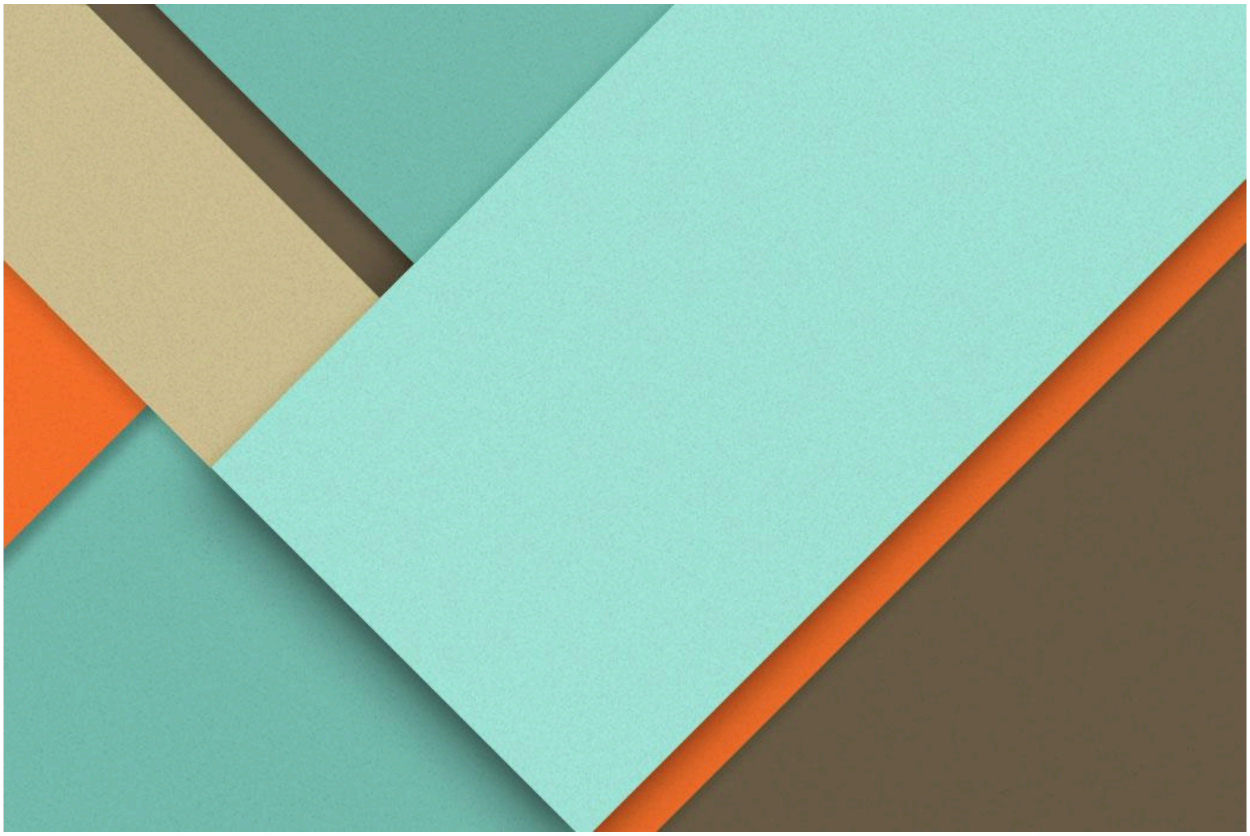


ECE 9063 Tesla Stock Price Prediction

Group:

Yanzhou Yang (yyang853@uwo.ca)

Zhiben Li (zli2285@uwo.ca)

Zhenyu Wu (zwu622@uwo.ca)

Tianqi Chen(tchen546@uwo.ca)

Yang He(yhe288@uwo.ca)

Western University

Course ECE 9063

Instructor: Dr. Katarina Grolinger

Overview

Stock, or capital inventory, consists of the shares by which a corporation distributes its ownership. Since a limited company needs to raise long-term funds, it issues shares to investors as part of the ownership certificate of the company's capital, so that shareholders can obtain stock dividends or cash dividends and share profits brought by the company's growth or trading market fluctuations. But also share the risk of what goes wrong.

The trend of the stock can also reflect the situation the company has operated so far. For example, If a company's stock has a consistent upward trend, things are usually going well for the business. However, if the stock seems to be on the decline, the business is likely going through some difficulties.

The world's first manufacturer of self-driving cars, Tesla is the leading manufacturer of electric vehicles in the United States. In 2018, Tesla rose to the top spot among manufacturers of electric vehicles worldwide and also became the sixth business to surpass $1 trillion in revenue in 2021, setting a benchmark for the new energy industry. Its CEO, Elon Musk, became the world's richest man in October. A few years ago, purchasing Tesla stock was a very wise move. However, along with many EV accidents, many people are now pondering whether now is a good time to purchase Tesla stock or whether to buy and when to sell.

