# Covid-19 Data Forecasting using Neucube

1st Taher Yunus Lilywala
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20180609@goa.bits-pilani.ac.in

2nd Arun Ganti
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190021@goa.bits-pilani.ac.in

3rd Vishal Sharma
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190036@goa.bits-pilani.ac.in

4th Sri Kiran Nunna
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190067@goa.bits-pilani.ac.in

5th Atharva Parashuram Karandikar
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190083@goa.bits-pilani.ac.in

6th Shubham Rajnish Gandhi
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190086@goa.bits-pilani.ac.in

7th Karthik Uppalapati
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190089@goa.bits-pilani.ac.in

8th Dhyana Chidvilas Rottela
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190093@goa.bits-pilani.ac.in

9th Rajath V
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190122@goa.bits-pilani.ac.in

10th Sarthak Chaudhary
*Dept of Computer Science*
*BITS Goa*
Goa, India
f20190125@goa.bits-pilani.ac.in

*Abstract*—This report documents the process of sourcing, preparing, cleaning and formatting the covid-19 dataset so that it can be fed into Neucube for predicting the third wave of covid-19. In this report we have tried to predict the number of people infected with COVID-19 using neucube software. We tried to keep our model similar to the SUTRA model presented in [4] and have used the dataset from "covid19india.org". We have outlined in detail our method of data preparation for feeding it to neucube and the model that was used by us. At the end we present the results obtained after training the model and have compared our results with the SUTRA model.

*Index Terms*—covid, neucube, spiking, neural, network

## I. LITERATURE SURVEY

The modelling of epidemics is an old field and there are several models that try to predict the future scenario majorly based on the number of people infected. One of the first such models is the SIS models shown in figure 1 which takes into consideration the number of people susceptible to the infection and number of people infected by the virus at any particular time. This model is very simple as it only considers two classes of individuals and does not take into account the complexity which might be present in the real world.

An improvement on the SIS model was the SIR model shown in figure 2 which is one of the widely used models for modelling a pandemic. Here an extra R (Recovered) term has been introduced which takes care of the number of people
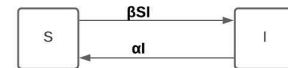


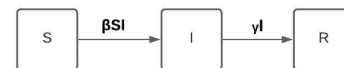Fig. 1. Compartment diagram for SIS model



Fig. 2. Compartmental diagram of SIR model

at any given time who have recovered from the infection. The equation for this model is given by:

$$\dot{S} = -\beta IS$$
$$\dot{I} = \beta IS - \gamma I$$
$$\dot{R} = \gamma I$$

While the SIR model is a good starting point, a more realistic model consists of an intermediate class called exposed(E) in between S and I. The individuals in the S group move to E after being exposed to the I, and move to I with some rate which is taken as a parameter. This model is called SEIR and is given by the following equation:
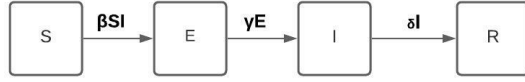
$$\dot{S} = -\beta IS,$$
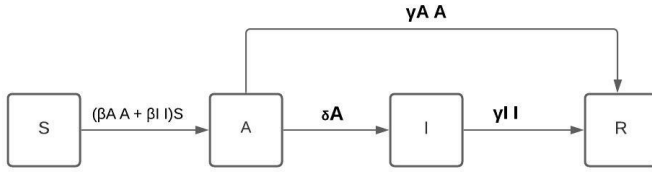
Fig. 3. Compartmental diagram of SEIR model



Fig. 4. Compartmental diagram of SAIR model

$$\dot{E} = \beta IS - \gamma E,$$
$$\dot{I} = \gamma E - \delta I,$$
$$\dot{R} = \delta I$$

For the particular case of COVID-19, we can consider asymptomatic(A) individuals to be a separate class for consideration which is the fundamental difference between SEIR model and the model that includes A known as SAIR model given by:

$$\dot{S} = -\beta_A AS - \beta_I IS,$$
$$\dot{A} = \beta_A AS + \beta_I IS - \gamma_A A\delta A,$$
$$\dot{I} = \delta A\gamma_I I$$
$$\dot{R} = \gamma_A A + \gamma_I I$$

