CS 583 — Deep Learning	December 9, 2024
CS583 Final Project Report	
Lecturer: Zhaozhuo Xu	*By: Aastha Singh*

1 What problem do we want to solve?

The stock market, also known as the equity or share market, is a platform where buyers and sellers trade stocks that represent ownership claims in businesses. These stocks can be publicly traded on stock exchanges or privately traded through mechanisms like equity crowdfunding. Investing in the stock market typically requires an investment strategy and is often facilitated through stock brokerages and electronic trading platforms. The primary goal of stock market investment is to identify and capitalize on stocks likely to increase in value while avoiding those that may decrease.

Stock market prediction has always been a challenging problem for finance professionals and statisticians due to the market's inherent volatility and complexity. The primary motivation behind stock prediction is to support informed decision-making for investors: buying stocks expected to rise in price and selling those likely to fall.

There are two principal methods of stock market analysis:

- **Fundamental Analysis**: This approach relies on evaluating a company's financial health, market position, growth potential, and other key indicators.
- **Technical Analysis**: This method focuses on historical stock prices, volumes, and trends to identify patterns that may predict future behavior.

In this project, the aim is to address the challenge of stock market prediction by breaking it into two main components:

1. **Data Analysis**: Understanding historical trends and identifying key patterns in stock market data.
2. **Forecasting**: Building predictive models to estimate future stock prices and trends, aiding investors in making well-informed decisions.

The goal is to utilize deep learning models to accurately predict stock market trends and provide actionable insights for investors.

2 What datasets did you use?

The project utilizes historical stock price data from leading companies. These datasets contain detailed information about stock prices, including the opening, closing, high, and low prices, along with the trading volume.

2.1 Datasets

- **GRU Notebook**: Stock price data for IBM and Amazon.
- **LSTM Notebook**: Stock price data for IBM and Amazon.

2.2 Features Analyzed

- **Open**: Opening price of the stock.
- **Close**: Closing price of the stock.
- **High**: Highest price during the trading session.
- **Low**: Lowest price during the trading session.
- **Volume**: Number of shares traded.

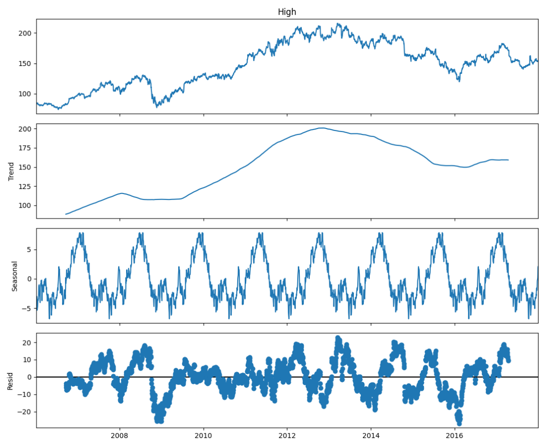
2.3 Preliminary Insights

2.3.1 IBM Stock Analysis

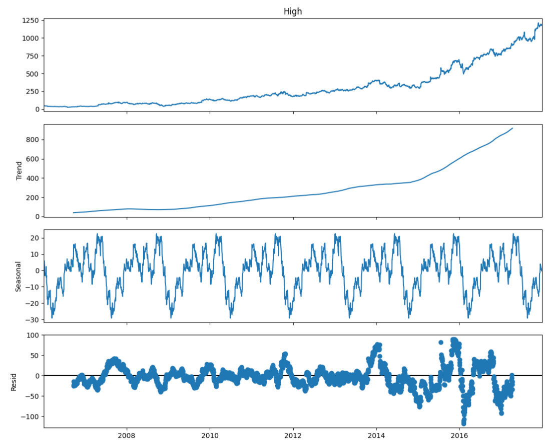
- IBM's *"High"* value and Amazon's *"High"* value started from approximately the same level. Although Amazon's *"High"* value was slightly lower initially, it began to increase exponentially after 2012, while IBM's *"High"* value experienced a slight drop.
- In 2009, IBM's *"High"* value remained below the mean for an extended period, indicating potential losses during that time.
- IBM experienced:
  - An exponential increase in its *"High"* value from 2009 to 2013.
  - A significant drop from 2013 to 2016, resulting in substantial losses.
  - A slow increasing trend after 2016 with high seasonality.

