Fulbright University Vietnam - Spring 2023

CS209 - Deep Learning For Artificial Intelligence

Student: Nguyen Thu Huyen - ID: 190033

Final Project Report

Deep Learning Application for Time Series Data - Stock Market

Introduction

Stock market is undoubtedly unstable, and more than often, unpredictable. This project includes multiple experiments to predict future values of stocks for multiple companies belonging to NASDAQ and Vietnam Stock Market, predict trading points and formulate a simple portfolio management strategy for investment in Vietnamese technology companies. The following parts of this report will cover research questions with experimental design, process and results, followed by discussion on challenges and difficulties throughout project execution.

Part 1: Nasdaq stock price prediction

In this part, I aimed at building a time-series model to predict the stock price of various companies in the NASDAQ stock market. A sample of one company (NVIDIA) was chosen to test and compare initial hypotheses.

Figure 1 demonstrates how stock price of the company changed in a selected time frame (2022-2023) in different time windows, smoothed by simple moving average for each window.

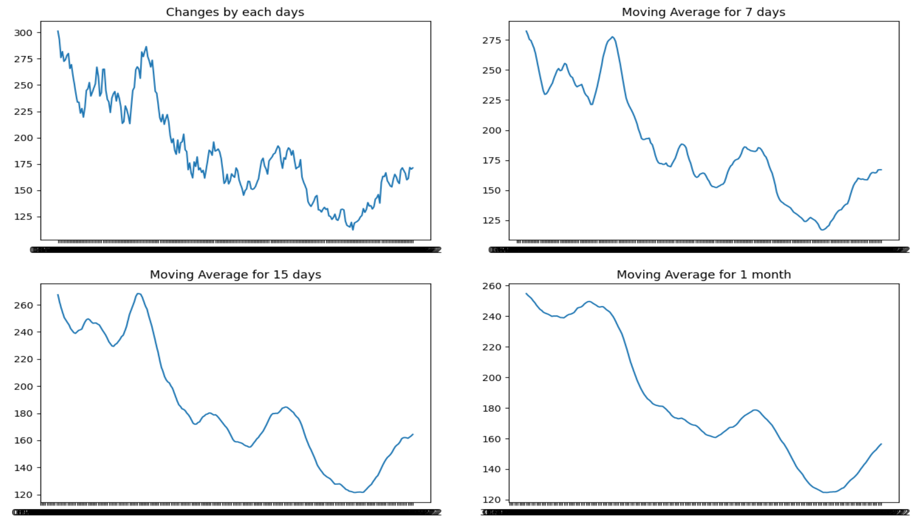
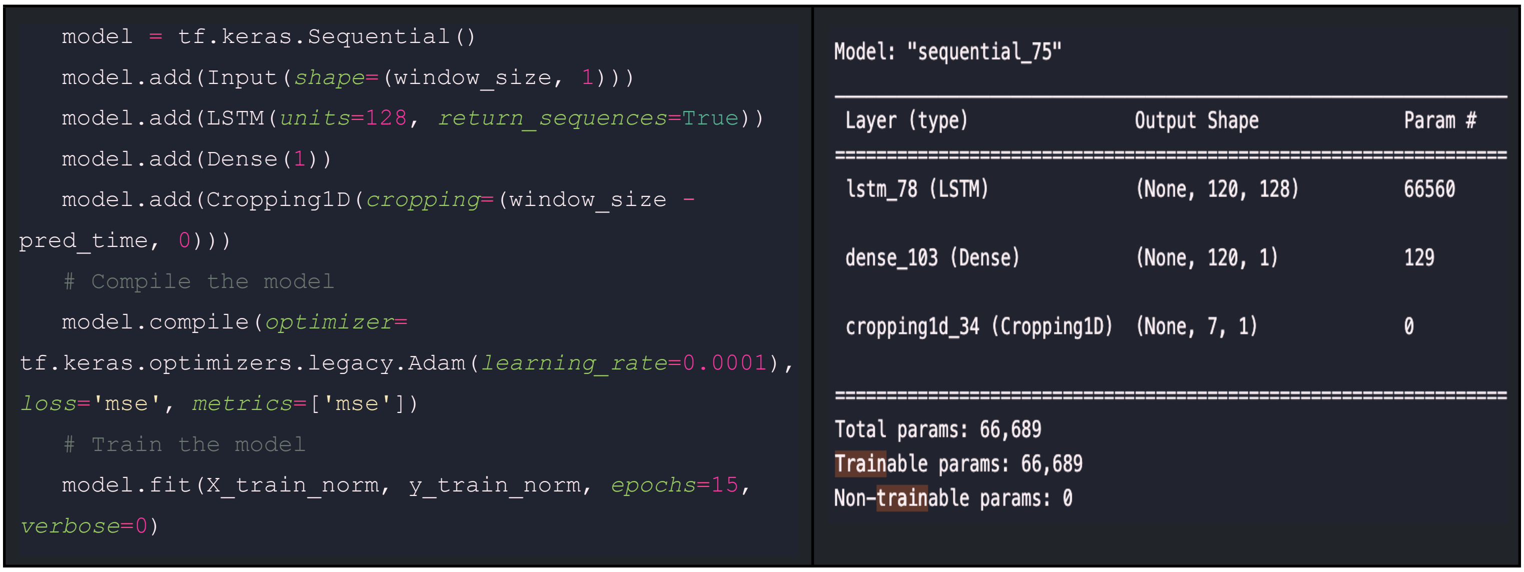


