COBRA on Sequential Models (LSTM, GRU, RNN, Transformer)

A Project Report Submitted for the Course

MA691

by

Rahul D (Roll No. 180102054)

Anish Kumar (Roll No. 180123003)

Bhargab Gautom (Roll No. 180123008)

Harsh Yadav (Roll No. 180123015)

Udandarao Sai Sandeep (Roll No. 180123063)

to the

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, INDIA

MA691 Fall 2021

DISCLAIMER

This work is for learning purpose only. The work cannot be used for publication or as commercial products etc without mentor's consent.

This has been done as course project under MA691 fall 2021.

Introduction

COBRA is a new method for combining several initial estimators of the regression function. In this project we try to apply the method on Sequential models to get better regression outputs. We have used 5 sequential models, a LSTM, a RNN, a GRU, a Transformer, and a custom model of GRU stacked over a LSTM as estimators. We are testing this model for predicting the Closing price of Bit-Coin.

Input Data

Our input data consists of bitcoin opening prices, closing prices, high prices and low prices on each day. For each day, we are also recording total volume, and market capitalization till each day. The time frame ranges from 29/4/13 till 6/7/21.

The starting 80% of the input data has been considered as training data. The next 10% has been reserved for validation, and the remaining 10% of the data has been used for testing.

Open, close, high, low price, volume, market capitalization were the columns of the initial data set. From the market capitalization till each day, we calculated the number of bitcoins available in the market. This was added as a new column in the data set, and market capitalization column was also dropped. Certain derived columns were further added to correlate between bitcoin price on the current day and the relative jump in the price from the previous day. This was done by adding the columns:

(High Price on day i) / (High Price on day i-1)

(Close Price on day i) / (Close Price on day i-1)

(Low Price on day i) / (Low Price on day i-1)

(Open Price on day i) / (Open Price on day i-1)

(Volume on day i) / (Volume on day i-1)

(Number of total bitcoins on day i)/ (Number of total bitcoins on day i-1)

To increase the stationarity of data, we have converted the bitcoin price and volumes into daily change rates. Thus, the learnings a model derives from our dataset have a higher validity for future predictions. The estimated data should have been the closing price, but instead in our model we would predicting the ratio of closing price of current day to the closing price of the previous day.

The initial input data is as follows:

	Date	High	Low	Open	Close	Volume	Marketcap
0	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.0	1.603769e+09
1	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.542813e+09
2	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.298955e+09
3	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.168517e+09
4	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.085995e+09

After adding the derived columns, we obtain the data as follows:

Date	High	Low	Open	Close	Volume	Numshares	High_ratio	Low_ratio	Open_ratio	Close_ratio	Volume_ratio	Numshares_ra
2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.109938e+07	0.996217	1.000373	1.071078	0.961672	0.808361	1.0003
2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.110313e+07	0.952086	0.803581	0.965278	0.841655	0.808361	1.0003
2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.110653e+07	0.897848	0.856683	0.837266	0.899308	0.808361	1.0003
2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.110992e+07	0.860892	0.857156	0.912958	0.929094	0.808361	1.0003
2013-05-04 23:59:59	115.000000	92.500000	98.099998	112.500000	0.0	1.111392e+07	1.063554	1.169406	0.923294	1.150895	0.808361	1.0003

Now, for training sequential model we have to set the input and the taget.

Input we chose is 25 days of data with all the aforementioned features, and the target is the column "Close_ratio", which is the relative jump of closing price of 26th day i.e closing price of 26th day / closing price of 25th day .

COBRA

We will be starting with an original data set D_n. It will be divided into two data sequences, D_k (starting k points) and D_l (remaining n-k points).

We have a set of M models which takes in a certain input, and gives an estimate r^*. These basic estimators – basic machines – are assumed to be generated using only the first subsample (D_k). Hence, each model predicts provides an estimation of $r^*(x)$ on the basis of D_k alone. Cobra provides a method to combine all the results of the model, and hence obtain a better estimate using the given models.

The goal of COBRA is to predict response for a new input point x. Using the threshold level as ε, we would create a ε-ball around the point (x_i,y_i), such that $|r_m(x_i) - r_m(x_i)| \leq \varepsilon$. For the corresponding y_i's, we would be taking the average (or some other function) for the prediction of x.

Let us consider the weighted average of the corresponding values of y_i. We define the collective estimator T_n to be:

$$T_n(\mathbf{r}_k(\mathbf{x})) = \sum_{i=1}^{\ell} W_{n,i}(\mathbf{x}) Y_i$$

where the random weights $W_{n,i}(x)$ take the form

$$W_{n,i}(\mathbf{x}) = \frac{\mathbf{1}_{\bigcap_{m=1}^{M}\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_i)| \leq \varepsilon_\ell\}}}{\sum_{j=1}^{\ell} \mathbf{1}_{\bigcap_{m=1}^{M}\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_j)| \leq \varepsilon_\ell\}}}$$

where ε_l has been chosen appropriately using a certain algorithm.

The restriction of the above ε_l -ball can be relaxed by imposing a fixed fraction $\alpha \in \{1/M, 2/M, \ldots, 1\}$ such that:

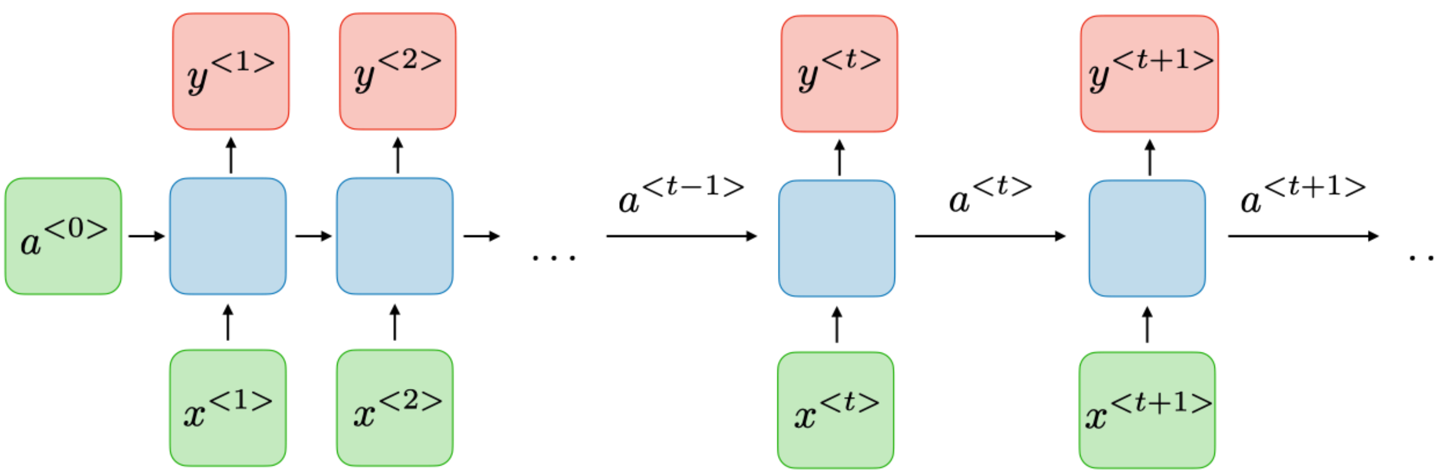
$$W_{n,i}(\mathbf{x}) = \frac{\mathbf{1}_{\{\sum_{m=1}^{M} \mathbf{1}_{\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_i)| \le \varepsilon_\ell\}} \ge M\alpha\}}}{\sum_{j=1}^{\ell} \mathbf{1}_{\{\sum_{m=1}^{M} \mathbf{1}_{\{|r_{k,m}(\mathbf{x}) - r_{k,m}(\mathbf{X}_j)| \le \varepsilon_\ell\}} \ge M\alpha\}}}$$

