Machine Learning for Artificial Intelligence

Student Name: Nguyen Thai Thao Uyen
Student ID: 200117

Overview

This project examines the performance of GRU and RNN LSTM models in predicting stock prices for Apple Inc on NASDAQ and ACB Bank on VNIndex. Two prediction scenarios were explored: a 1-day horizon and a 7-day horizon, using a window size of 30 days. However, predictions over a 30-day horizon were not included due to their lower accuracy. Throughout the project, various adjustments were implemented to minimize the mean squared error (MSE) and mitigate the risk of overfitting.
The output of this prediction is then used for portfolio management and trading point identification.

Model Evaluation

RNN Model

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 30, 50)            10400

 dropout_4 (Dropout)         (None, 30, 50)            0

 lstm_5 (LSTM)               (None, 50)                20200

 dropout_5 (Dropout)         (None, 50)                0

 dense_4 (Dense)             (None, 32)                1632

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 32,265
Trainable params: 32,265
Non-trainable params: 0
_________________________________________________________________
```

```
  # Compile and train the model with Mean Squared Error loss function
    RNN_model.compile(optimizer= tf.keras.optimizers.Adam(learning_rate=0.001),
                      loss='mse',
                      metrics=['mse'])
    history = RNN_model.fit(X_train_norm, y_train_norm,
validation_data=(x_val,y_val), epochs=100, batch_size=32)
```

The model evaluation for Predicting Stock Prices Over a One-Day Horizon on NASDAQ AAPL

```
AVERAGE MSE ACROSS ALL FOLDS:  0.003152991183969972
MSE on the test set:  0.003021315828854927
```

The model evaluation for Predicting Stock Prices Over a 7-Day Horizon on NASDAQ AAPL

```
AVERAGE MSE ACROSS ALL FOLDS:  0.029864001072995177
MSE on the test set:  0.2711378886740058
```

The model evaluation for Predicting Stock Prices Over a One-Day Horizon on VNIndex ACB

```
AVERAGE MSE ACROSS ALL FOLDS:  0.002768636803054983
MSE on the test set:  0.0016302128441067893
```

The model evaluation for Predicting Stock Prices Over a 7-Day Horizon on VNIndex ACB

```
AVERAGE MSE ACROSS ALL FOLDS:  0.016613561113413482
MSE on the test set:  0.398472585676969
```

GRU Model

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 gru (GRU)                   (None, 30, 32)            3360      
                                                                 
 gru_1 (GRU)                 (None, 30, 32)            6336      
                                                                 
 gru_2 (GRU)                 (None, 32)                6336      
                                                                 
 dropout_6 (Dropout)         (None, 32)                0         
                                                                 
 dense_6 (Dense)             (None, 1)                 33        
                                                                 
=================================================================
Total params: 16,065
Trainable params: 16,065
Non-trainable params: 0
_________________________________________________________________
```

```
# Compile and train the model with Mean Squared Error loss function
   GRU_model.compile(optimizer= tf.keras.optimizers.Adam(learning_rate=0.001),
                     loss='mse',
                     metrics=['mse'])
   history = GRU_model.fit(X_train_norm, y_train_norm,
validation_data=(x_val,y_val), epochs=100, batch_size=32)
```

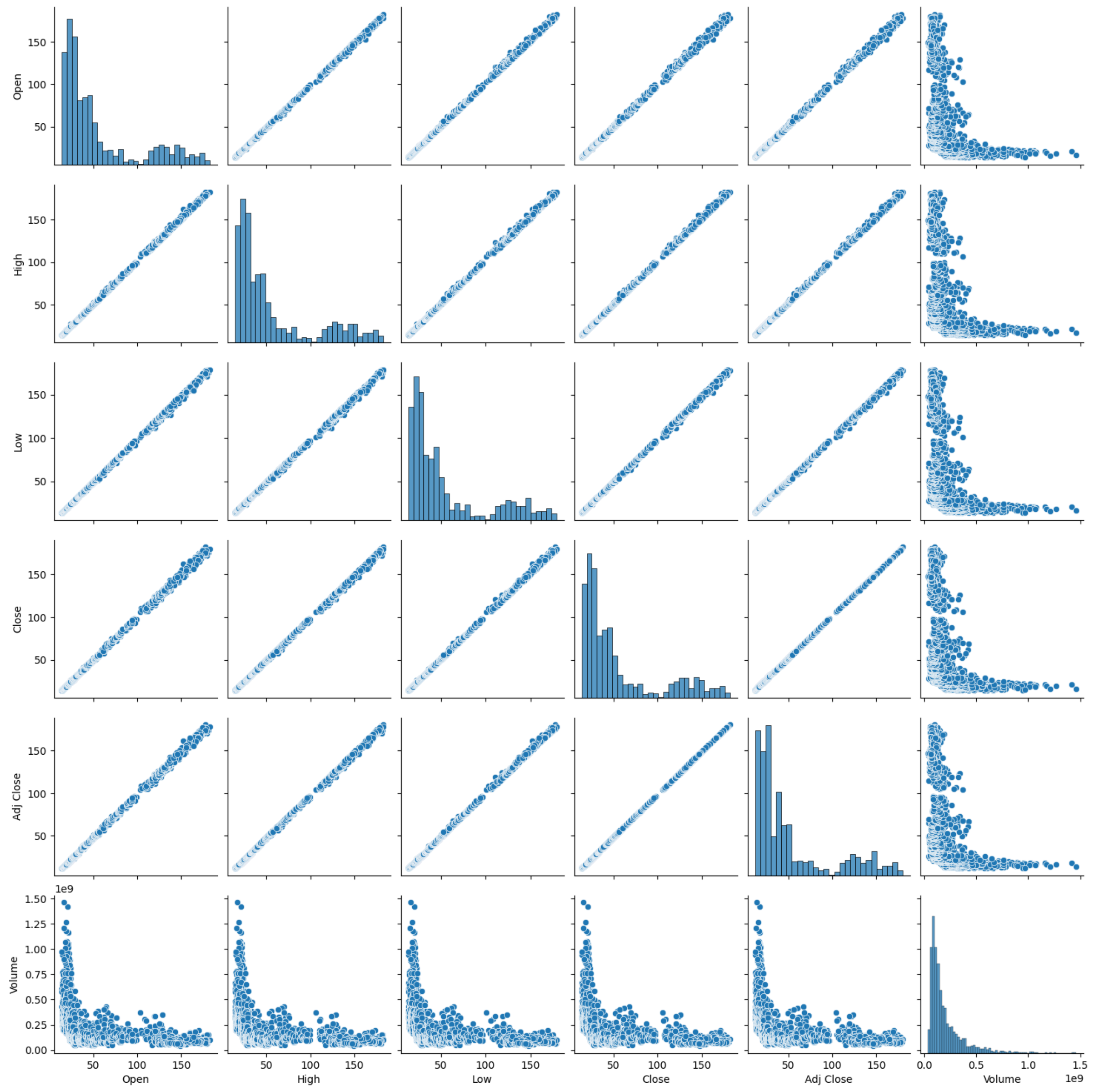
The model evaluation for Predicting Stock Prices Over a 1-Day Horizon on VNIndex ACB

