Final Project

Stock Price Prediction

Student Name: Hiep Vo Dang

Part 1: Abstract

This project addresses the complex challenge of forecasting stock closing prices using deep learning models. The study's foundation is built upon extracting, transforming, and loading (ETL) raw stock trading data into a PostgreSQL database, setting the stage for detailed analysis. Multiple downstream modules experiment a diversity of deep learning models, ranging from Multilayer Perceptrons (MLP) to advanced architectures like Convolutional Neural Networks (Conv1D), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Bidirectional networks. These models are systematically trained and evaluated with the aim of accurately predicting short-term stock price movements.

The findings indicate that while uni-structured models were insufficient for this task, the Bidirectional model combining both LSTM and GRU layers notably excelled, yielding an impressive Mean Absolute Error (MAE) of 0.64% for AAPL stocks, 0.81% for META stocks, and 1.41% for TSLA stocks over a five-day prediction period. This result underscores the potential of utilizing sophisticated deep learning techniques in financial market prediction, offering significant insights and tools for investors and analysts.

Part 2: Introduction

The field of financial analysis has long grappled with the challenge of predicting stock market movements. This project attempts to explore the application of deep learning techniques in forecasting stock closing prices, a task of considerable importance and complexity in financial trading. The primary focus is to deploy the sophisticated capabilities of deep learning models to process and analyze historical stock market data, aiming to predict short-term movements of individual stocks.

The primary stakeholders of this research are financial analysts and investors engaged in the stock market. These individuals and entities depend heavily on precise and timely predictions of stock prices to make judicious investment decisions. The volatility and uncertainty inherent in the stock market pose significant challenges, making stock price prediction a notoriously difficult task. Traditional predictive methods, which typically lean on fundamental and technical analysis, may fall short in capturing the complex patterns of stock market data.

This project, therefore, seeks to address this gap by offering more sophisticated and perhaps more accurate predictive tools. By leveraging the power of deep learning, it aims to provide stakeholders with enhanced forecasts of stock market movements, facilitating more informed and strategic investment choices. The ultimate goal is to develop a solution that not only augments the decision-making process for investors and analysts but also contributes to the broader field of financial analysis through the application of advanced computational techniques.

The project is anchored around several key research questions:

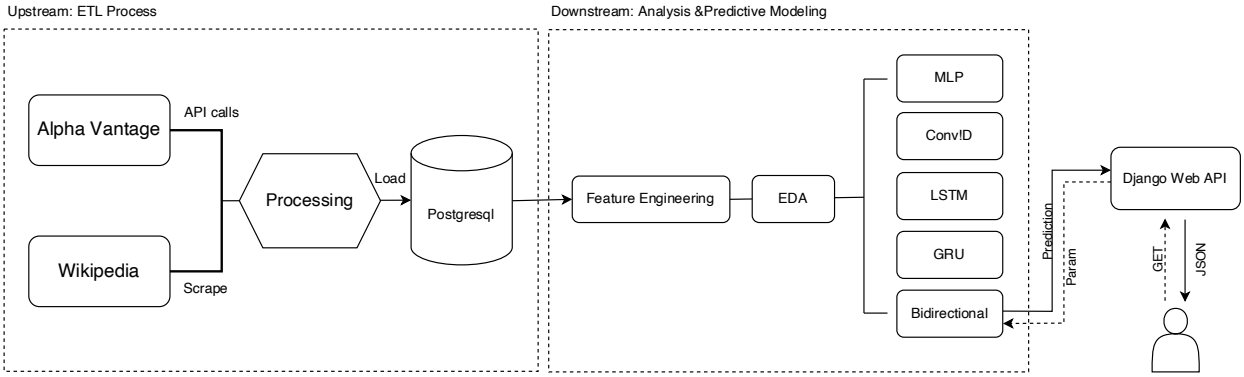
1. What are the typical behavioral patterns of individual stock returns?
2. Which deep learning models are most effective in modeling stock price movements?
3. How can we predict stock prices for the next five days with a degree of reliability?

These questions are pivotal in guiding the research towards meaningful insights and practical outcomes.

