<u>Susceptible - Exposed - Infectious – Recovered – Died (SEIRD) model</u>

Target:

a. Prediction of the COVID-19 cases per day at a hyperlocal.
b. Predict the peak for the places at hyperlocal level for COVID-19.

Introduction of the proposed Model:

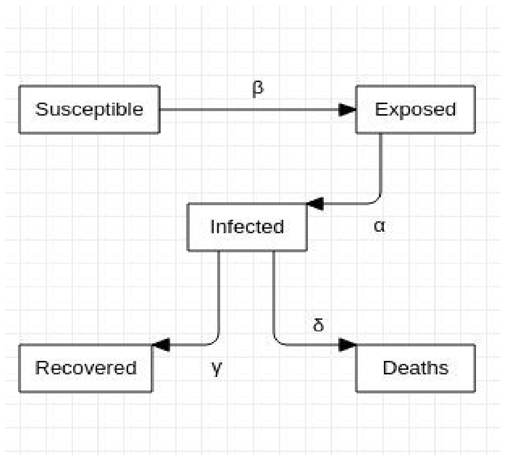
The S-E-I-R-D is a mathematical model with some assumptions about the dependence of each of the factors on the other factors such as:

a. Alfa → onset rate = 0.2
b. Gamma → Removal Rate = 0.1
c. M → Mortality Rate = 0.029 (depends upon the area)
d. N → epidemic size (This value is calculated by the optimizing algorithm, but initial guess is 7.45842445e+03)
e. P → identification rate (assumed 0.09)
f. Beta → Infection rate (calculated by the optimizing algorithm, but the initial guess is 0.01)

The values S (Susceptibility), E (Exposed), I (Infected), R (recovered), D (deaths) are depend upon these multiplication factors as well some ordinary differential equations as shown:

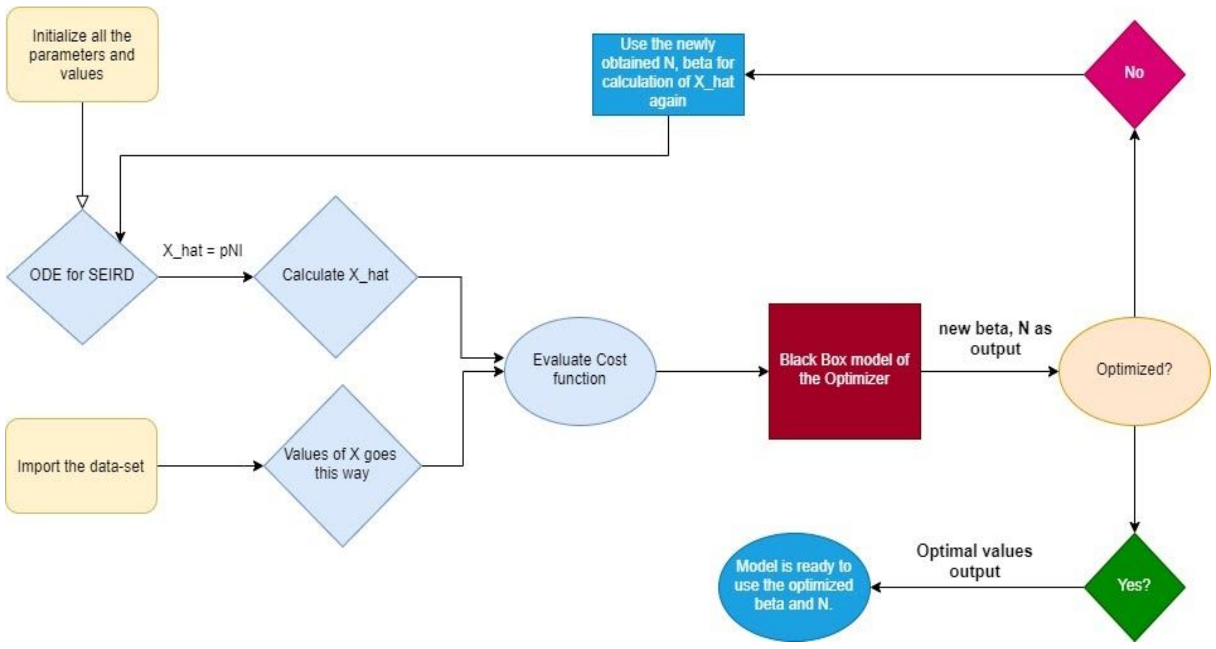
$$\begin{gathered}
dS(t)/d(t) = -\beta S(t)I(t), \\
dE(t)/d(t) = \beta S(t)I(t) - \alpha E(t), \\
dI(t)/d(t) = \alpha E(t) - \gamma I(t) - MI(t), \\
dR(t)/d(t) = \gamma I(t), \\
dD(t)/d(t) = MI(t)
\end{gathered}$$