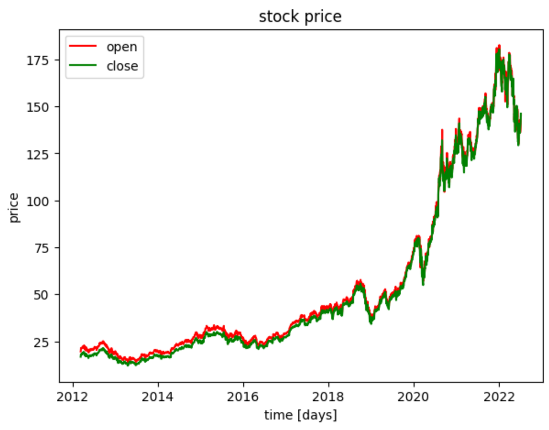
```
AVERAGE MSE ACROSS ALL FOLDS:  0.0027686368030549983
MSE on the test set:  0.0016302128441067893

AVERAGE MSE ACROSS ALL FOLDS:  8.930193120439057e-05
MSE on the test set:  8.1905252812876e-05
```

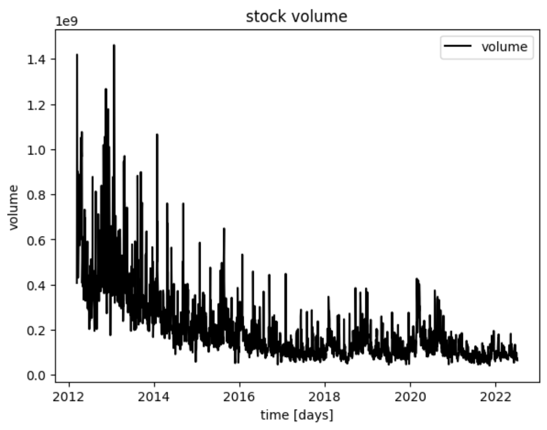
Nasdaq stock price prediction

(Nasdaq dataset) NASDAQ - APPLE (AAPL)

