City University of Hong Kong

Semester B (2024-2025)

SDSC4107

Financial Engineering and Analytics

Time Series Analysis on DJIA 30 – IBM

Group 23

Name	SID
KYDYR Meiirbek	57161931
AIDARBEKOV Duman	57514520

11 April 2025

Table of contents

Part 1. Introduction

1.1 Background

The financial market, particularly stock trading, plays a pivotal role in the global economy. Among the various indices that track stock performance, the Dow Jones Industrial Average (DJIA) encompasses 30 of the largest publicly traded companies in the United States. While our initial plan was to analyze all 30 DJIA stocks, the complexity of managing and interpreting such a huge dataset proved to be a significant challenge. As a result, we chose for a more focused approach by selecting IBM as our subject for analysis. IBM is a long-standing leader in the technology sector and offers a rich dataset for understanding market behavior, investor sentiment, and economic trends over time. Analyzing its stock price over a 12-year period provides insights into both the company's performance and broader market dynamics.

1.2 Motivation

Our motivation arises from the increasing complexity and volatility of financial markets. Investors and analysts are continually seeking advanced methods to predict stock price movements accurately. Traditional statistical methods often fall short in capturing the intricate patterns inherent in financial time series data. By leveraging deep learning algorithms, we aim to enhance predictive accuracy and provide a more robust framework for stock price prediction. This approach not only aligns with current trends in data science but also addresses the growing need for sophisticated analytical tools in finance.

1.3 Objectives

The primary objective of this project is to conduct a comprehensive time series analysis of IBM's stock price data, focusing on key components such as price movement, trend, seasonality, and residuals. Furthermore, we aim to implement and compare the effectiveness of various machine learning algorithms, including Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks, in predicting future stock prices. Through this analysis, we seek to understand financial data engineering and provide actionable insights to navigate the complexities of stock market fluctuations.

Part 2. Data Collection and Preprocessing

2.1 Data Source and Collection

Dataset was taken from the Kaggle website and consists of 3020 rows of IBM stock price in USD. There are 7 columns:

1. Date: in YYYY-MM-DD format, starting from 3rd January 2006 till the end of 2018.
2. Open: price of the stock when market opens
3. High: highest price reached during the day
4. Low: lowest price reached during the day
5. Close: closing price of the stock
6. Volume: number of shares traded
7. Name: the stock ticker in the financial market.

2.2 Data Preprocessing

After investigation, we found that there is one row with missing values, and we decided to remove it. Also, we drop the Name column, as it has no additional information, except the name of a stock.

We decided to create visualizations first, before making any new features for the machine learning algorithms.

