Modeling Tesla Stock Trends with ARIMA and Recurrent Neural Network LSTM

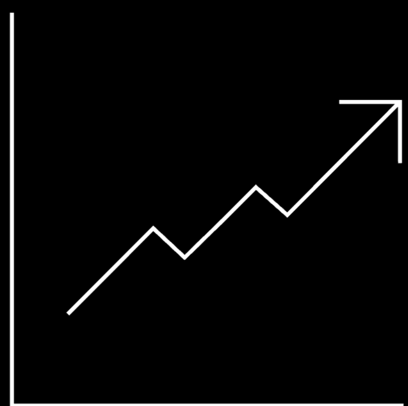
Julie Seo

Sangmyeong Lee

Objective

- To forecast Tesla Stock Prices using two different time series modeling:
    - LSTM (Recurrent Neural Network)
    - ARIMA (Classical Statistical model)
- To compare the performance of traditional statistical forecasting vs. deep learning methods
    - Are good evaluation metric(e.g. RMSE) indicate good performance in real-world trading?
    - Does neural network always outperform than classical statistic method?

Dataset



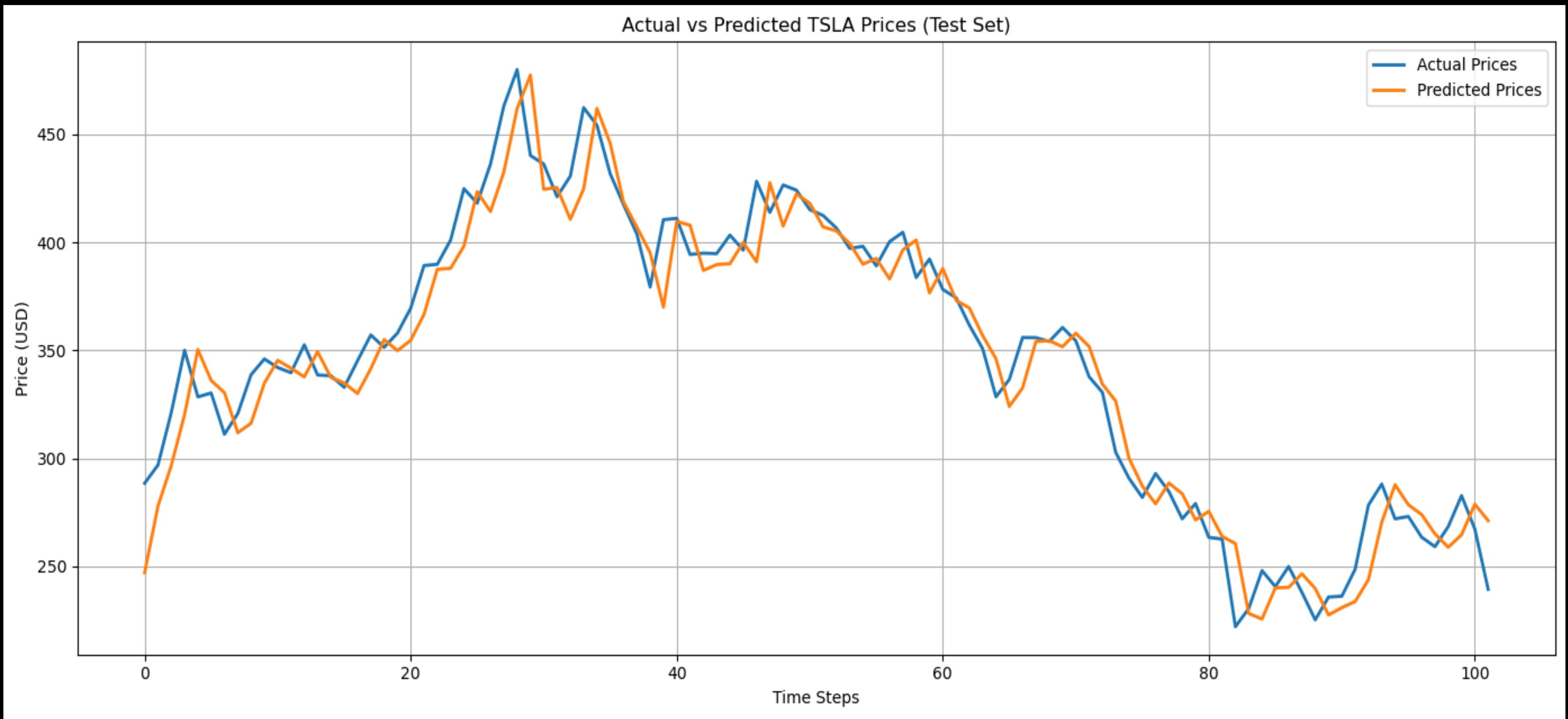
- **Source**: yfinance API from Yahoo Finance
- **Stock**: Tesla (TSLA)
- **Time Range**: Jan 1, 2021 – April 4, 2025
- **Train, test**: 80% for training, 10% for validation, 10% for testing