Asymptomatically speaking (α --> 1), the behaviour remains the same.

The 5 basic estimators used for the CoBRA is as follows:

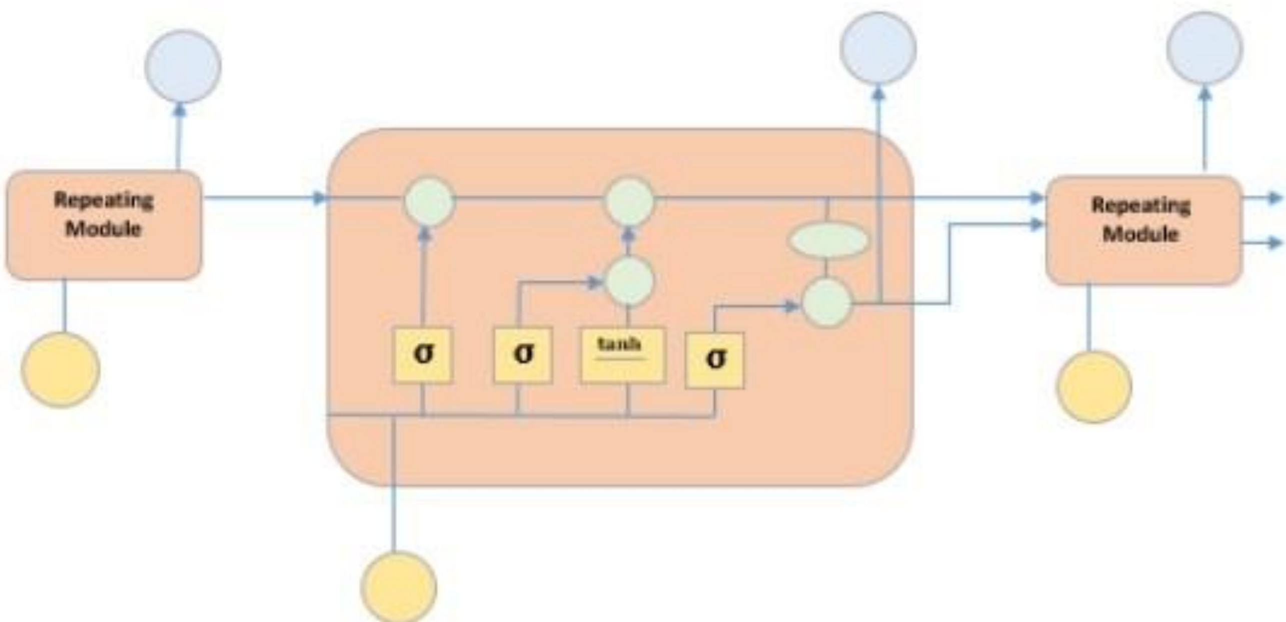
<u>1) Recurrent Neural Networks</u>

A neural network is a layered structure connected nodes. It is a combination of various algorithms which enables us to perform complex operations on data. RNN is a class of neural networks which performs very well on temporal data. It uses their reasoning from previous experiences to inform the upcoming events. It is widely used in Natural language processing and speech recognition.



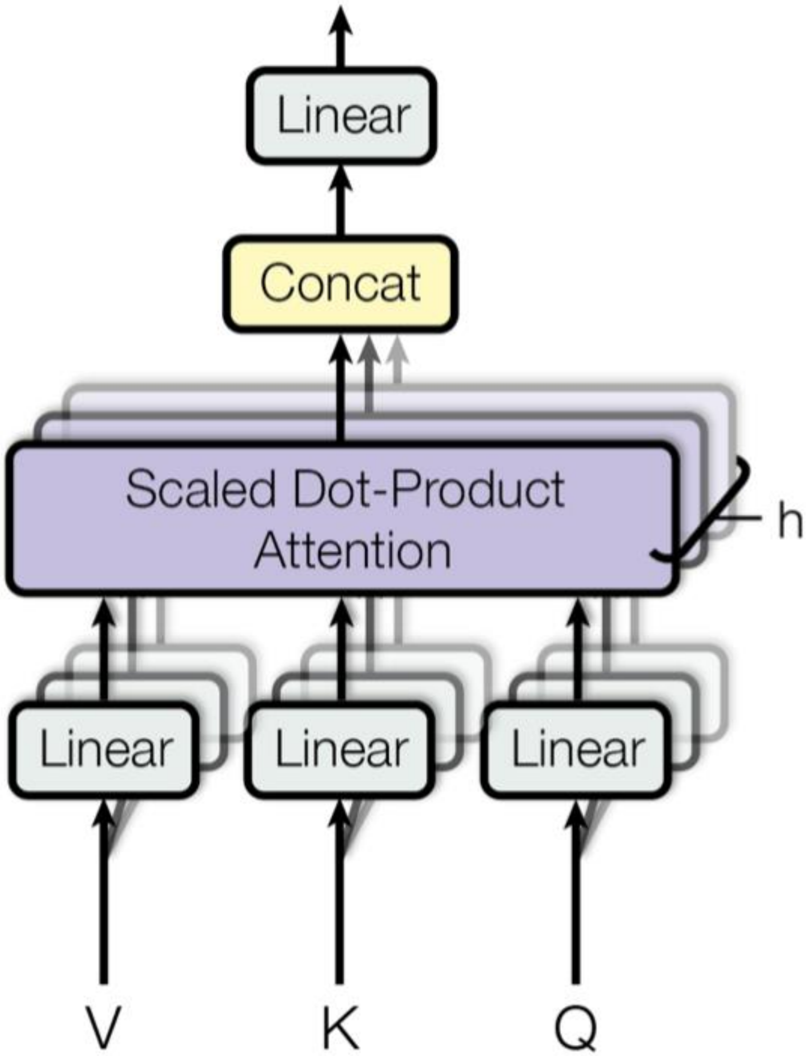
2) LSTM Model

Long Short-Term Memory model (LSTM) is a class of recurrent neural network. It is capable of learning long term dependencies in data. This can be done using 4 interactive layers as shown in the following figure.