Figure 1. Closing price changes of NVIDIA during 2022-2023 in different time windows.

To target the main question, which is, to predict the price changes for the company, I first compared a Convolutional 1D model with LSTM for one day predicted based on previous 30-day history. As expected, the LSTM model outperformed Conv1D and thus was selected as the final model for prediction. Final LSTM model has structure as below:



```
    model = tf.keras.Sequential()
    model.add(Input(shape=(window_size, 1)))
    model.add(LSTM(units=128, return_sequences=True))
    model.add(Dense(1))
    model.add(Cropping1D(cropping=(window_size -
pred_time, 0)))
    # Compile the model
    model.compile(optimizer=
tf.keras.optimizers.legacy.Adam(learning_rate=0.0001),
loss='mse', metrics=['mse'])
    # Train the model
    model.fit(X_train_norm, y_train_norm, epochs=15,
verbose=0)
```

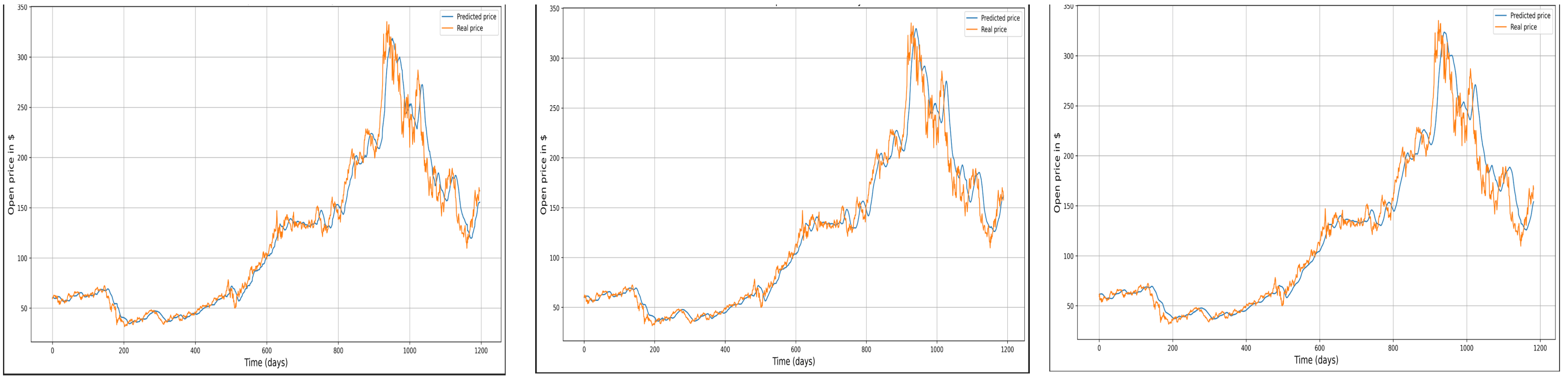
```
Model: "sequential_75"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_78 (LSTM)              (None, 120, 128)          66560

 dense_103 (Dense)           (None, 120, 1)            129

 cropping1d_34 (Cropping1D)  (None, 7, 1)              0

=================================================================
Total params: 66,689
Trainable params: 66,689
Non-trainable params: 0
```