LSTM Model 1 vs. LSTM Model 2

Aspect	Model 1	Model 2	Explanation & Reasoning
Target Variable	TSLA_Close (normalized)	TSLA_target (next-day close price)	Model 1 predicts the actual closing price (scaled), Model 2 explicitly defines a target-shifted column to predict the **next day close price**.
Features Used	Close, MA10, MA20	Open, High, Low, Close, Volume, NASDAQ Index	Model 1 focuses on technical indicators. Model 2 includes **external index**, trying to model TSLA's behavior with broader **market context and price action**.
Input Sequence	60 days	30 days	Model 1 uses a **longer memory.** Model 2 is more **responsive to recent fluctuations**.
Normalization	Close, MA10, MA20 (MinMaxScaler)	Full feature + target scaling	Model 1 is simpler and avoids leaking future info. Model 2 standardizes **everything.**
Architecture	LSTM(128) → LSTM(64)	LSTM(64) → LSTM(32)	Model 1 is **deeper** and potentially more expressive. Model 2 is **lighter** – this may reduce overfitting, especially given the higher input dimensionality.

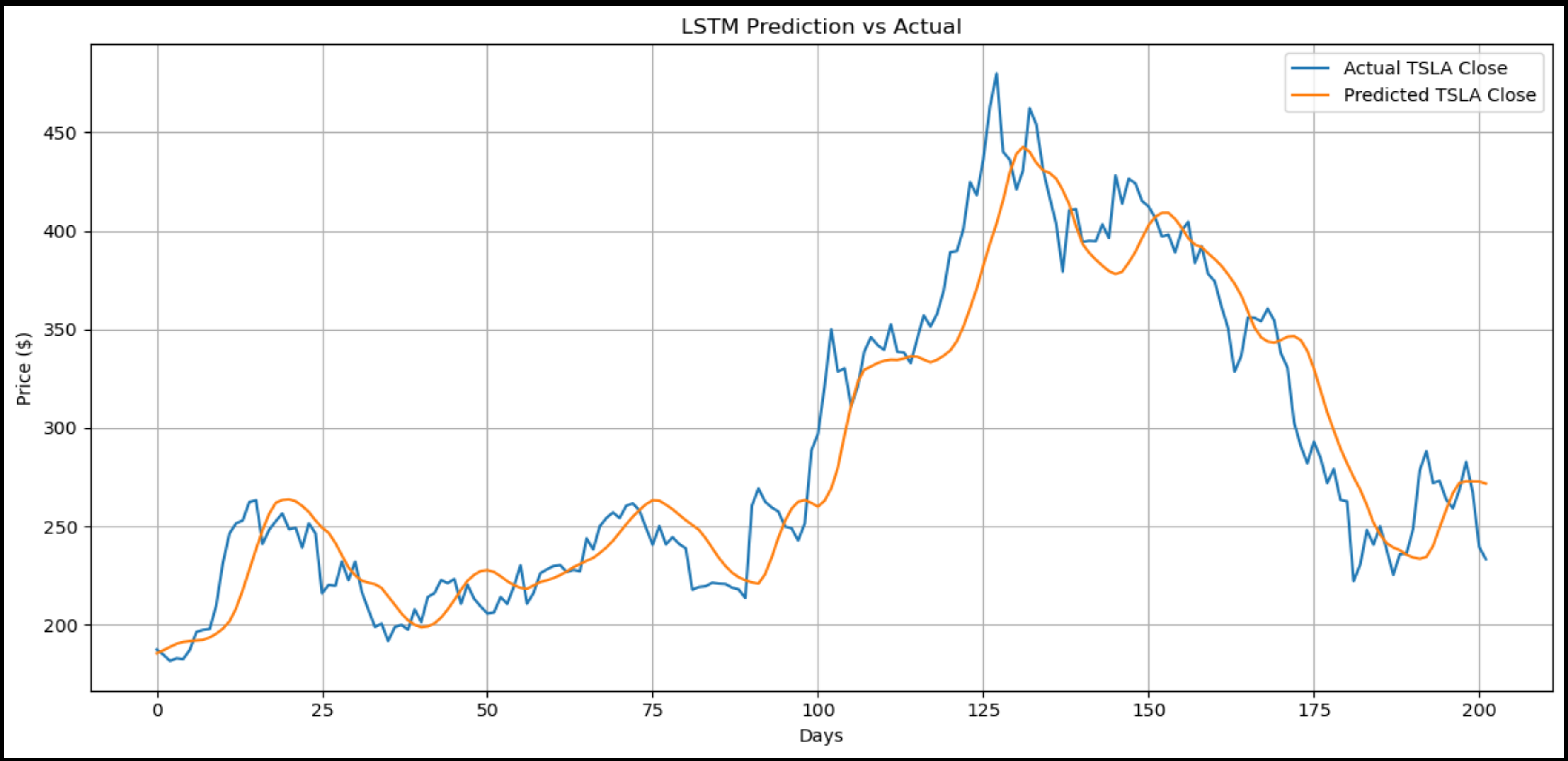
Dropout Regularization	None	0.2 after each LSTM layer	Model 2 adds **dropout** to prevent overfitting, more generalization — especially with more features. Model 1 doesn't, which might explain its **slightly better training fit**, but less generalization.
Validation Split	Manual x_val, y_val	validation_split=0.1 in Keras	Model 1 uses **explicit split.** Model 2 uses 'Keras' **automatic slicing**.
RMSE (Test Set)	16.63	24.51	Model 1 has a better RMSE — due to **simpler features + less overfitting**.
Relative RMSE	4.71%	8.51%	
Directional Accuracy	53.37%	55.22%	Model 2 slightly better.
Naive RMSE	15.35	12.96	Naive benchmarks (predict tomorrow = today)

LSTM Model 1



Actual vs Predicted TSLA Prices (Test Set)
Actual Prices
Predicted Prices
Price (USD)
Time Steps
450
400
350
300
250
0
20
40
60
80
100

LSTM Model 2


LSTM Prediction vs Actual
Actual TSLA Close
Predicted TSLA Close
Price ($)
Days
450
400
350
300
250
200
0
25
50
75
100
125
150
175
200

ARIMA

Raw Plot

Variable used: TSLA_Close (daily closing prices)