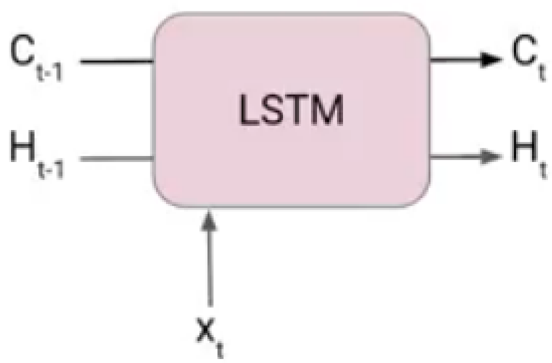
3) Transformer Model

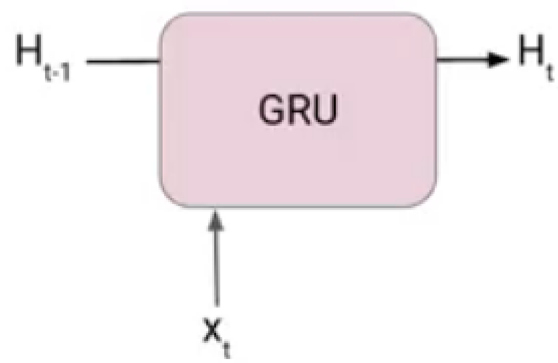
A transformer is a deep learning model that adopts the method of self-attention. Transformers, like RNNs, are designed to handle sequential data. The attention mechanism provides the requires context for each position in the input mechanism, hence eliminating the need for processing the data in order. The specific model is used in our case is Multi-head attention model. It runs through an attention mechanism several times parallelly. The independent outputs are then concatenated and linearly transformed in the output dimension.


Linear
Concat
Scaled Dot-Product
Attention
h
Linear
Linear
Linear
V
K
Q

4) GRU Model

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, which is improvised version of LSTM, since it has lesser number of model parameters, but does not compromise on the model performance. Like LSTM, GRU does not have a separate cell state. It has only one Hidden state. Hence, due to its simpler architecture, GRUs are much faster to train.



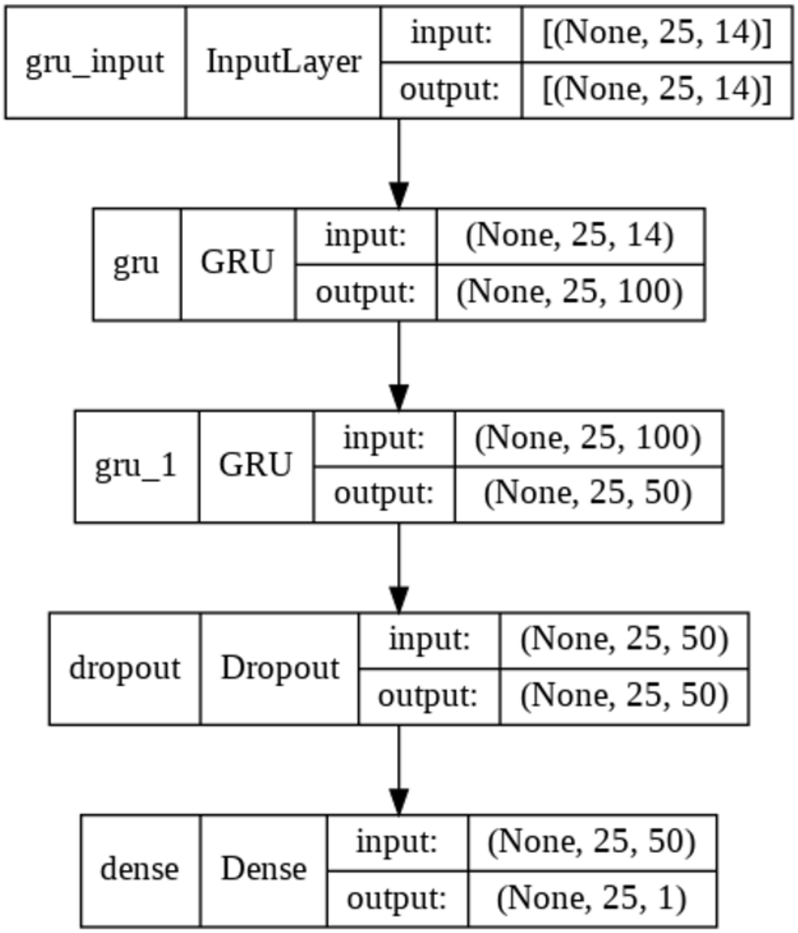


5) Hybrid Model

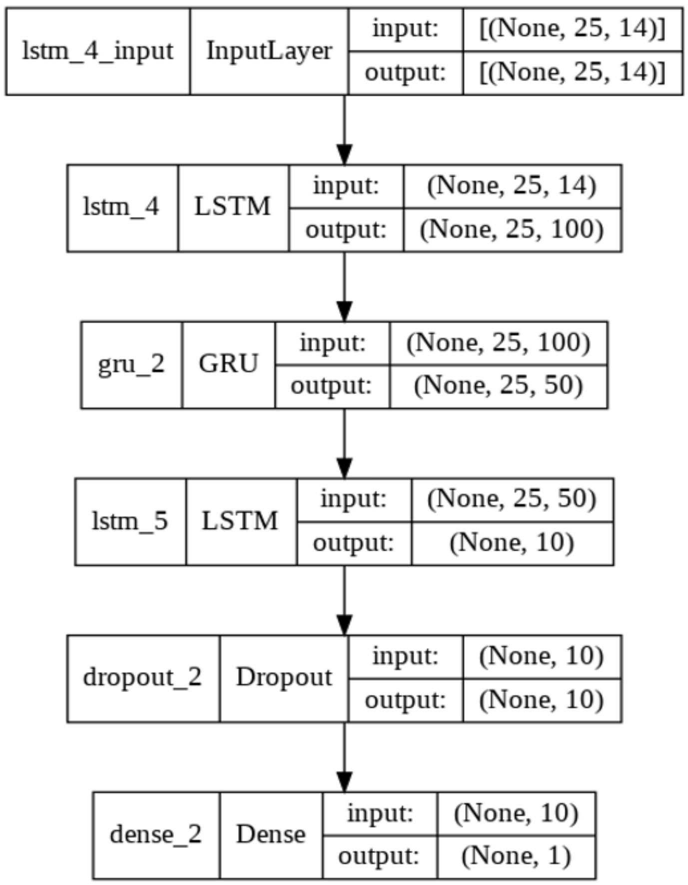
The 5th model used is the hybrid model, which is a combination of LSTM model and GRU Model stacked on each other.

