FINTECH – Homework 4

I) Plot Charts for the period of 2018

To study the movements of a stock price, the traders usually use several types of graphs and indicators. In this question we had to plot a Candlestick chart, a KD line chart and a volume bar chart. These charts have several utilities:

- **Candlestick chart with 2 moving average lines (10 and 30 days)**: it represents the 4 information about the stock price (Open, High, Low, Close). If the candle is red the stock price has dropped during the day and if green, it has risen. The moving average lines represent the closing price calculated with the following formula : $MA(i) = \frac{Close(i-n)+Close(i-n+1)+\cdots+Close(i)}{n}$ (MA of n days). It smooths out short-term fluctuations of the curve and highlight longer-term trends or cycles.
- **KD line chart:** it attempts to predict turning points by comparing the closing price of a security to its price range. When “D” is above 80 and decrease bellow, the stock is oversell and the trader should probably sell stocks. On the opposite, if “D” is below 20 and rise above, the trader should buy stocks.
- **Volume bar chart:** it represents the number of transactions related to the stock per day.

In this question, I plot these curves for the dataset on the year of 2018. *(Figure 1)*

II) Creation of the input data for the RNN

In this question, we add different features to the basic data: 'Moving Average 10 days', 'Moving Average 30 days', 'K' and ‘D’. I have chosen to add 6 more features that are commonly used in stock prediction:

- Relative Strength Index (RSI) with a time period of 9 days and 14 days
- Exponential Moving Average (EMA) with a time period of 30 days
- Moving Average Convergence/Divergence (MACD) with commons parameters (fastperiod=12, slowperiod=26, signalperiod=9)

Then, I normalise the data to rescale it between 0 and 1 according to the following formula.