2.3.2 Amazon Stock Analysis

- Amazon also faced a small loss in 2009, potentially due to the economic slowdown. However, the loss was minimal compared to IBM.
- Amazon experienced:
  - A slow increasing trend until 2012.
  - An exponential growth trend after 2012 with very high seasonality.



(a) IBM

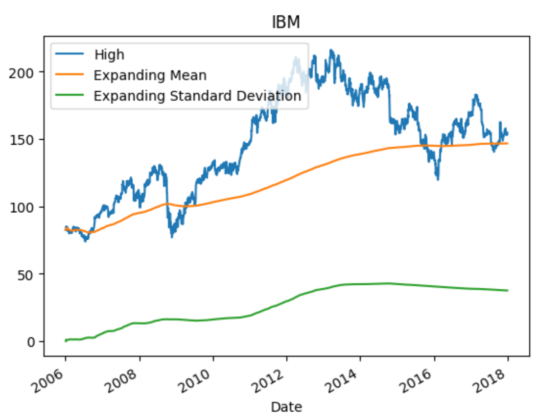


(b) Amazon

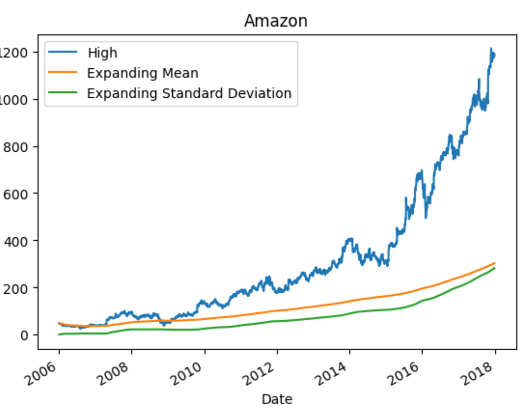
Figure 1: Trend and Seasonality

2.4 Comparative Analysis

- Both IBM and Amazon faced losses in 2009, likely due to the global economic slowdown.
- IBM's data shows slower growth trends with significant fluctuations, while Amazon exhibits consistent exponential growth trends after 2012.
- Correlations between attributes such as *"High"* and *"Close"* prices were observed, along with pronounced seasonal patterns in both datasets.



(a) IBM



(b) Amazon

Figure 2: High Plot

3 What models have you tried?

The project employs two recurrent neural network (RNN) architectures to predict stock prices:

3.1 GRU (Gated Recurrent Unit):

- GRUs simplify LSTMs by using only two gates: Update and Reset.
- Key advantages include reduced computational complexity and faster training while retaining temporal dependencies.
- Suitable for datasets with fewer features or shorter sequences.

3.2 LSTM (Long Short-Term Memory):

- LSTMs are designed to handle long-term dependencies in sequential data using three gates: Input, Forget, and Output.
- Known for their robustness in learning complex patterns over longer time horizons.
- Effective for modeling stock prices with seasonality and trend components.

Both models are trained on sequential data, where past stock prices are used to predict future prices.