<u>Fig1: ODE for different processes Of SEIRD model</u>



<u>Fig2: Interpretation for different processes Of SEIRD model</u>

Workflow of the SEIRD Model


Initialize all the parameters and values
Use the newly obtained N, beta for calculation of X_hat again
No
ODE for SEIRD
X_hat = pNI
Calculate X_hat
Evaluate Cost function
Black Box model of the Optimizer
new beta, N as output
Optimized?
Import the data-set
Values of X goes this way
Model is ready to use the optimized beta and N.
Optimal values output
Yes?

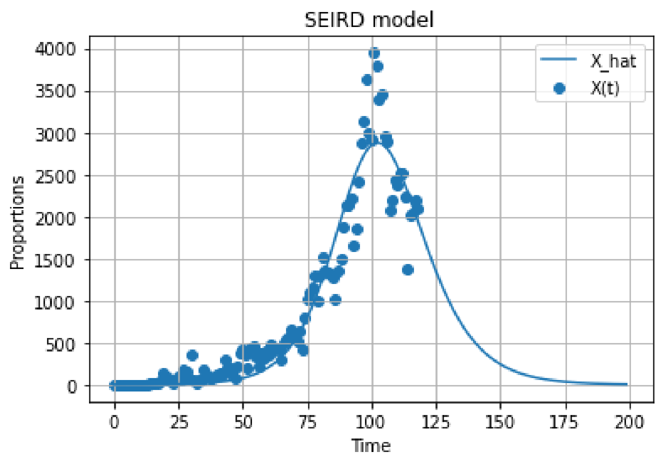
Here, the **Black Box model of the optimizer used is:**

[Broyden–Fletcher–Goldfarb–Shanno algorithm](#) **(BFGS algorithm)** using the python library [SCIPY](#).

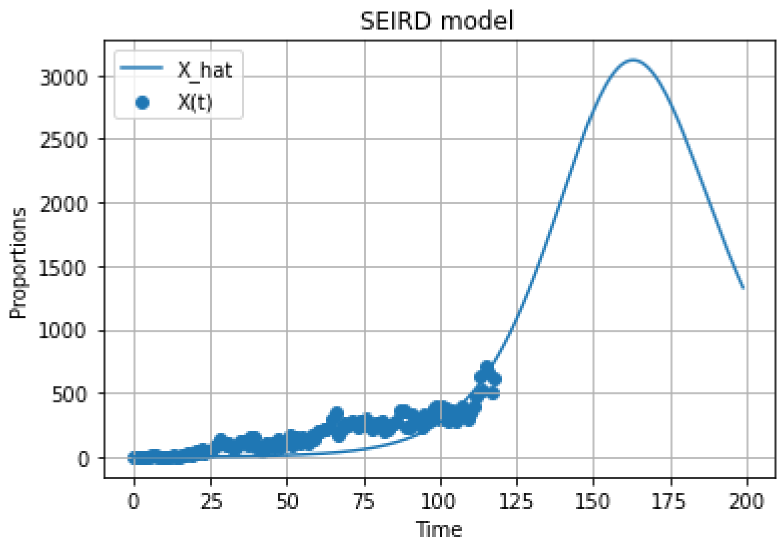
Code to the model will be available at this link after being done with the alterations.

Several Insights out of the model:

1. Predictions for Delhi:



2. Predictions for Rajasthan:



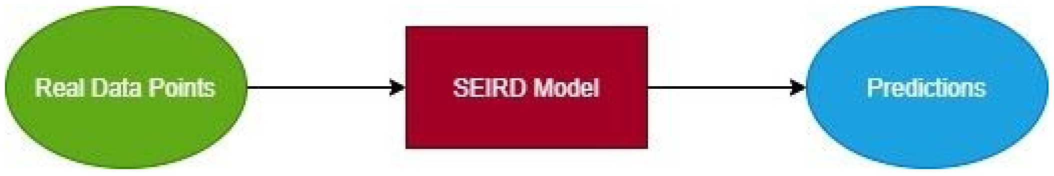
*Here: Time = 0 means the date: **2020-03-14***

Note: In the workflow explained above, to make sure that the epidemic size doesn't go beyond a desirable limit, we decided to put a cap of 400000 on the maximum value of the epidemic size i.e. N.

So far, we have discussed the intuition behind using this model, mathematics and relations of the system S-E-I-R-D and its implementation based on the real data-set to make it more accurate. Now, we will focus on the implementation of this model in the system.