Part 3. Seasonal Decomposition

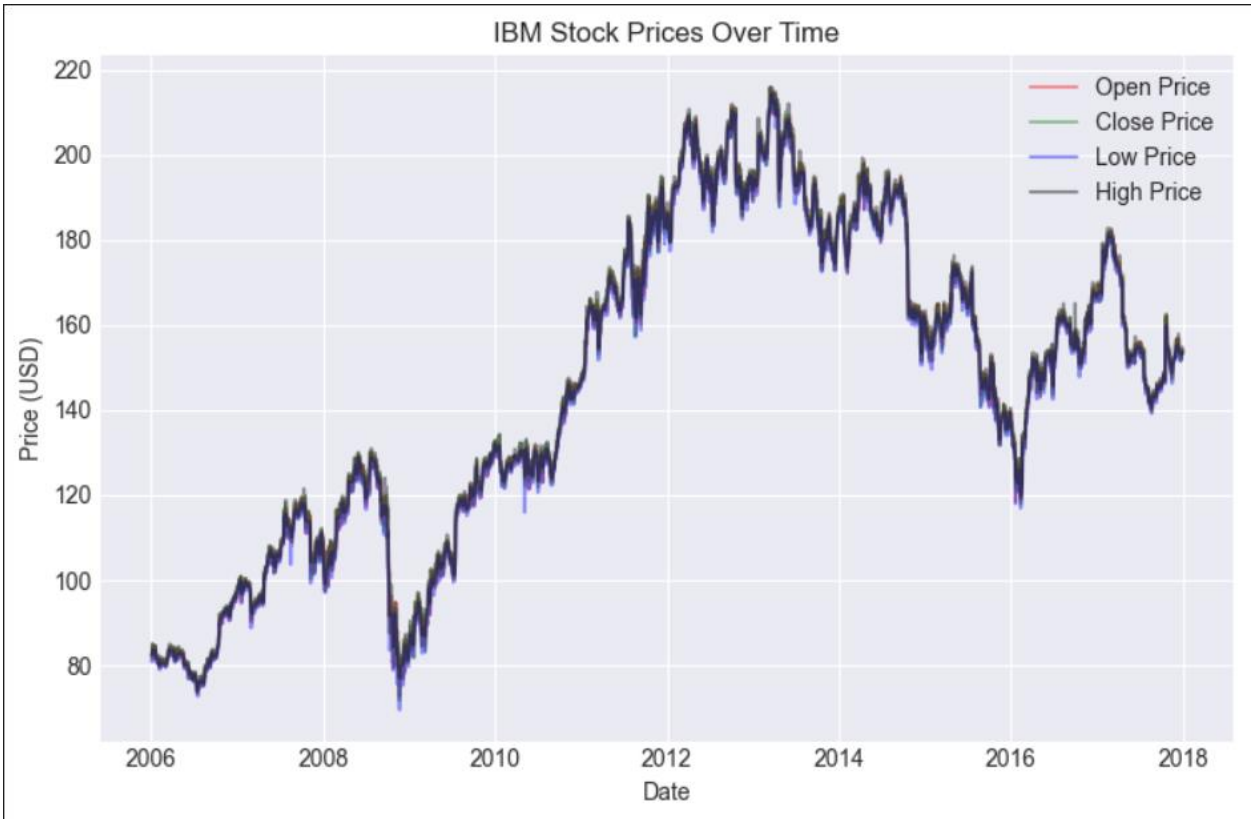


Figure 1.

Figure 1 shows the stock price movement during the twelve-year period. We can clearly see the fluctuations in all four price categories, with notable peaks and troughs. Here is a big downfall around 2008 and 2009, which indicates the impact of subprime mortgage crisis. Also, there is another downward trend between 2014 and 2016, which may be related to IBM's financial results and strategic decisions, which raised concerns among investors.