Goals

Forecast Tesla stock price by applying machine learning models, and evaluate its performance.

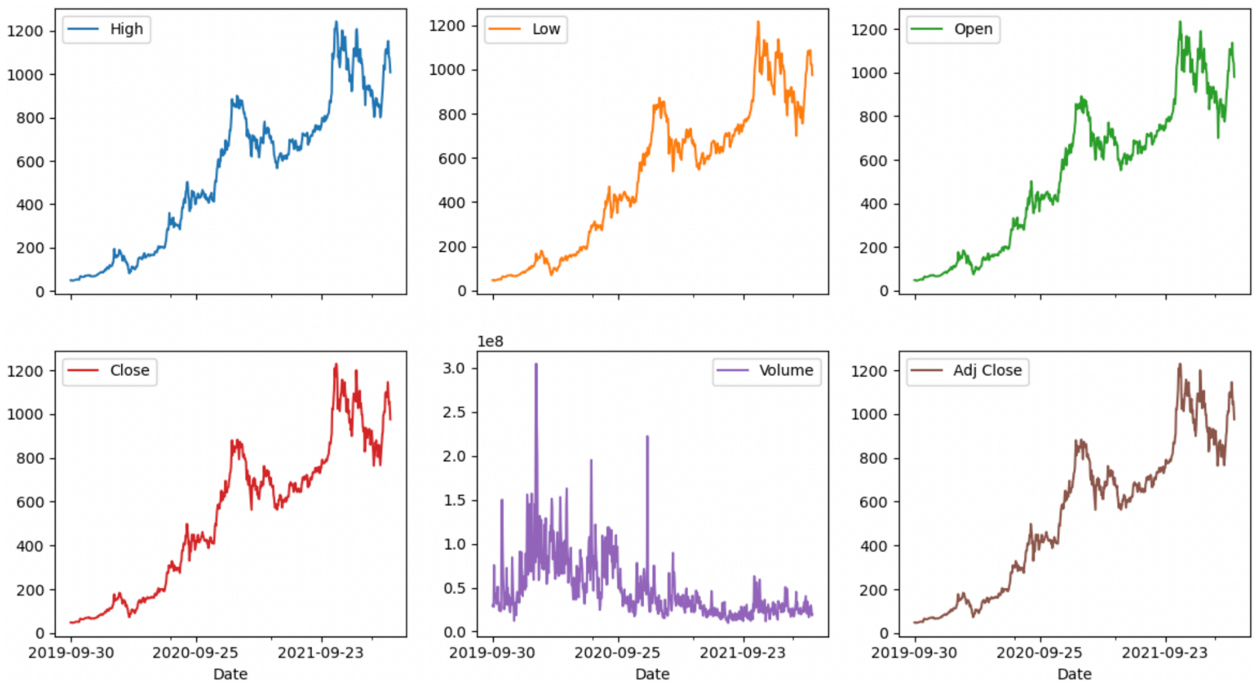
Data Set

Our dataset consists of the historical monthly price in US dollars of Tesla Inc. from 9/30/2019 to 4/11/2022. It is arranged in 6 columns: Date, Highest price, Lowest price, Opening price, Closing price, and Stock volume.

The link of our dataset is provided below:

https://www.kaggle.com/datasets/jillanisofttech/tesla-stock-price

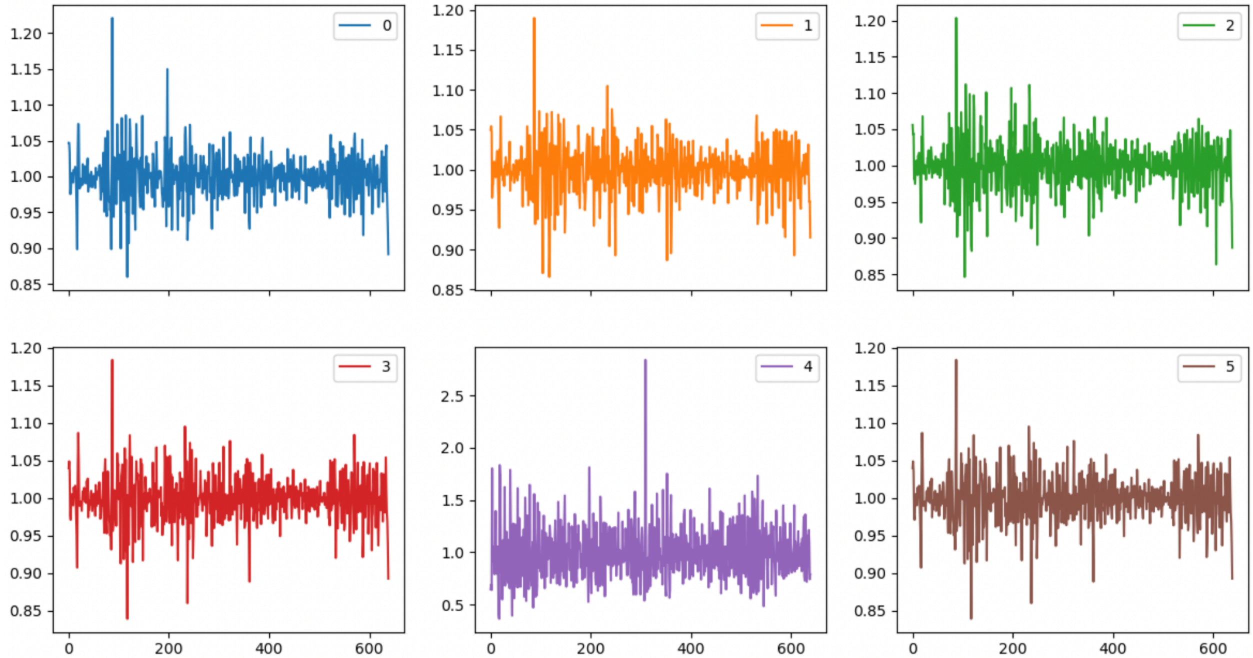
Here are the six attributes in data according to the date, we can see that except for the volume, every other attribute has a trend in it, so we need to detrend them.