Finally coming to the SUTRA model which we tried emulating. SUTRA stands for Susceptible, Undetected, Tested and Recovered Approach. In SUTRA individuals who are infected by the virus but aren't detected yet are classified as U, and T considers the individuals who have been identified to be infected and hence been isolated from others. The primary novelty in SUTRA was to estimate the changes in parameters and shifts in the active region using statistical analysis which helps in better modelling of real life scenarios. According to [4], SUTRA can make reliable predictions so long as the parameters related to it don't change. The primary equations used by SUTRA are as follows:

$$\dot{S} = -\beta SU,$$
$$\dot{U} = \beta SU - N_T - \gamma_U U,$$
$$\dot{T} = N_T - \gamma_T T,$$
$$\dot{R}_U = \gamma_U U,$$
$$\dot{R}_T = \gamma_T T,$$

In SUTRA model the input taken is S,T,R which are then used to estimate U and as a whole these values are used to make predictions about the number of infected. Some of the drawbacks of SUTRA model are:
This model uses too many parameters and can cause overfitting of the curve. It does not consider the behaviour of the virus on geographical diversity and social diversity. It does not take into account the new variants of covid-19 and their characteristics.

## II. DATA SOURCE

The data was sourced from COVID19 India [1], an open-source initiative maintained by dedicated volunteers, which collects data from state bulletins and other official sources [3], and aggregates it. The data used is documented state-wise with daily monitoring. The data had multiple formats including API endpoints, JSON files as well as CSV files. All of the files had multiple parameters that could be used for training, including daily count, cumulative count of cases, recovered, deceased, and change in cases per day for each state. All of these parameters were available in time-stamp-based data as well as raw data without time stamps. There were other data files that included the records of each test and each patient as well. The data was curated for the purpose of maintaining the website [1], and every aspect of the data was available as open-source.

The data we decided to consider included time-stamp based COVID statistics from January 2020 to Ovtober 31, 2021. The API was deprecated after the 31st of October 2021, so the data available is restricted till the end of October. We used the state-wise daily data in the model to train the spiking neural network

## III. DATA PREPARATION

The raw data obtained had all the state-wise occurrences of number of confirmed, recovered and deceased cases put in a single file. This raw data had to be processed into two different data formats.

### A. Data set 1

The data set 1 consists of just one csv file. This data set has been obtained by summing up all the respective attributes from all the state-wise data sets to represent the data country-wide.

```python
import pandas as pd
import numpy as np
df_tested = pd.read_csv('Confirmed.csv')
df_recovered = pd.read_csv('Recovered.csv')
df_deceased = pd.read_csv('Deceased.csv')

states = []
numOfCol = df_recovered.columns.size
for i in range(3, numOfCol):
  states.append(df_recovered.columns[i])

indexes = ['Recovered', 'Deceased',
    'Confirmed']
df_country = pd.DataFrame(0, index =
    np.arange(1, df_deceased.shape[0]),
    columns=indexes)

for state in states:
  df_country['Recovered'] +=
      df_recovered[state]
for state in states:
  df_country['Deceased'] +=
      df_deceased[state]
```

```
for state in states:
    df_country['Confirmed'] += df_tested[state]

df_country.to_csv('Country.csv', header =
    None, index = None)
```

## B. Data set 2

The data set 2 should consist of 29 state-wise csv files. The raw data has been split into three csv files (number of confirmed cases, recovered and deceased).

```
import os
import pandas as pd
cwd = os.path.abspath('')
files = os.listdir(cwd)
print(cwd)
print(files)
df = pd.DataFrame()
for file in files:
    if(file.endswith('.csv') &
        file.startswith('state_wise_daily')):
        df = df.append(pd.read_csv(file))

#print(df.head())

df1 = df.iloc[::3]
df2 = df.iloc[1::3]
df3 = df.iloc[2::3]

df3.to_csv('Deceased.csv')
df1.to_csv('Confirmed.csv')
df2.to_csv('Recovered.csv')
```