Data Exploration

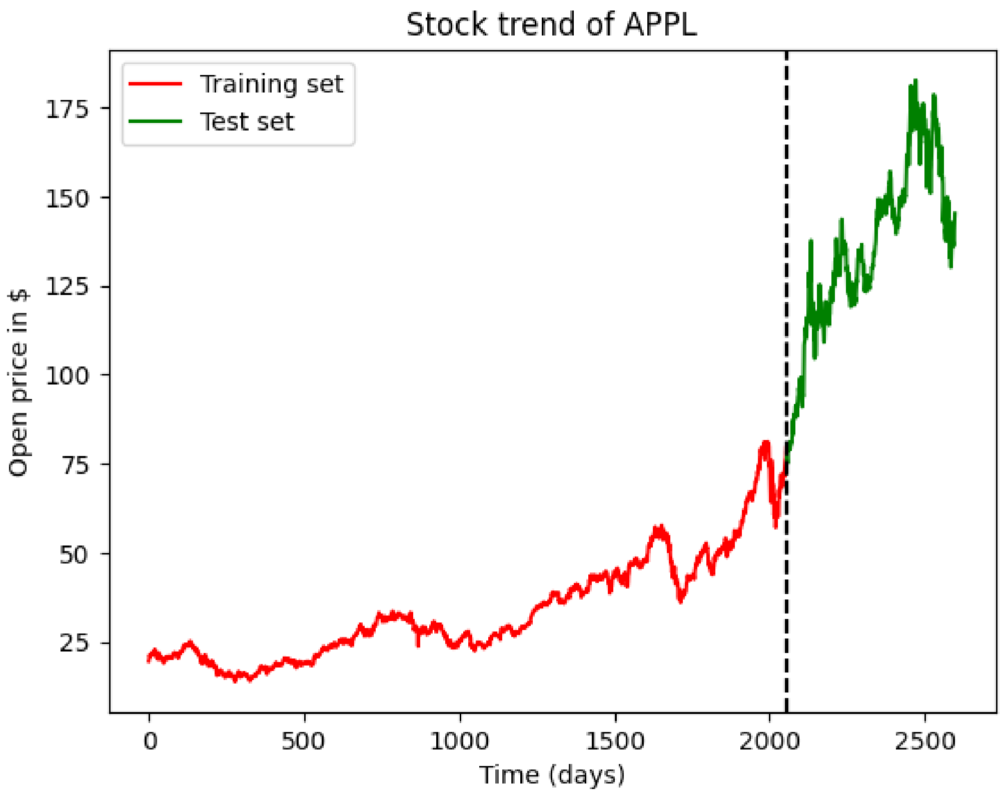




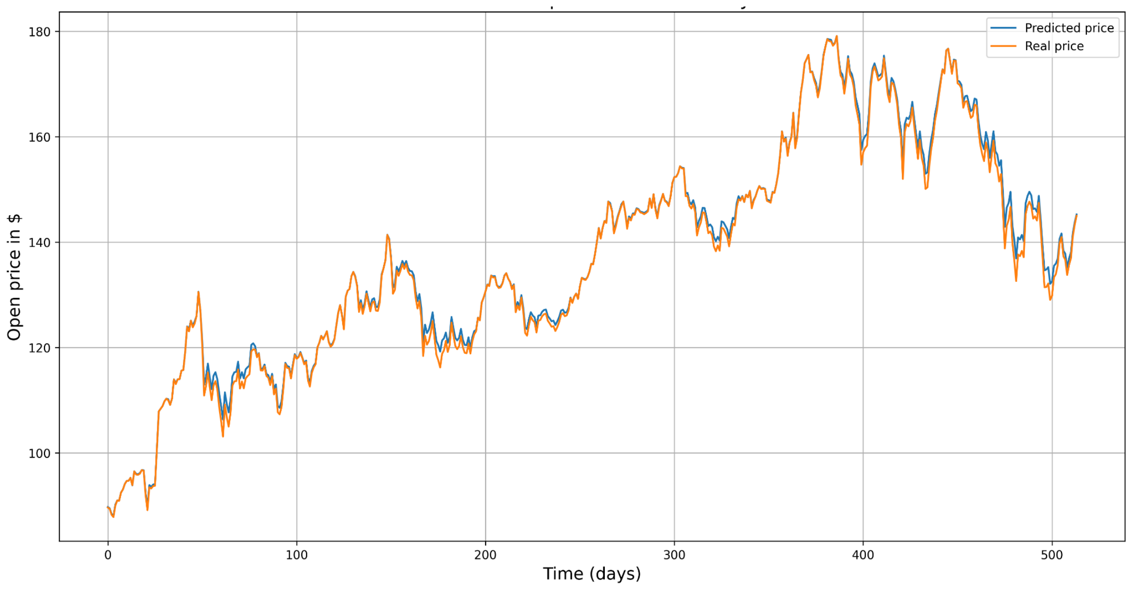


Open price prediction with window size = 30 to predict the 31st trading day

1. 80% training 20% testing
2. Use of cross validation (5 k folds)
3. Visualize the result



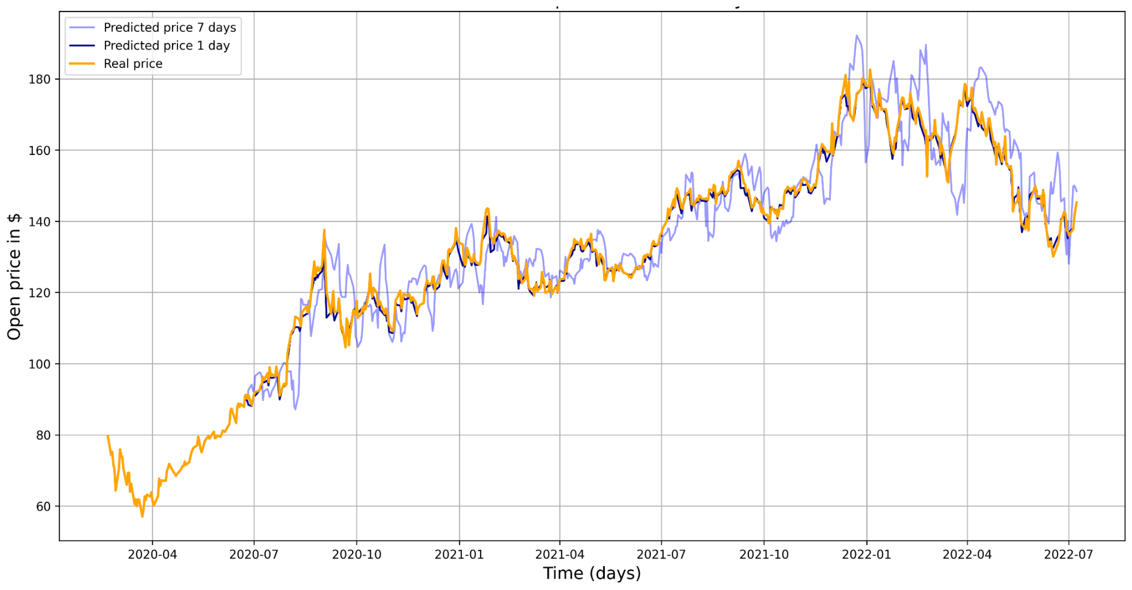
Apple Adj. Stock Trend Prediction (1 day)



The model evaluation for Predicting Stock Prices Over a One-Day Horizon on NASDAQ AAPL

```
AVERAGE MSE ACROSS ALL FOLDS:  0.0031529911839699972
MSE on the test set:  0.003021315828854927
```

Apple Adj. Stock Trend Prediction (1 day and 7 day)