While both Conv1D and LSTM perform well on next-day prediction, my experiment focused on which time window would be suitable to predict stock price for next 7-day prediction window. Therefore, I applied the same model with the same train-test split strategy for 30, 60 and 120 days windows. Performance on test set (train-test split 80-20 ratio) for each time windows is as follow:



Time window 30 days validation MSE: 0.2680392861366272
test MSE: 0.20683531219350865

Time window 60 days validation MSE: 0.11675182729959488
test MSE: 0.11503071208437919

Time window 120 days validation MSE: 0.06443630158901215
test MSE: 0.05788760879692795

Figure 2. Open price prediction on test set for NVIDIA on 30, 60 and 120 day time windows (from left to right), seven-day period

Based on observation, I moved on with the 120-day window strategy train-test split for cross-validation. On the notion that cross validation for time series data is highly dependent upon time window, I chose expanding window split as instructed. Folding was executed with help of sklearn’s TimeSeriesSplit module, which divides the pre-splitted train samples into 5 folds. After iteration of cross-validation, I selected the best-performing model (with min MSE) and saved to the “saved_model/nasdaq” folder.

Test performance for cross validation for NVIDIA was as follow:

```
On fold 0 - MSE on the test set:  0.06150350933617205
On fold 1 - MSE on the test set:  0.0579309338294184
On fold 2 - MSE on the test set:  0.06305126993034632
On fold 3 - MSE on the test set:  0.052216148954271305
On fold 4 - MSE on the test set:  0.05548103844506735
Overall MSE:  0.05803658009905509
```

I further iteratively applied the model to 9 other companies on NASDAQ Technology 100 Sector (^NDXT), with following performance.

Ticker	Company	Average MSE
ADSK	Autodesk Inc	0.0599
AAPL	Apple Inc	0.0466
FTNT	Fortinet Inc	0.0970
AMZN	Amazon	0.0613
MSFT	Microsoft Corporation	0.0423
NXPI	NXP Semiconductors NV	0.0691
QCOM	Qualcomm Inc	0.0538
INTC	Intel Corporation	0.0538
CTSH	Cognizant Technology Solutions Corp	0.0751

Figure 3. Test result for NASDAQ 100 Companies

Overall, test set MSE for 7-day prediction of selected NASDAQ companies, which are established organization, often fall in range 0.03 - 0.06

Part 2: Vietnam stock price prediction

This parts aim at answering basic questions listed in project description, including:

- Company filtering by data points and industry
- Visualizing price changes in certain time windows
- Stock price prediction for different companies (closing price) for a selected time window and prediction period

Experiment details and results are discussed as follow:

Following the result of Part 1, I selected a time window of 120 days for training and 7 days for testing. In order to perform cross validation for window sizes of 120 days, we need a large amount of data points for the total data sample. Therefore, I chose only companies with 1200 data points. Secondly, out of personal interest, I selected companies in Technology sector. Post-filtering, there are 28 out of 30 companies selected.

To first test the model performance, I selected the first instance of all filtered companies, which is CKV (COKYVINA Jsc.). Closing price changes for 1 day, 1 week, 15 days and 1 month periods from January 2022 to last day of data collection (Feb 28, 2023) are demonstrated below:

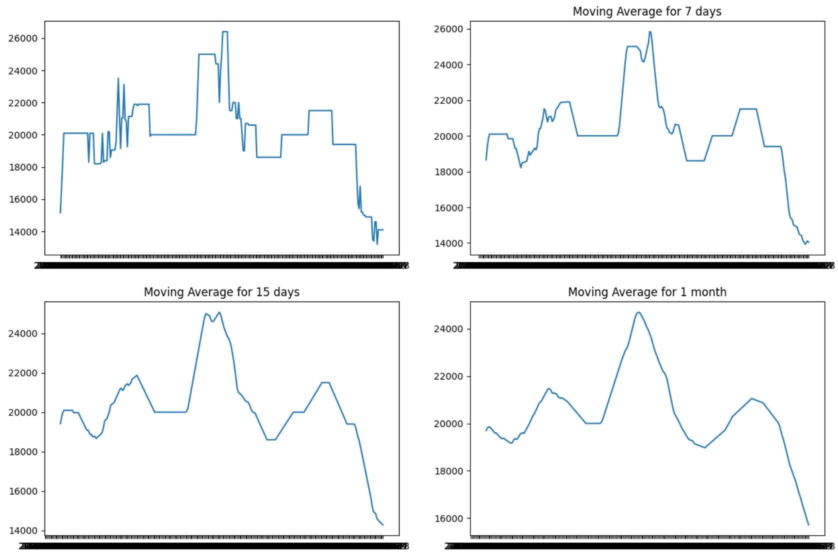


Figure 4. Price change in different time windows of COKYVINA Jsc. (Jan 2022 - Feb 2023)

The prediction process is similar to NASDAQ prediction, with similar LSTM model architecture, iterated for different train and test time windows (120 train - 7 test, 60 train - 3 test, 30 train - 1 test). Visually, on plot observation, sampling and training on 30-1 time windows gave better fit to test data. However, based on MSE on test set, the 120 train - 7 predict set gabe best MSE out of three (0.1068), while 30-1 window gave unexpectedly high MSE. This may be deducible to the volatility of the company's performance on daily basis and unstability Vietnam Stock Market. Details for experiment result on Figure 5.

Same strategy for cross validation as in previous part was applied with 5 fold expanding window cross validation, sampled for the same company on 120-7 window. Average MSE for cross validation is observed at 0.1343. Model training on selected companies showed that prediction accuracy is highly influenced by the company's own performance, as out of 28 samples, lowest MSE observed for 120-7 window was 0.034 while the highest of all was 0.2634.

Due to time limitation, I had written code to train model on all filtered companies, but only sample cross-validation. The same cross validation process will be saved for the following part.

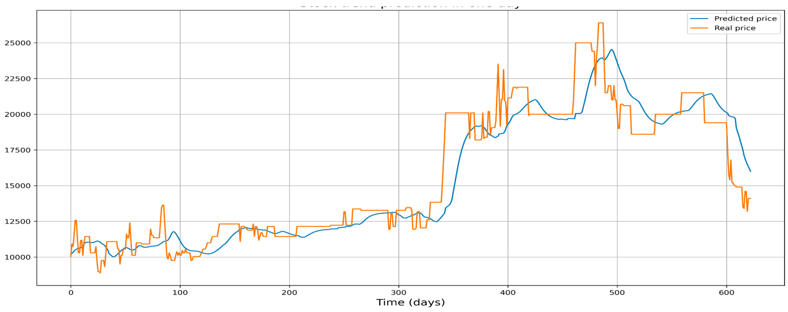
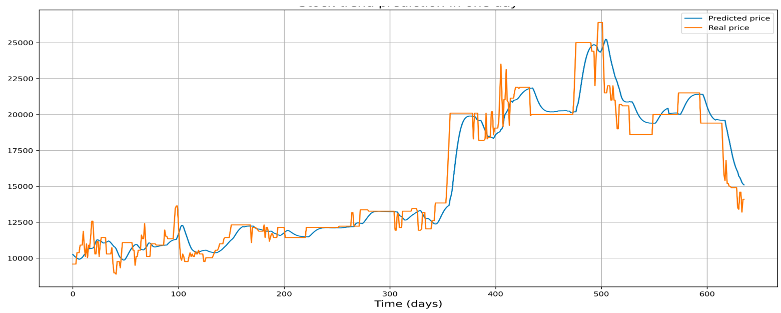
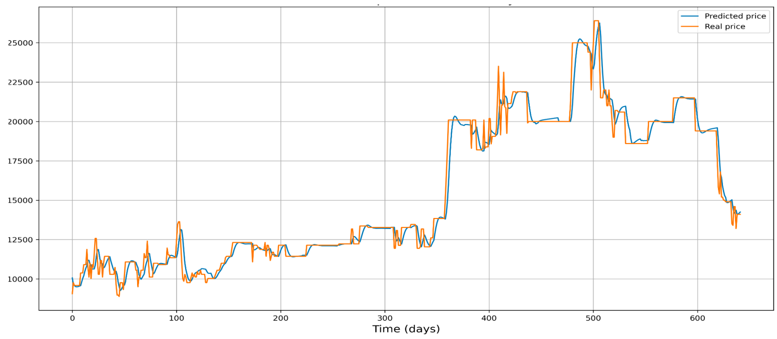
Train - test (days)	Test MSE	Plot on test result
120 - 7	0.1068	
60 - 3	0.1194	
30 - 1	1558.2385	