Using these three csv files, 29 state-wise csv files have been generated.

```
import os
import pandas as pd

df_tested = pd.read_csv('Confirmed.csv')
df_recovered = pd.read_csv('Recovered.csv')
df_deceased = pd.read_csv('Deceased.csv')

states = []
numOfCol = df_recovered.columns.size
for i in range(3, numOfCol):
  states.append(df_recovered.columns[i])

dataFrames = []

for i in range(len(states)):
  temp_df =
      df_tested[states[i]].rename('Tested')
  temp_df = temp_df.to_frame()
  temp_df =
      temp_df.join(df_recovered[states[i]].rename('Recovered'))
  temp_df =
      temp_df.join(df_deceased[states[i]].rename('Deceased'))
  dataFrames.append(temp_df)

for i in range(len(states)):
    dataFrames[i].reset_index(drop=True,
        inplace=True)
    dataFrames[i].to_csv('sam'+str(i+1)+'_'+states[i]+'.csv',
        header = None, index = None)
```
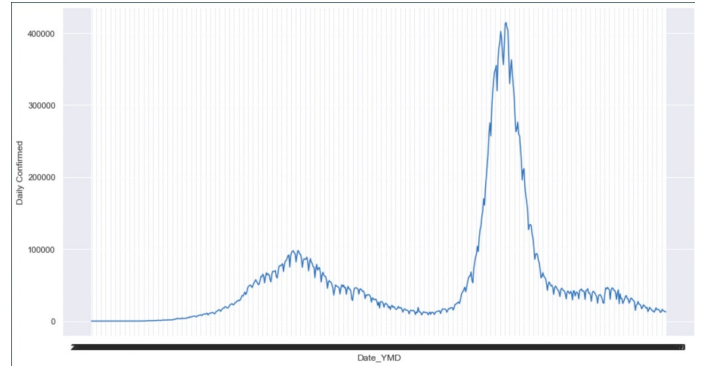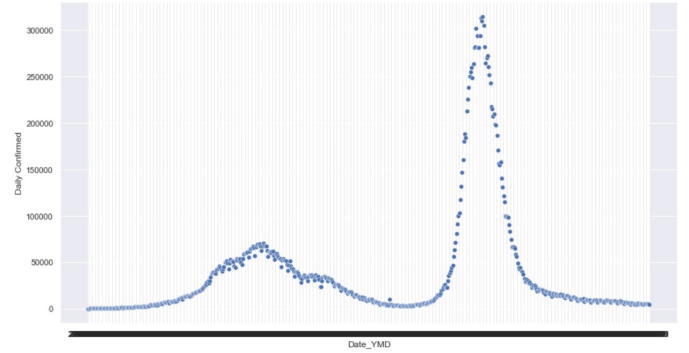


Fig. 5. Country plot



Fig. 6. Modified country plot

## IV. DATA ANALYSIS

The initial assumption based on the data sourced that Kerala and Maharashtra were outliers has been resolved by visualising the data after manipulating it. Daily Confirmed cases all over the country has been plotted against dates. Then the graph was plotted again after removing Kerala's and Maharashtra's contribution. [insert "picture modified country plot"]. These observations indicate that Kerala and Maharashtra were not outliers.

## V. THE MODEL AND RESULTS

Neucube is a windows-based software that makes use of the Brain Inspired Spiking Neural network architecture [2] to hep deal with temporal or spatio-spectro temporal data across domains. It uses SNNs and neuromorphic hardware.

It is used for data mining, pattern recognition and predictive data modeling, all of which involves complex and large data.