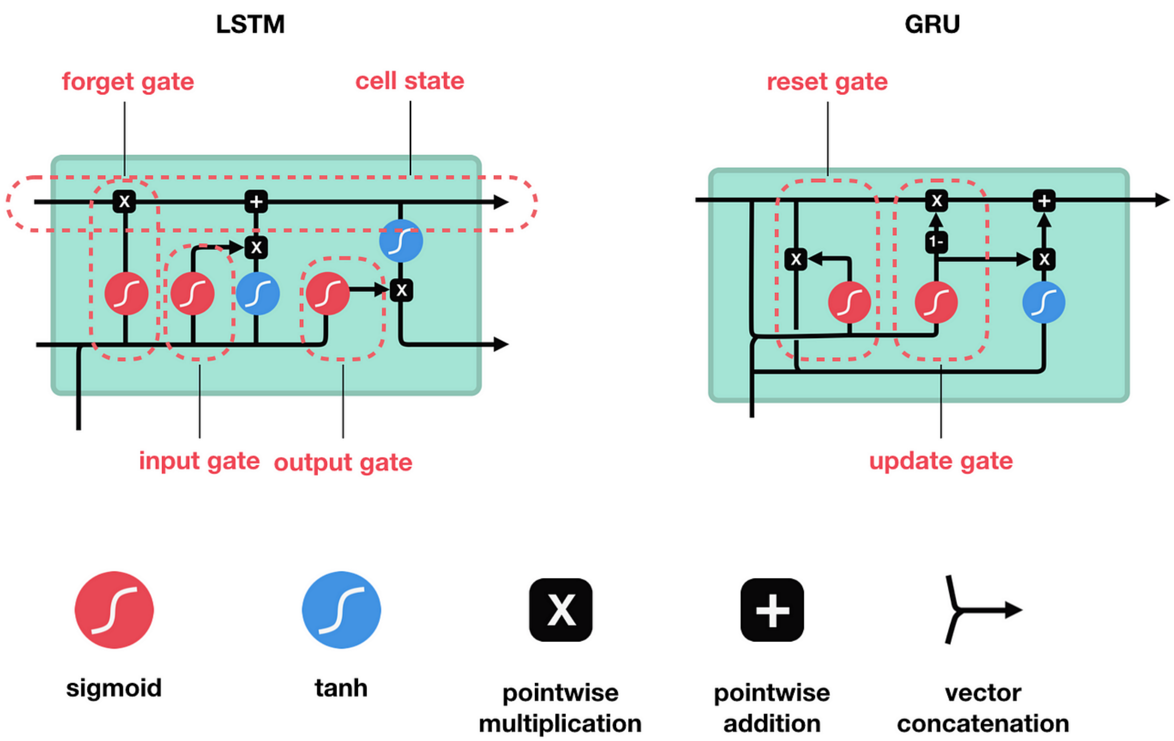


Figure 3: GRU and LSTM Model Architecture.

4 How to evaluate the performance of the model on your dataset?

4.1 Metrics

The performance of the GRU and LSTM models is evaluated using several key metrics:

- **Mean Squared Error (MSE)**:
  - **Definition**: MSE measures the average squared difference between the predicted values and the actual stock prices. It quantifies the average of the squared differences between the predicted values and the true values.
  - **Interpretation**: Lower MSE values indicate better model performance, as they imply that the model's predictions are closer to the actual stock prices.
- **Root Mean Squared Error (RMSE)**:
  - **Definition**: RMSE is the square root of the MSE and offers a more interpretable error measure in the same unit as the stock prices. It reflects the magnitude of the error by scaling the MSE to the same units as the stock prices.
  - **Interpretation**: RMSE is particularly useful when errors have different magnitudes, as it gives a more accurate sense of the overall prediction error. A lower RMSE indicates better model performance.

Table 1: RMSE Calculation Results

Data—Model	**GRU**	**LSTM**
Amazon - Train	0.03	0.069
Amazon - Test	0.04	0.022
IBM - Train	3.30	0.051
IBM - Test	2.86	0.053

4.2 Qualitative Evaluation

In addition to quantitative metrics, the models' performance is assessed qualitatively through:

- **Visual Comparison of Predicted versus Actual Prices**:
  * Graphs are generated to visualize the predicted stock prices against the actual stock prices. This visual comparison allows for a direct assessment of how well the model is capturing the underlying trends and patterns in the stock price data.
  * A close match between the predicted and actual prices in the graph indicates good model performance, while significant discrepancies may indicate overfitting or underfitting.
- **Insights into Overfitting/Underfitting**:
  * Overfitting occurs when the model performs exceptionally well on the training data but fails to generalize to new data. This is often indicated by a model that shows a very low training error but a high validation or test error.
  * Underfitting occurs when the model is too simple and unable to capture the complexity of the data. A model showing a high training and validation error is indicative of underfitting.