Detailed view of each architecture used are as follows:

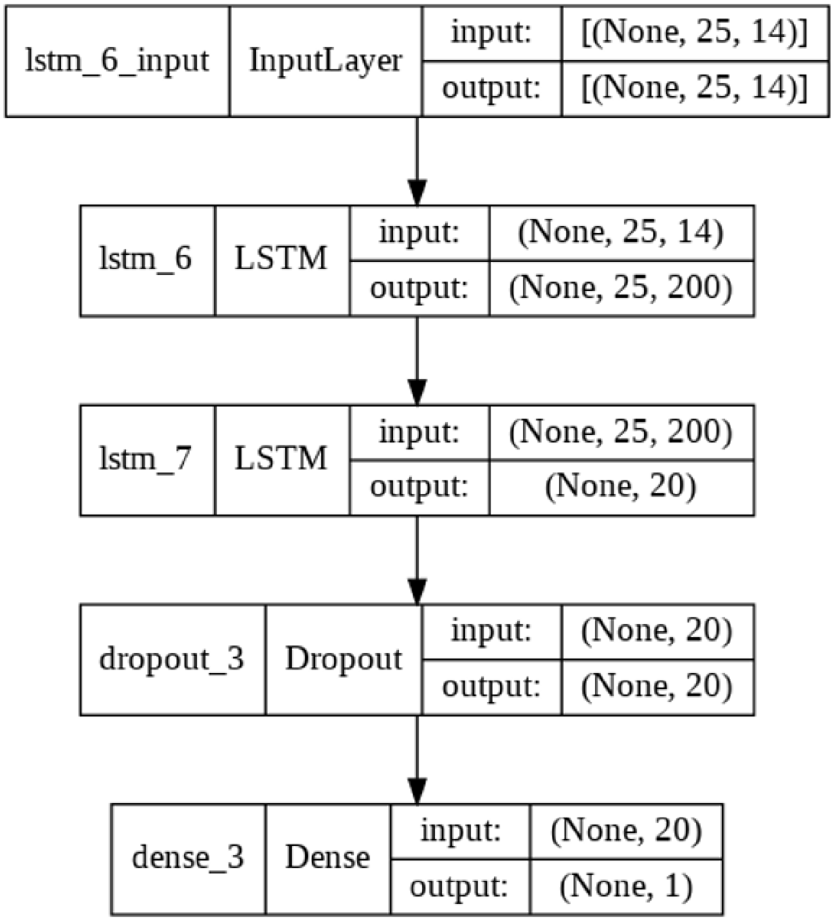
GRU



HYBRID

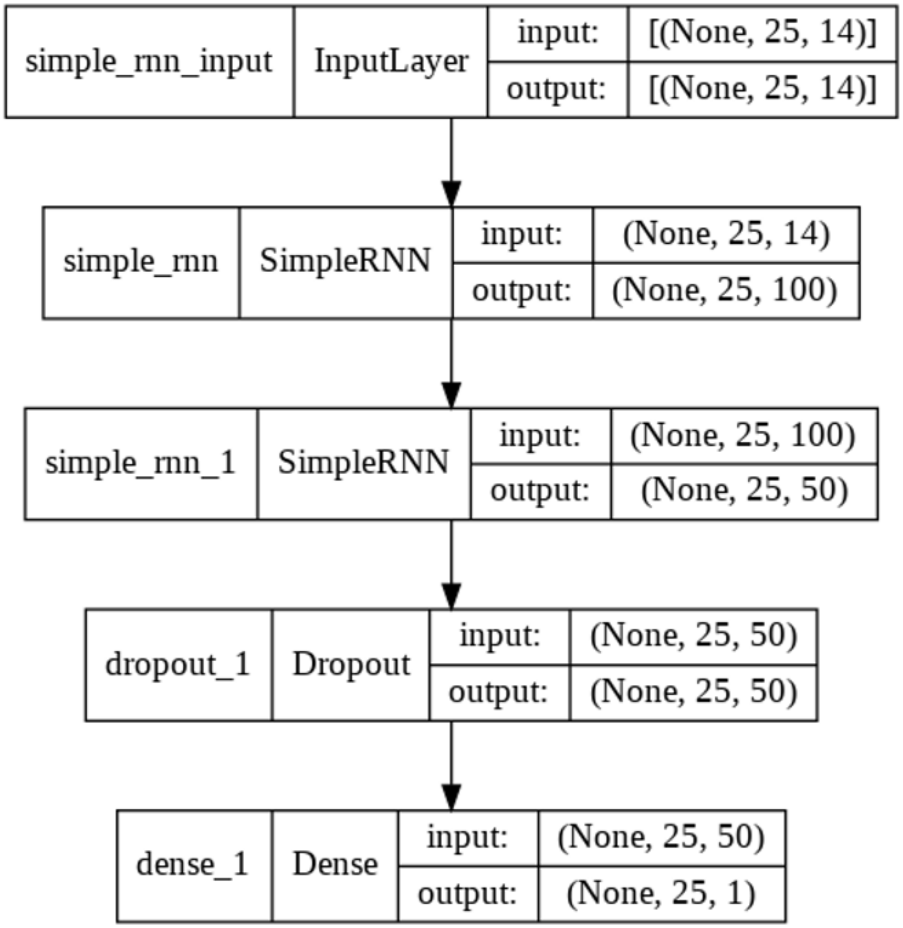

lstm_4_input
InputLayer
input:
[(None, 25, 14)]
output:
[(None, 25, 14)]
lstm_4
LSTM
input:
(None, 25, 14)
output:
(None, 25, 100)
gru_2
GRU
input:
(None, 25, 100)
output:
(None, 25, 50)
lstm_5
LSTM
input:
(None, 25, 50)
output:
(None, 10)
dropout_2
Dropout
input:
(None, 10)
output:
(None, 10)
dense_2
Dense
input:
(None, 10)
output:
(None, 1)