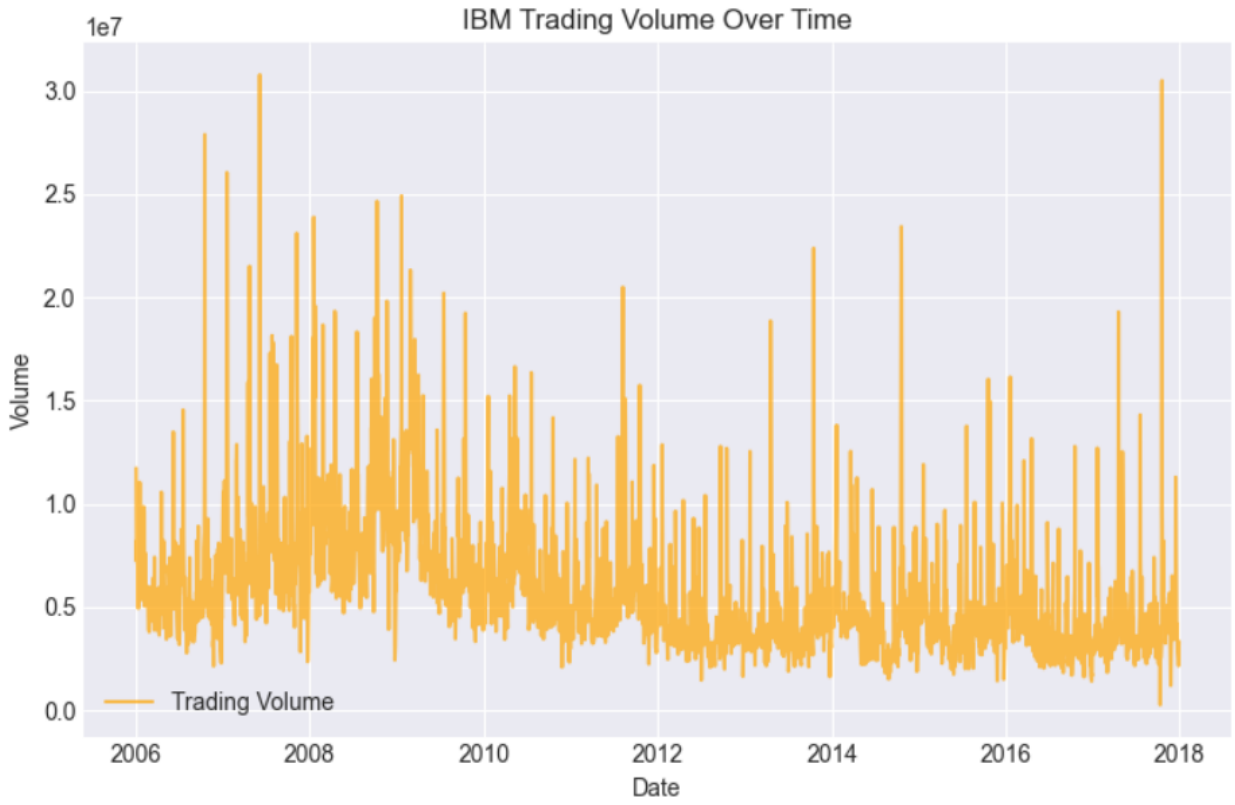


Figure 2.

Figure 2 reveals significant fluctuations in IBM's trading volume over time, with high peaks in time before 2008. The overall trend shows a decline in volume in the following years, indicating reduced trading activity. Despite these fluctuations, there are periods of relatively consistent trading, reflecting changes in investor interest and market conditions.

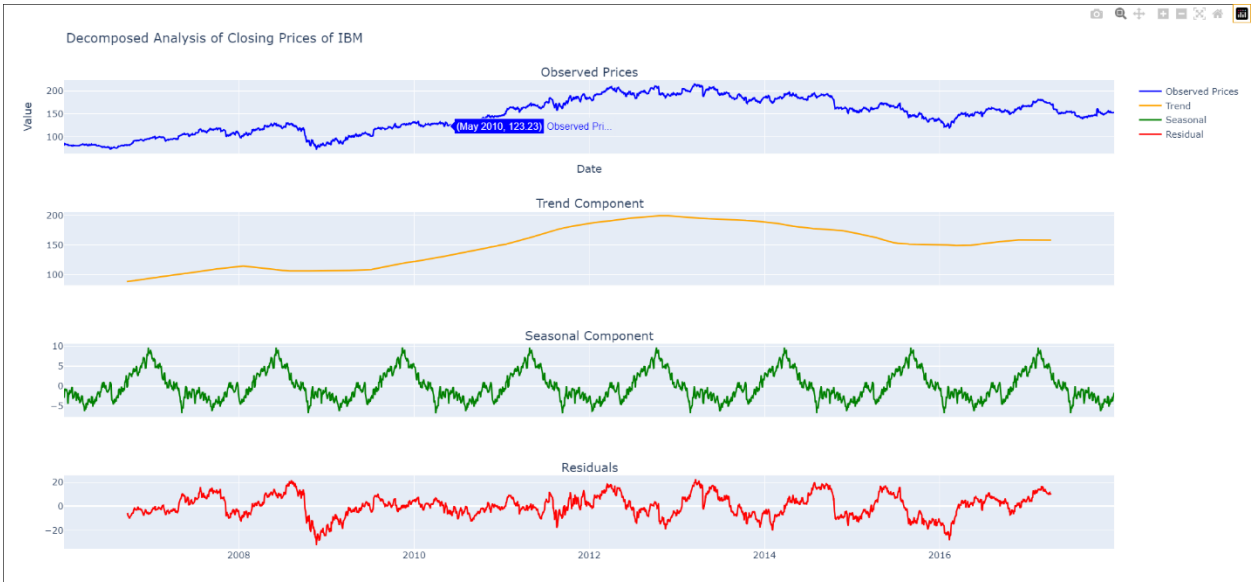


Figure 3.

Figure 3 consists of four graphs, where the closing price of IBM is decomposed in analysis. The first graph shows the actual closing prices, while the second graph indicates an overall trend. We can clearly see there is a gradual growth in stock prices from under 100 to over 200, then a decrease and fluctuations around 150.

The third graph illustrates the seasonal component, which highlights the predictable, recurring patterns in IBM's stock prices. These patterns suggest that there are specific times when the stock tends to rise or fall, likely influenced by market cycles and seasonal business factors. The regular peaks and troughs indicate these seasonal trends, providing valuable insights for forecasting future price movements.