There are 5 broad states in which neucube works First the dataset has to be loaded, then the input data is encoded into spike trains Then the resevoir has to be initialized for which we used default parameters. The we train the reservoir using default parameters after which we train and verify the regressor.
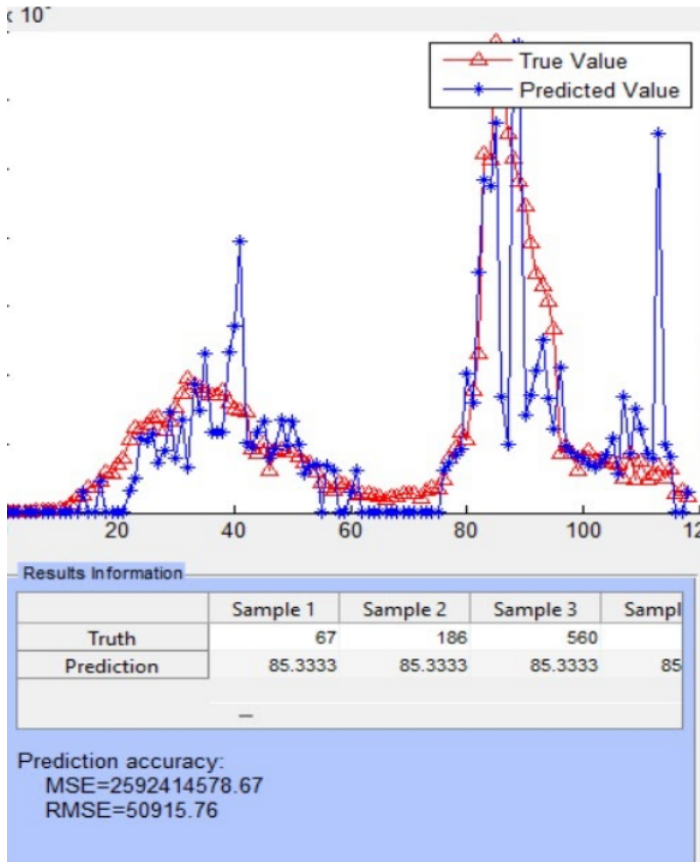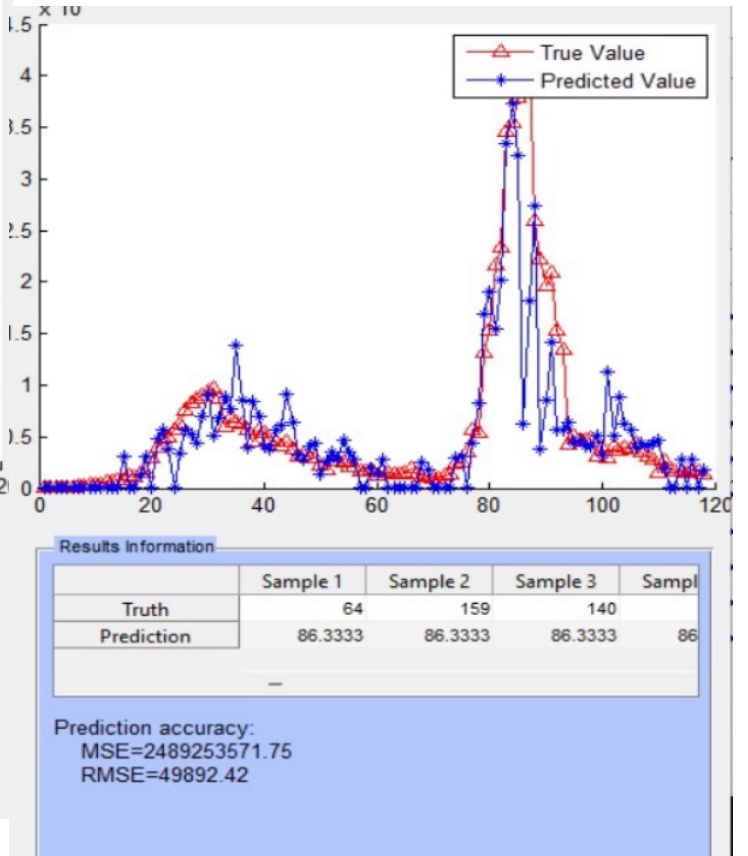
Fig. 7. mw 0.5 5

| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 67 | 186 | 560 | |
| Prediction | 85.3333 | 85.3333 | 85.3333 | 85 |

Prediction accuracy:
MSE=2592414578.67
RMSE=50915.76



Fig. 8. mw 0.5 0.8

| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 64 | 159 | 140 | |
| Prediction | 86.3333 | 86.3333 | 86.3333 | 86 |

Prediction accuracy:
MSE=2489253571.75
RMSE=49892.42

NeuCube offers 4 different spike encoding algorithms : (threshold representation, step forwarding, bens spiker algorithm, moving window). All 4 spiking algorithms have been tested. It was observed that BSA did not perform as good as the others.

