Student Name: Thanh Tam Vo
Student ID: 103487596

COS30018 - Intelligent Systems
Learning Summary Report

Contents

1 Introduction

In this learning summary report, I would like to shortly show my effort during the course of COS30018. The report contains the data preprocessing techniques that we applied, the implementation result for using basic deep learning model such as RNN, LSTM, GRUs, and my research topic.

2 System Architecture

In every task, user works directly with the `.ipynb` file. Below is the list of files corresponding to task:

- `TaskB2.ipynb` accessible at Github
- `TaskB3.ipynb` accessible at Github
- `TaskB4.ipynb` accessible at Github
- `TaskB5.ipynb` accessible at Github
- `TaskB6.ipynb` accessible at Github
- `TaskB7.ipynb` accessible at Github

2.1 Before you can run

Before you can run, navigate to the folder `\COS30018\ProjectAssessment\TaskB1`, then:

- Open **PowerShell** as Administrator Privilege
- Execute this Command Line: `Set-ExecutionPolicy RemoteSigned`
- Execute this Command Line: `python3 -m venv myenv`
- Execute this Command Line: `myenv \Scripts \activate` to activate the virtual environment
- Execute this Command Line: `.\requirements.bat`

2.2 User Configuration

There are 6 coding tasks in total and user can configure the hyperparameters such as:

- Stock Ticker
- Start/End Date in Training/Testing Dataset
- Training Ration
- Scale Data Flag
- Scaling Method
- Desired Deep Learning Model and training parameters such as: the number of hidden layers, number of layers, number of training epochs, etc.

The Figure 1 is the area where user can config as they want

Figure 1: Configure Hyperparameters

2.3 Executing

For example, after configuring every, I would like to run the Task B4, then:

1. navigate to the folder `\COS30018\ProjectAssessment\TaskB4`
2. Open the file `TaskB4.ipynb` in Visual Studio Code
3. Choosing the kernel from `\COS30018\ProjectAssessment\TaskB1 \myenv \Scripts \python.exe`
4. Choose **Run All**

3 Data Processing Technique

3.1 Min-Max and Standard Scaler

The Data is scaled by the Min Max and Standard Scaler from the open-source library **scikit-learn** as in Figure 2

```
def DataScaler(stock_data, scaling_method=SCALING_METHOD):

    DatasetScaler = None
    ColumnScalers = {

    }
    if scaling_method == "MinMax":
        DatasetScaler = preprocessing.MinMaxScaler()


    elif scaling_method == "Standard":
        DatasetScaler = preprocessing.StandardScaler()

```

Figure 2: Using `MinMaxScaler()` and `StandardScaler()`

3.2 Data Spliter

After splitting the dataset, I implemented the `train_test_split()` function to split the dataset into 4 sets:

- X training set
- Y training set
- X testing set
- Y testing set

The number of instances in the *X Training Set* depends on the `TRAINING_RATIO`, which is configured by the user.

In Figure 3, the Apple Inc. Stock price is scaled into the range of [0, 1] after applying our data preprocessing technique.

```
+ Code  + Markdown
[12]  scaledStockData.head(10)                                                    Python
```

Date	Open	High	Low	Close	Adj Close	Volume
2014-01-02	0.035064	0.033398	0.034352	0.034218	0.034218	0.054690
2014-01-03	0.034908	0.034425	0.035039	0.033746	0.033746	0.057842
2014-01-06	0.034160	0.032675	0.032604	0.032880	0.032880	0.099644
2014-01-07	0.033912	0.033126	0.034437	0.034236	0.034236	0.045036
2014-01-08	0.034963	0.034514	0.034983	0.035435	0.035435	0.062468
2014-01-09	0.036569	0.035707	0.035732	0.035155	0.035155	0.053176
2014-01-10	0.036207	0.034747	0.034284	0.034122	0.034122	0.078268
2014-01-13	0.034813	0.033527	0.032613	0.032063	0.032063	0.085468
2014-01-14	0.033020	0.033175	0.033500	0.034085	0.034085	0.063496
2014-01-15	0.035107	0.033383	0.033887	0.033571	0.033571	0.078216