The data used for this project will be sourced from two primary platforms: Alpha Vantage and Wikipedia. Alpha Vantage, a leading provider of APIs for historical and real-time stock data, will be utilized to acquire comprehensive raw stock trading information. This data includes essential market indicators such as daily open, high, low, close prices, as well as trading volumes for individual stocks. In addition to Alpha Vantage, the project will leverage Wikipedia, particularly for extracting the list of companies included in the S&P 500 index.

Part 3: Research Approach

The foundation of this project is built upon robust data acquisition and processing. The project utilizes the Alpha Vantage API (https://www.alphavantage.co/documentation/) to obtain raw stock trading data, including daily open, low, high, clode price and trading volume. This comprehensive dataset is further enriched by scraping the list of S&P 500 companies, along with the Global Industry Classification Standard (GICS) sector classification, from Wikipedia (https://en.wikipedia.org/wiki/List_of_S%26P_500_companies). An ETL (Extract, Transform, Load) pipeline is developed to streamline the processing of this data into a PostgreSQL database. This pipeline not only facilitates efficient data handling but also ensures the integrity and consistency of the data being fed into the deep learning models.



Downstream modules encompass Exploratory Data Analysis (EDA) and Feature Engineering steps, where the engineered features are fed to a wide range of predictive models. Each model is carefully designed to analyze historical data, learning patterns and trends that are not readily apparent through conventional statistical methods. The project explores multiple deep learning architectures, assessing their efficacy in understanding and predicting the nonlinear and complex movements within the stock market. The models we plan to implement and assess include:

1. Multilayer Perceptron (MLP)
2. 1D Convolutional Neural Network (Conv1D)
3. Long Short-Term Memory (LSTM)
4. Gated Recurrent Unit (GRU)
5. Bidirectional Models

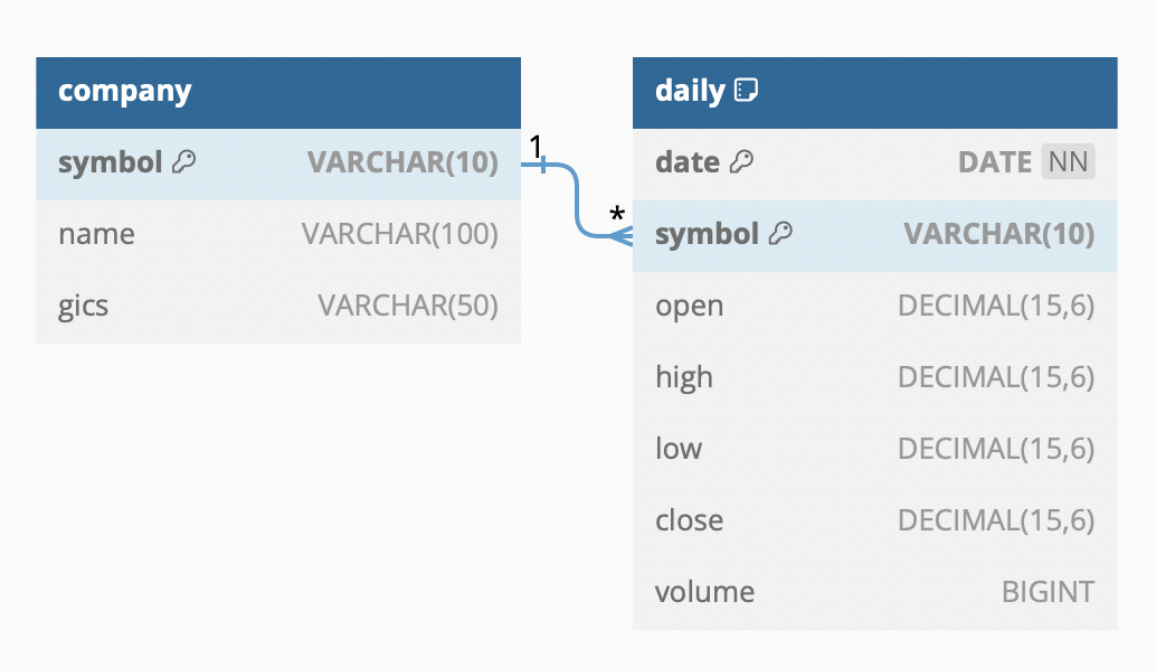
The efficacy of each model will be assessed based on its ability to accurately predict short-term movements in stock prices. Key performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) will be employed to train and evaluate each model.