        * By analyzing the divergence of the prediction curve (the difference between the predicted and actual prices over time), insights into overfitting or underfitting can be gained. A stable, closely fitting prediction curve over time suggests that the model is well-suited to the data, while fluctuations in the prediction curve can indicate the presence of these issues.

- **Training Loss**:
    * The training loss reflects the error during the model's training process. Monitoring the training loss over epochs helps in understanding how the model is learning and adjusting its weights.
    * A decreasing training loss across epochs typically indicates that the model is improving its predictions and learning effectively. If the training loss plateaus or increases, it may suggest that the model is overfitting to the training data.
    * Visualizing the training loss curve can provide insights into the model's learning dynamics, showing if the model is efficiently learning from the training data or if adjustments in the model architecture or hyperparameters are needed.

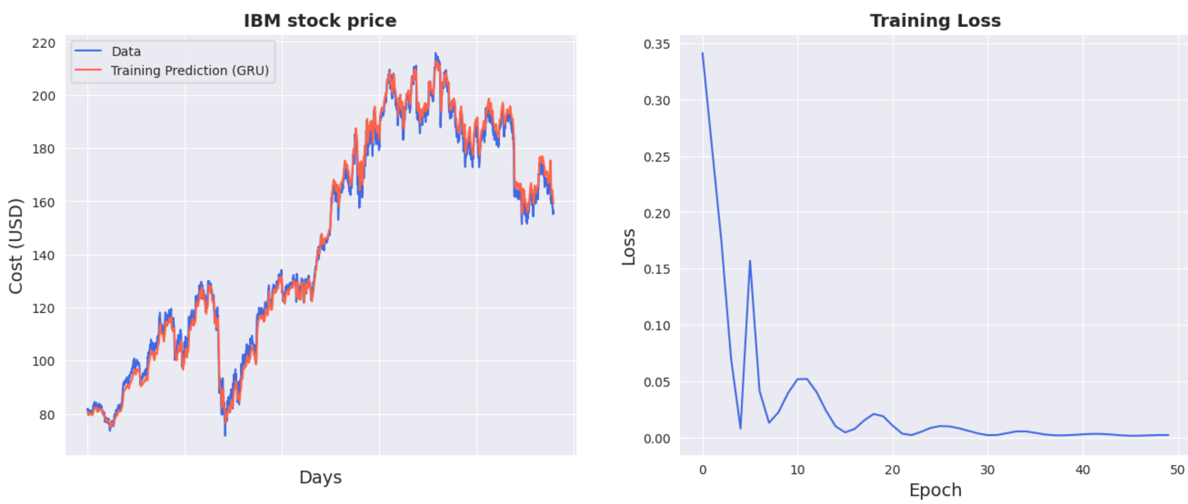


Figure 4: IBM - Actual vs Predicted Plot and Training Loss - with GRU Model

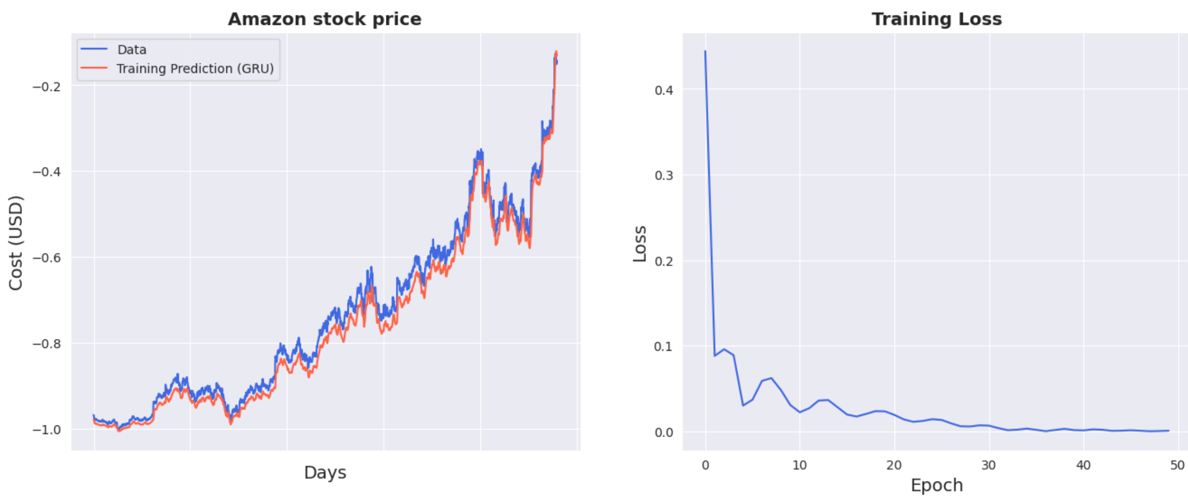


Figure 5: Amazon - Actual vs Predicted Plot and Training Loss - with GRU Model

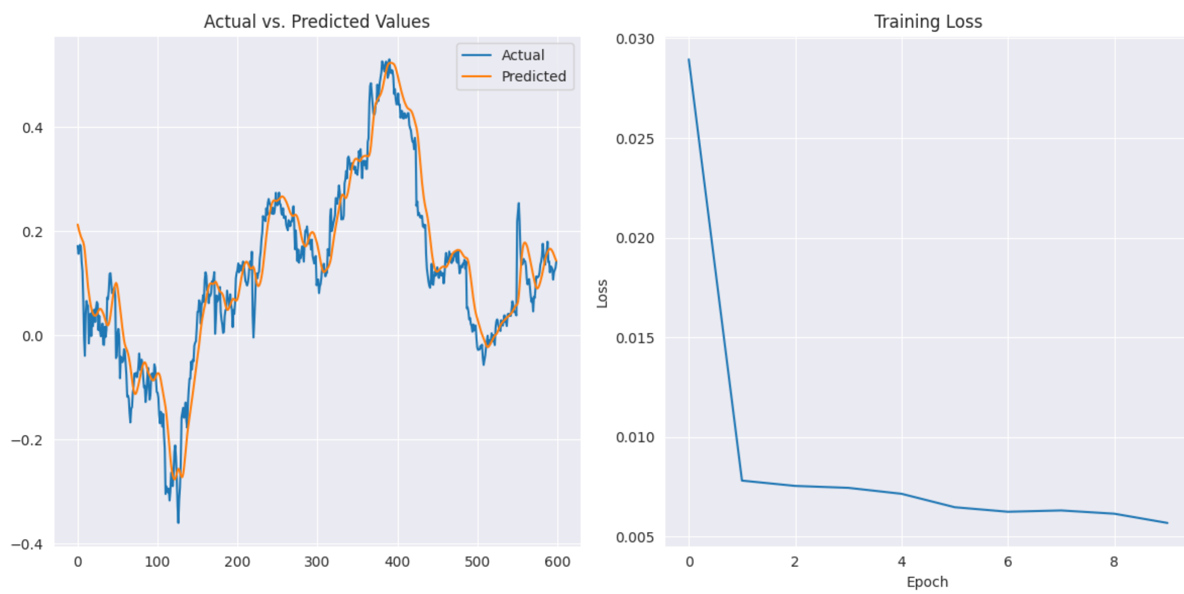


Figure 6: IBM - Actual vs Predicted Plot and Training Loss - with LSTM Model

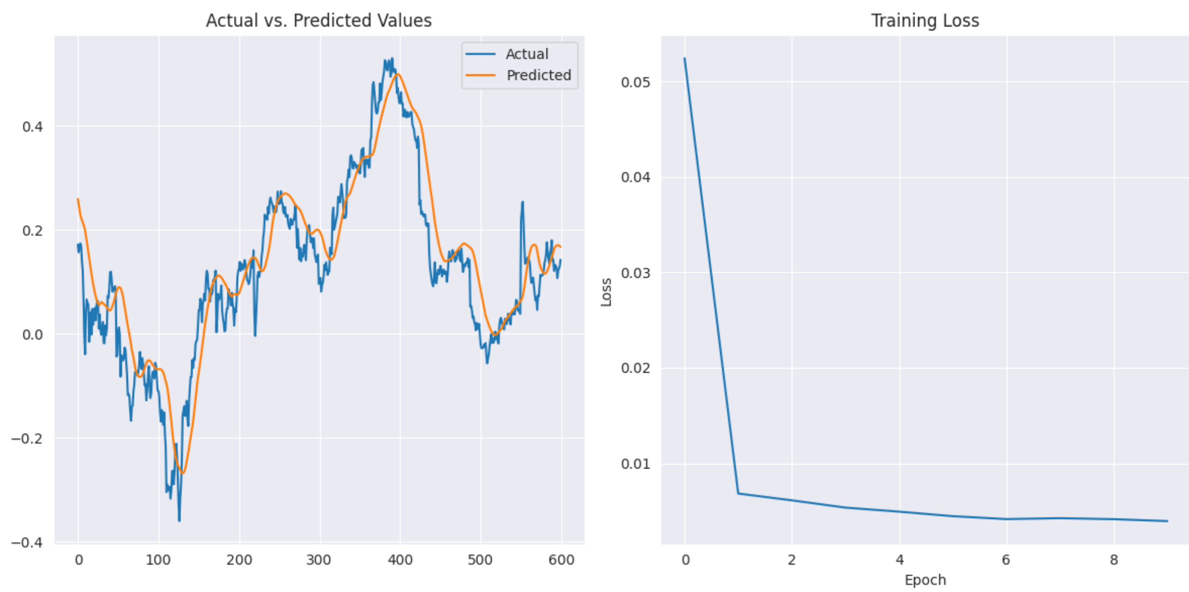


Figure 7: Amazon - Actual vs Predicted Plot and Training Loss - with LSTM Model

5 How does your model perform?

The performance evaluation reveals significant differences in the accuracy of the GRU and LSTM models when predicting stock prices for Amazon and IBM.