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

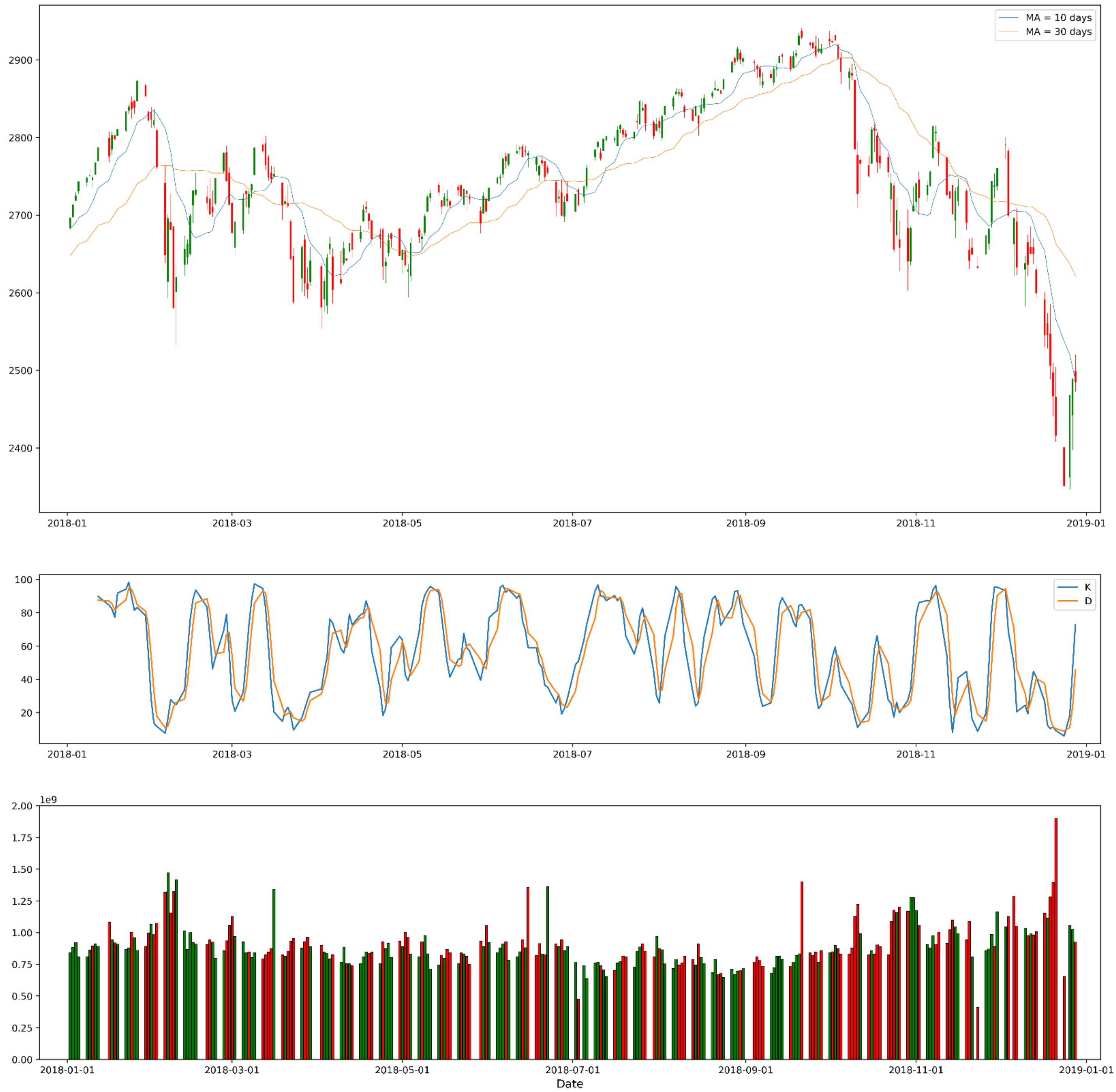

Figure 1: Candlestick Chart (with 2 moving averages lines of 10 and 30 days) - KD line chart – Volume bar chart for the year 2018

III) Creation of the RNN input

A RNN needs a very specific input in comparison with other deep learning models. Indeed, to predict a value, the model needs the values of the n previous samples on the timeline (n= time step). So, it is needed to regroup the data by packs of n values to match to a target corresponding to the last value of each pack. In the problem statement, the time step is 30 and I decided to have an input of 15 *(figure 2)*.

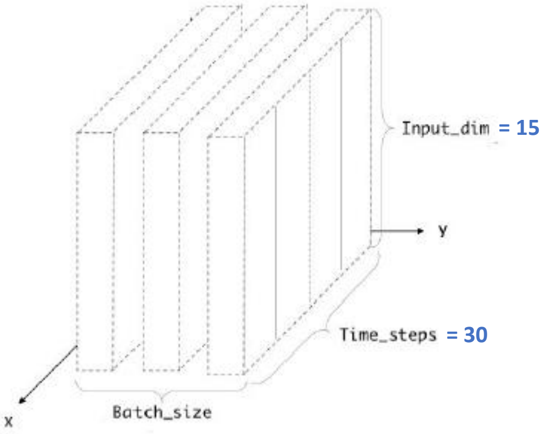

Figure 2: RNN input

Finally, I split the data into the training and validation set to get:

- Training = 1994-2016
- Validation = 2017-2019

IV) Construction of a RNN model with SimpleRNN layer

The first RNN model I implement has a *simpleRNN* layer. According to the subject of the exercise, the loss is defined as the "mean_squared_error" and the optimizer used is "Adam". The model has the following structure:

- *SimpleRNN* layer with 100 units
- *Dropout* layer with a rate=0.3
- *Dense* layer with 1 unit corresponding to the output of the model

We achieve the following results with a batch size of 20 and 15 epochs.