Data preprocessing:

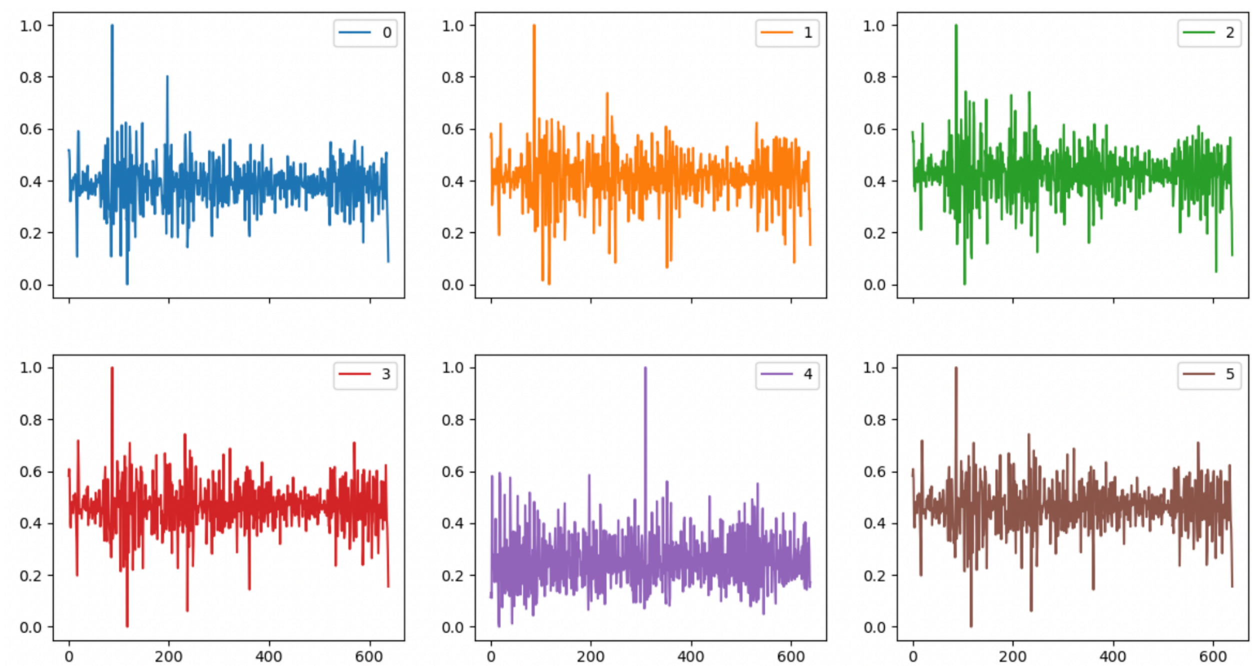
Detrending (Seasonal decompose):

Time series detrending is optional. Nevertheless, stationary time series simplify training models considerably. Detrending time series can be done in a variety of methods. Here we used a multiplicative model, so we need to divide the trend from the values.



Normalize:

After detrend, we use minmax normalization to range the data from 0 to 1.



Date set was divided into 80% training set and 20% test set. The first 511 data points will be utilized as our train data and the remaining data points will be used as our test data because the sequence length is 639.