The whole application interfaces with the users through a Django Web API, such that the users can submit a GET request, specifying parameters, the application makes predictions and responds to the users with a JSON.

Part 4: ETL Process

4.1. Database Schema

The database schema designed for this project plays a crucial role in efficiently organizing and storing the stock market data sourced from Alpha Vantage and Wikipedia. The database, named sp500, is structured to facilitate easy access and manipulation of data, essential for the deep learning models used in this study. This schema is comprised of two primary tables: company and daily.



Company Table: The company table is a central component of the database, storing key information about companies listed in the S&P 500 index. This data, scraped from Wikipedia's "List of S&P 500 companies," includes the symbol, name, and gics (Global Industry Classification Standard) of each company. The symbol serves as the primary key, ensuring that each entry in the table is unique. The name

field holds the full name of the company, while the gics field contains the sector classification, providing an insight into the industry segment each company belongs to. This table forms the backbone of our analysis, linking the company-specific data to their corresponding stock market activities.

Daily Table: The daily table is designed to capture the day-to-day fluctuations in the stock market as provided by the Alpha Vantage API. It includes fields such as date, symbol, open, high, low, close, and volume. The date and symbol fields together form a composite primary key, ensuring each record uniquely identifies the stock's daily performance. The symbol field in this table is a foreign key referencing the symbol in the company table, establishing a relational link between the two tables. Each of the stock's daily metrics (open, high, low, close) is stored as a decimal value, and volume is recorded as an integer, representing the total number of shares traded that day. Checks are in place to ensure that these values are non-negative, maintaining the integrity of the data.

4.2. ETL Pipeline

The BaseETL class forms the architectural backbone of all of the ETL processes. As an abstract class, it sets out the standard framework for ETL operations across different concrete processes. It initializes a placeholder for the extracted data, ensuring a standard structure across different ETL processes.

The class outlines three main methods: extract(), transform(), and load(), each of which is defined as an abstract method, forcing concrete ETL classes to implement. The extract() and transform() methods are designed to return the class instance (self). This implementation enables the chain of method calls for streamlined and efficient operations.

The Company class, extending BaseETL, is responsible for scraping the list of S&P 500 companies from Wikipedia. The Daily class focuses on fetching and processing daily stock data from the Alpha Vantage API.

The implementation of the ETL pipeline is initiated in the main block of the script. It begins with updating the S&P 500 company list, followed by fetching and processing daily stock data for each company. This process iterates over all the companies in the S&P 500.

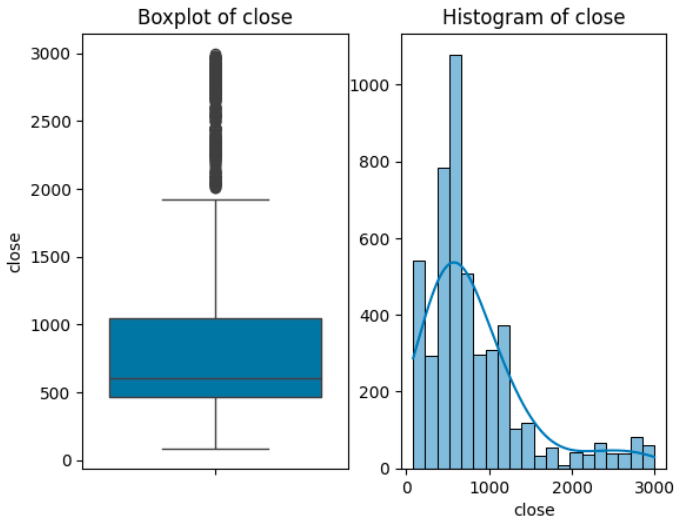
```python
# get all symbols from S&P 500
symbols = pd.read_sql(
    """
    select distinct "symbol"
    from "company";
    """,
    con=engine,
).squeeze().to_list()

# get trading data each of symbols
daily = Daily()
for symbol in symbols:
    daily.extract(symbol=symbol, size='full').transform().load()
```