Figure 3: Apple Inc. Stock price dataframe

4 Experimented Machine Learning Techniques

4.1 Single Prediction

The problem we received is predicting the stock price of a company in the next day (Apple Inc. is the case in our experiment) given N observation days. The experiment is conducted with 3 models:

- Recurrent Neural Networks (RNNs)
- Long Short-term Memory (LSTM)
- Gated Recurrent Units (GRUs)

4.1.1 Observing the last 10 days

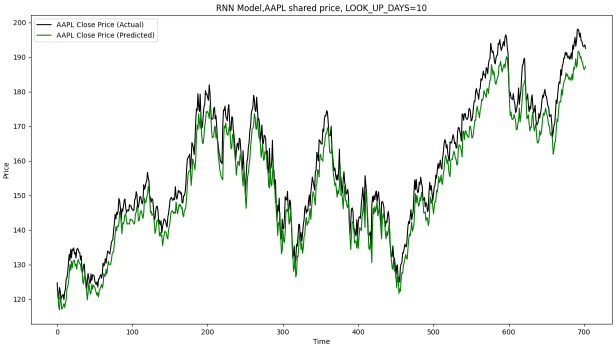

Figure 4: Predicted Result using RNNs Model

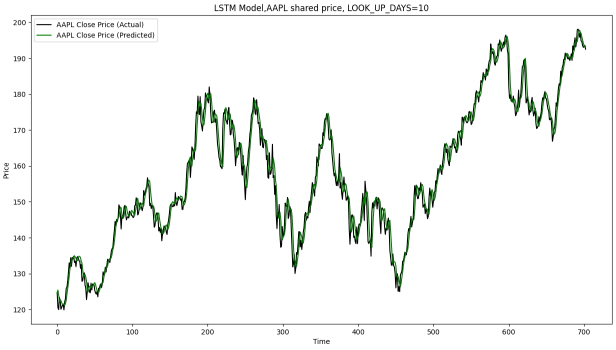

Figure 5: Predicted Result using LSTM Model

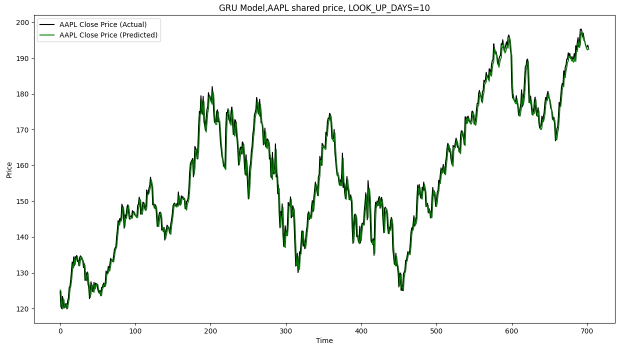

Figure 6: Predicted Result using GRU Model

4.1.2 Observing the last 60 days

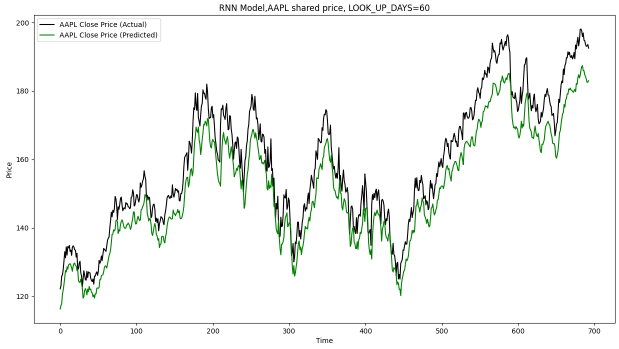

Figure 7: Predicted Result using RNN Model

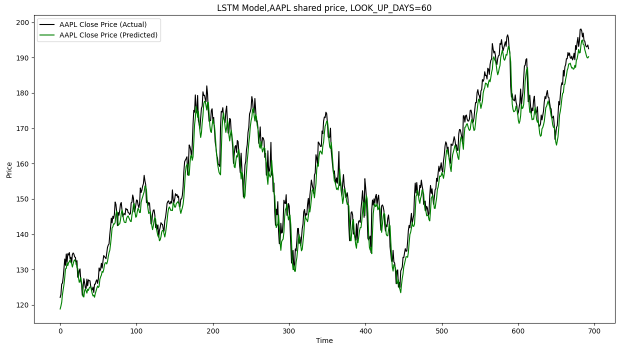