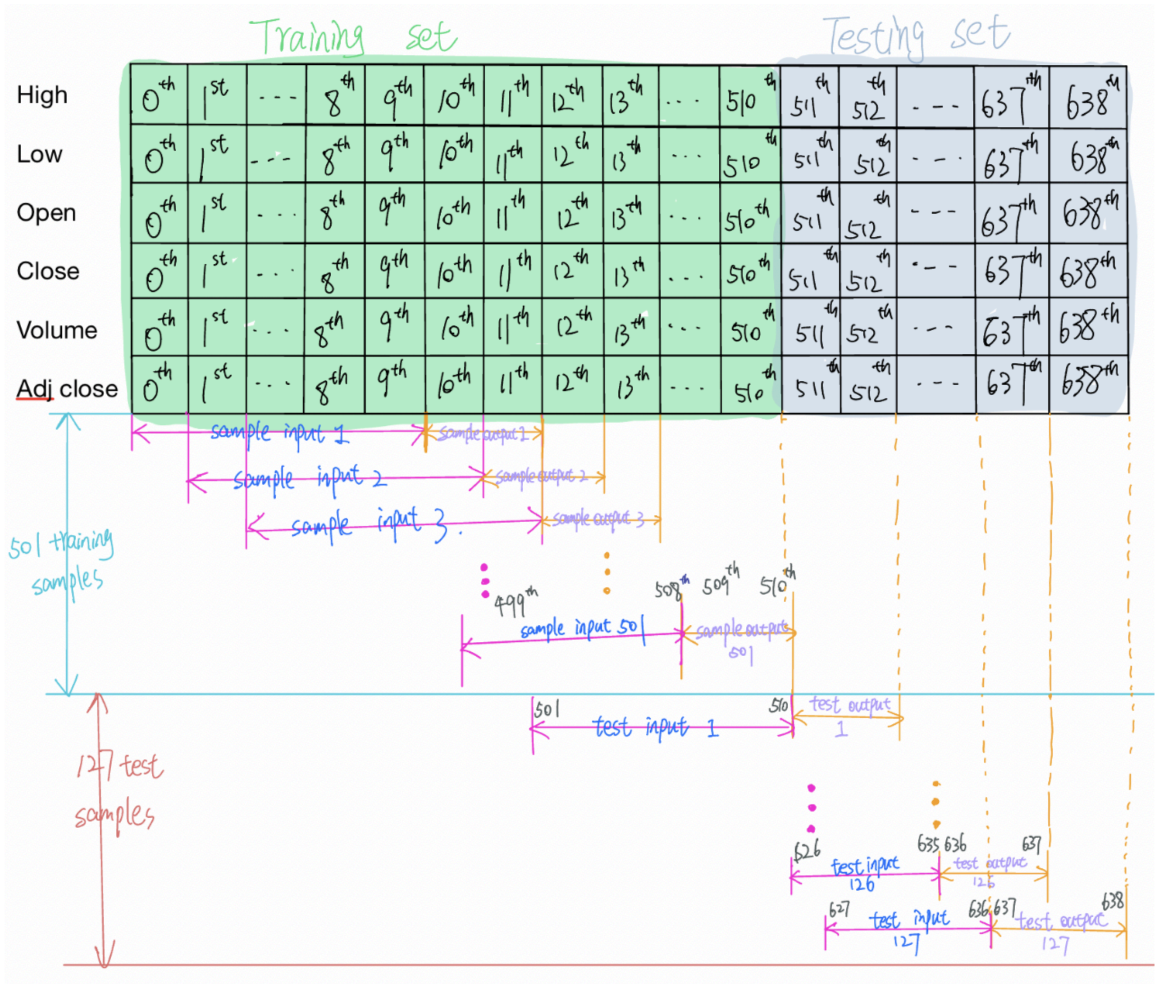
Training dataset:

```
            0         1         2         3         4         5
0    0.517535  0.568116  0.587027  0.582300  0.114900  0.582300
1    0.517413  0.581761  0.550466  0.608062  0.131655  0.608062
2    0.479109  0.558485  0.553859  0.566995  0.111598  0.566995
3    0.319760  0.305658  0.378279  0.400786  0.581183  0.400786
4    0.321985  0.356063  0.382409  0.382271  0.226007  0.382271
..        ...       ...       ...       ...       ...       ...
506  0.344729  0.382421  0.411332  0.452027  0.183865  0.452027
507  0.435307  0.425928  0.477553  0.470100  0.476088  0.470100
508  0.396169  0.412081  0.429558  0.459706  0.234667  0.459706
509  0.351188  0.398991  0.391748  0.459475  0.161382  0.459475
510  0.407297  0.425268  0.420622  0.486950  0.278290  0.486950

[511 rows x 6 columns]
```