Part 5: Data Preparation

5.1. Inherent Skewness in Stock Price Data

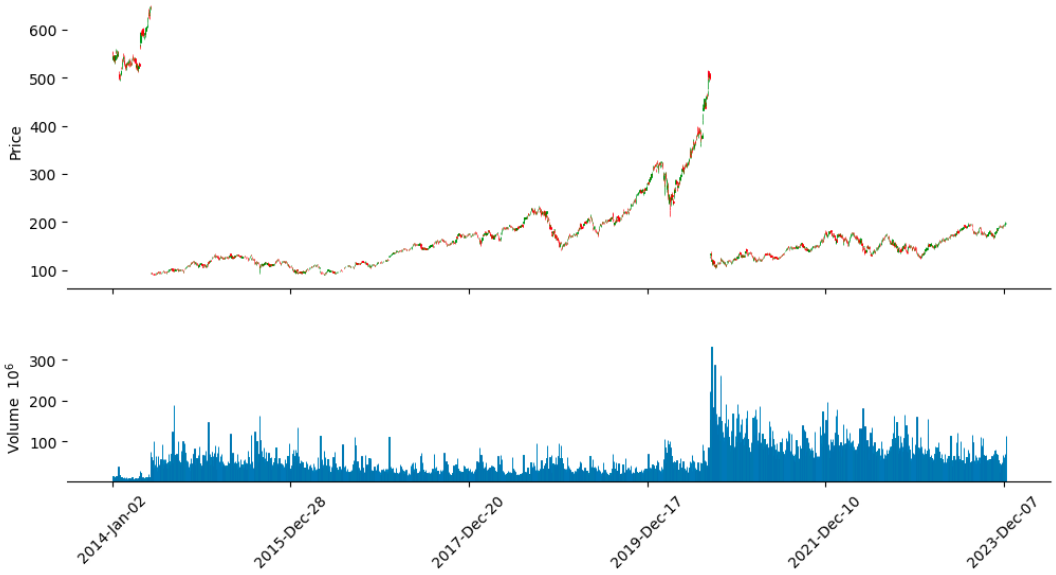
Skewness in stock price data is a well-documented phenomenon in financial analysis, presenting unique challenges in predictive modeling. Stock prices typically exhibit a right-skewed distribution, a characteristic that must be addressed during the data preparation phase to ensure effective analysis and modeling.



To address the inherent skewness in stock prices, feature engineering plays a pivotal role, particularly through the transformation of raw stock prices into log returns. This method involves calculating the logarithm of the ratio between consecutive prices, effectively normalizing the data. Such normalization results in a more symmetric distribution around zero.

5.2. Artificial Data Shifts

One important aspect when dealing with financial data is that stock prices and volumes are subject to occasional shifts due to stock splits, reverse stock split, dividends, etc. These shifts, while significantly impacting the price, do not reflect actual changes in trading patterns. For investors trading around these events, such adjustments have no net effect. For example, if a stock is splitted by a 2:1 ratio, each of their stocks will become 2 but the price will be adjusted downward by 50%, meaning zero net effect. The same applies for dividends.



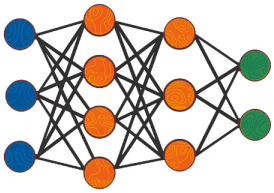
During the data preparation phase, artificial shifts in stock prices can be treated as extreme values. These are often identified as log returns (for prices) and log rate of change (for volume) that fall below the 0.001 quantile or exceed the 0.999 quantile, indicating significant deviations from typical movements. By categorizing these artificial shifts as extreme values, they can be systematically isolated and removed prior to our EDA phase. This ensures that the EDA is free from distortions caused by non-market factors.

The StockPrice class is a key component of the data preparation phase. This class is designed to manage and transform the stock price data into a tf.data.Dataset for feeding into various keras models.