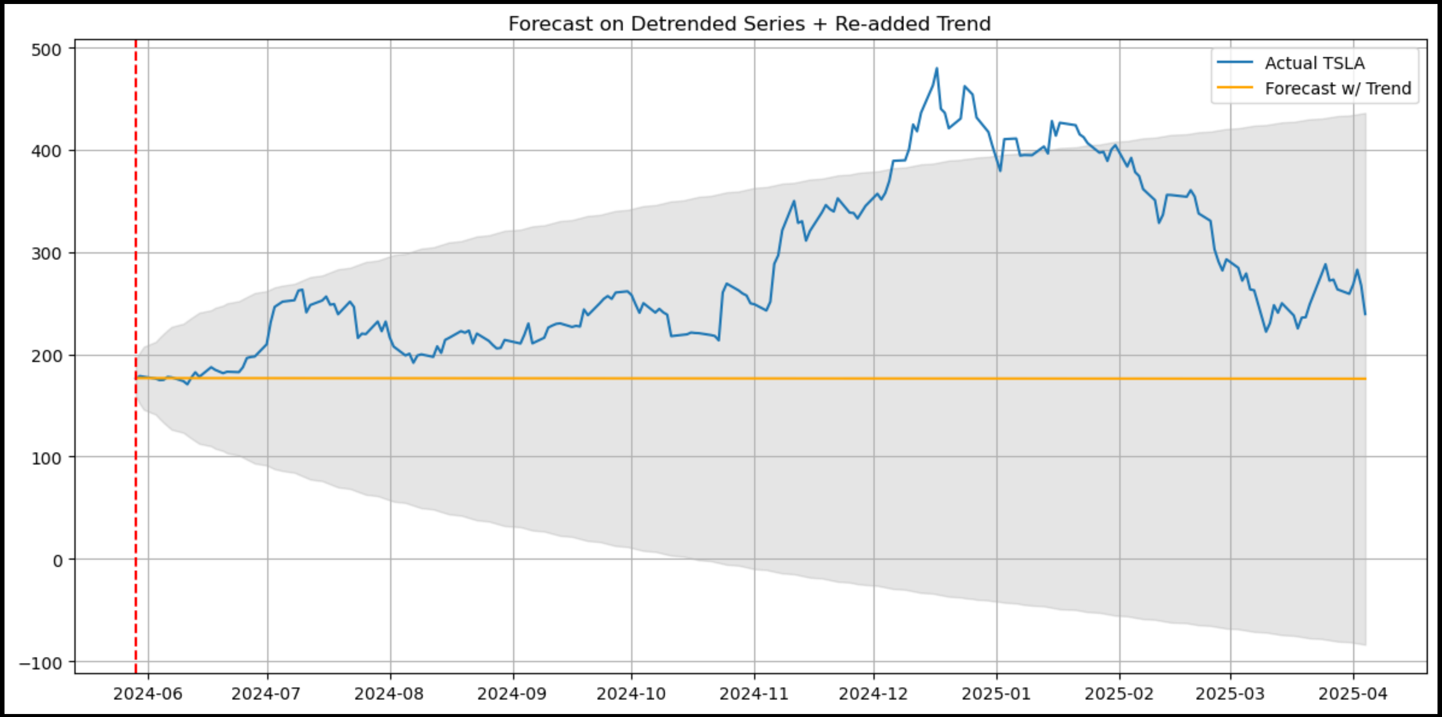
➢ High volatility, Non-stationarity appearance

Detrending

- Fit Linear Regression: $$TSLA_{Clost_t} = \beta_0 + \beta_1 t$$
- Residual (detrended TSLA price): $$Z_t = X_t - \beta_o + \beta_1 t$$