Figure 5. LSTM test prediction for COKYVINA Jsc. in different train test windows

Part 3: Vietnam trading point prediction

In order to answer the key questions of “What is a good signal to buy and sell stock?”, I take a personal, naive approach to this problem. My approach includes prediction based on historical price patterns, trading volumes and simple moving average as a technical indicator to inform buying and selling decision.

Specifically, I manually generated 2 labels “buy” and “sell” to indicate whether a data point (1 day) is a good time to buy / sell compared to the previous 7 day. The labeling task followed rather simple logic:

- For time t, within k=7 day:
    + if highest possible price of t (column [“High”]) is higher than highest price points of (t-7) day before, it is considered a selling point -> sell = 1, else = 0
    + if lowest possible price of t (column [“Low”]) is lower than lowest price points of (t-7) day before, it is considered a buying point -> buy = 1, else = 0

Interactive candle plot for price history of sample company (CKV) was used to easily observe ground truth for buying and selling point to inspect my reasoning. While I have yet derived a way to prove the mechanism’s validity, ground truth observation showed that this is an acceptable baseline logic for beginners as I am.

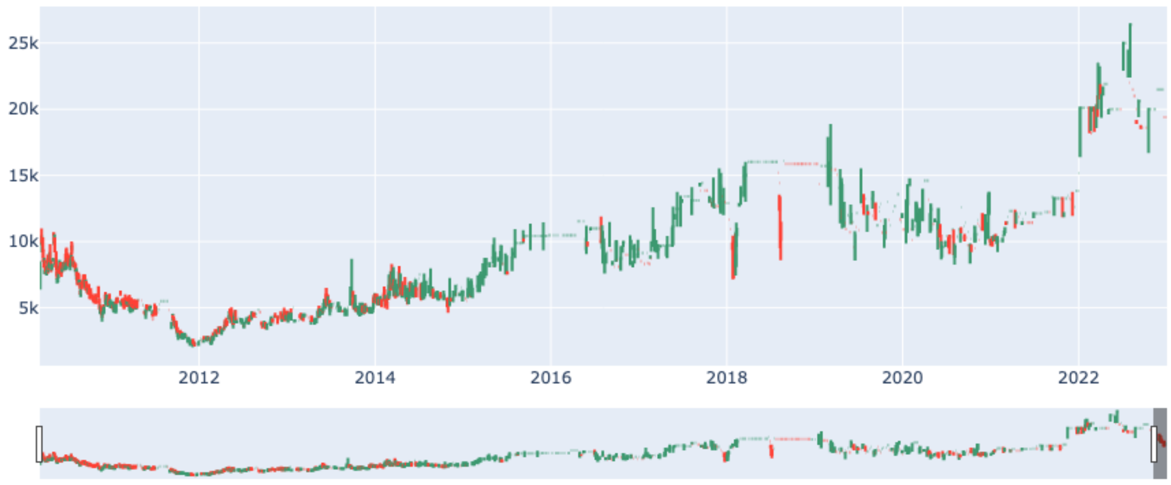


Figure 5. Candle plot for price history of COKYVINA Jsc.

