LSTM2 REPORT

Data Description

The dataset used for this analysis consists of **historical stock price data** for a specified ticker symbol. The data was retrieved using the `yfinance` library, which collects stock market data from **Yahoo Finance**.

The dataset begins on **January 1, 2020**, and continues up to the current date at the time the code is executed. For the purpose of this model, we focused solely on the **closing price** of the stock for each valid trading day.

```
# Print first few rows to confirm data is loaded correctly
print("Stock Data Preview:\n", data.head())
```

```
YF.download() has changed argument auto_adjust default to True
[*********************100%***********************]  1 of 1 completed
Stock Data Preview:
 Price            Close
Ticker           COST
Date
2020-01-02  267.592499
2020-01-03  267.812775
2020-01-06  267.886292
2020-01-07  267.463928
2020-01-08  270.530151
```

Tuning Parameters

The key tuning parameters used in our LSTM model are:

- **Learning Rate:** Controls how quickly the model updates its weights during training. A smaller learning rate results in slower, more precise updates, while a larger rate speeds up learning but risks overshooting the optimal solution.
- **Batch Size:** Defines the number of samples the model processes before updating its weights. Smaller batch sizes can lead to more responsive learning, while larger batches make updates smoother but less sensitive to short-term fluctuations.

- **Number of LSTM Layers:** Specifies the depth of the model. Adding more layers can allow the model to capture more complex patterns but also increases the risk of over fitting and longer training times.

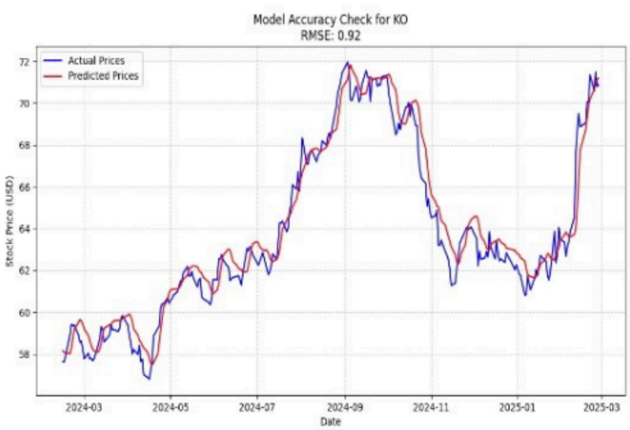
Impact of Tuning Parameters

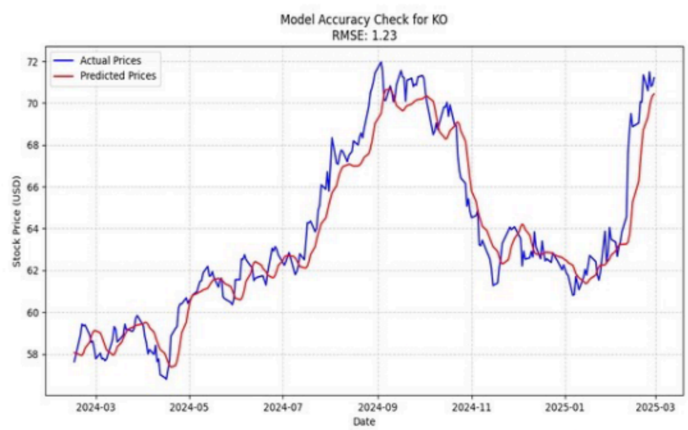
We used **Root Mean Squared Error (RMSE)** to evaluate model performance because it directly measures the average magnitude of prediction errors. RMSE penalizes larger errors more heavily, making it suitable for assessing how well the model captures stock price trends. A lower RMSE indicates closer alignment between predicted and actual values. In all plots shown after this, the **modified model outputs are shown on the right**.

Learning Rate

Table 1: Comparison of Default and Modified Learning Rate Settings

Configuration	Learning_Rate	LSTM_Layers	Epochs	Batch_Size	RMSE
Default	0.001	2	50	32	0.92
Modified (Lower LR)	0.0005	2	50	32	1.23