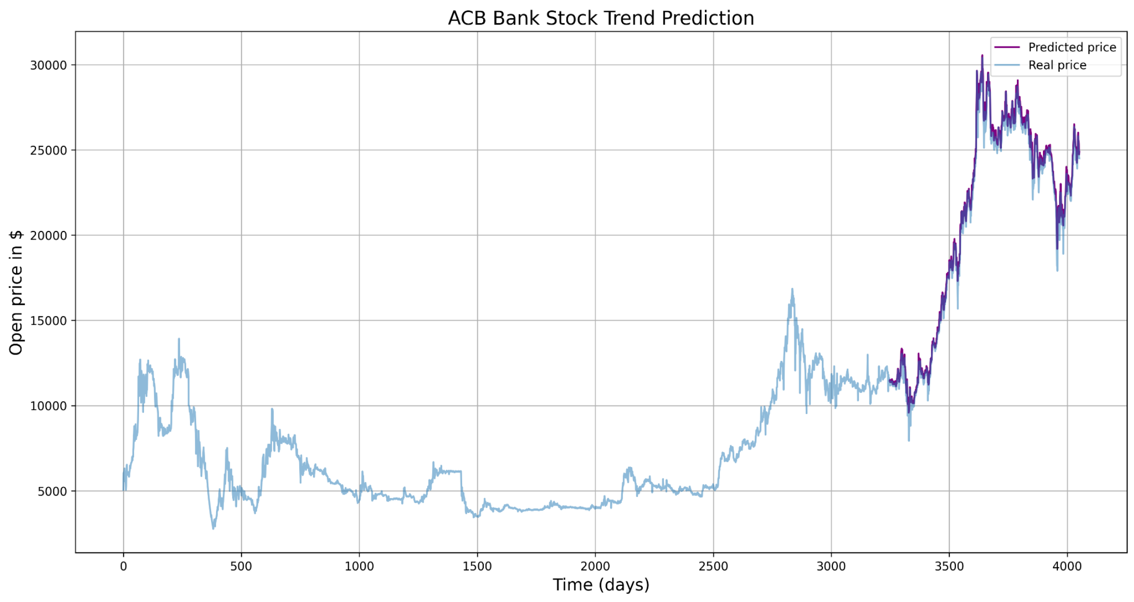


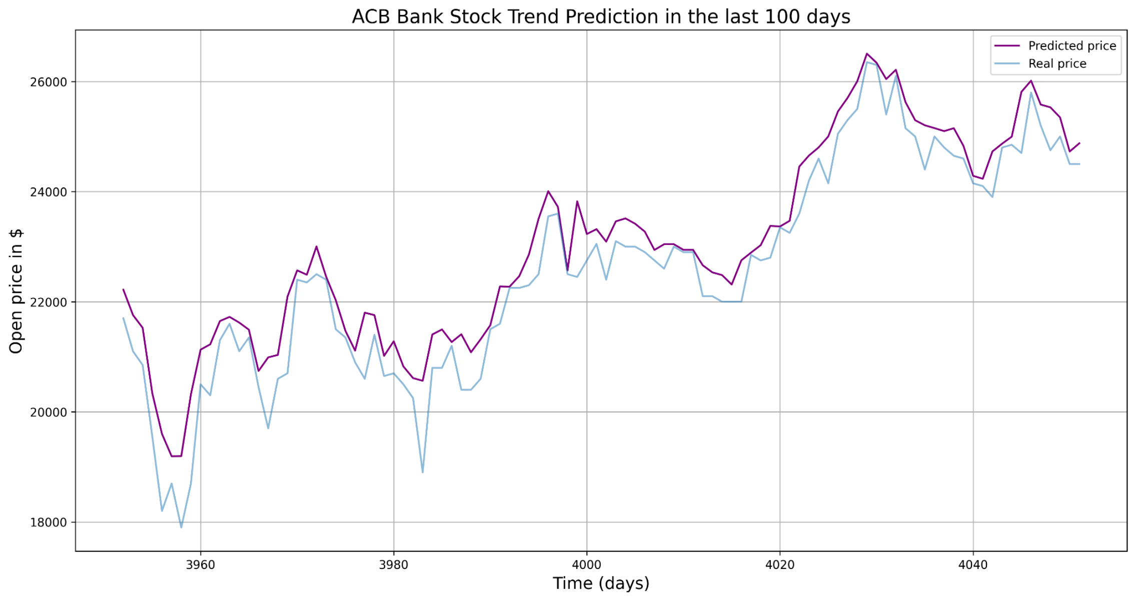
The graph illustrates the performance of the LSTM model in predicting NASDAQ AAPL stock prices over different time horizons. The purple line represents the model's 7-day horizon prediction, while the dark blue line represents the 1-day horizon prediction. Comparing these predictions to the actual prices (depicted by the orange line), it becomes evident that the 1-day horizon prediction aligns more closely with the real prices. This discrepancy suggests a potential issue of overfitting, highlighted by the substantial gap observed between the mean squared error (MSE) of cross-validation k-folds and the MSE on the test set.