The fourth graph presents the residuals, representing the random fluctuations in the stock prices that remain after accounting for the trend and seasonal effects. Ideally, the residuals should appear random and centered around zero, indicating that there are no discernible patterns left. However, if we observe any patterns in this graph, it suggests that there may be additional factors affecting the stock price that have not been captured by the model.

Part 4: Models training

As it is illustrated in Figure 4, we have taken records before 2017 for the training set. So, our goal is to predict ‘High’ prices after 2017.

Figure 4.

4.1 Feature Engineering

To enhance predictive accuracy and market analysis, we engineered some new features as follows:

- Moving Averages:
  - 10-day Moving Average, ‘MA-10’: average closing prices of the last 10 days.
  - 50-day Moving Average, ‘MA-50’: average closing prices of the last 50 days.
- Relative Strength Index, ‘RSI’: the ratio of average gains to average losses
- Bollinger Bands:
  - The upper band: 10-day moving average plus two standard deviations of the closing prices over the same period.
  - The lower band: moving average minus two standard deviations of the closing prices over the same period.
- 10-day Volume Moving Average, ‘Volume_MA_10’: average volume of the last 10 days,
- Daily Returns, ‘Daily_Return’: percentage change in closing prices from one day to the next

Next, we employed cross-validation to get the best subset of features, which provide the best performance for both validation and training sets. It was used for training of different models further:

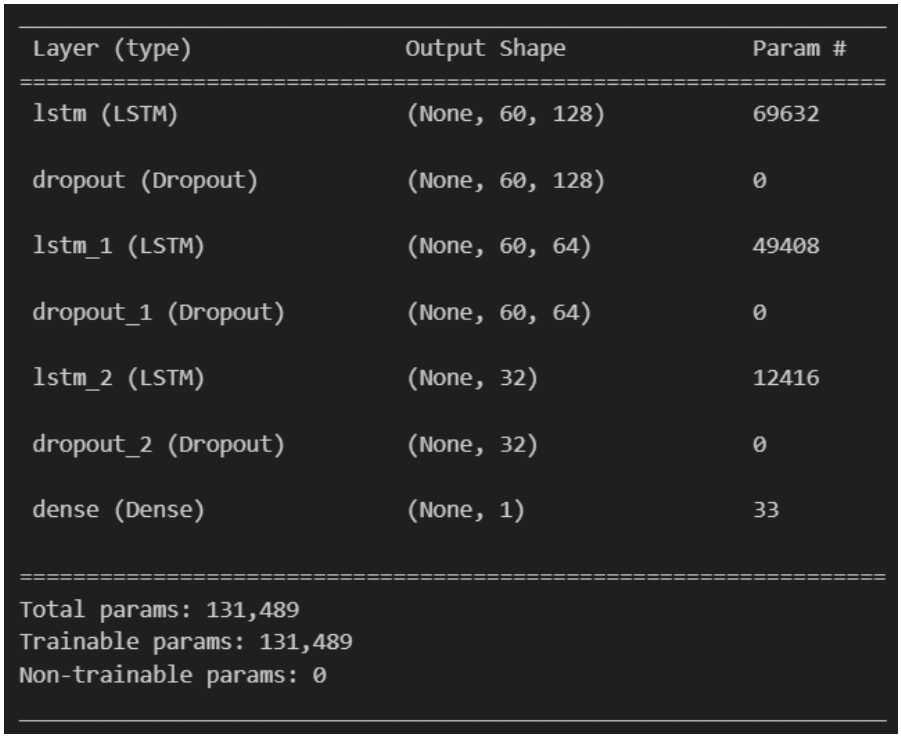
best_subset = ['Open', 'High', 'Low', 'Close', 'MA_10', 'RSI', Volume]

4.2 Models Description and Evaluation Metrics

For the Sentiment Analysis, we trained three different deep learning models: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Recurrent Neural Network (RNN), which are specifically designed for sequential data. To access models' accuracy and effectiveness, the following metrics were used for evaluation: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The models were trained using a time step of 60, ensuring optimal capture of temporal dependencies in the data.