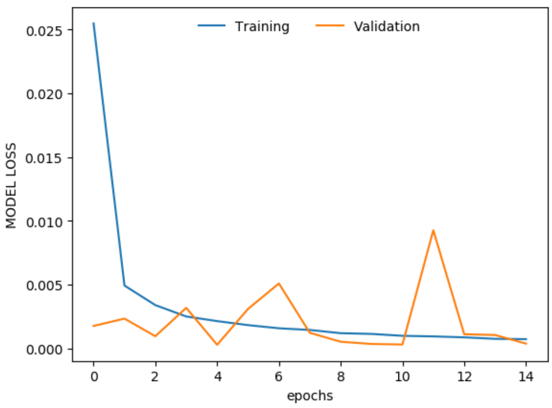

Figure 3: Loss curve of the simple RNN model

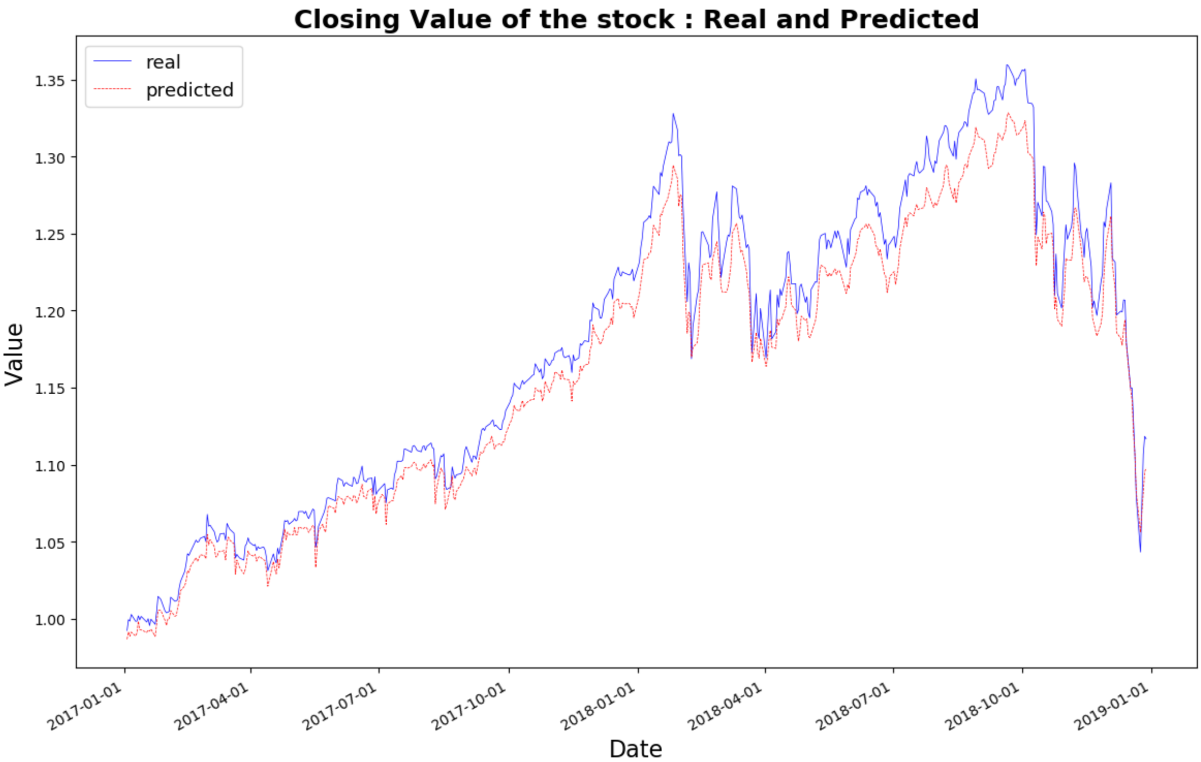

Figure 4: Results for the simple RNN model

V) Construction of a RNN model with a LSTM layer

We replace in this model the *simpleRNN* layer with a *LSTM* layer with 100 units. The batch size is 20 and there are 15 epochs.