(need to install plotly (`!pip install plotly`) on Jupyter Notebook to be able to view the interactive candle plot)

The buy and sell values are fed into 2 different LSTM binary classification models as y labels. Below are listings of experiments and results I have tried for buying point prediction task:

Model	**Layers & Params**	**Inputs**	**Test loss**	**Test accuracy**

LSTM
(0)
	Single LSTM 128 units,
	1 Dense, Cropping 1D
	learning_rate=0.0001
	epochs =15
		["Open", "High", "Low", "Close","Volume"]	0.7541	0.4778

GRU
(0)
	GRU 128 units
	1 Dense, Cropping 1D
	learning_rate=0.0001
	epochs =15
		["Open", "High", "Low", "Close","Volume"]	7.7070	0.4971

GRU
(2)
	GRU 128 units
	1 Dense, Cropping 1D, Dropout(0.5)
	learning_rate=0.0001
	epochs =30
		["Open", "High", "Low", "Close","Volume"]	7.7573	0.4971

GRU
(3)
	GRU 128 units
	1 Dense, Cropping 1D, Dropout(0.5)
	learning_rate=0.0001
	epochs =10
		["Open", "High", "Low", "Close","Volume"]	7.7572	0.4971

GRU
- SMA
	GRU 128 units
	1 Dense, Cropping 1D, Dropout(0.5)
	learning_rate=0.0001
	epochs =10
		["Open", "High", "Low", "Close","Volume"] as Simple Moving Average for 7-day window	7.5804	0.4971

Overall, GRU performed better in this task compared to LSTM, yet my attempts to add dropout layer, increase number of epochs seemed to no avail, as the model stopped improving after the 10th epoch.

My attempt at feature engineering by converting the price values to Simple Moving Average at 7 days windows also led to no improvement on the model’s performance. Test prediction of the GRU (3) model had loss at 7.4290 and accuracy at 0.5184.

In explanation for the unfruitful attempt, I suppose that another method to more accurately formulate the approach is needed, with the help of more technical indexes to aid the decision.

Part 4: Vietnam stock portfolio management

In order to figure the question of which companies to keep and to sell in one's portfolio, I first looked into the problem by common finding risk and potential technical indicators.

Relative Strength Index (RSI) is a common momentum indicator that compares a company security's strength on days when prices increase with its strength on days when prices turns lower. According to market technicians, oversold reading by the RSI in an uptrend is higher 30 and score lower than 70 suggests overbuying [1]. To be more sure of the signals, traders can adjust the RSI to set 20 and 80 as oversold and overbought levels [2]. The calculation of RSI is as follow:

$$\mathrm{RSI} = 100 - \left[\frac{100}{1 + \frac{n_{up}}{n_{down}}} \right]$$

where:
n_{up} = average of n-day up closes
n_{down} = average of n-day down closes
(most analysts use 9 - 15 day RSI)

Based on this finding, my strategy to decided on companies to hold and get rid of in portfolio will be informed by RSI index calculated from predicted price for 7-day future.

To calculate the price prediction for companies and start an initial portfolio, I filter tech companies in Vietnam stock market with the same strategy as in previous parts, which got me an initial list of 28 companies. Input parameters include Open, Close, High and Low prices to predict for 120 day train and 7-day output.

A single LSTM model with 128 units, 1 Dropout layer with 0.5 probability, 1 Dense Layer, 1 Cropping1D layer and Adam optimizer (learning rate 0.0001, 10 epochs) was used for this prediction task. Expanding window cross-validation with 5 time series folds was used to validate train result and select best model (minimum MSE). Average MSE of all folds were calculated and compared to a MSE threshold of 0.07. Companies with lower MSE than threshold will be selected to initial profile. For each company, best trained models by cross validation will be saved in "saved-model/vnstock" folder so that it can be later called to predict latest data.