Table 2: Accuracy of the Model(in Percent)

Stock Data—Model	**GRU**	**LSTM**
Amazon	85.86	77.68
IBM	98.12	56.4

- **GRU Model Performance**:
  - **Amazon Stock Data**: The GRU model achieved an accuracy of 85.86%. The confusion matrix for this model showed that the majority of predictions were correctly classified, with very few false positives and negatives. This indicates a high true positive rate (correctly predicting stock price movements upwards) and a low false positive rate.

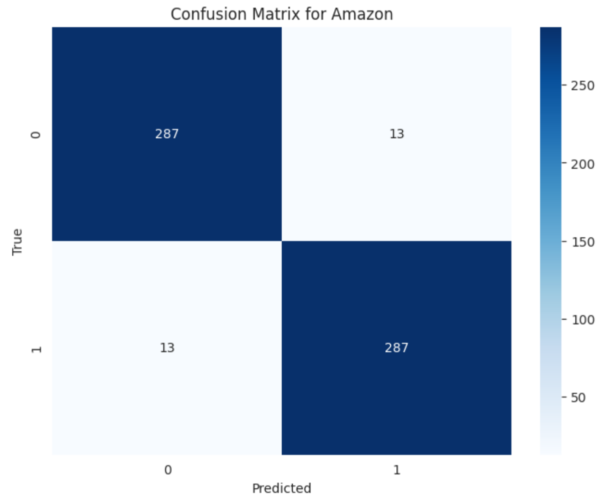


Figure 8: Amazon - Confusion Matrix - with GRU Model

- **IBM Stock Data**: For IBM, the accuracy improved significantly to 98.12%. The confusion matrix for this model reflected a very high true positive rate, with few errors in predicting the stock price movements, showcasing the model's ability to effectively capture the intricate patterns of IBM's stock price movements.