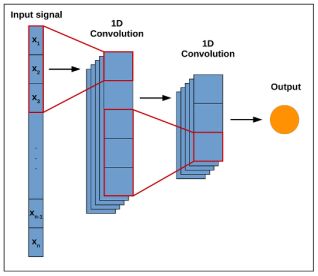
Part 6: Modeling & Results

I have chosen to explore 5 sets of neural network models:

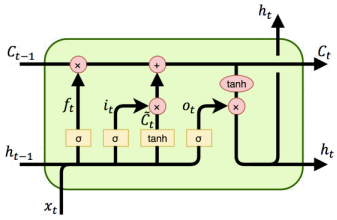
Multilayer Perceptron



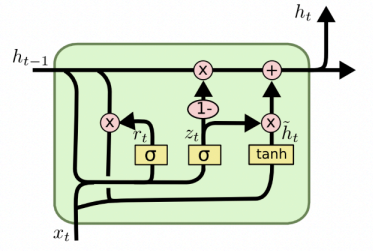
1-D Convolutional Neural Network



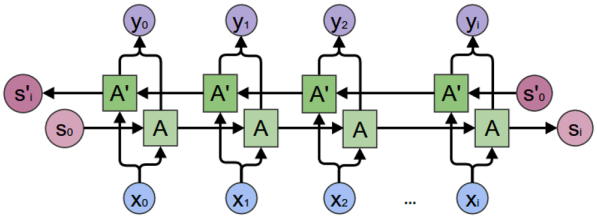
Long-Short Term Memory



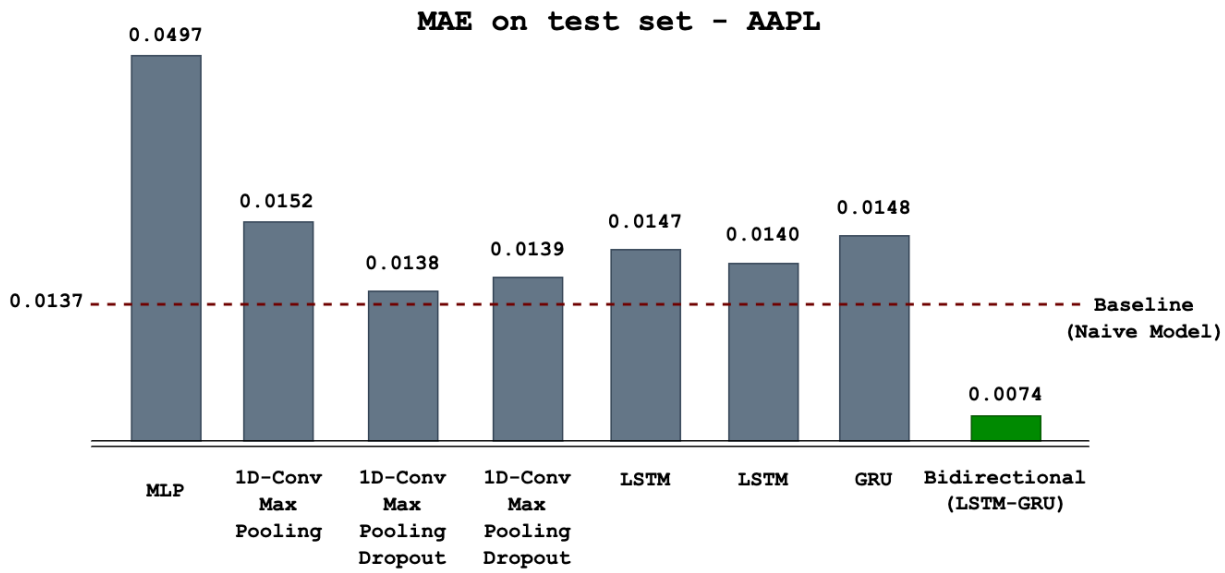
Gated Recurrent Unit



Bidirectional



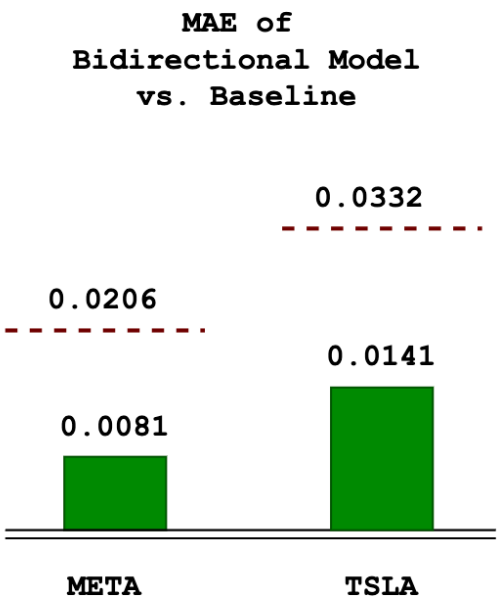
MLP models were used under the assumption that temporal information is not significant in analyzing stock prices. This over-simplified approach was proven to be ineffective in stock price prediction. The results of MLP models with various configurations did not surpass the simple baseline model.



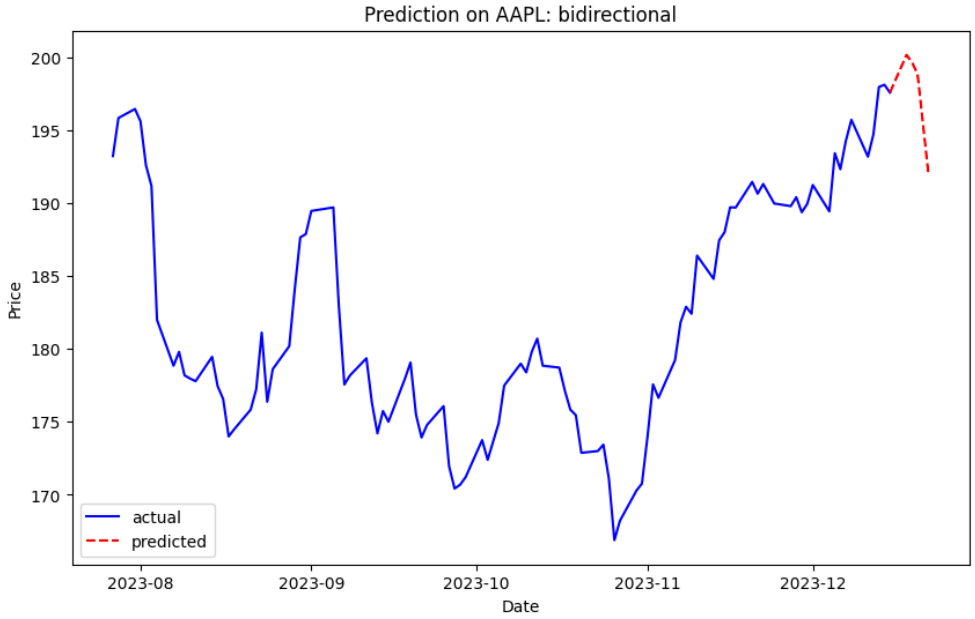