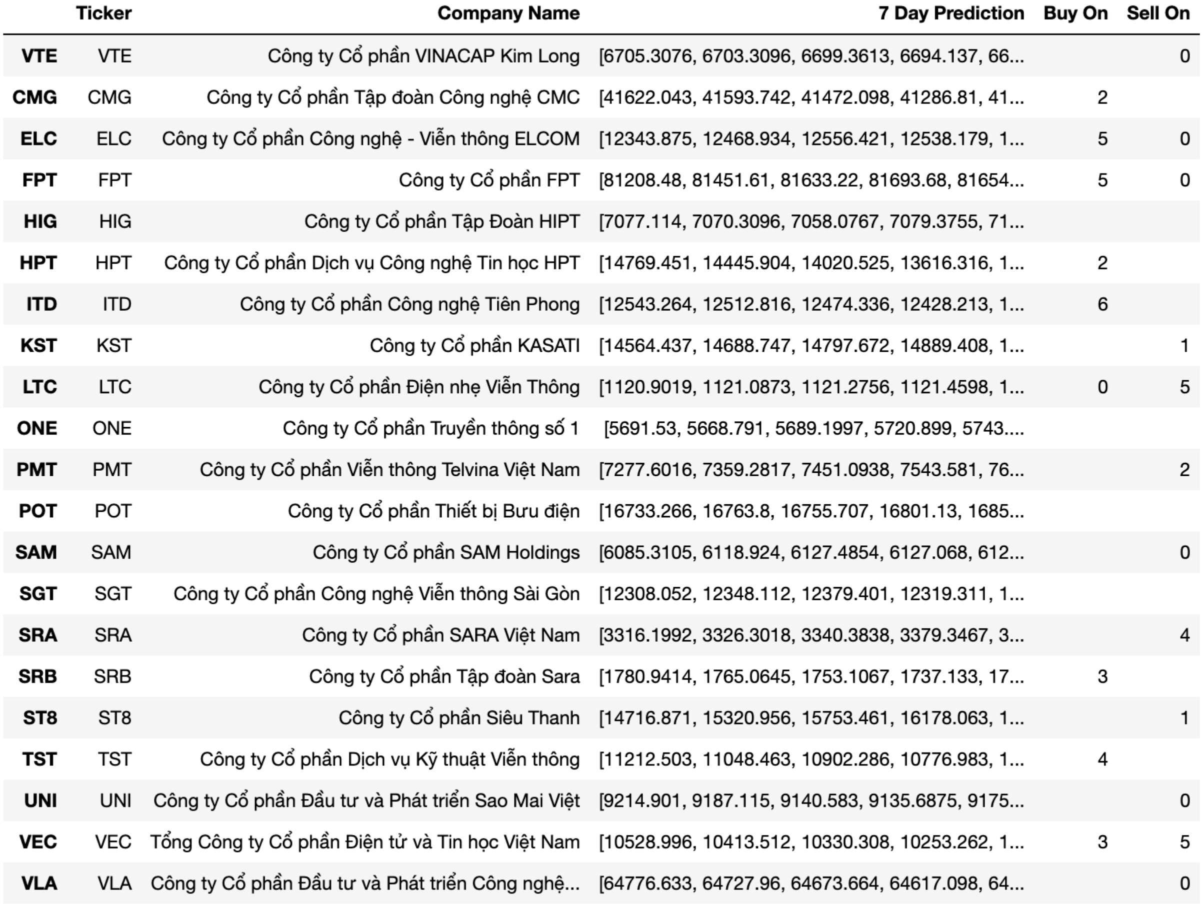
After training with cross-validation for 28 companies in the filtered list, 21 companies were selected to the portfolio. Portfolio management task for the following 7 days by the last time data was collected (Feb 28, 2023) is applied for this list with these steps:

- Predict stock price for 7 days after Feb 28, 2023

[1] Constance M. Brown. "Technical Analysis for the Trading Professional." McGraw Hill Professional, 2012.
[2] VnExpress, https://vnexpress.net/chi-bao-rsi-la-gi-4481781.html, Accessed May 20, 2023.

- Calculate RSI index for 7 predicted days company using predicted prices. Calculation was performed using pandas-ta[3], an open-source, easy to use library for stock technical analysis.
- Selling and buying recommendation was based on RSI index:
    - Recommend buying in day with RSI < 20 and has lowest RSI in the time window (index 0-6)
    - Recommend selling in day with RSI > 80 and has highest RSI in the time window (index 0-6)

For each company, the output result will include Company name, Ticker, predicted price for upcoming 7 days, date recommended for selling and date recommended for buying. The whole profile was saved in a .csv file named “my_portfolio.csv”.