Figure 8: Predicted Result using LSTM Model

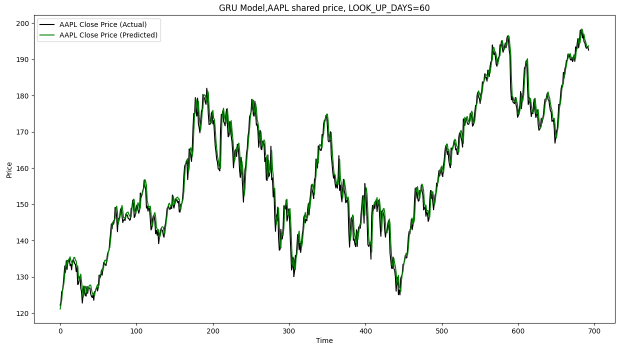

Figure 9: Predicted Result using GRU Model

4.2 Comments

The results above are the output from RNNs, LSTM, and GRUs with these configuration:

- `NUMBER_OF_LAYER = 2`
- `NUMBER_OF_HIDDEN_UNITS = 200`
- `NUMBER_OF_EPOCHS = 25`
- `LOSS_FUNCTION = "mean_squared_error"`
- `OPTIMIZER = "adam"`
- `DROP_OUT_RATE = 0.2`

4.3 Multistep Prediction

The problem we received is predicting the stock price of a company in the next K sequential days (Apple Inc. is the case in our experiment) given N observation days. Also, the model need to consider other factors such as: Open Price, High Price, Low Price, Adjusted Close Price, and Volumne. The experiment is conducted with 3 models:

- Recurrent Neural Networks (RNNs)
- Long Short-term Memory (LSTM)
- Gated Recurrent Units (GRUs)