Integration of S-E-I-R-D into the system:

To integrate the model with the backend and the frontend, it can be considered as a black-box model, which takes input as the real data, outputs the results.



What we want is:

Take the data of a particular day, feed that to the data-base, using that data-base, the model should be trained at midnight. The obtained predictions would then be displayed on the frontend.
Refer to the diagram below for better understanding the deployment process.

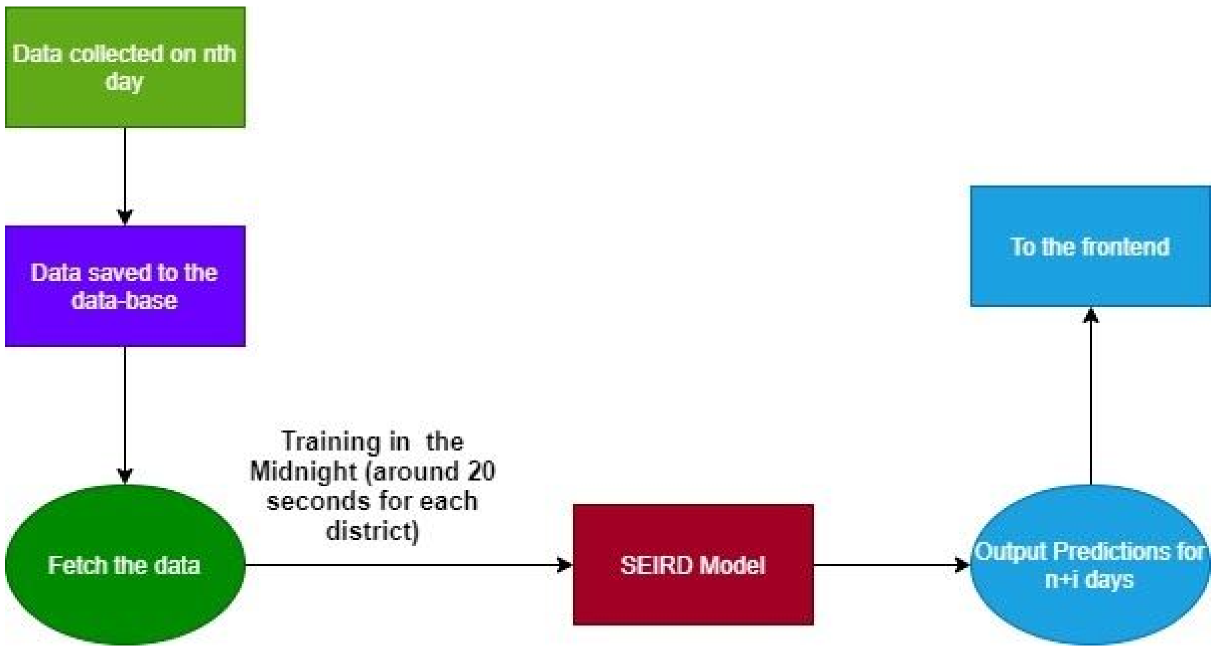

Data collected on nth day
Data saved to the data-base
Fetch the data
Training in the Midnight (around 20 seconds for each district)
SEIRD Model
Output Predictions for n+i days
To the frontend

Fig: Deployment process to the system.

Database required for the model:

The data-base shall be divided into two categories:

1. **Data required for the training**
2. **Data obtained as results which will be used by the frontend.**

Data required for the training:

The structure of the data is as follows:

District	**Date**	**Cases**	**Cumulative**
BADRADRI KOTHAGUDEM	7/15/2020	7	98
BADRADRI KOTHAGUDEM	07-08-2020	5	47

Here, Cases means the new cases on a particular date, cumulative means total cases till particular date.

Since, in the data-set given, we have many missing values for Cases, but we do have Cumulative cases, so we will be using that column for finding the cases on a given date, i.e.

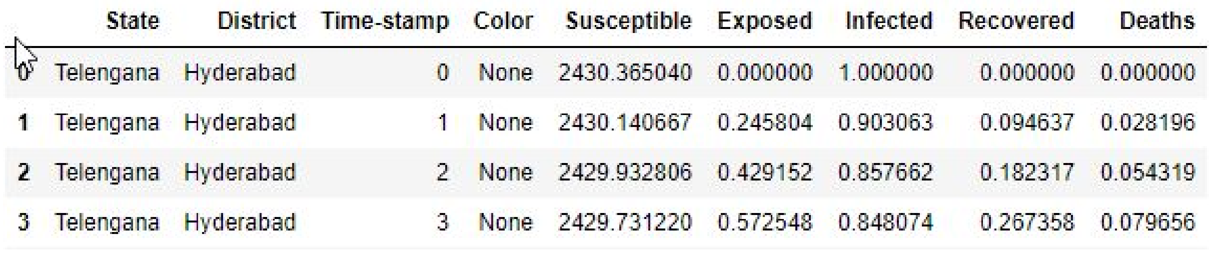
today’s case = cumulative case yesterday - Cumulative cases Today
(the above formula can be implemented in the python code of the model)

So, finally the columns which we are using for training:
Date, District and Cumulative.