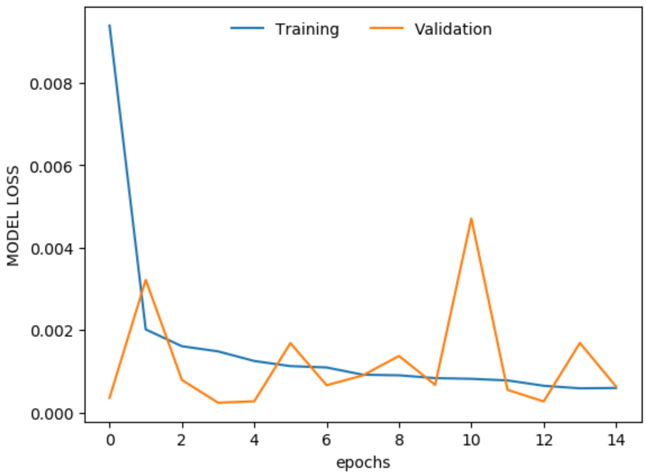

Figure 5: Loss curve of the LSTM model

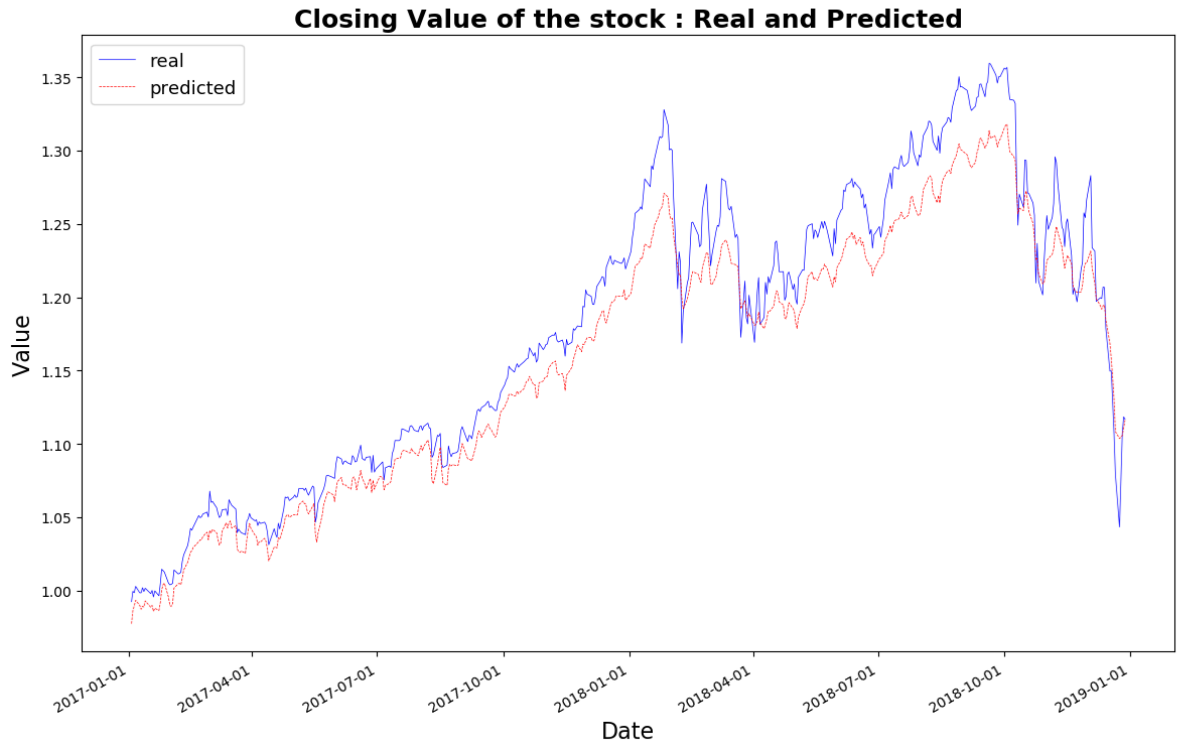

Figure 6: Results for the simple LSTM model

VI) Construction of a RNN model with a GRU layer

In this last model, I substitute the *simpleRNN* layer with a *GRU* layer with 100 units. The batch size is 20 and there is 15 epochs.