We replicated the suutra[ref] model, we discovered that 7-day window did not perform well, so we switched to a 30 day window.

Model was also explored by training with different format of data ( normalized input-data and normalized labels, normalized input-data and unnormalized labels, unnormalized input-data and unnormalized labels). Training the model with unnormalized data resulted in high errors, whereas normalized data showed diminished errors. Normalized input data with normalized labels worked better over the other combinations.

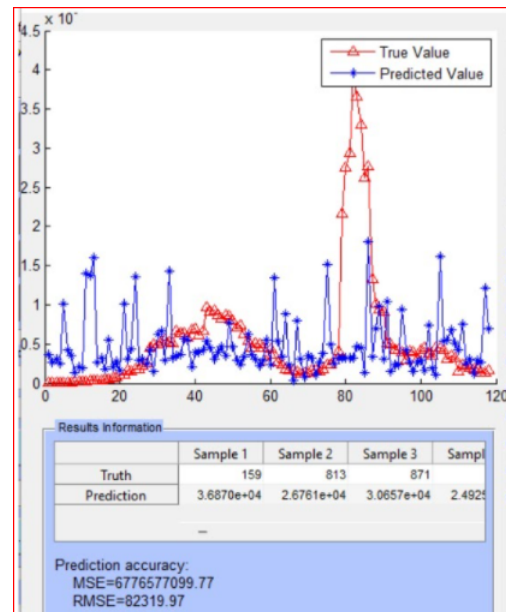For Bsa, default filter which is the ss filter has been used. The train test split was 0.8-0.2 for all tests.



| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 159 | 813 | 871 | |
| Prediction | 3.6870e+04 | 2.6761e+04 | 3.0657e+04 | 2.4929 |

Prediction accuracy:
MSE=6776577099.77
RMSE=82319.97

Fig. 9. bsa 0.5

Default Training parameters:

**Training Parameters**

| | | | |
|---|---|---|---|
| Potential Leak Rate | 0.002 | STDP Rate | 0.01 |
| Firing Threshold | 0.5 | Training Iteration | 1 |
| Refractory Time | 6 | LDC Probability | 0 |

OK      Cancel

Default classifier parameters:

**Classifier Parameters**

| | |
|---|---|
| Classifier | deSNNs |
| Mod | 0.8 |
| Drift | 0.005 |
| K | 3 |
| Sigma | 1 |

Fig. 10. Model



**Results Information**

| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 0.0026 | 0.0022 | 0.0030 | 0 |
| Prediction | 0.0026 | 0.0026 | 0.0026 | 0 |

Prediction accuracy:
MSE=0.00
RMSE=0.05

Fig. 12. Moving window 0.5 5



**Results Information**

| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 0.0022 | 0.0031 | 0.0034 | 0 |
| Prediction | 0.0024 | 0.0024 | 0.0024 | 0 |