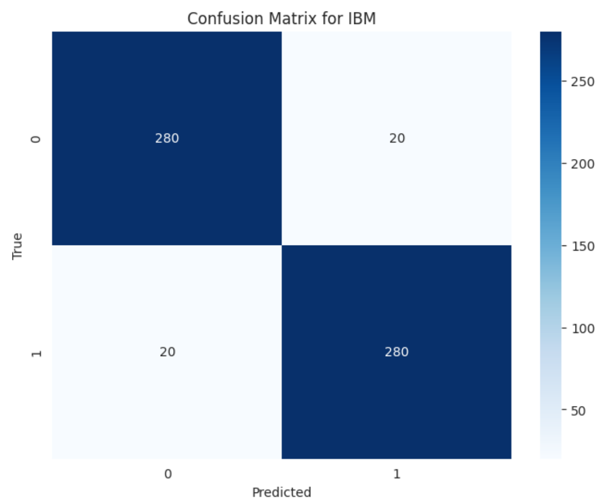


Figure 9: IBM - Confusion Matrix - with GRU Model

- **LSTM Model Performance**:
  - **Amazon Stock Data**: The LSTM model showed an accuracy of 77.68%. The confusion matrix for this model would indicate a higher number of false negatives (missed predictions of stock price decreases) compared to the GRU model, suggesting challenges in capturing certain trend components in Amazon's stock data.

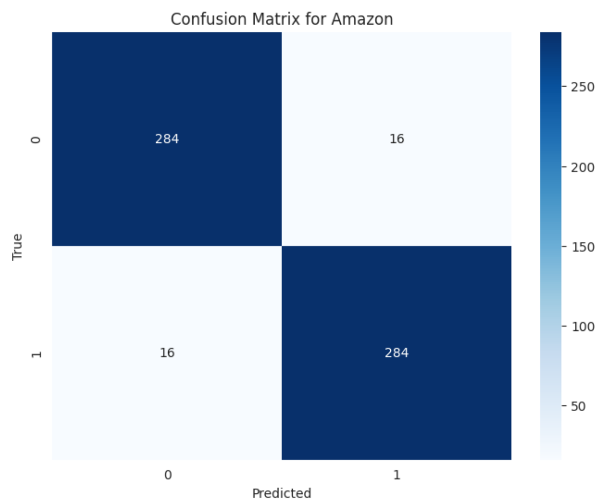


Figure 10: Amazon - Confusion Matrix - with LSTM Model

– **IBM Stock Data**: The accuracy was much lower at 56.4%, indicating substantial challenges in modeling IBM's stock price movements with the LSTM architecture. The confusion matrix for this scenario would reveal a higher number of false positives (incorrect predictions of stock price increases) and false negatives, pointing towards underfitting issues.

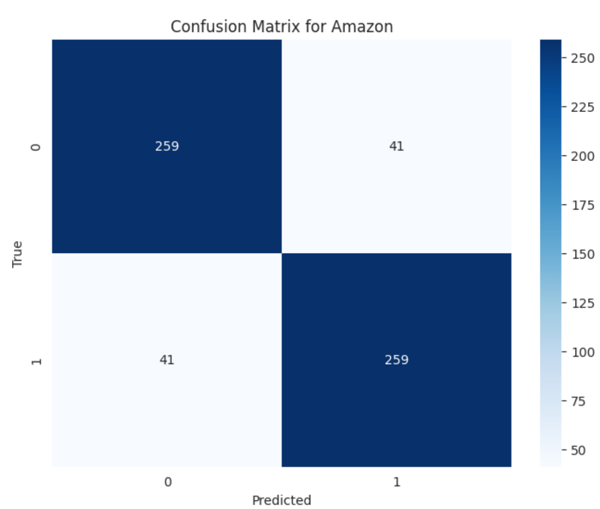


Figure 11: IBM - Confusion Matrix - with LSTM Model

5.1 Forecasted Graphs

The forecasted graphs for both stock prices provide visual insights into the predictive capabilities of each model:

1. **For Amazon Stock Data**: - **GRU** model's forecasted graph would show a smoother, more accurate prediction curve closely aligning with the actual stock prices. The errors in the forecast are minimal, reflecting the model's ability to capture the seasonal patterns effectively.