LSTM

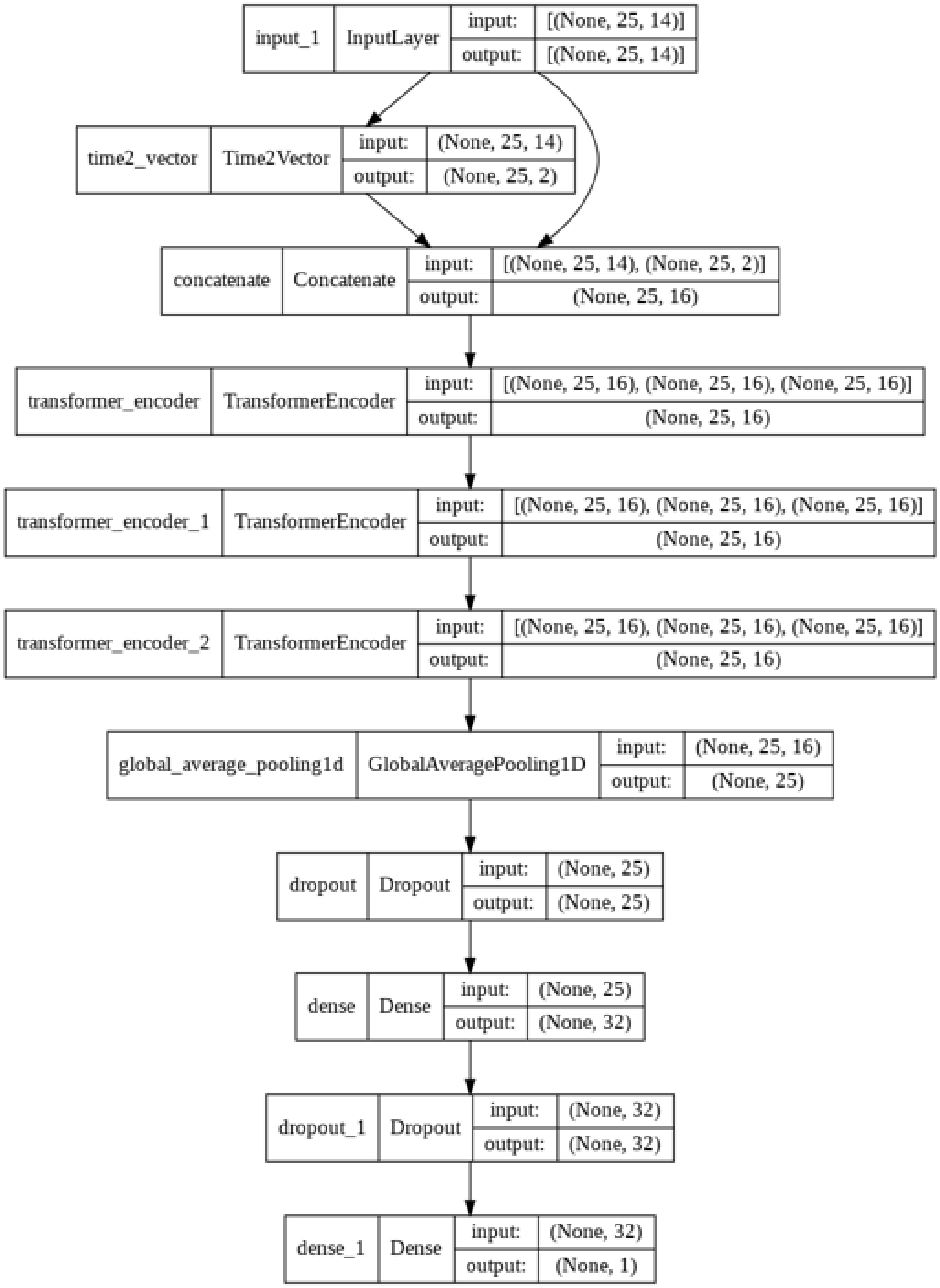


lstm_6_input
InputLayer
input:
[(None, 25, 14)]
output:
[(None, 25, 14)]
lstm_6
LSTM
input:
(None, 25, 14)
output:
(None, 25, 200)
lstm_7
LSTM
input:
(None, 25, 200)
output:
(None, 20)
dropout_3
Dropout
input:
(None, 20)
output:
(None, 20)
dense_3
Dense
input:
(None, 20)
output:
(None, 1)

RNN


simple_rnn_input
InputLayer
input:
[(None, 25, 14)]
output:
[(None, 25, 14)]
simple_rnn
SimpleRNN
input:
(None, 25, 14)
output:
(None, 25, 100)
simple_rnn_1
SimpleRNN
input:
(None, 25, 100)
output:
(None, 25, 50)
dropout_1
Dropout
input:
(None, 25, 50)
output:
(None, 25, 50)
dense_1
Dense
input:
(None, 25, 50)
output:
(None, 25, 1)

Transformers



Results

The results of our implementation of above-described models (i.e. LSTM, RNN, GRU, Transformer, Hybrid of LSTM & GRU and COBRA) are presented below. For each model the plots of prediction on Training, Validation and Testing data were obtained which are shown below:

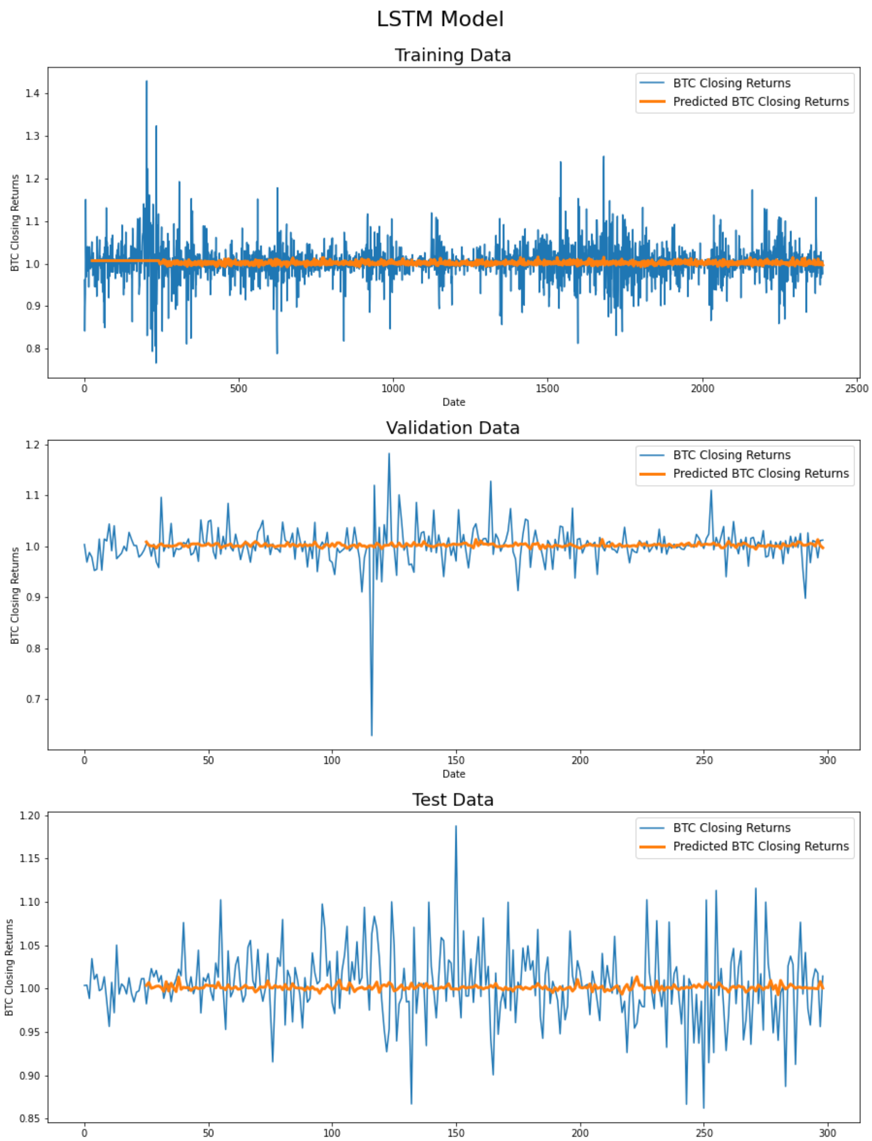
The metric used for evaluation are:

MSE -> Mean squared error

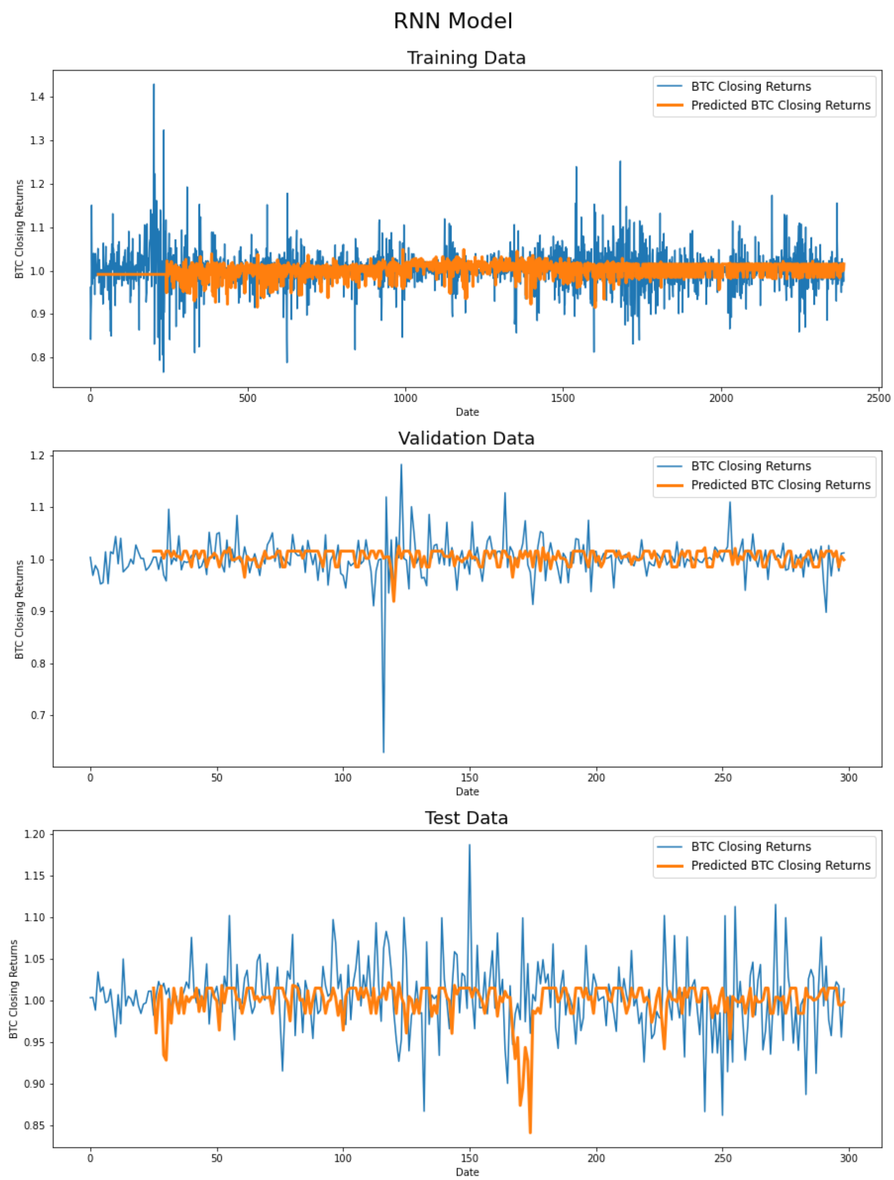
MAE -> Mean Absolute error

Models	Training Data		Validation Data		Test Data	
	MSE	MAE	MSE	MAE	MSE	MAE
LSTM	0.0018,	0.0269,	0.0016,	0.0236,	0.0020,	0.0324,
RNN	0.0024	0.0328	0.0019	0.0276	0.0025	0.0371
Transformer	0.0018,	0.0269	0.0016	0.0233	0.0019	0.0320
GRU	0.0025	0.0346	0.0017	0.0251	0.0044	0.0439
Hybrid	0.0028	0.0307	0.0018	0.0253	0.0022	0.0342
COBRA	0.0019	0.0279	0.0016	0.0233	0.0020	0.0321