Below you can observe structure and parameters used for each model:

4.3 LSTM



Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 128)	69632
dropout (Dropout)	(None, 60, 128)	0
lstm_1 (LSTM)	(None, 60, 64)	49408
dropout_1 (Dropout)	(None, 60, 64)	0
lstm_2 (LSTM)	(None, 32)	12416
dropout_2 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 131,489
Trainable params: 131,489
Non-trainable params: 0

Figure 5.

Training Parameters:

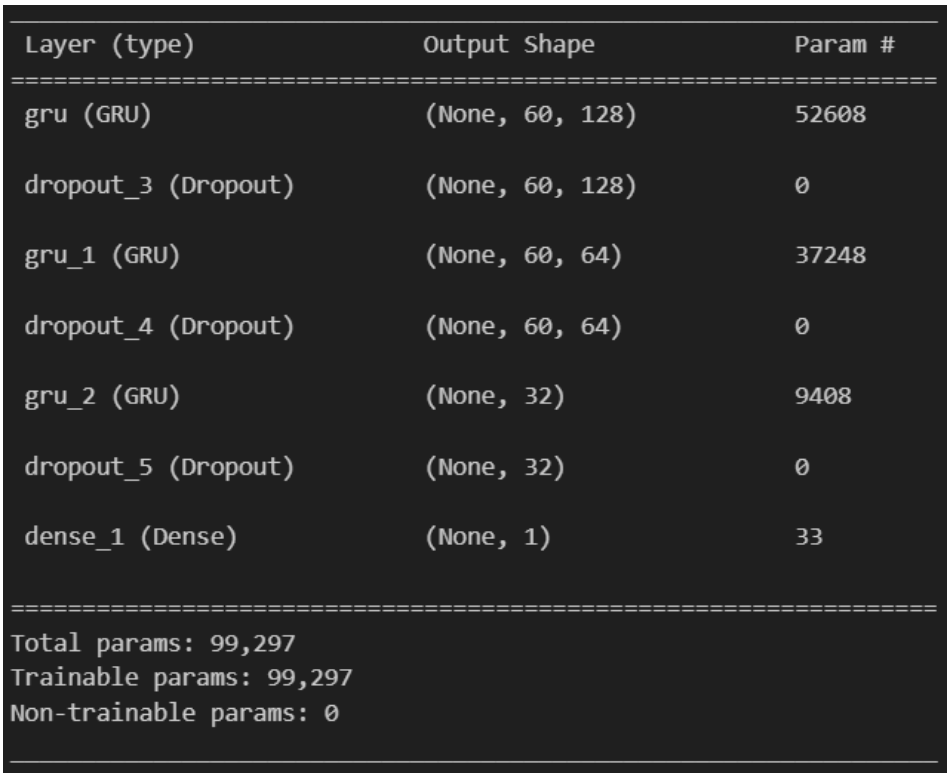
- Optimizer: Adam with learning rate 0.001.
- Loss Function: Mean Squared Error (MSE).
- Batch Size: 32.
- Epochs: 50 (early stopping with patience=10).
- Validation Split: 20%.

Feature Scaling:

- Price Features: MinMaxScaler (feature_range=(0.1, 0.9)) + clipping to [0.05, 0.95].

– Volume Feature: RobustScaler (quantile_range=(10, 90)) + clipping to [-3, 3].

4.4 GRU



Layer (type)	Output Shape	Param #
gru (GRU)	(None, 60, 128)	52608
dropout_3 (Dropout)	(None, 60, 128)	0
gru_1 (GRU)	(None, 60, 64)	37248
dropout_4 (Dropout)	(None, 60, 64)	0
gru_2 (GRU)	(None, 32)	9408
dropout_5 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

Total params: 99,297
Trainable params: 99,297
Non-trainable params: 0

Figure 6.

