is memorizing noise.

38. Importance of Simplicity

Simple architectures often generalize better.

39. Computational Cost

Deep models require more time than ARIMA.

This matters in real applications.

40. Reproducibility

Using fixed seeds helps reproducibility.

41. Visualization Insights

Graphs reveal trend capture better than numbers.

42. Interpretation Caution

Prediction does not equal certainty.

It reflects probability patterns.

43. Multi-Factor Nature of Markets

Prices depend on many hidden variables.

44. Information Flow Concept

Markets digest information continuously.

45. Behavioral Influence

Investor psychology plays a role.

46. News Saturation

Too many headlines can dilute signal.

47. Neutral Headlines

Many financial headlines are neutral.

48. Model Comparison Fairness

All models see the same data splits.

49. Practical Learning Outcome

The project improves understanding of time-series ML.

50. Methodology Value

A structured workflow matters as much as model choice.