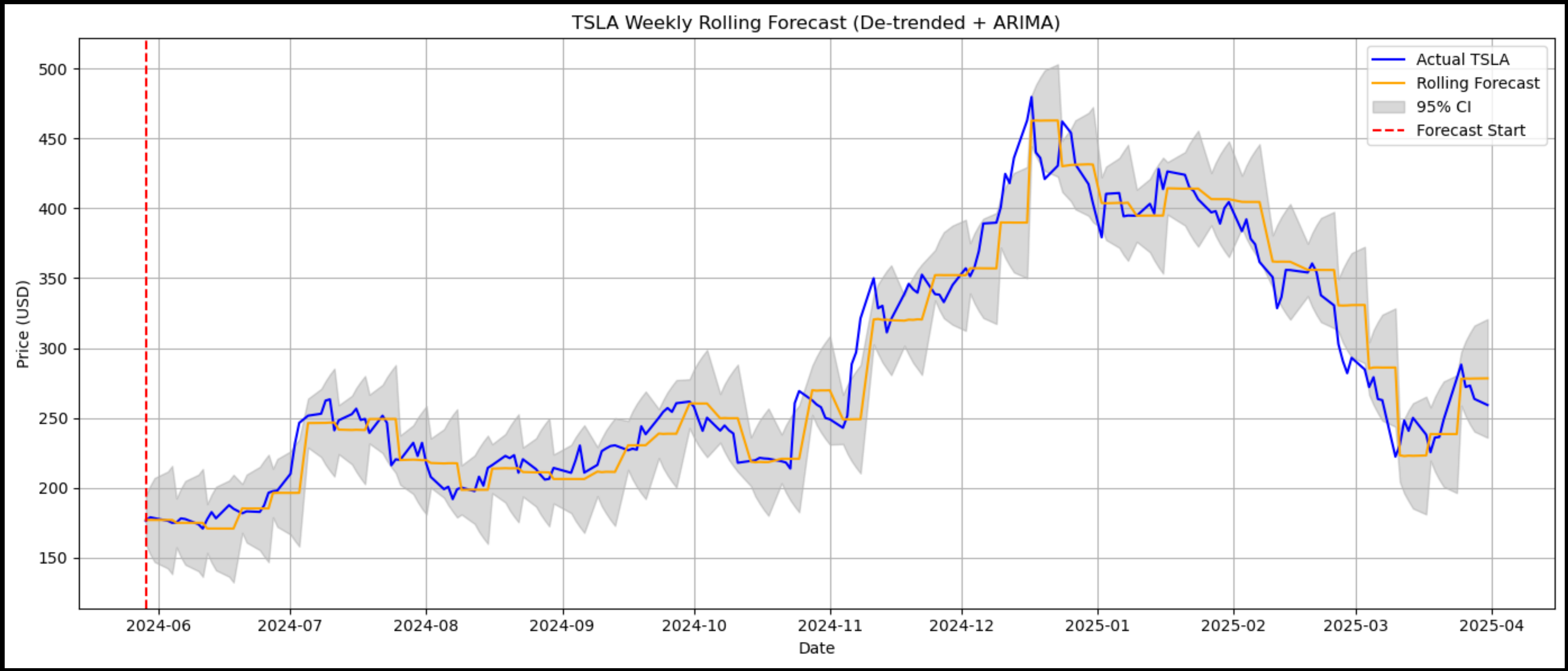
ARIMA on Detrended Series

- Fitted ARIMA(0,1,0) on detrended TSLA data
- Forecasted is made on residuals, then trend is re-added



TSLA weekly Rolling Forecast (de-trended + ARIMA)

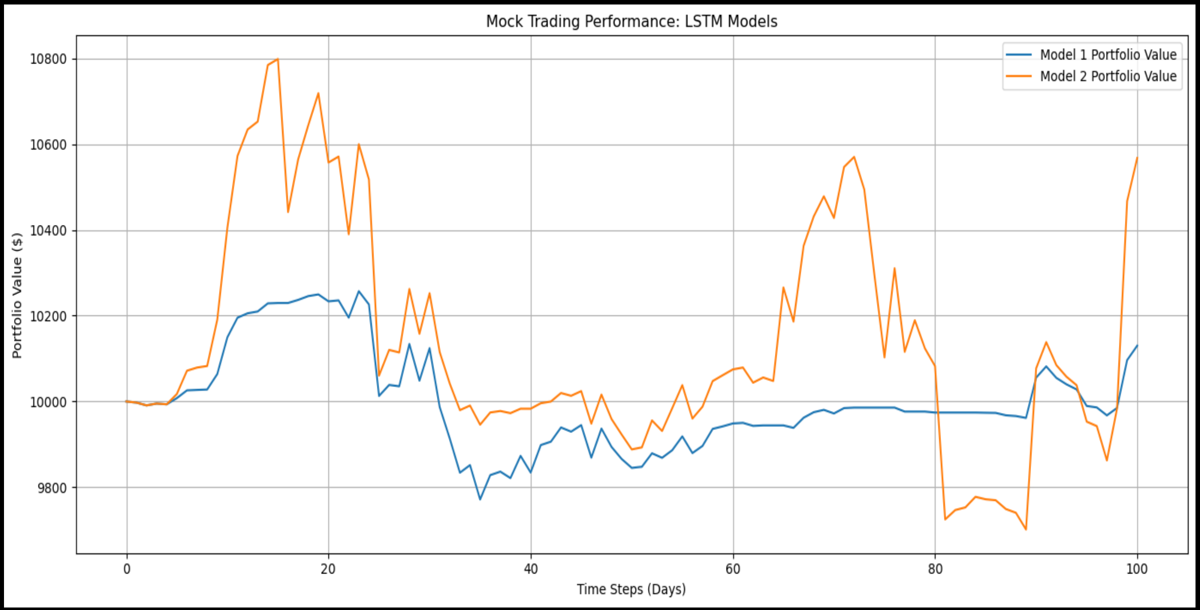
- Applied rolling ARIMA model (weekly updates) to residuals
  - Each forecast is updated using the most recent data