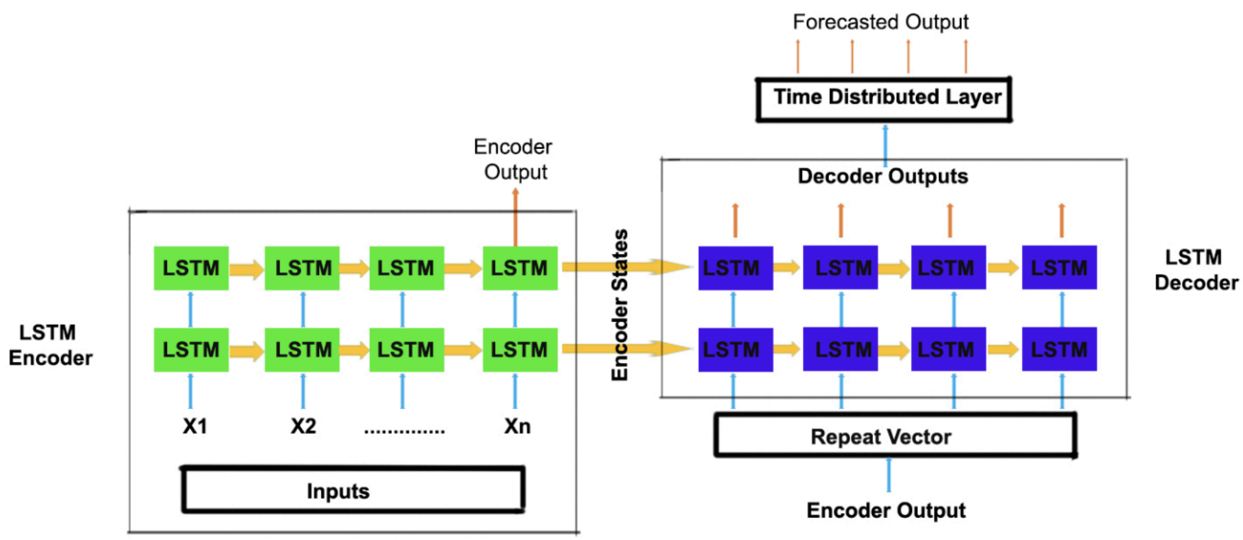
Testing dataset:

```
            0         1         2         3         4         5
511  0.370755  0.398452  0.449811  0.440404  0.246860  0.440404
512  0.372743  0.393165  0.400578  0.444020  0.198398  0.444020
513  0.390221  0.410666  0.424574  0.468360  0.395637  0.468360
514  0.367743  0.411023  0.430560  0.447673  0.188811  0.447673
515  0.341120  0.393258  0.406403  0.427531  0.129875  0.427531
..        ...       ...       ...       ...       ...       ...
634  0.508112  0.511107  0.566305  0.498423  0.297448  0.498423
635  0.352228  0.371830  0.436908  0.411407  0.342812  0.411407
636  0.272101  0.289586  0.309836  0.376669  0.291817  0.376669
637  0.194179  0.291632  0.279165  0.289502  0.152761  0.289502
638  0.087877  0.152573  0.112997  0.155261  0.171497  0.155261

[128 rows x 6 columns]
```

Truncate: We separate our dataset to a subset via sliding the input window, length is equal to 10 and the length of output window is 2 time steps. In result, we will have a (501, 10, 6) matrix for training input, a (501, 2, 6) matrix for training output, a (127, 10, 6) matrix for testing input, and a (127, 2, 6) matrix for testing output. Then we finish the data preprocessing.



Model: Sequence to Sequence LSTM Model:



The graph shown above demonstrates the Seq2Seq LSTM model:

This model is a unique class of recurrent neural network architecture

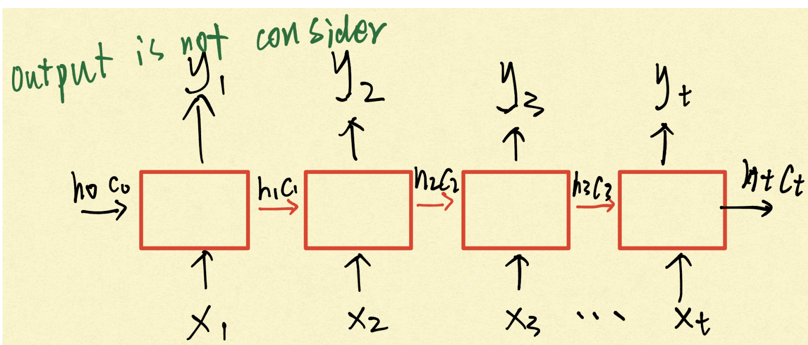
Encoder-Decoder Architecture:

Exactly what it says, it consists of two components (Encoder and Decoder):

Encoder:

- These two components can be LSTM or GRU models.
- After reading the input sequence, the encoder compiles the data into internal state vectors or context vectors (namely hidden state and cell state vectors regarding the LSTM model). We only keep the internal states after discarding the encoder's outputs. In order to aid the decoder in making precise predictions, this context vector aims to include the data for all input elements.

- $h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}X_t)$ this equation shown the process of operation of hidden states



The data are read by the LSTM one sequence at a time. We therefore say that LSTM reads the input in time steps of length 't' if the input is a sequence of that length.