Vietnam stock price prediction

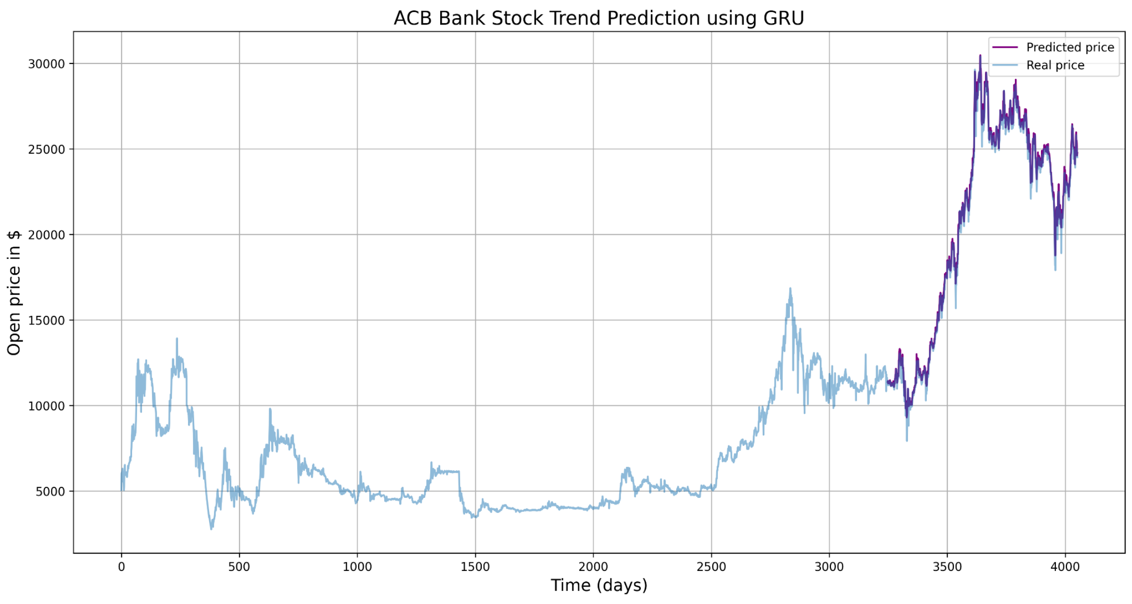
on the dataset of Asia Commercial Bank (Ngân hàng thương mại cổ phần Á Châu)

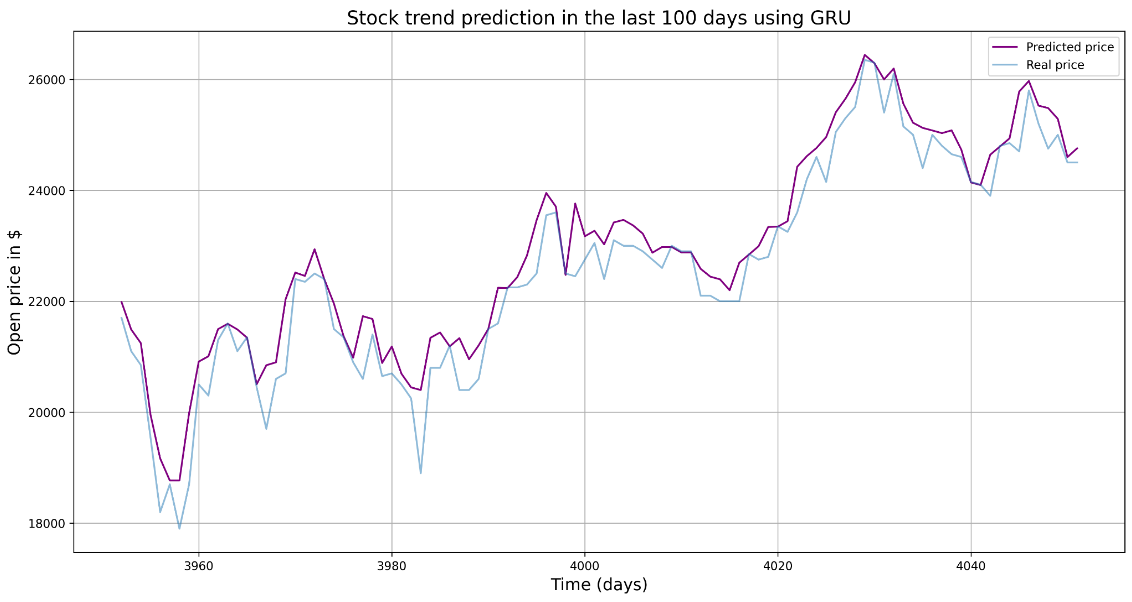
Prediction using LSTM



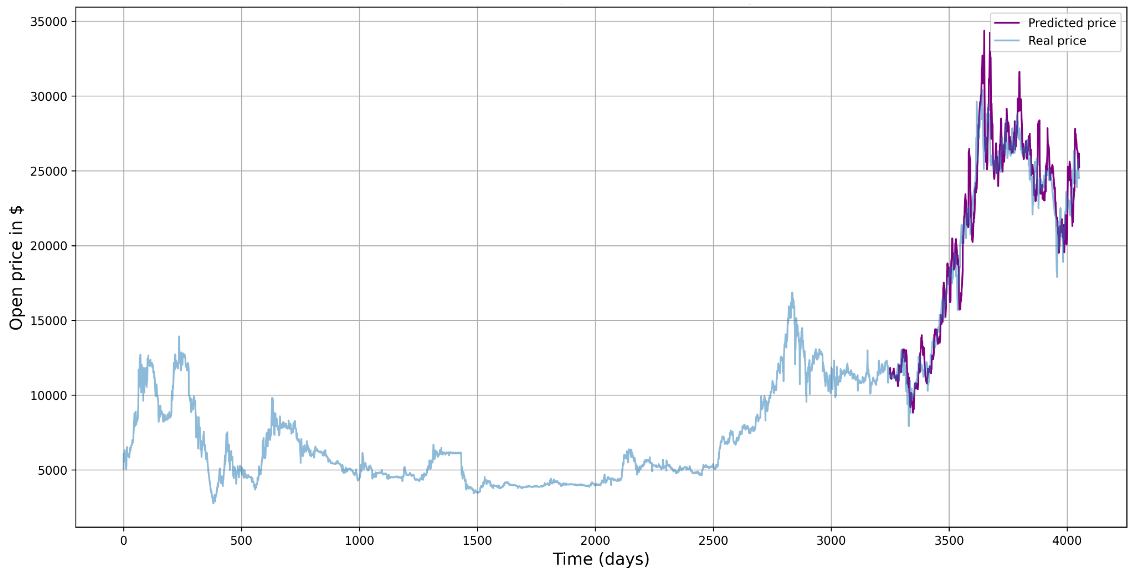


Prediction using GRU





Predicting Stock Prices Over a 7-Day Horizon on VNIndex ACB



Predicting the stock prices for the next seven days, starting from the most recent trading day.