4.3.1 Observing the last 10 days

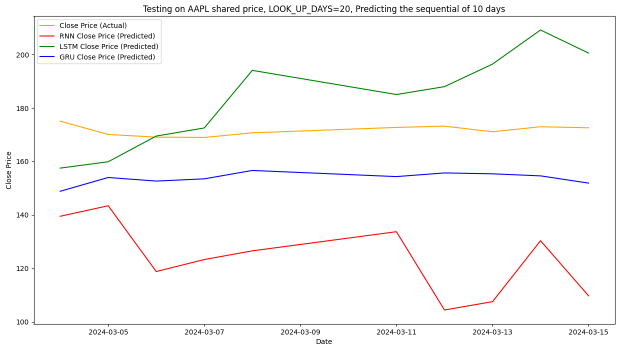

Figure 10: Predicted Result in the next 10 days

4.3.2 Observing the last 20 days

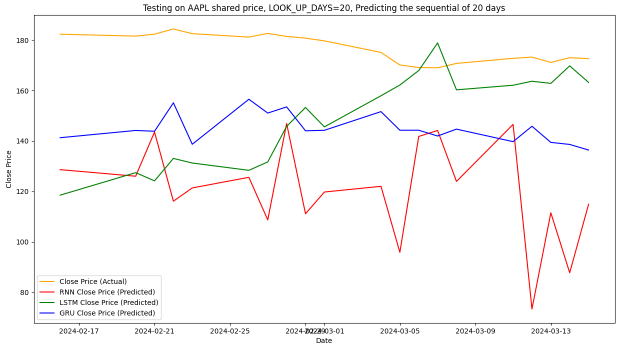

Figure 11: Predicted Result in the next 20 days

4.3.3 Observing the last 30 days

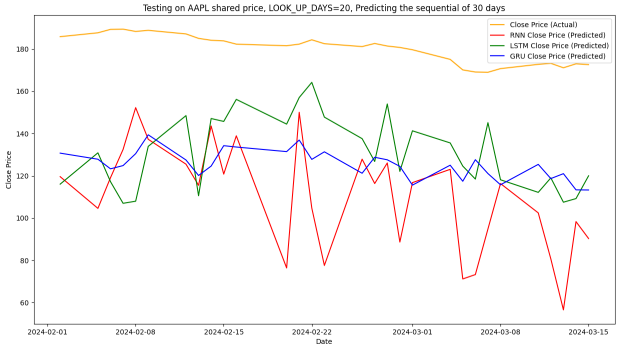

Figure 12: Predicted Result in the next 30 days

4.4 Comments

The predicted result from the above model is not good as we expected. The predicted values is quite different from the original values but overall, the GRU line somehow demonstrates the pattern of the original price. The more length in sequence, the more fluctuations we get. This problem will be solved in my extension task: Improving the long-term pattern forecasting of stock price.

5 Extension

5.1 Simple Ensemble Learning with ARIMA model

We take the average values from the ARIMA models with three different models: RNNs, LSTM, GRUs to get the final result:

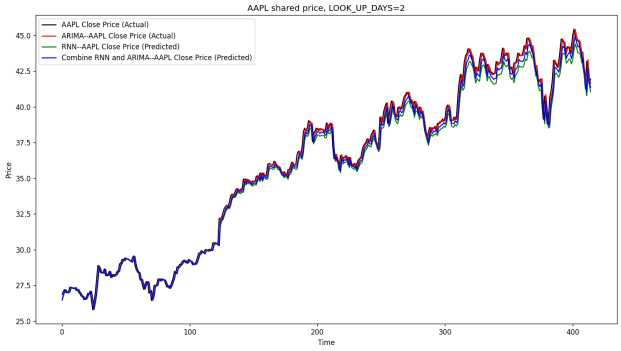

Figure 13: Predicted Result when combine ARIMA and RNN

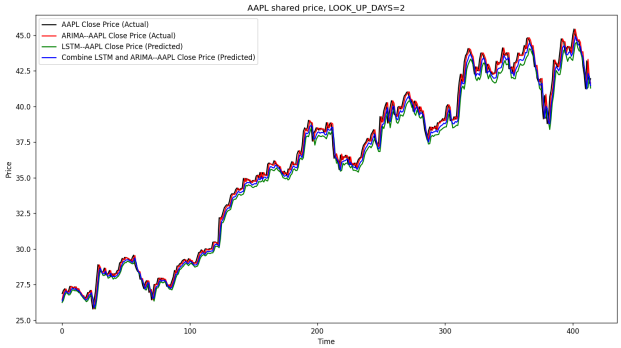

Figure 14: Predicted Result when combine ARIMA and LSTM

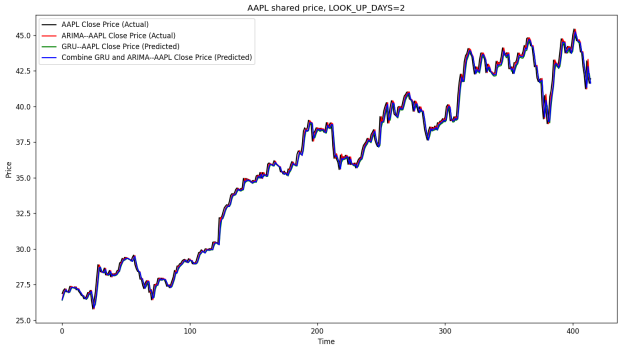

Figure 15: Predicted Result when combine ARIMA and GRU

5.2 Comprehensive Extension and Independent Research

My entire paper for this Comprehensive Extension and Independent Research can be found via this link. In this report, I will shortly explain my implementation from the paper *Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks*