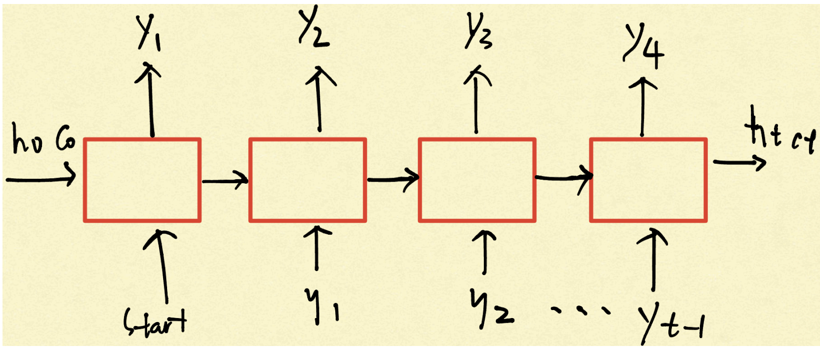
- X represents input sequence
- h represents the hidden state and c represents the cell state. The internal state of LSTM is the combination of hidden state and cell state
- Y is output sequence

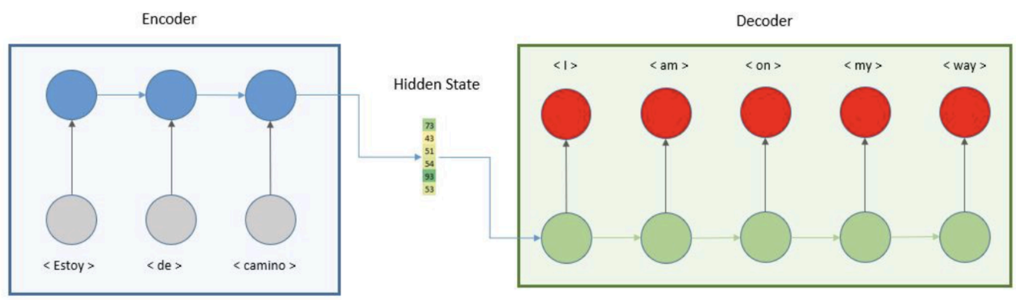
Decoder:

- The first stats of the decoder is the last hidden state h and cell state c from the encoder.
- A collection of LSTM units, each of which forecasts an output at a given time step (t), y t.
- Each recurrent unit receives a hidden state from the prior unit and generates both an output and a hidden state of its own.
- The hidden state h i is calculated with the following formula:

$h_t = f(W^{(hh)} h_{t-1})$

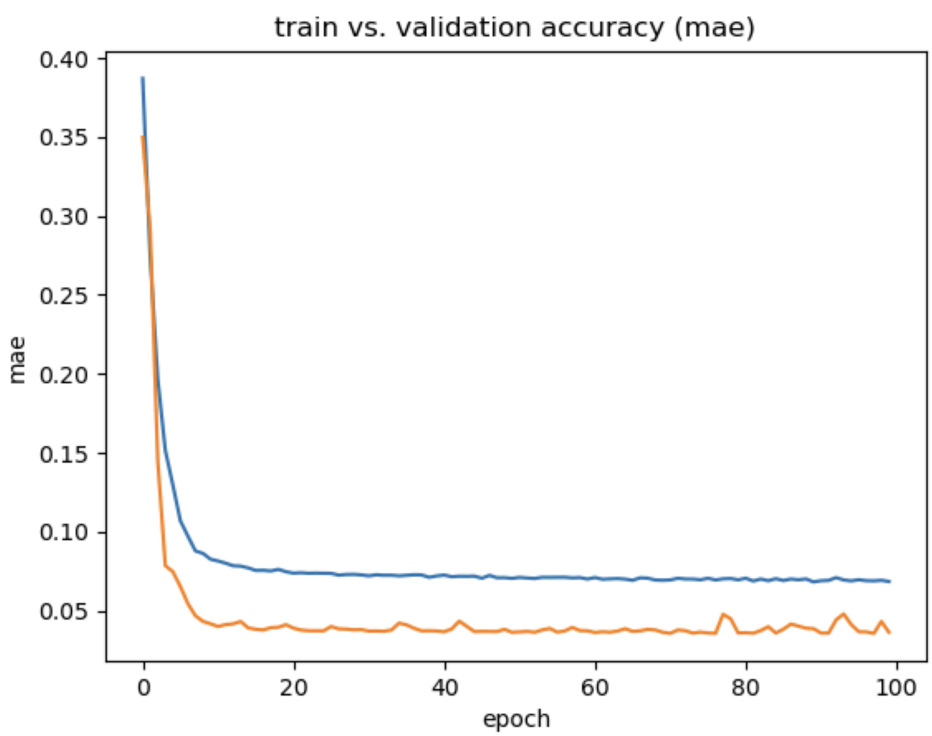
- The out y can be calculate using: $y_t = softmax(W^S h_t)$