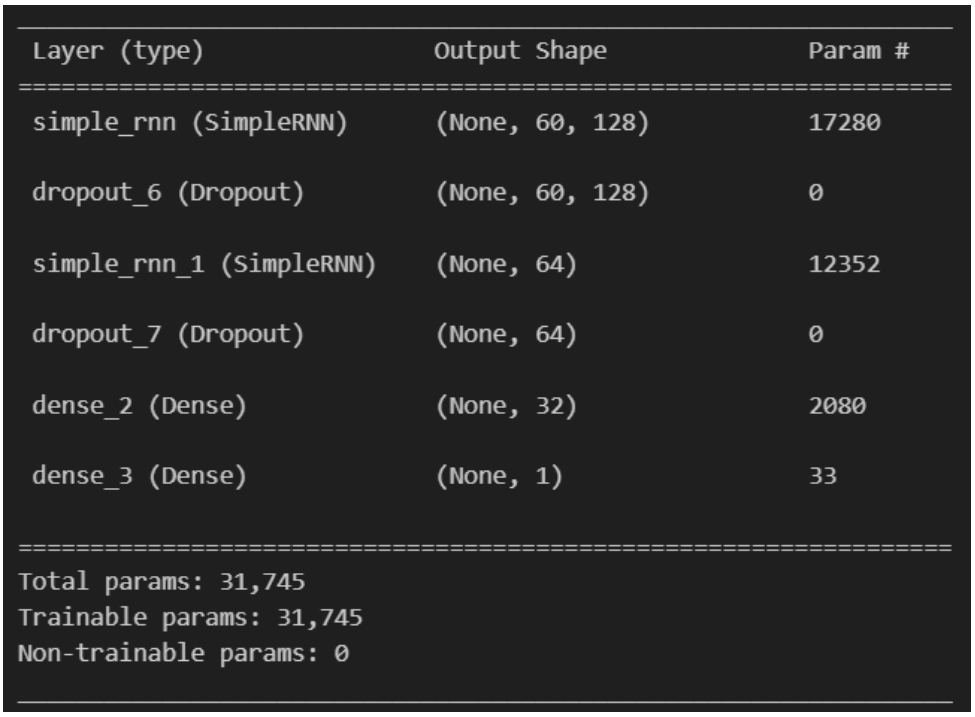
Training Parameters:

– Optimizer: Adam with learning rate 0.001.
– Loss Function: Mean Squared Error (MSE).
– Batch Size: 32.
– Epochs: 50 (early stopping with patience=10).
– Validation Split: 20%.

Feature Scaling:

– Price Features: MinMaxScaler (feature_range=(0.1, 0.9)) + clipping to [0.05, 0.95].
– Volume Feature: RobustScaler (quantile_range=(10, 90)) + clipping to [-3, 3].

4.5 RNN



Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 60, 128)	17280
dropout_6 (Dropout)	(None, 60, 128)	0
simple_rnn_1 (SimpleRNN)	(None, 64)	12352
dropout_7 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 1)	33

Total params: 31,745
Trainable params: 31,745
Non-trainable params: 0

Figure 7.

Training Parameters:

- Optimizer: Adam with learning rate 0.001.
- Loss Function: Mean Squared Error (MSE).
- Batch Size: 32.
- Epochs: 50 (early stopping with patience=10).
- Validation Split: 20%.

Feature Scaling:

- Price Features: MinMaxScaler (feature_range=(0.15, 0.85)) + clipping to [0.10, 0.90].
- Volume Feature: RobustScaler (quantile_range=(20, 80)) + clipping to [-2, 2].