The other 4 sets of models are particularly well suited in the context of time-series. Conv1D models are considered the most lightweight compared to LSTM, GRU and, Bidirectional. It was employed with the expectation that the local pattern it learns from the stock prices might help in predicting future price values. However, Conv1D architectures alone were not able to outperform the baseline.

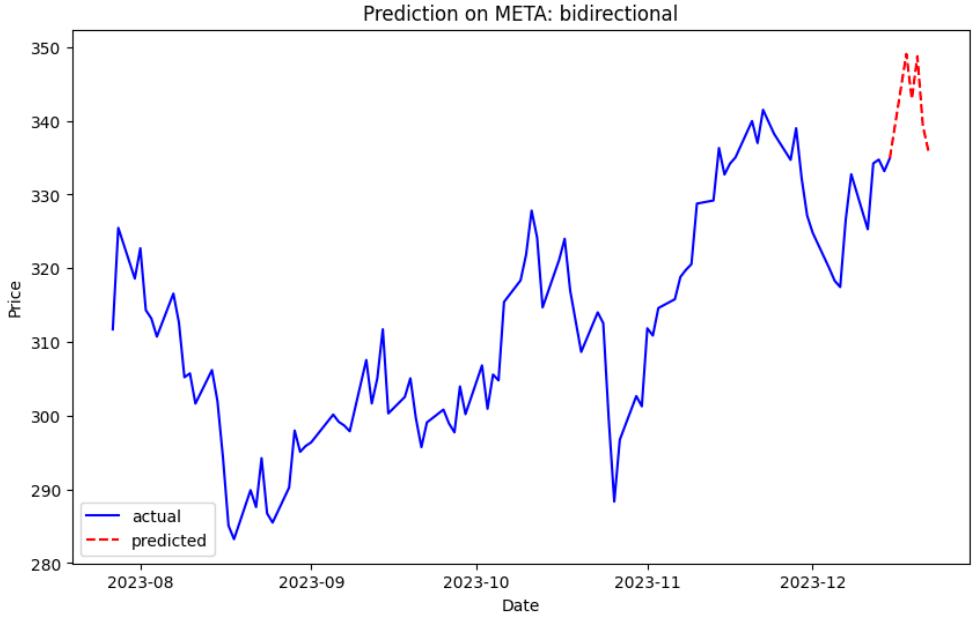
LSTM and GRU models are known to be very powerful. I tried these models with a lot of passion since it can capture dependencies of data values across very long time frames. But they failed to improve the performance maybe because processing the information only in one direction is not enough in understanding stock prices.