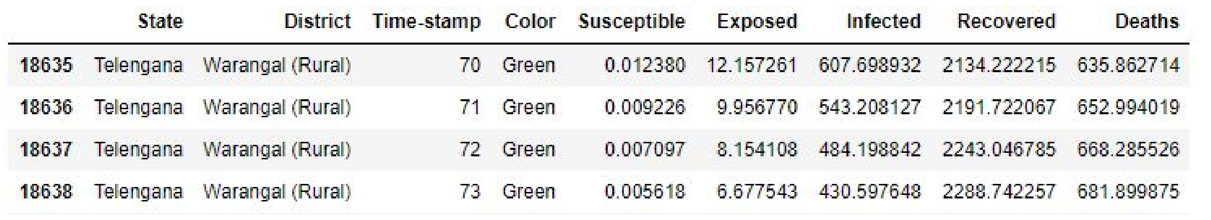
Output Data:
When we let these columns input in the function, the output would be the following predictions for any date:

1. Susceptible population
2. Exposed population
3. No. of Infections
4. Recoveries
5. Deaths

The output variables of the data frame would be like this: **(Note: The date at Time-step = 0 is May-1-2020.)**



	State	District	Time-stamp	Color	Susceptible	Exposed	Infected	Recovered	Deaths
0	Telengana	Hyderabad	0	None	2430.365040	0.000000	1.000000	0.000000	0.000000
1	Telengana	Hyderabad	1	None	2430.140667	0.245804	0.903063	0.094637	0.028196
2	Telengana	Hyderabad	2	None	2429.932806	0.429152	0.857662	0.182317	0.054319
3	Telengana	Hyderabad	3	None	2429.731220	0.572548	0.848074	0.267358	0.079656

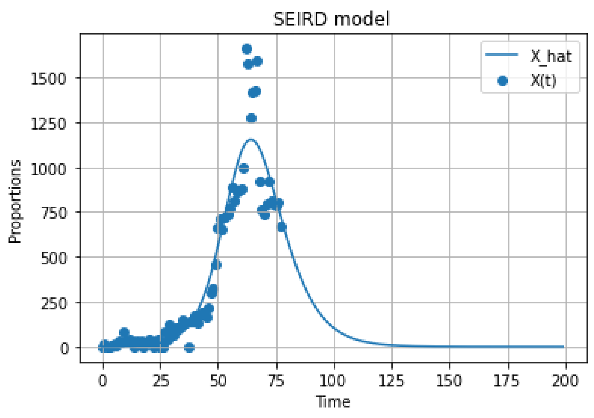


	State	District	Time-stamp	Color	Susceptible	Exposed	Infected	Recovered	Deaths
18635	Telengana	Warangal (Rural)	70	Green	0.012380	12.157261	607.698932	2134.222215	635.862714
18636	Telengana	Warangal (Rural)	71	Green	0.009226	9.956770	543.208127	2191.722067	652.994019
18637	Telengana	Warangal (Rural)	72	Green	0.007097	8.154108	484.198842	2243.046785	668.285526
18638	Telengana	Warangal (Rural)	73	Green	0.005618	6.677543	430.597648	2288.742257	681.899875

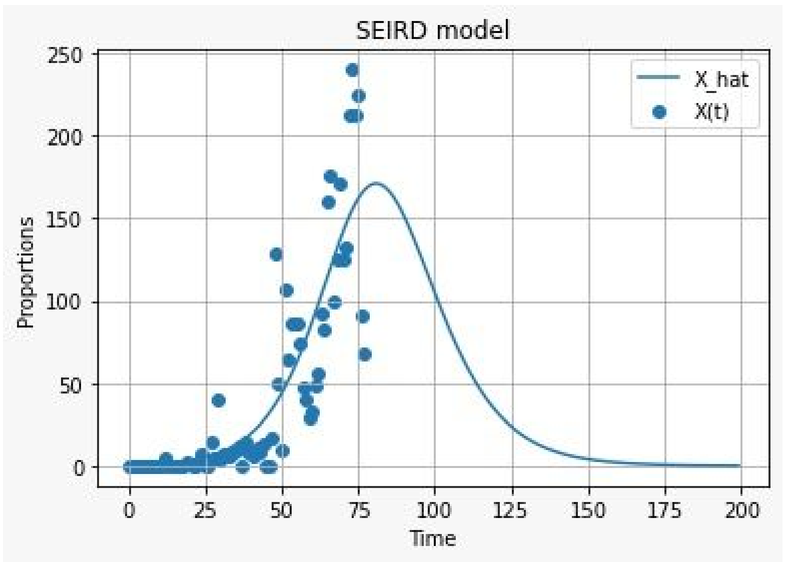
We are predicting infected cases for the next 50 days if we change lockdown status. The color column represents change in the lockdown status.
Color column will have ‘None’ for all the initial predictions for 365 days..