Model Evaluation

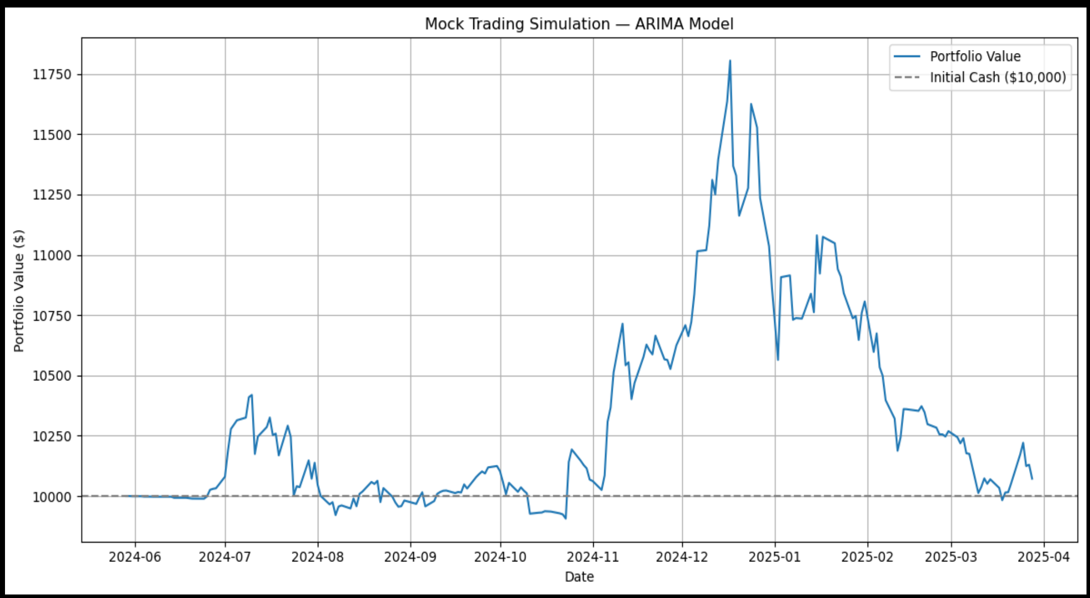
Evaluation Metrics	Result
RMSE	19.88
MAPE	5.06%
MAE	14.46
Directional Accuracy	48.33%
Relative RMSE	7.05%
Naïve RMSE	12.50

LSTM Mock Trading



- Initial cash: $10,000
- **Model 1** Final Portfolio Value: $9,991.46
  - Total profit: -$8.54
  - Profit percentage: -0.0854%
- **Model 2** Final Portfolio Value: $10,567.73
  - Total profit: $567.73
  - Profit percentage: 5.6773%

ARIMA Mock Trading



- Initial cash: $10,000
- Final Portfolio Value: $10,072.18
  - Total Profit: $72.18
  - Profit Percentage: 0.72%

Conclusions

- LSTM model 1 had the lowest RMSE but produced a smallest trading profit
- LSTM model 2 had higher RMSE but achieved the highest profit in mock trading
- ARIMA model had moderate accuracy but made a small profit
- Statistical accuracy (like RMSE) doesn't guarantee better trading outcomes.
    - directional accuracy and feature design matter
- Neural networks aren't always better

Limitations

- Limited input futures
  - Macro Economic indexes such as GDP, CPI, inflation rates and other external factors are not reflected
- Market conditions can shift drastically (past trends may not reflect the future)
- Directional accuracy remains low despite the low RMSE/MAE
- Mock trading doesn't account for the transaction costs, or real-world latency


Thank you!