Part 5: Prediction and Evaluation

5.1 LSTM

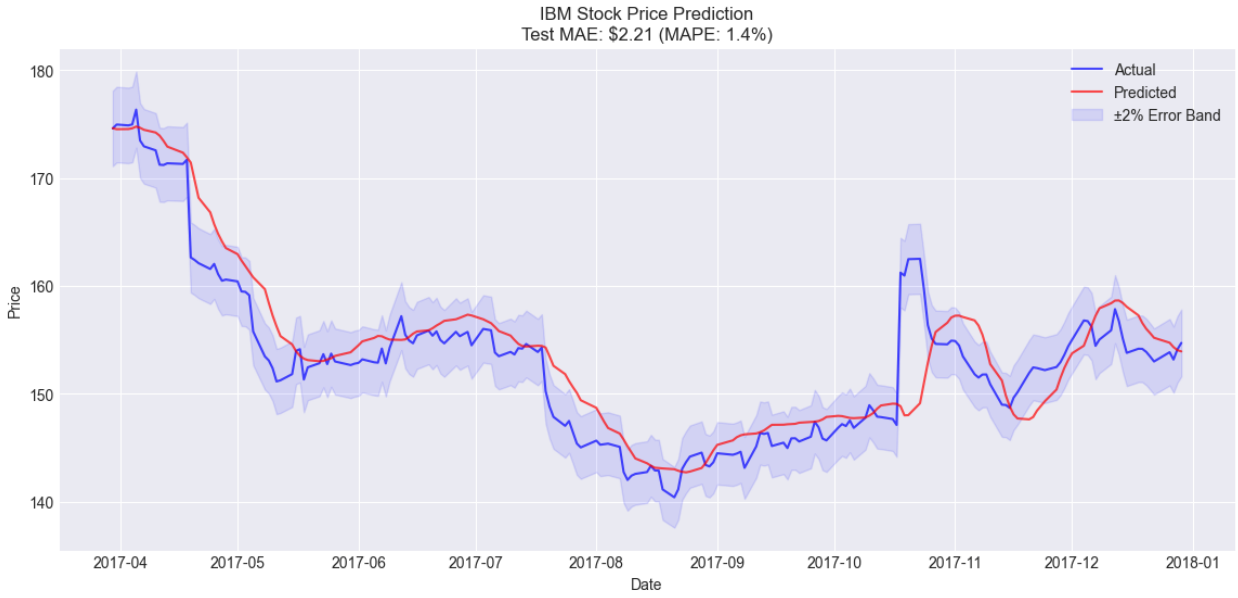


Figure 8.

	RMSE	MAE	MAPE
Train	3.66	2.76	1.90
Test	3.18	2.21	1.43

Table 1.

5.2 GRU

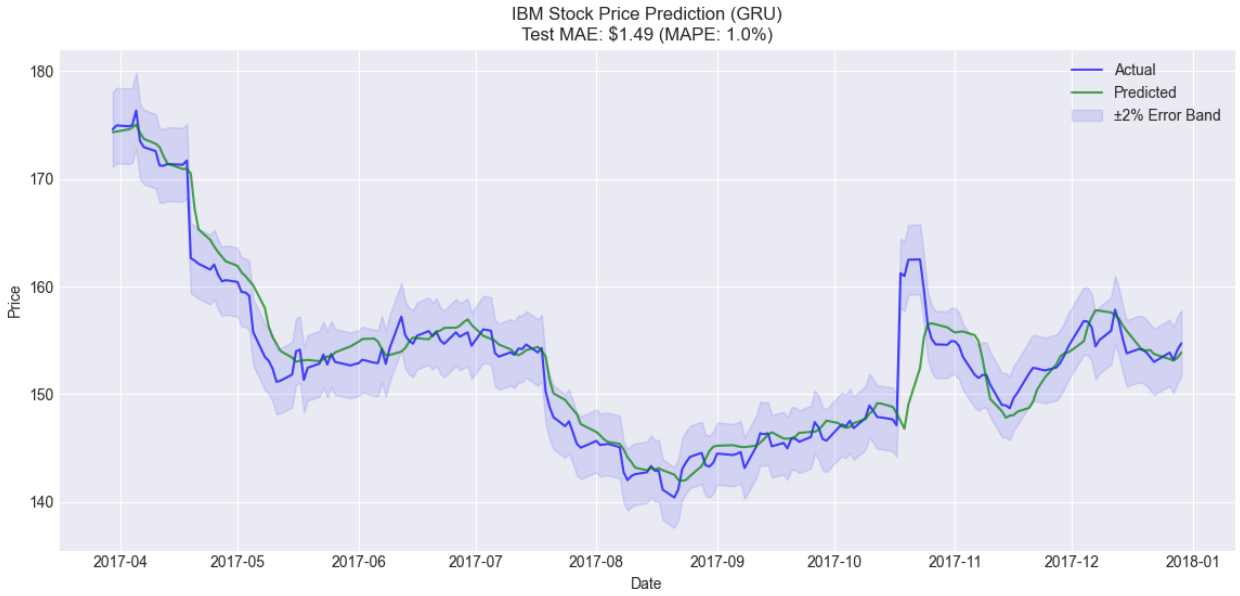


Figure 9.

	RMSE	MAE	MAPE
Train	2.77	2.05	1.41
Test	2.50	1.49	0.97

Table 2.