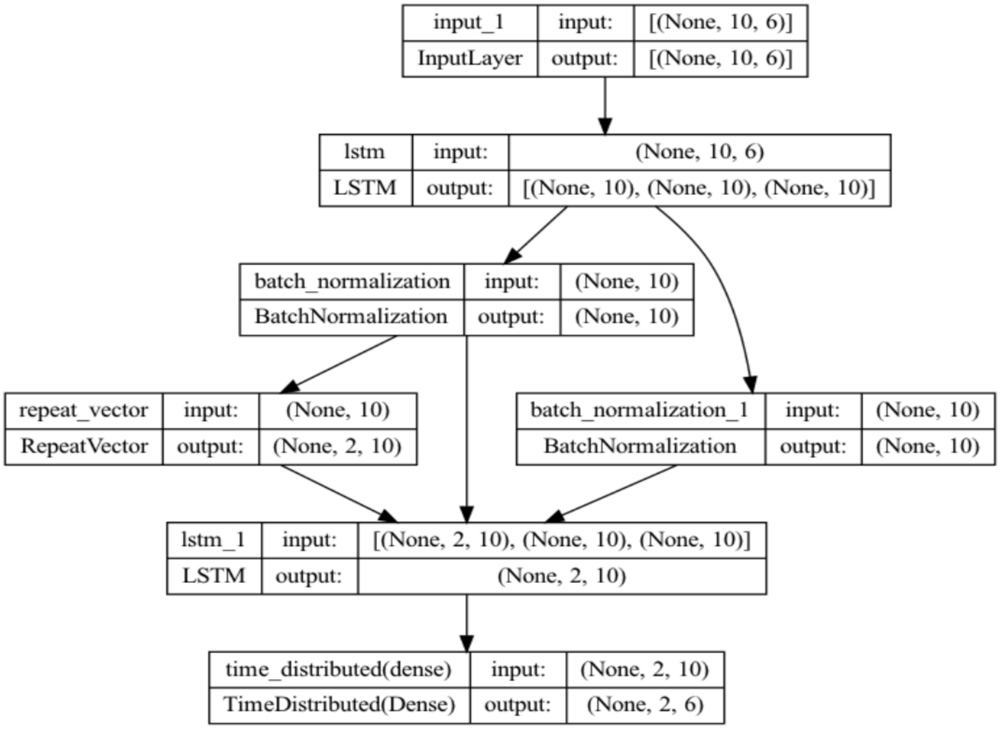


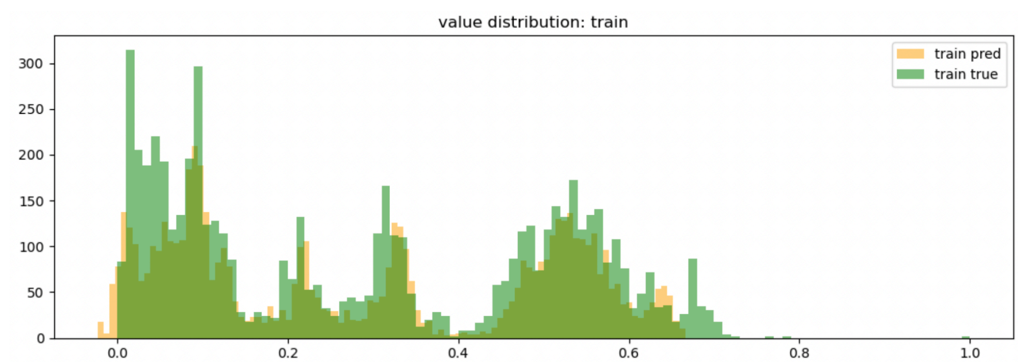
```
==========================================================================================
 input_1 (InputLayer)           [(None, 10, 6)]      0           []

 lstm (LSTM)                    [(None, 10),         680         ['input_1[0][0]']
                                 (None, 10),
                                 (None, 10)]

 batch_normalization (BatchNorm  (None, 10)          40          ['lstm[0][0]']
 alization)

 repeat_vector (RepeatVector)   (None, 2, 10)        0           ['batch_normalization[0][0]']

 batch_normalization_1 (BatchNo  (None, 10)          40          ['lstm[0][2]']
 rmalization)

 lstm_1 (LSTM)                  (None, 2, 10)        840         ['repeat_vector[0][0]',
                                                                  'batch_normalization[0][0]',
                                                                  'batch_normalization_1[0][0]']

 time_distributed (TimeDistribu  (None, 2, 6)        66          ['lstm_1[0][0]']
 ted)

==========================================================================================
Total params: 1,666
Trainable params: 1,626
Non-trainable params: 40
```