Prediction accuracy:
MSE=0.00
RMSE=0.04
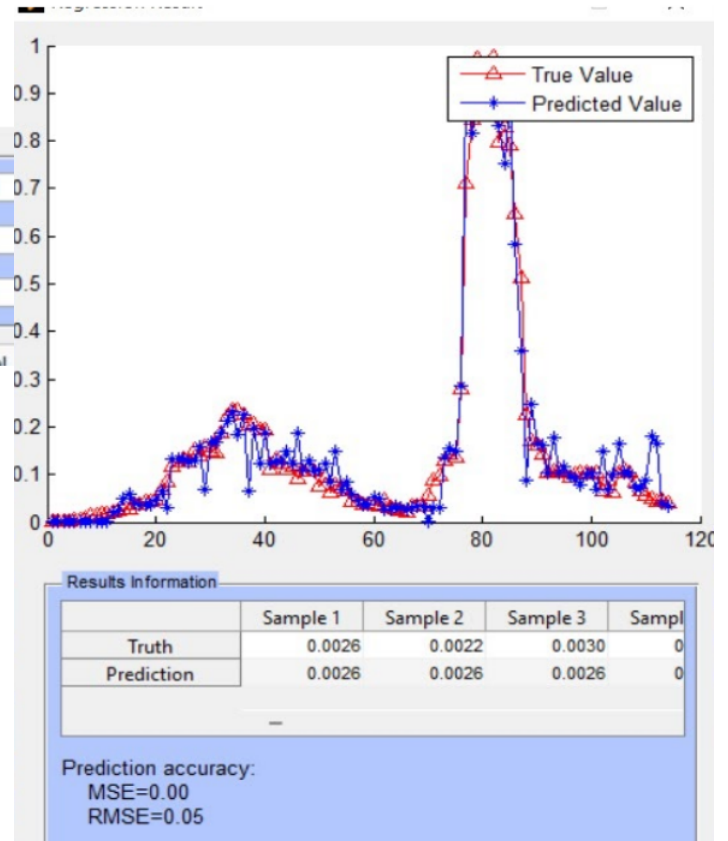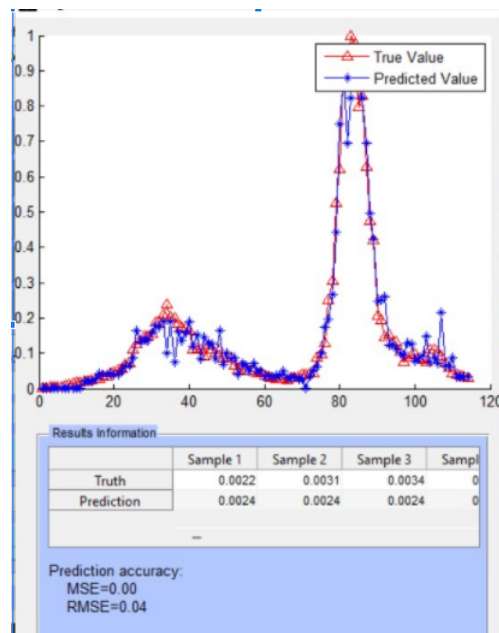
Fig. 11. Self forwarding (spiking threshold = 0.6)



**Results Information**

| | Sample 1 | Sample 2 | Sample 3 | Sampl |
|---|---|---|---|---|
| Truth | 0.0021 | 0.0033 | 0.0034 | 0 |
| Prediction | 0.3145 | 0.0054 | 0.0409 | 0 |

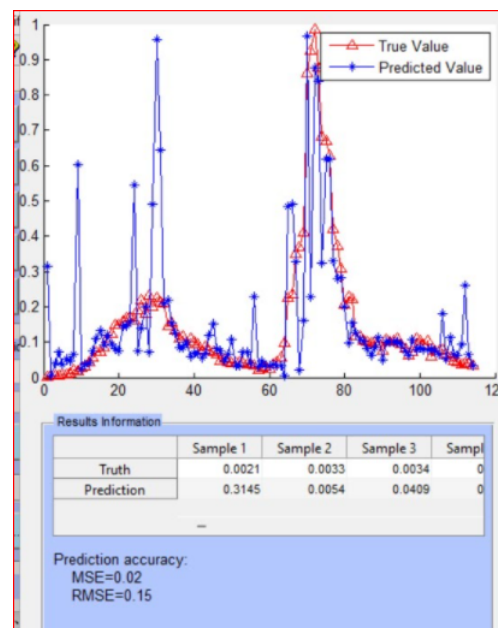Prediction accuracy:
MSE=0.02
RMSE=0.15
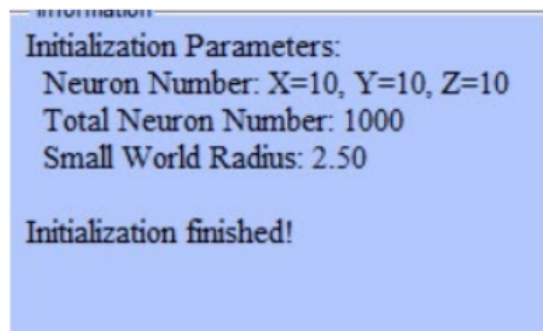
Fig. 13. bsa (spiking threshold : 0.5, filter : ss )