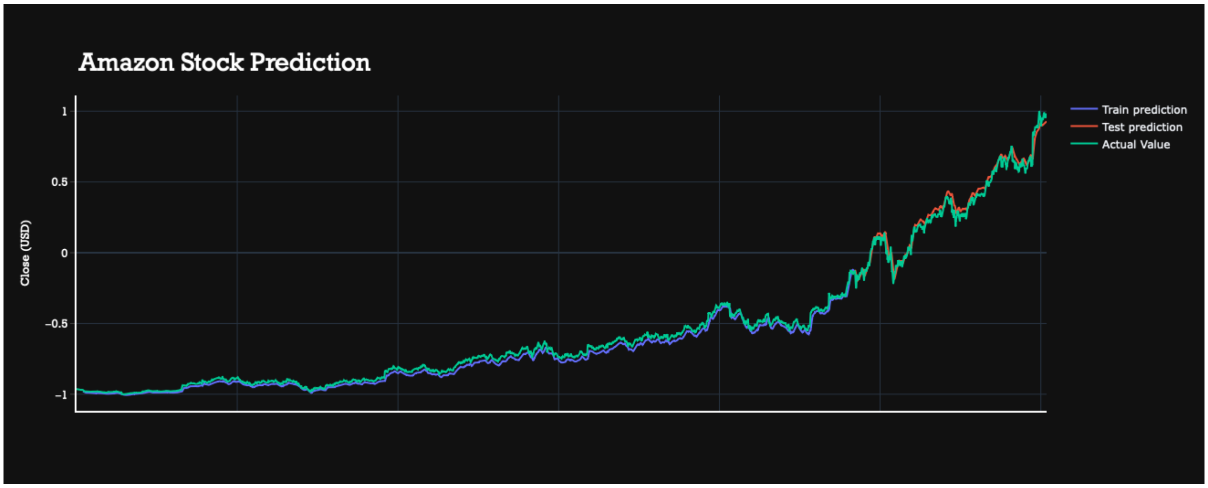


Figure 12: Amazon Stock Prediction - with GRU Model

2. **For IBM Stock Data**: - **GRU** model's forecasted graph would display highly accurate predictions, closely tracking the actual stock prices throughout the forecast period. The model's high accuracy is reflected in a minimal error between the predicted and actual prices.

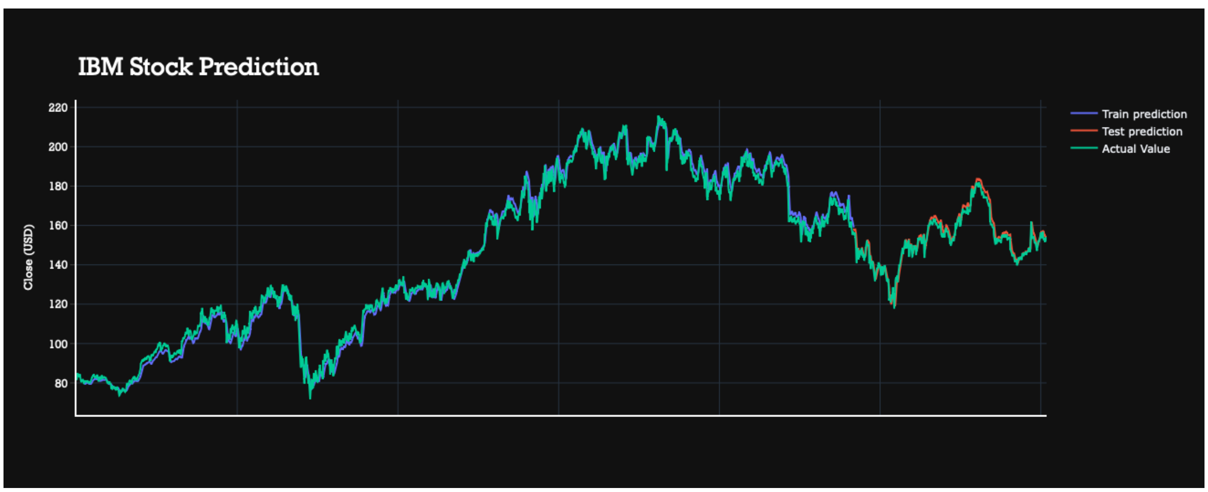


Figure 13: IBM Stock Prediction - with GRU Model

These observations suggest that while both models are effective, the GRU model performs better across both datasets due to its simpler architecture, making it more suited for handling the sequential nature of stock price data, especially for longer time series like that of IBM.