Methodology

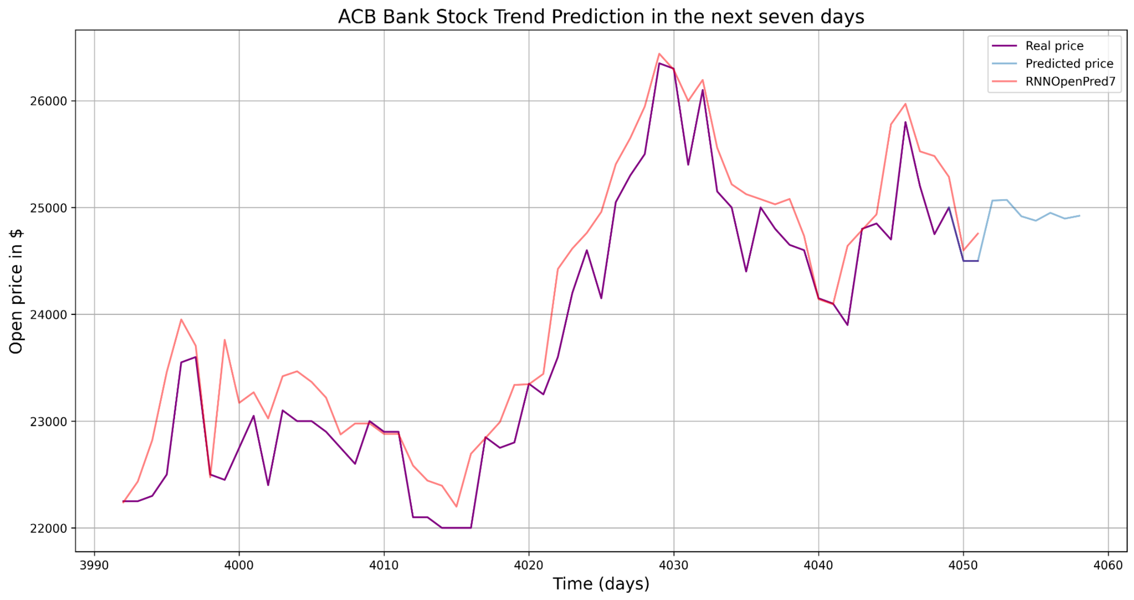
In brief, the RNN model is used in a for-loop where the predicted value will be added to the dataset for continuous prediction of the day after. This is done 7 times to get the prediction.

1. Copy the original dataset (vndf) into vndf_pred7.
2. Sort the data in vndf_pred7 by date in ascending order and reset the index.
3. Iterate over a range of seven days.
4. Retrieve the *ACTUAL open prices* for the last seven days from the 'Open' column of vndf_pred7.
5. Normalize the last seven days of open prices by subtracting the mean and dividing by the standard deviation.
6. Reshape the normalized data to match the input shape required by the prediction model.
7. Use the trained RNN_model to generate a prediction for the next day's open price based on the normalized data.
8. Denormalize the predicted price by multiplying with the standard deviation and adding the mean of the last seven days of open prices.
9. Convert the 'TradingDate' column in vndf_pred7 to a DatetimeIndex.
10. Append the *predicted next day's open price to the actual price dataset* and corresponding date to vndf_pred7.
11. Print the predicted next day's open price.

12. Retrieve the resulting dataset, excluding the last seven appended rows, which correspond to the predictions.

Output

```
1/1 [==============================] - 0s 406ms/step
Predicted next day's open price: 24611.234
1/1 [==============================] - 0s 18ms/step
Predicted next day's open price: 24788.926
1/1 [==============================] - 0s 17ms/step
Predicted next day's open price: 24627.023
1/1 [==============================] - 0s 17ms/step
Predicted next day's open price: 24630.445
1/1 [==============================] - 0s 19ms/step
Predicted next day's open price: 24624.293
1/1 [==============================] - 0s 16ms/step
Predicted next day's open price: 24685.316
1/1 [==============================] - 0s 34ms/step
Predicted next day's open price: 24687.996
```



Trading point identification

Methodology

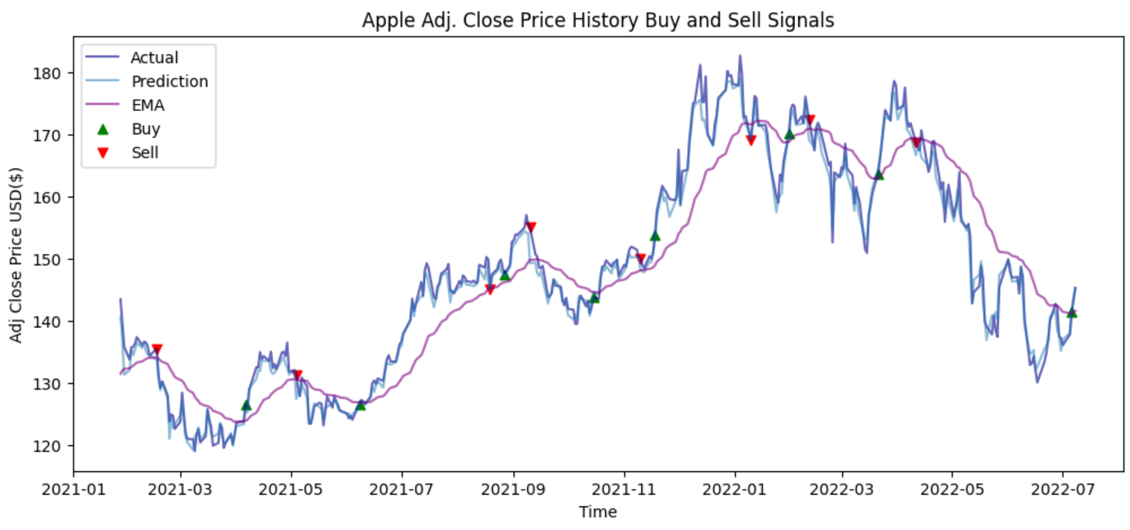
This project identifies trading points using Exponential Moving Average (EMA) in combination with predicted values obtained from a Recurrent Neural Network (RNN),