Fig. 14. Initial parameters
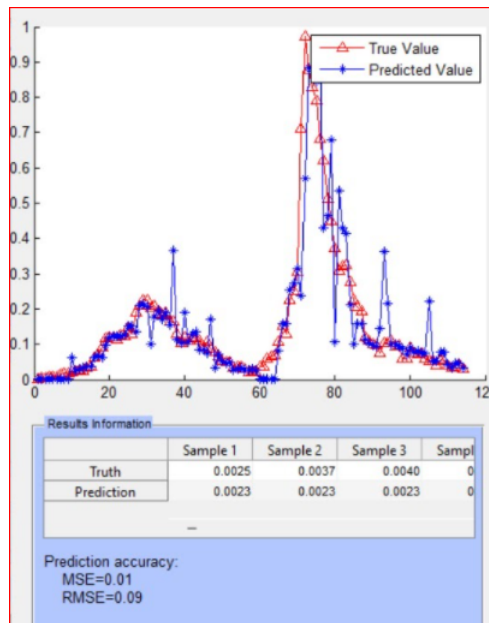


Fig. 15. Thresholding representation:(spiking threshold: 0.5)

REFERENCES

[1] *Coronavirus in India: Latest Map and Case Count*. URL: https://www.covid19india.org/.

[2] Nikola K. Kasabov. *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer, 2019.

[3] StriKe. *Covid-19 Sources*. June 2020. URL: https://telegra.ph/Covid-19-Sources-03-19.