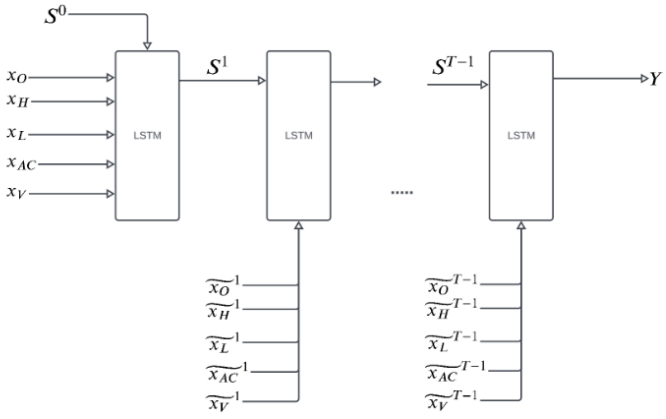

Figure 16: Implementation Of Paper

Given a sequence of T records, if we are not in the first stage, the input features will be transformed by a function bias. Again, my paper in the link below will explain more details. Below is some results from the Extension Task:

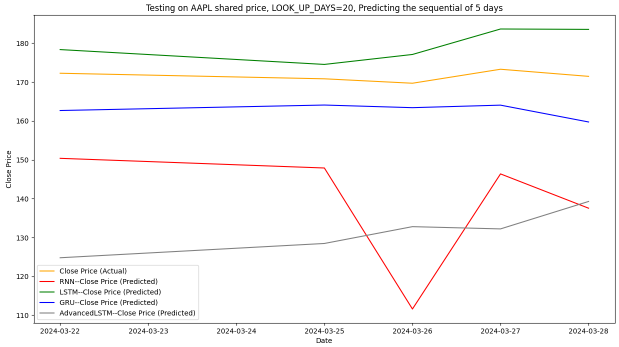

Figure 17: Predicting the Stock Pattern in the next 5 days

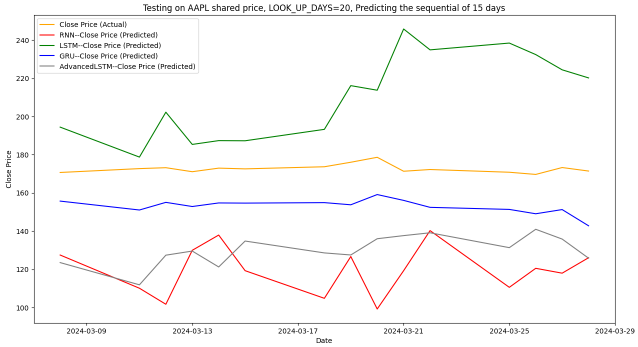

Figure 18: Predicting the Stock Pattern in the next 15 days

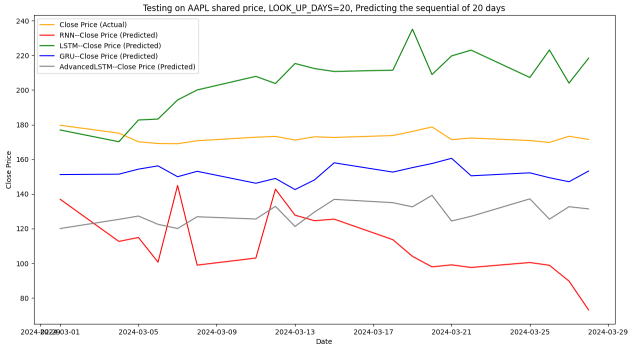

Figure 19: Predicting the Stock Pattern in the next 20 days

6 Demonstrations

Please follow this link to get my presentation

7 Critical Analysis and Conclusion

In the implementation for the Comprehensive Extension And Independent Research, there are not any absolute metrics to evaluate the result because we are talking about the pattern, if we are dealing with a simple regression problem, some metrics such as Mean Squared Error, Mean Absolute Error and R2 Score would be great. But now we are facing with the pattern forcasting in time series making myself confused to choose the approriate evaluation metrics.

This course plays an essential role in my future career because I actually started to train a model, especially the sequence model which requires knownledge in deep learning such as: Artificial Neural Networks, Backpropagation Algorithm, and Backpropagation Through Time, etc.