	Ticker	Company Name	7 Day Prediction	Buy On	Sell On
VTE	VTE	Công ty Cổ phần VINACAP Kim Long	[6705.3076, 6703.3096, 6699.3613, 6694.137, 66...		0
CMG	CMG	Công ty Cổ phần Tập đoàn Công nghệ CMC	[41622.043, 41593.742, 41472.098, 41286.81, 41...	2	
ELC	ELC	Công ty Cổ phần Công nghệ - Viễn thông ELCOM	[12343.875, 12468.934, 12556.421, 12538.179, 1...	5	0
FPT	FPT	Công ty Cổ phần FPT	[81208.48, 81451.61, 81633.22, 81693.68, 81654...	5	0
HIG	HIG	Công ty Cổ phần Tập Đoàn HIPT	[7077.114, 7070.3096, 7058.0767, 7079.3755, 71...		
HPT	HPT	Công ty Cổ phần Dịch vụ Công nghệ Tin học HPT	[14769.451, 14445.904, 14020.525, 13616.316, 1...	2	
ITD	ITD	Công ty Cổ phần Công nghệ Tiên Phong	[12543.264, 12512.816, 12474.336, 12428.213, 1...	6	
KST	KST	Công ty Cổ phần KASATI	[14564.437, 14688.747, 14797.672, 14889.408, 1...		1
LTC	LTC	Công ty Cổ phần Điện nhẹ Viễn Thông	[1120.9019, 1121.0873, 1121.2756, 1121.4598, 1...	0	5
ONE	ONE	Công ty Cổ phần Truyền thông số 1	[5691.53, 5668.791, 5689.1997, 5720.899, 5743....		
PMT	PMT	Công ty Cổ phần Viễn thông Telvina Việt Nam	[7277.6016, 7359.2817, 7451.0938, 7543.581, 76...		2
POT	POT	Công ty Cổ phần Thiết bị Bưu điện	[16733.266, 16763.8, 16755.707, 16801.13, 1685...		
SAM	SAM	Công ty Cổ phần SAM Holdings	[6085.3105, 6118.924, 6127.4854, 6127.068, 612...		0
SGT	SGT	Công ty Cổ phần Công nghệ Viễn thông Sài Gòn	[12308.052, 12348.112, 12379.401, 12319.311, 1...		
SRA	SRA	Công ty Cổ phần SARA Việt Nam	[3316.1992, 3326.3018, 3340.3838, 3379.3467, 3...		4
SRB	SRB	Công ty Cổ phần Tập đoàn Sara	[1780.9414, 1765.0645, 1753.1067, 1737.133, 17...	3	
ST8	ST8	Công ty Cổ phần Siêu Thanh	[14716.871, 15320.956, 15753.461, 16178.063, 1...		1
TST	TST	Công ty Cổ phần Dịch vụ Kỹ thuật Viễn thông	[11212.503, 11048.463, 10902.286, 10776.983, 1...	4	
UNI	UNI	Công ty Cổ phần Đầu tư và Phát triển Sao Mai Việt	[9214.901, 9187.115, 9140.583, 9135.6875, 9175...		0
VEC	VEC	Tổng Công ty Cổ phần Điện tử và Tin học Việt Nam	[10528.996, 10413.512, 10330.308, 10253.262, 1...	3	5
VLA	VLA	Công ty Cổ phần Đầu tư và Phát triển Công nghệ...	[64776.633, 64727.96, 64673.664, 64617.098, 64...		0

Conclusion

Even though not all steps of this projects are successful, I had a lot of time experimented with deep learning models and got a first grasp of processing, understanding and formulate approaches to answer research questions regarding time series data. This experience also

[3] https://github.com/twopirllc/pandas-ta

provided me opportunity to understand stock market, stock data and find out useful libraries and techniques for data science with a lot of improvements to be made in the future.

Appendix

Submisison folder follows this structure:

```
├── 200154-project-notebooks
│   ├── Final-project-DL4AI-Part1-NASDAQ.ipynb
│   ├── Final-project-DL4AI-Part2-VNSTOCK.ipynb
│   ├── Final-project-DL4AI-Part3-Trading-Point-Prediction-VNSTOCK.ipynb
│   └── Final-project-DL4AI-Part4-Portfolio Management-VNSTOCK.ipynb
├── my_portfolio.csv
├── Final-project-DL4AI-Report.pdf
└── saved_model
          ├── nasdaq_CV: saved models for NASDAQ price predictions (Part 1)
          │   ├── folders containing model assets and history…
          │   └── ….
          └── vnstock: saved models for Vietnamstock price predictions (Part 4)
              └──  folders containing model assets and history…
```