[4] *SUTRA: A Novel Approach to Modelling Pandemics with Applications to COVID-19*. URL: https://arxiv.org/pdf/2101.09158.pdf.
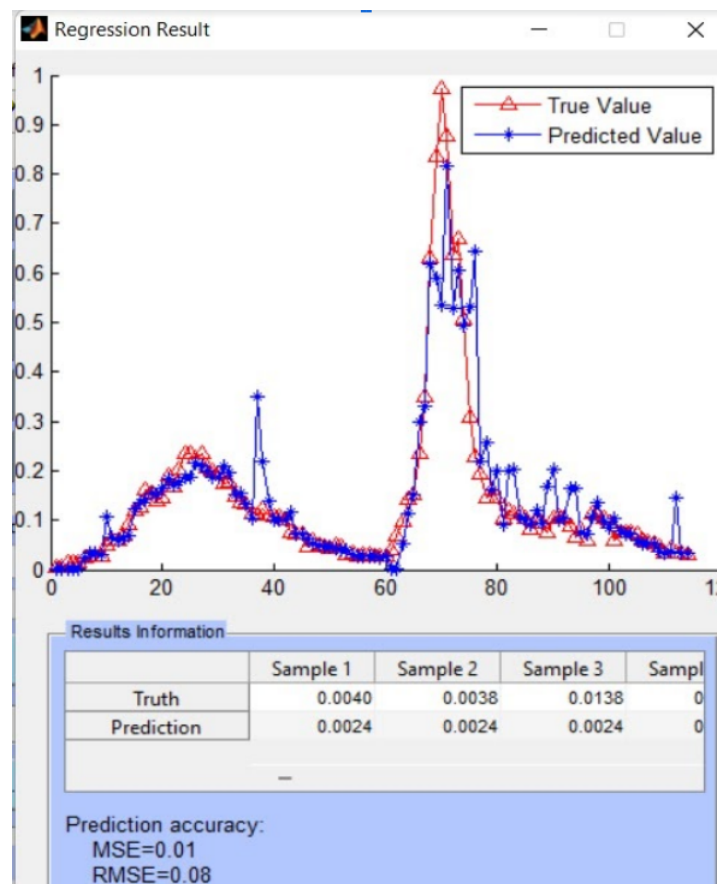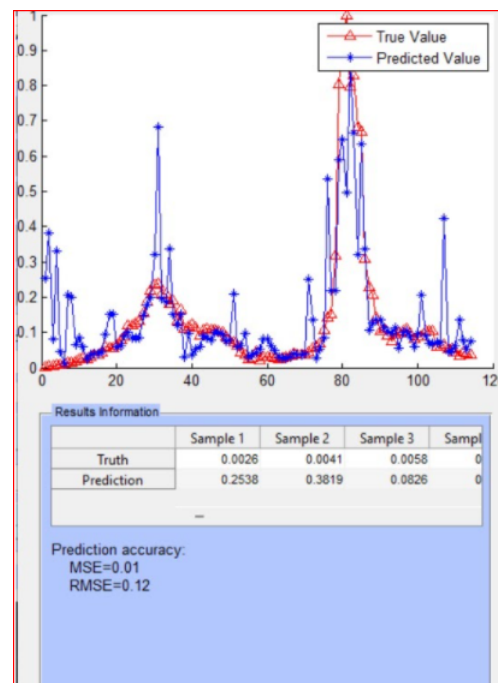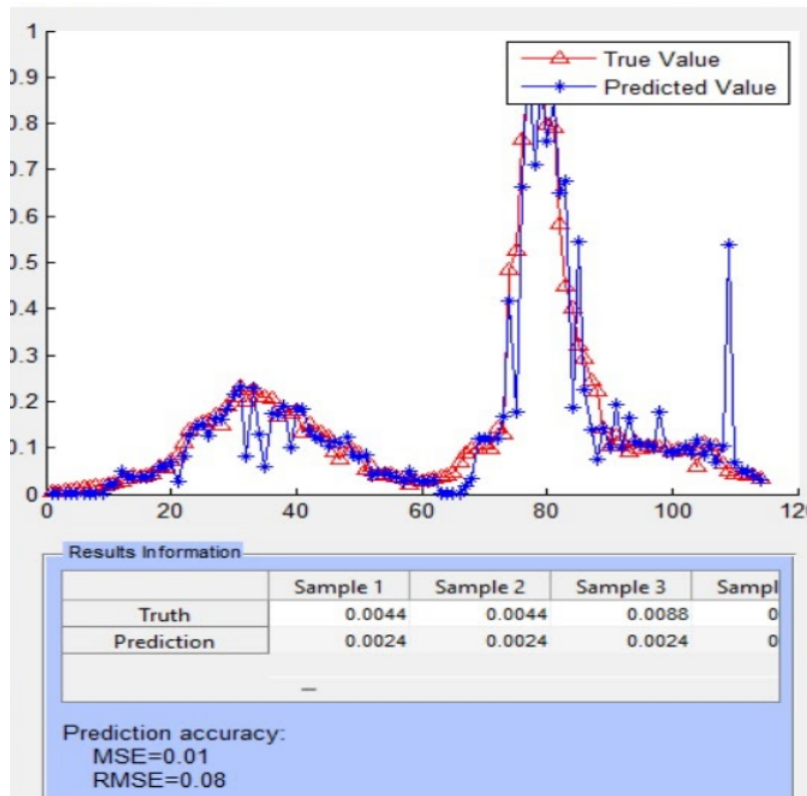
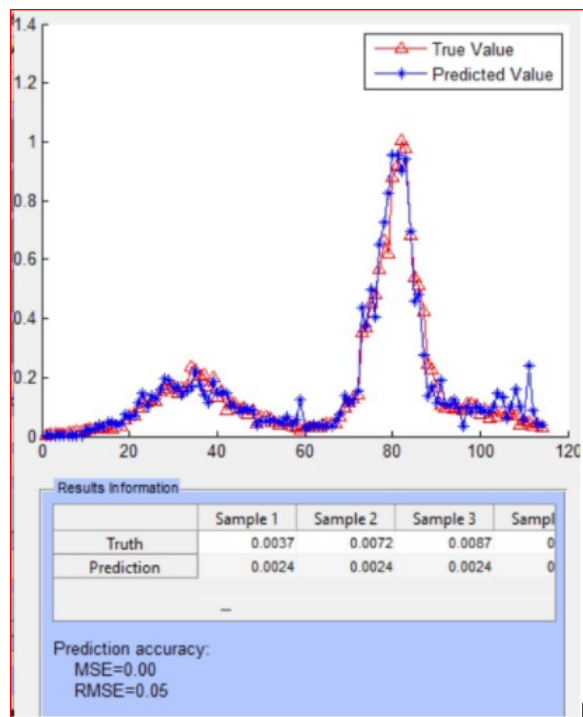Fig. 16. tr 0.5



Fig. 17. bsa 0.5 ss filter

Fig. 18. mw threshold



Fig. 19. fs 0.5