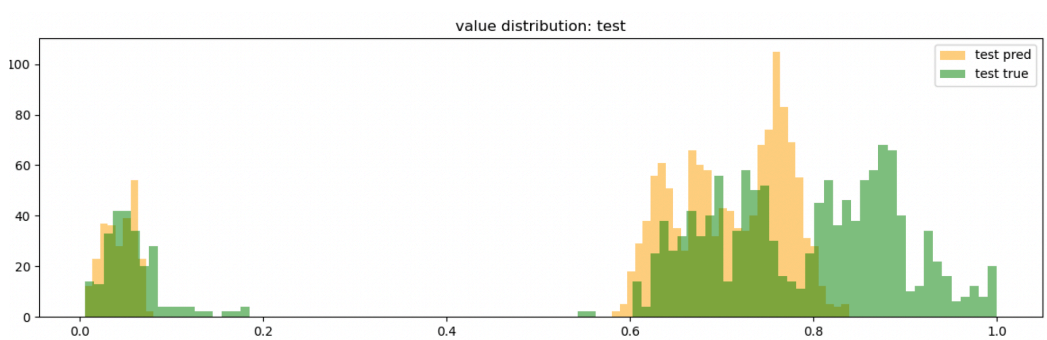
Loss function:

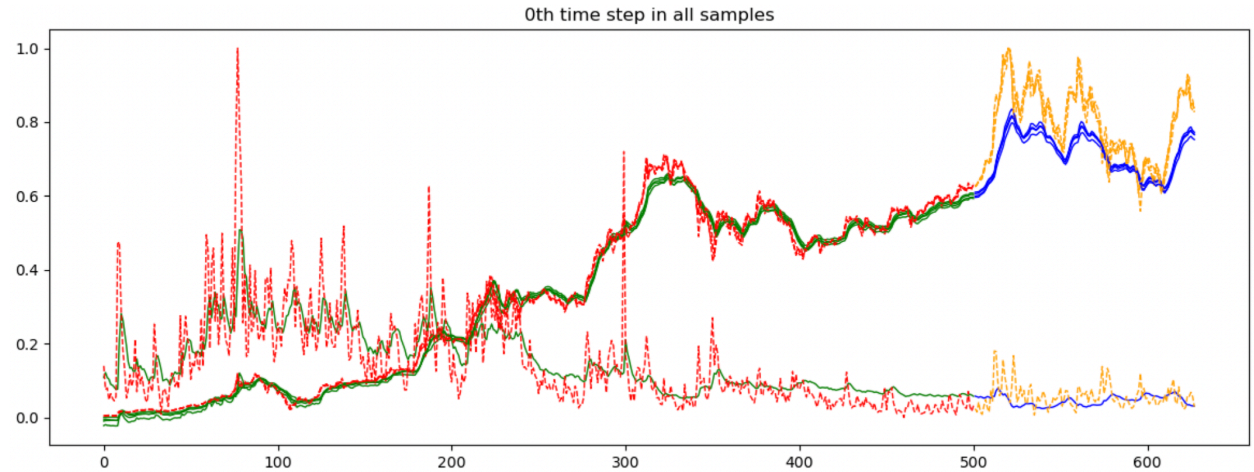


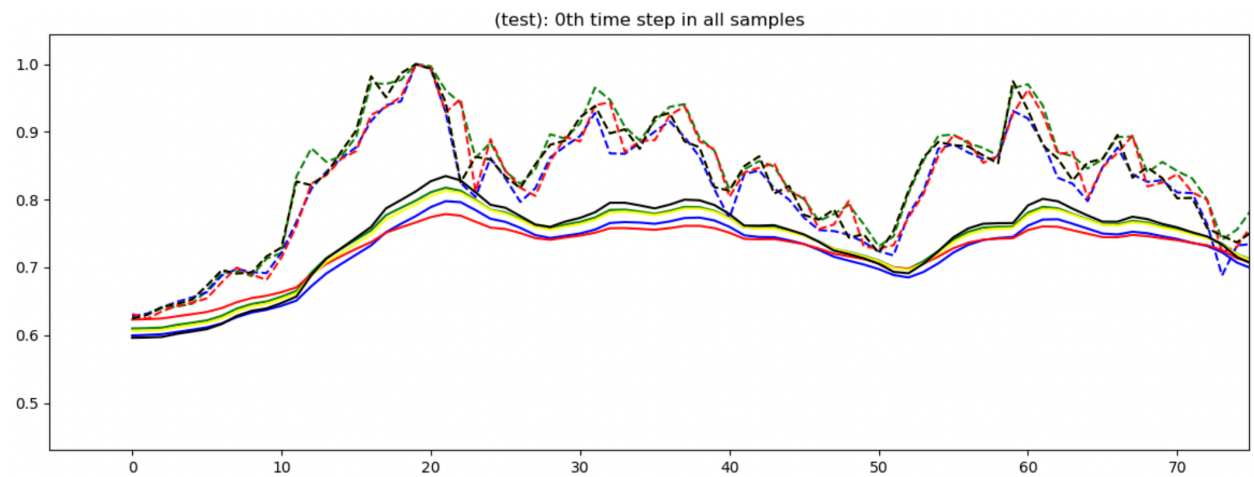
Implement model into our dataset:








0th time step in all samples
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
600


(test): 0th time step in all samples
1.0
0.9
0.8
0.7
0.6
0.5
0
10
20
30
40
50
60
70