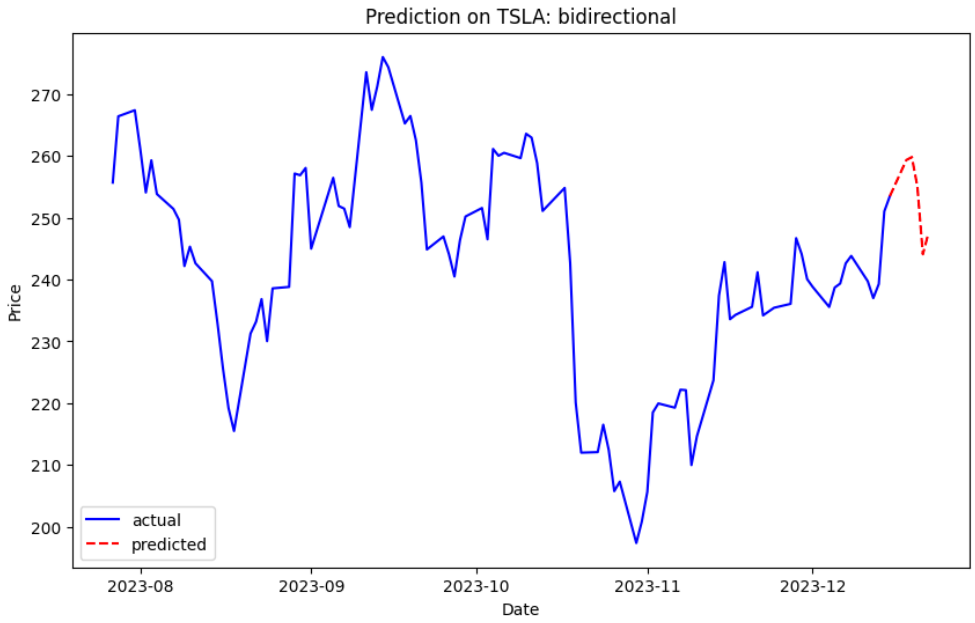
Bidirectional models, are those require the highest computational expense, surprisingly worked really well. My bidirectional model of one LSTM layer going forward and one GRU going backward is chosen as the recommended model in predicting stock prices. For AAPL stocks, it gave an MAE of 0.64% on the unseen data for 5 predictions ahead in the future, which is equivalent to only 0.64% / 5 = 0.128% error per day. And for META stocks, it gave an MAE of 0.81%, which is 0.81% / 5 = 0.162% error per day. For TSLA stocks, we obtained an MAE of 1.41%, which is 1.41% / 5 = 0.282% error per day. These results are much better than what is expected.



Here is the prediction over the next 5 days of different stocks:







I also built a Web API that exposes the model to real users. The API takes some parameters as input the return the stock price prediction as JSON. It asks for the stock symbol you want to predict, the backward is the number of days in the past you want consider for making prediction. And forward is the number of days into the future you want to predict.

Example:
GET http://10.144.9.21:8000/stock/?symbol=IBM&backward=100&forward=10