- **LSTM Model**

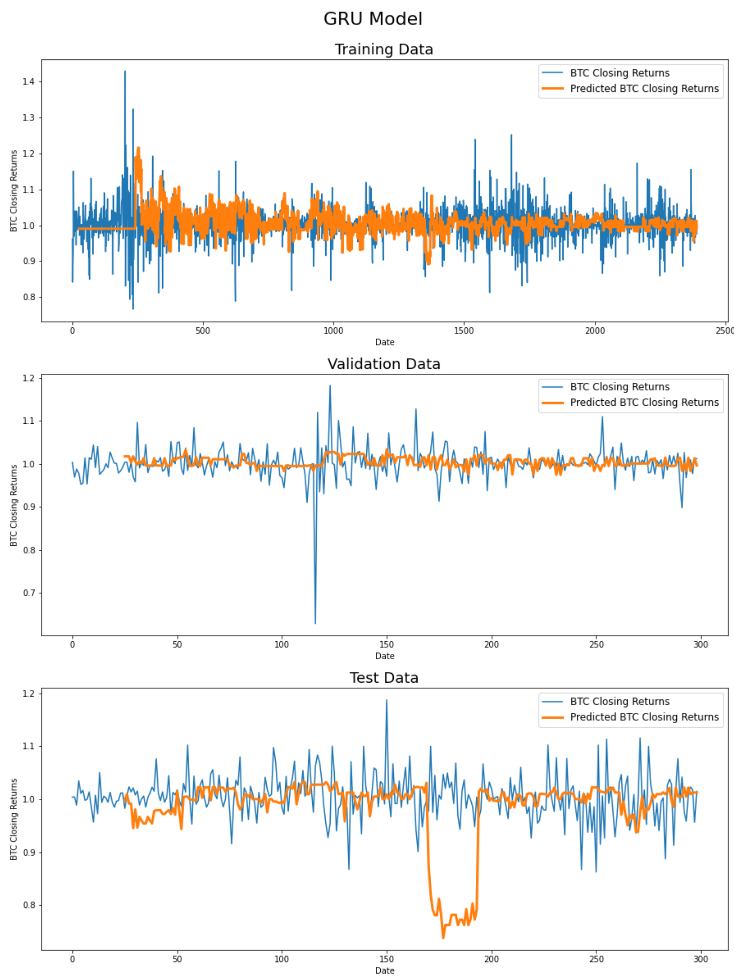

LSTM Model
Training Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Validation Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Test Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date

- **RNN Model**



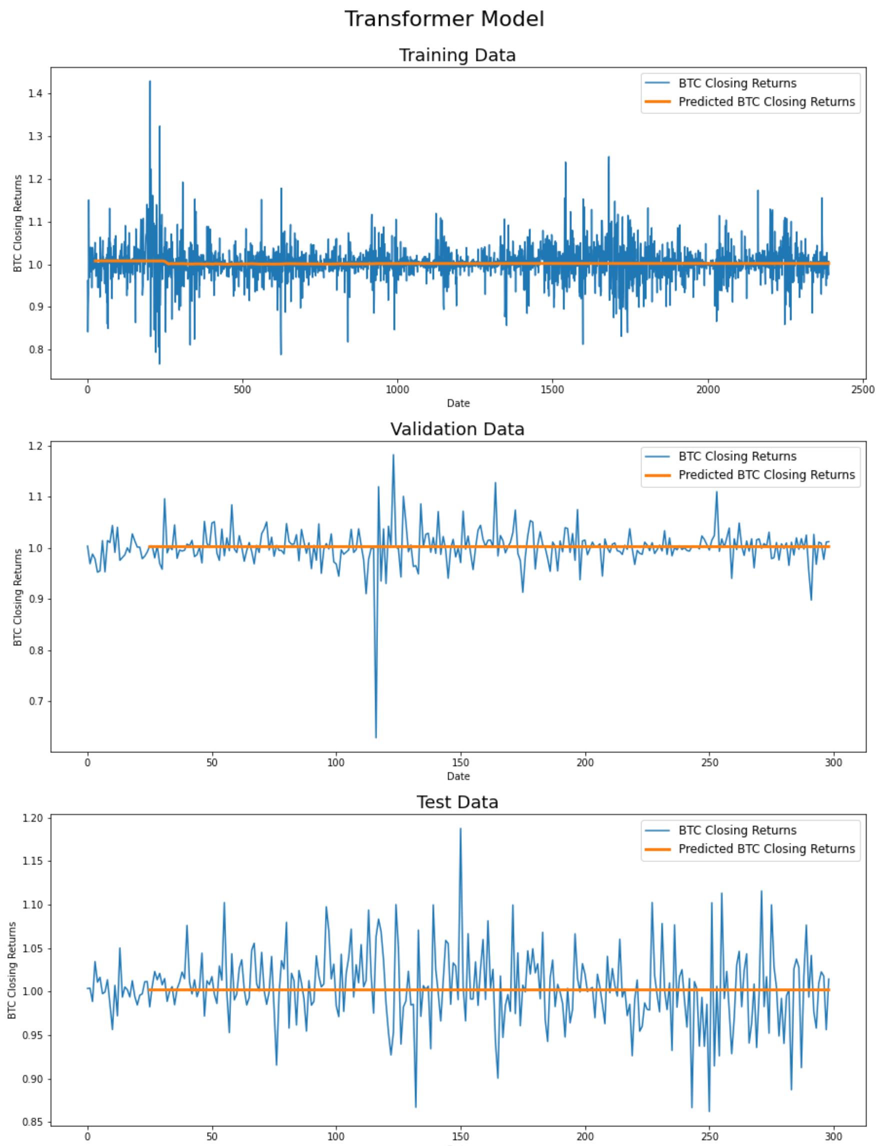
RNN Model
Training Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Validation Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Test Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date