First, the EMA is calculated on the historical data by selecting a specific period, in this project it is 30 days. Then, the program Initializes the EMA with the first closing price in the historical dataset. Then, for each subsequent day, EMA calculation is done using the formula, which involves considering the previous EMA value. This process continues until EMA is computed for all historical data points.

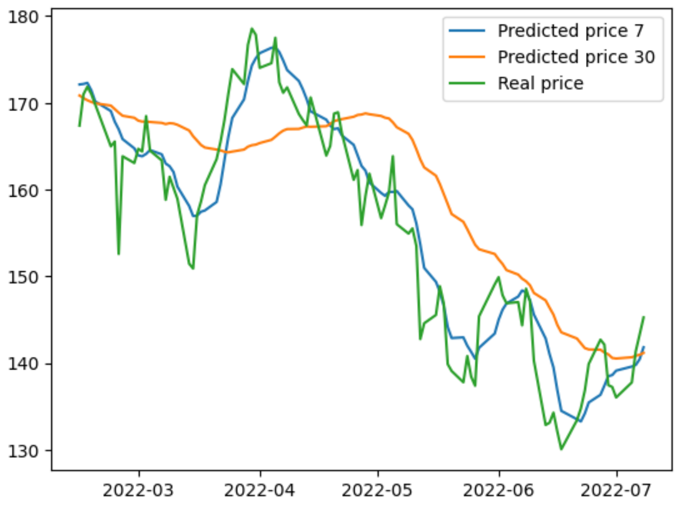
Next, this program utilizes the trained RNN model to generate predictions for future prices. This is done by providing the historical data as input to the model and ensuring that the predicted values correspond to the appropriate days in the historical dataset. With the predicted values at hand, it is then compared to the EMA line.

Compare the predicted values against the EMA line to determine their positioning. If the predicted values consistently lie above the EMA line, it might indicate a bullish trend, potentially signaling a buying opportunity. Conversely, if the predicted values consistently fall below the EMA line, it could suggest a bearish trend, indicating a potential selling opportunity. This is the conditions I have set in my buySell() function in the code.

Finally, since I have noticed the frequent trading points. In other words, there were cases where buying and selling points were very close to each other and the profit was not much. To eliminate buy and sell signals that are too close to each other,I have modified the buySell() function to check the time difference between consecutive signals and filter out those that fall within a specific time range.



Comparing with trading points generated from Simple moving Average (SMA)



In brief, the Exponential Moving Average (EMA) method differs from the Simple Moving Average (SMA) method when finding trading points in terms of weighting and responsiveness to recent data. The EMA places more weight on recent prices, giving them greater significance in the calculation. This means that EMA responds more quickly to price changes compared to SMA, which considers all prices equally. As a result, EMA may provide earlier trading signals and adapt faster to market trends. However, the choice between EMA and SMA depends on the specific trading strategy and the trader's preferences.

Portfolio Management

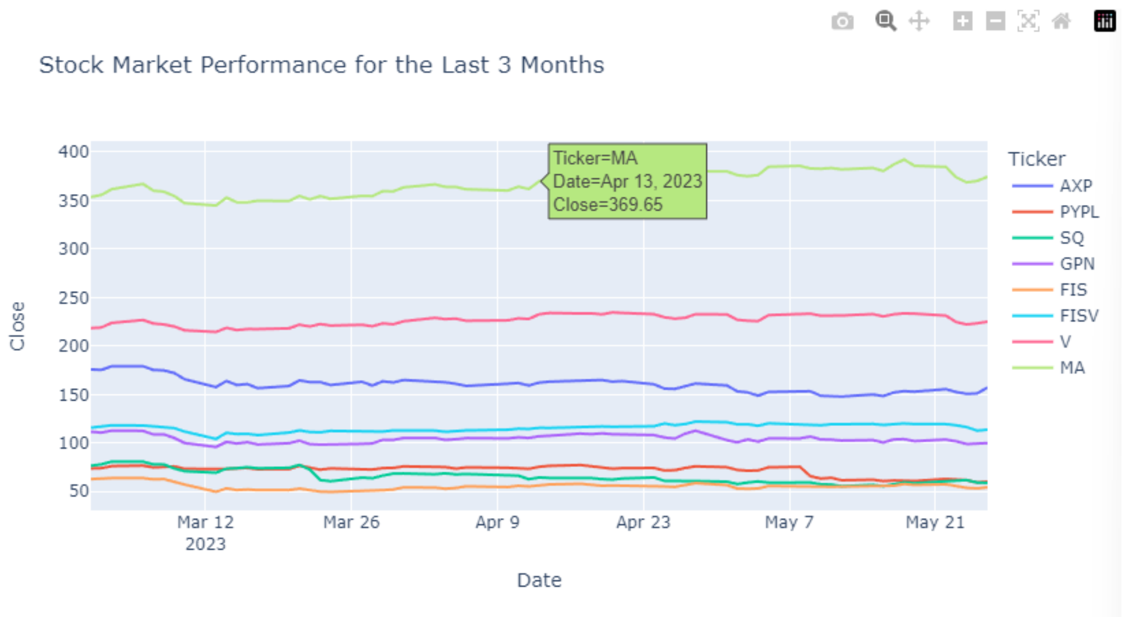
Since the trading point identification was done on ACB Bank, I continue with the banking sector for the portfolio management. Hence, my ticker choices for the portfolio is based on the banking industry in Vietnam.