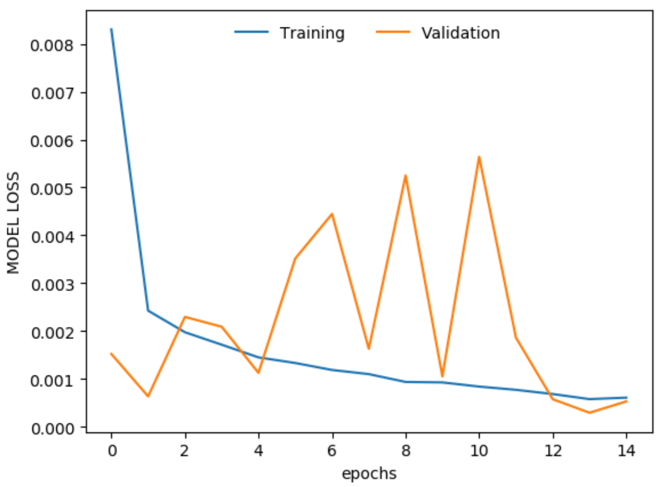

Figure 7: Loss curve of the GRU model

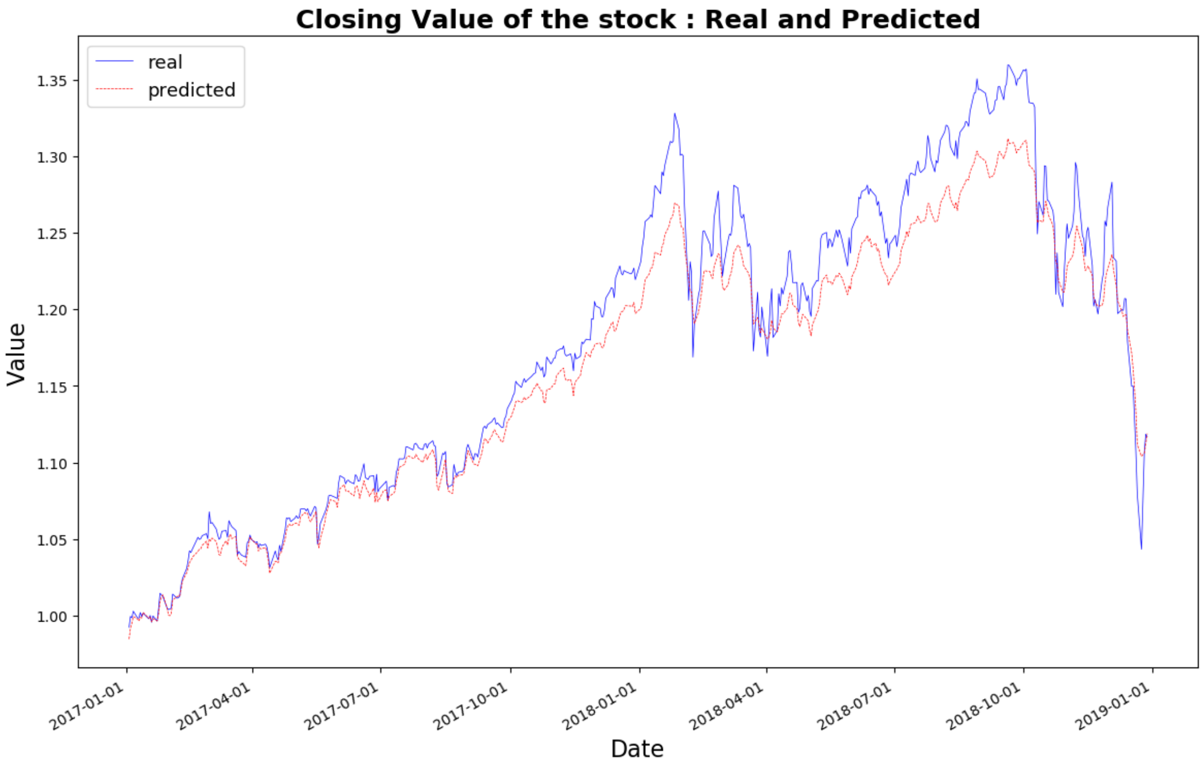

Figure 8: Results for the GRU model

VII) Comparison of the results

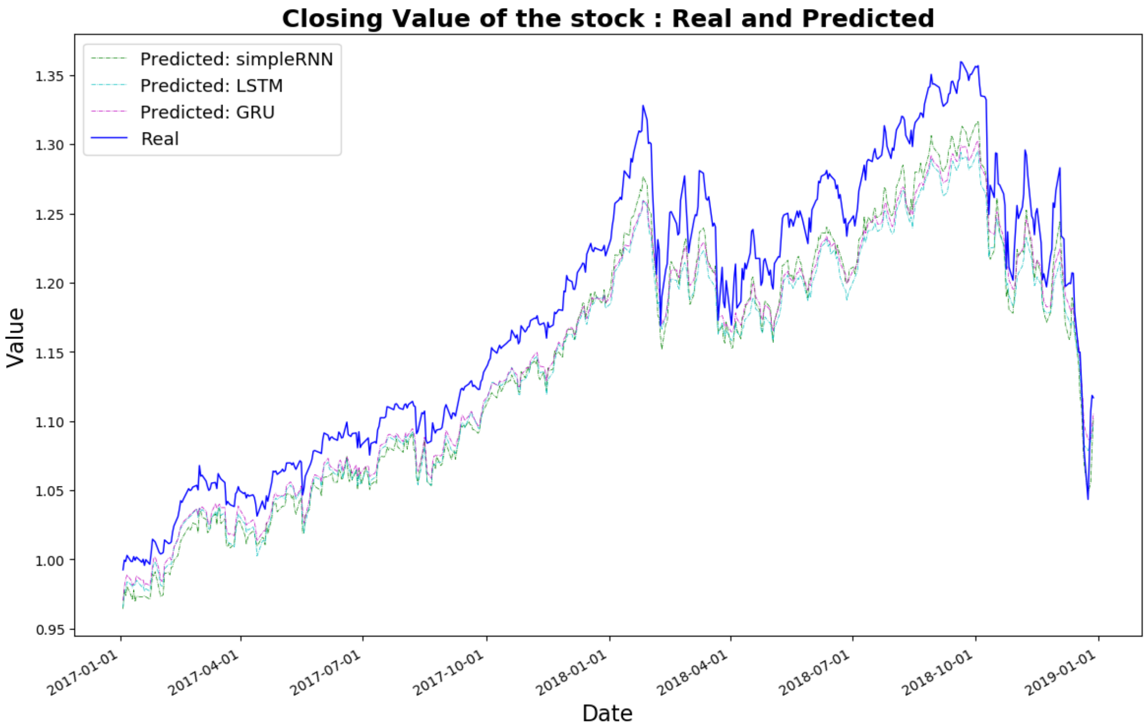

Figure 9: Result of the 3 different methods with equivalent parameters

Let's first compare the loss curves of the different models together. The training loss of the different functions decrease fast and are very similar the one from the others. However, the final loss of the GRU and LSTM is slower than the simpleRNN one which show us that these 2 models have a better prediction. But it seems like for the GRU model, the training curve is a little more below the validation curve than for the other model. It is then possible that this model overfit a little more than the others.

For the different models, the prediction is not bad. At the beginning, the GRU model achieve a slightly better result than the other models. But at the end, it is the simpleRNN one which have a result closer than the real curve.