- **GRU Model**

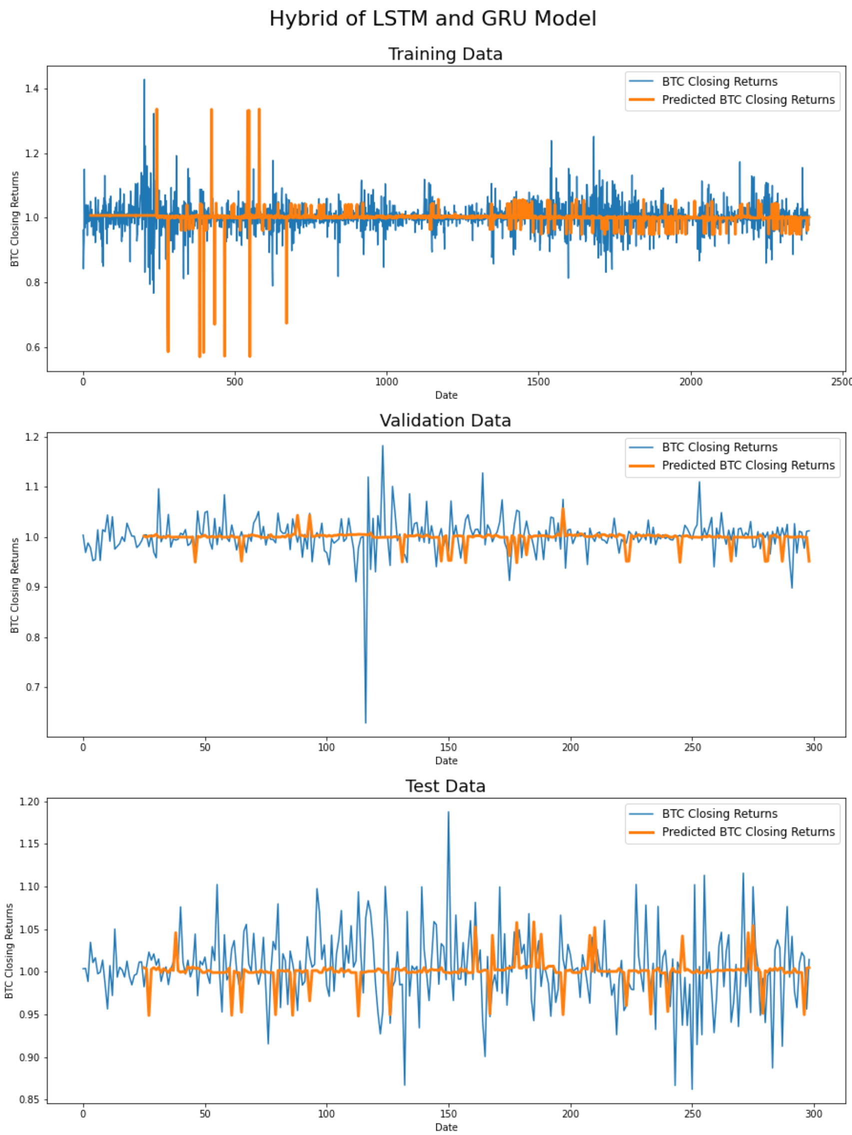


GRU Model
Training Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Validation Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date
Test Data
BTC Closing Returns
Predicted BTC Closing Returns
BTC Closing Returns
Date

- **Transformer Model**

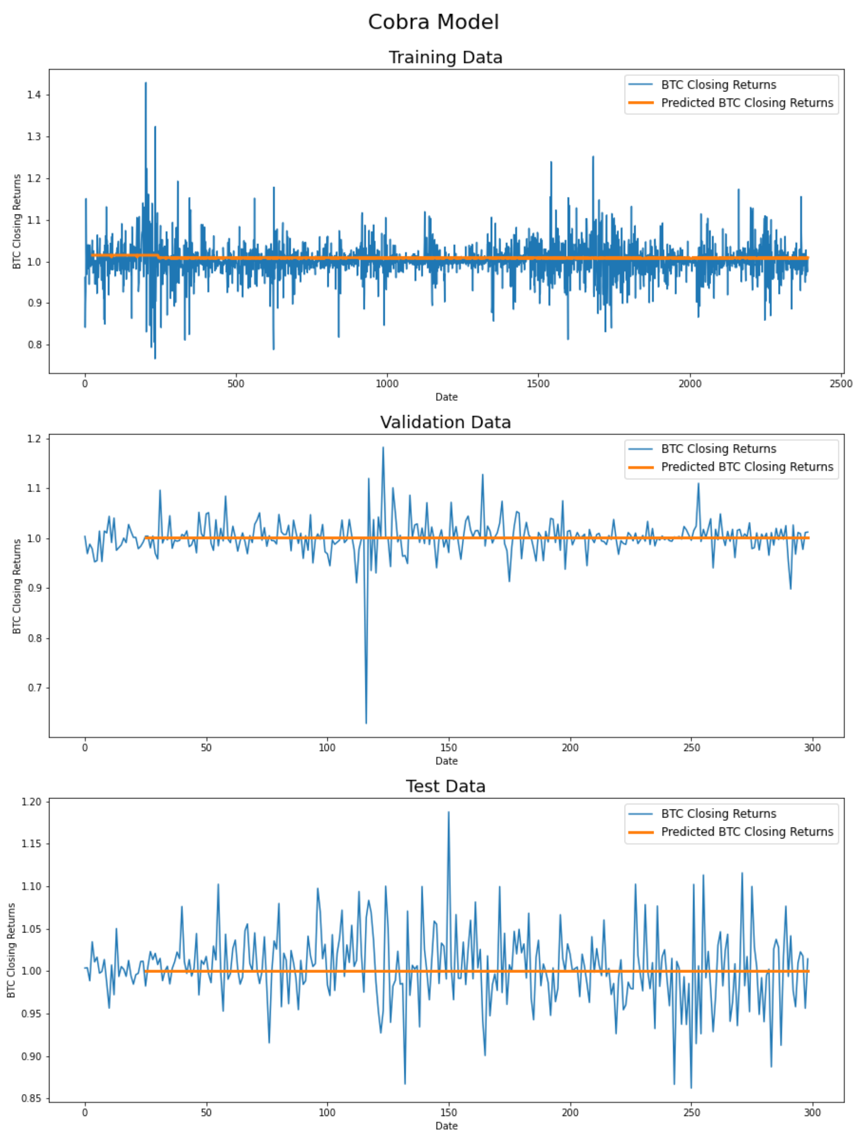


- **Hybrid of LSTM and GRU Model**



- **COBRA Model**

Using the predictions from all the aforementioned models and applying COBRA on them we obtained the following result:



Bibliography

[1] Biau, Gérard & Fischer, Aurélie & Guedj, Benjamin & Malley, James. (2015). COBRA: A combined regression strategy. Journal of Multivariate Analysis. 146. 10.1016/j.jmva.2015.04.007.

[2] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[3] Alfarraj, Motaz & Alregib, Ghassan. (2018). Petrophysical-property estimation from seismic data using recurrent neural networks. 2141-2146. 10.1190/segam2018-2995752.1.

[4] Sherstinsky, Alex. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena. 404. 132306. 10.1016/j.physd.2019.132306.

[5] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need.