5.3 RNN

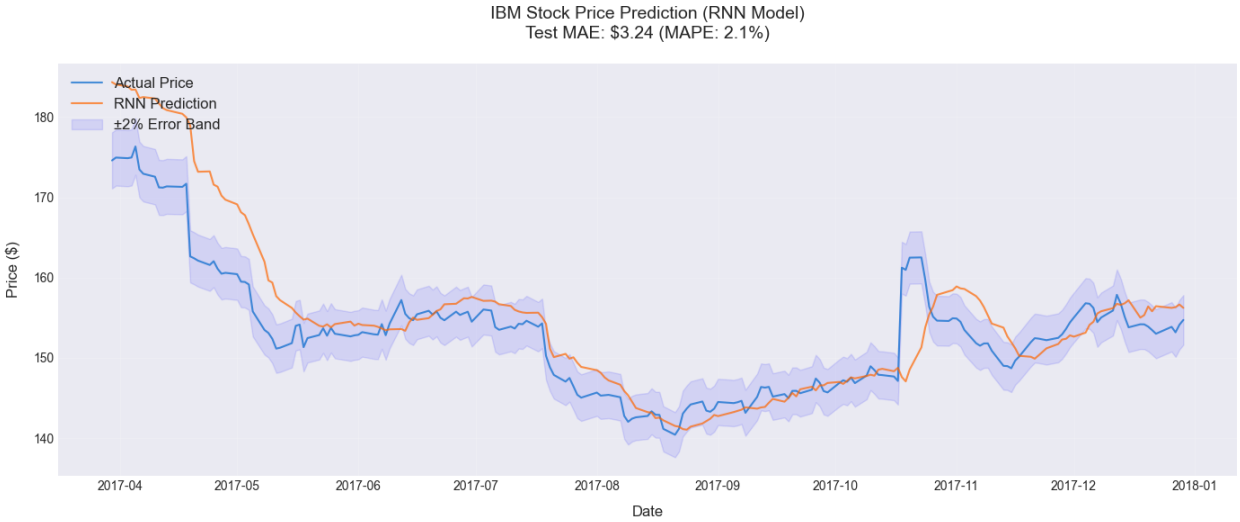


Figure 10.

	RMSE	MAE	MAPE
Train	7.21	5.78	4.44
Test	4.36	2.83	1.79

Table 3.

5.4 Discussion

The evaluation outcomes for the three models—LSTM, GRU, and RNN—highlight notable differences in performance when assessed on both training and test datasets.

GRU emerged as the top performer, recording the lowest error metrics (RMSE: 2.77 for training, 2.50 for testing; MAE: 2.05 for training, 1.49 for testing; MAPE: 1.41% for training, 0.97% for testing). Its reliable performance indicates robust generalization abilities with minimal signs of overfitting.

LSTM showed satisfactory results but was slightly surpassed by GRU. Its higher RMSE values (3.66 for training, 3.18 for testing) and MAPE (1.90% for training, 1.43% for testing) reflect a marginally lower accuracy in predictions compared to GRU.

RNN displayed the least effective performance, with considerably higher error rates (RMSE: 7.21 for training, 4.36 for testing; MAPE: 4.44% for training, 1.79% for testing). The significant disparity between training and test metrics indicates possible underfitting or challenges in capturing the temporal dependencies within the data.

Part 6: Conclusion

In this project, we tested different time series models such as LSTM, RNN, and GRU to describe the pattern and forecast future IBM stock values using historic data. A significant conclusion drawn from this research is that accurately predicting future stock prices solely based on historical data is often difficult and, at times, unfeasible. While these models can effectively recognize patterns and seasonal trends in past data, they tend to struggle with forecasting abrupt market changes or reversals, which frequently occur in financial markets.

For instance, the models are adept at recognizing patterns, such as a decline in stock prices. However, in the absence of supplementary external data, including correlated stocks, news sentiment, or other external market factors, they cannot anticipate a possible recovery or increase in prices. The models we used simply follow the trends they've learned from the past, making it hard to predict unpredictable events such as price reversals.

This emphasizes the significance of utilizing richer datasets, which may include correlated stocks, news sentiment, or various external market influences. While the primary aim of this project was to practice time series forecasting through models such as LSTM, RNN, and GRU, it underscores the fundamental limitations of depending exclusively on historical data for predicting stock prices. Nevertheless, this work brought a valuable opportunity to gain experience in applying these sophisticated deep learning models for time series analysis.