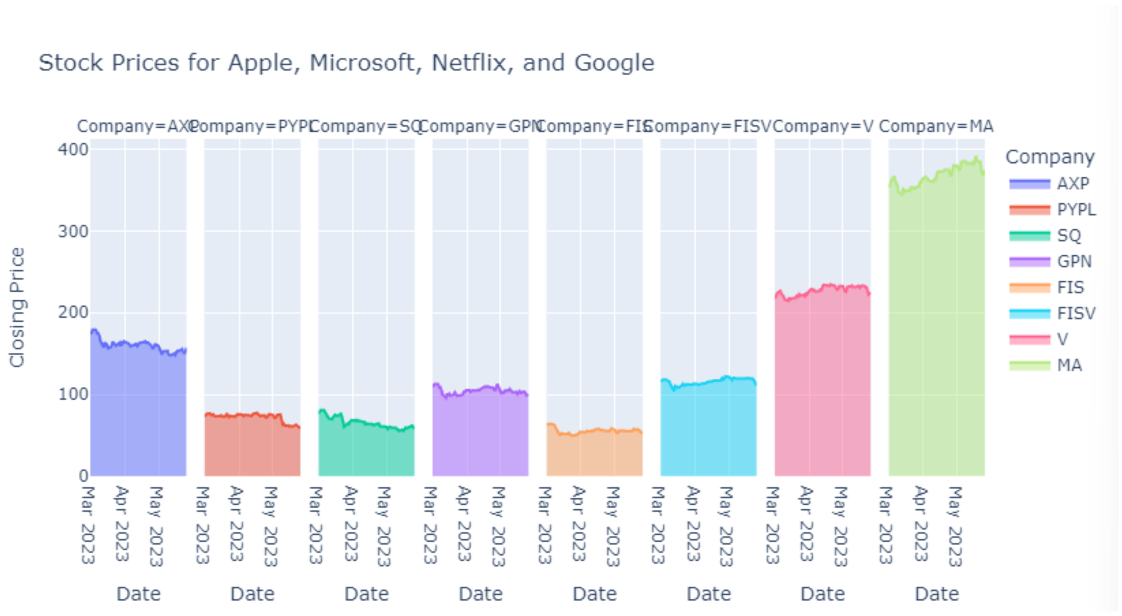
The ticker list is retrieved from the assigned data in the Industry Analysis folder.

```
['ACB', 'VCB', 'BID', 'CTG', 'VPB', 'TCB', 'MBB', 'SSB', 'STB', 'HDB']
```

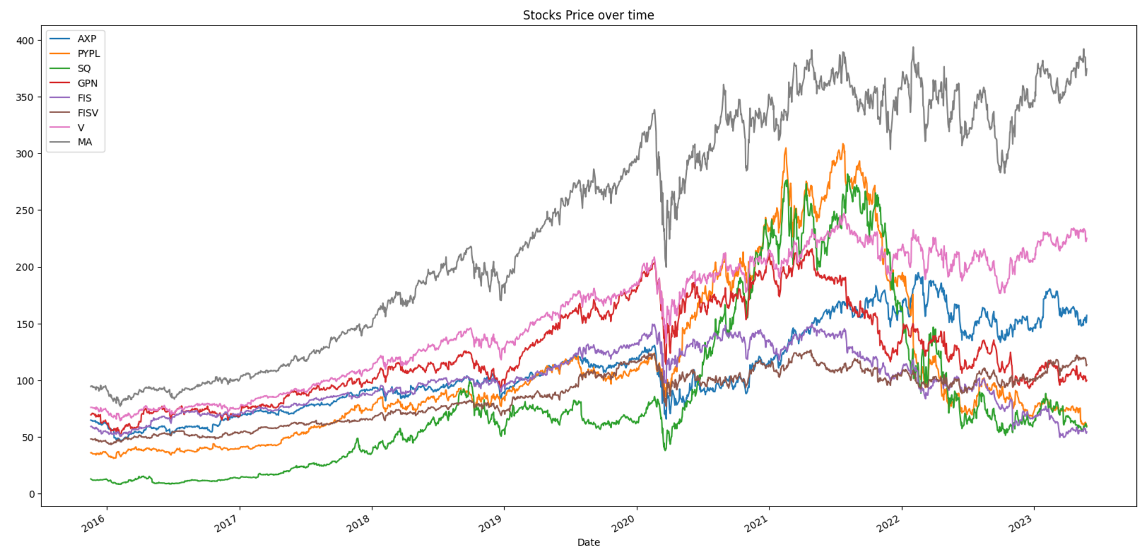
The tickers I have chosen for NASDAQ are:

```
tickers = ['AXP','PYPL','SQ','GPN','FIS','FISV','V','MA']
# Apple, Microsoft, Netflix, and Google
```





Stock Prices for Apple, Microsoft, Netflix, and Google
Company=AXP
Company=PYPL
Company=SQ
Company=GPN
Company=FIS
Company=FISV
Company=V
Company=MA
Closing Price
0
100
200
300
400
Mar 2023
Apr 2023
May 2023
Date
Company
AXP
PYPL
SQ
GPN
FIS
FISV
V
MA



Stocks Price over time
AXP
PYPL
SQ
GPN
FIS
FISV
V
MA
0
50
100
150
200
250
300
350
400
2016
2017
2018
2019
2020
2021
2022
2023
Date