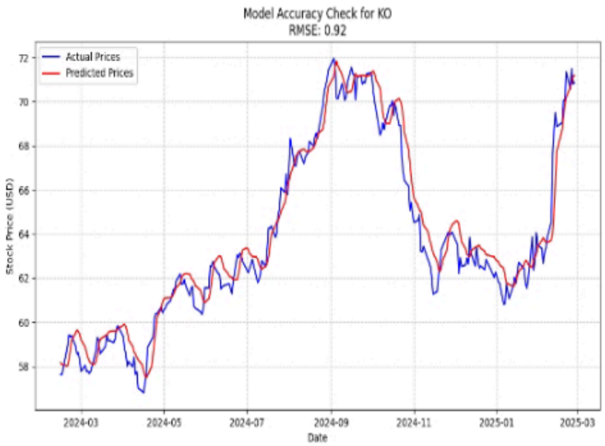


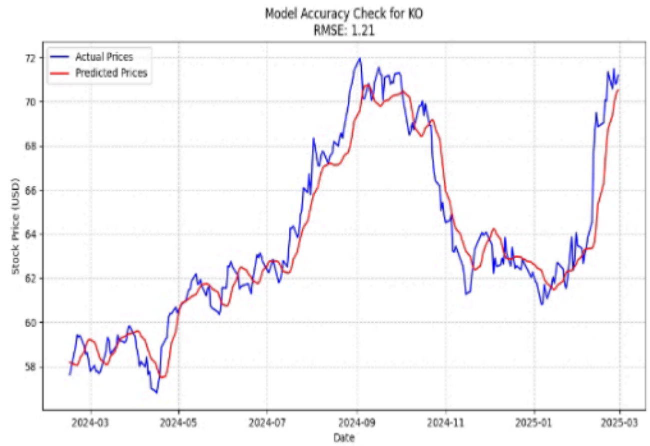
Lowering the learning rate to **0.0005** slowed convergence, preventing the model from fully learning stock price patterns within **50 epochs**, leading to underfitting. The smaller updates caused less responsiveness to market fluctuations, reducing prediction accuracy. As a result, RMSE increased from **0.92** to **1.23**.

Batch Size

Table 2: Comparison of Default and Larger Batch Size Settings

Configuration	Learning_Rate	LSTM_Layers	Epochs	Batch_Size	RMSE
Default	0.001	2	50	32	0.92
Modified (Larger Batch)	0.001	2	50	64	1.21



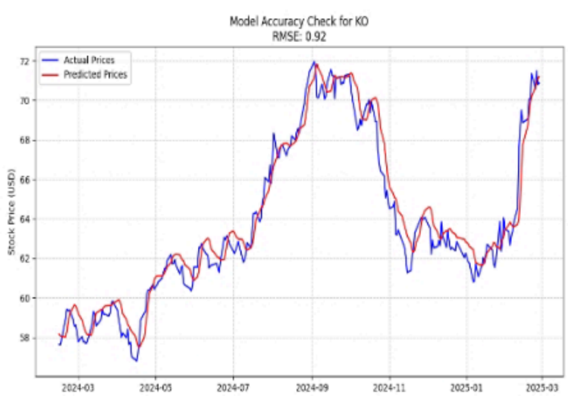


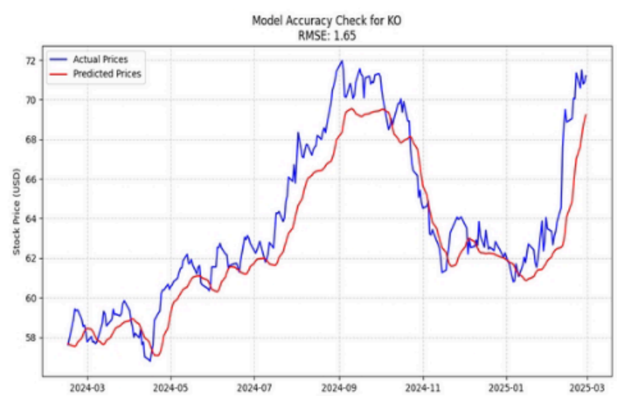
Increasing the batch size from **32** to **64** led to a higher RMSE ($\mathbf{0.92} \rightarrow \mathbf{1.21}$), indicating a slight drop in accuracy. The larger batch size made training smoother but reduced the model's ability to capture fine stock price fluctuations. This suggests that batch size = 32 provides better generalization for this dataset.

Amount of Layers

Table 3: Comparison of Default and Extra LSTM Layer Settings

Configuration	Learning_Rate	LSTM_Layers	Epochs	Batch_Size	RMSE
Default	0.001	2	50	32	0.92
Modified (Extra Layer)	0.001	3	50	32	1.65





Adding an extra LSTM layer increased RMSE from 0.92 to 1.65, reducing model accuracy. The deeper network may have overfitted or struggled to generalize stock price patterns effectively. Additionally, we observe that the predicted values in the modified model appear smoothed out, indicating a loss of sensitivity to short-term price fluctuations.

Limitations of Using LSTM for Stock Price Prediction

- **Prone to Overfitting:** LSTM's can memorize historical price patterns but often fail to generalize to future market conditions, especially in the presence of noise or sudden shifts.
- **Limited Ability to Capture External Factors:** Stock prices are heavily influenced by external events such as political decisions, economic announcements, and unexpected global events. Since LSTMs operate purely on numerical inputs, they do not inherently account for these external drivers unless explicitly included through additional data sources like news sentiment.
- **Sensitive to Market Volatility and Noise:** Highly volatile stocks, particularly in the tech sector (e.g., Microsoft, Nvidia, Apple), exhibit abrupt price movements triggered by events like earnings reports or product launches. LSTM's may misinterpret these as meaningful trends, reducing forecasting accuracy.
- **Computationally Expensive:** Training LSTM's, especially deeper architectures, demands considerable processing power and time. This can be a limitation for rapid iteration or when working with large-scale datasets.

Sources

A beginner's guide to loss functions for regression algorithms. (2022, November 18). DataMonje. https://datamonje.com/regression-loss-functions/

LABS, S. (2019, March 21). *Understanding deep learning: Dnn, rnn, lstm, CNN and R-CNN.* Medium. https://medium.com/@sprhlabs/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff

Understanding LSTM networks. (n.d.). colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/