This output dataframe can be used by the frontend now.

The input and output data frames should be updated every 24 hours, the out dataframe is to be updated by the model itself as the columns shown are the predictions.
Please mark the point that the Output data frame has to be different from the input data frame.
Example output for the District GHMC of Telangana:



Predictions for the district RANGAREDDY:



← **Modifications to the model from next page** →

Modifications to the Model:

Agenda Behind the modifications:

A. Number of cases should be highly affected by the number of movements inside the area. Further we can assume that these cases would be a function of Lockdown.

B. Cases should be dependent upon the migration (influx-outflux) to the city.

C. Lockdown color dependent Infected cases prediction

Based on these three conditions the alterations done are defined as:

$$dS/dT = -beta*S*I$$

$$dE/dT = (beta*S*I - alfa*E)+mi*S*I$$

$$dI/dT = alfa*E - gamma*I - M*I +(1-mm[t])*E*I$$

OR

$$dI/dT = alfa*E - gamma*I - M*I +(1-colors[color])*E*I$$

$$dR/dt = (gamma*I)$$

$$dD/dt = (M*I)$$

**** Alterations done are marked in highlighted. Where mi = %migration (influx / outflux) and mm[t] = % lockdown with time.**

Explanations to the modifications:

1. **Addition of "mi*S*I" to the dE/dT equation:**

The rate which people would be exposed to this disease is dependent upon how much the migration population is. More the migrant population, higher will be chances of their migration to their home-town during lockdown, and larger should be the exposed.

How much steeper the exposed people would be, it depends upon how many people are susceptible and out of them, how many are actually infected. This results that migration (mi) multiplied by the Susceptible and Infections should be added to the exposed to increase the rate of exposure to the disease.

2. **Addition of "(1-mm[t])*E*I" to the dI/dT equation:**

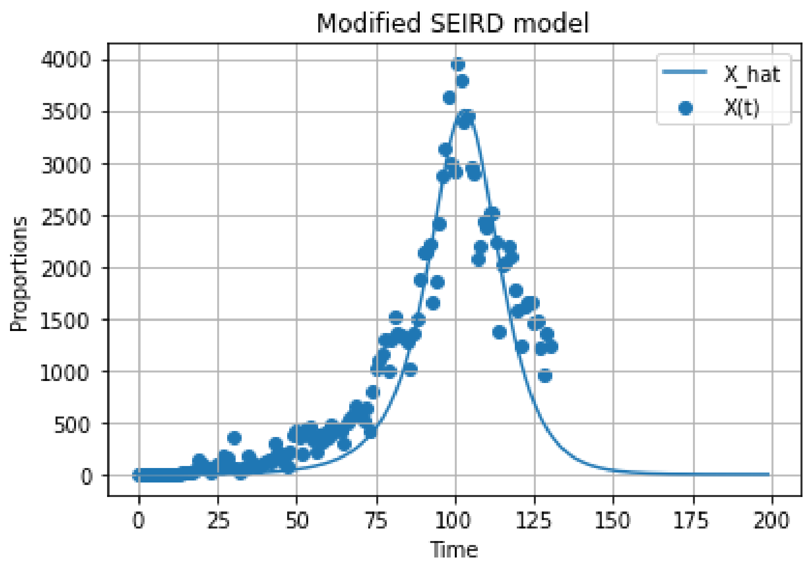
The rate at which infections would increase should depend upon the lockdown conditions as well, lockdown affects the movement inside the area. The "mm[t]" is the lockdown status for each day as a function of time (time ~ day). Which means 1-mm[t] would denote the unlock %. More the unlock %, higher will be the chance of getting exposed to the disease, which clearly means that higher the exposed and Infectious (E*I) and higher the % unlock (1-mm[t])*E*I, larger should be the infections.

3. **Lockdown color dependant prediction +(1-colors[color])*E*I**

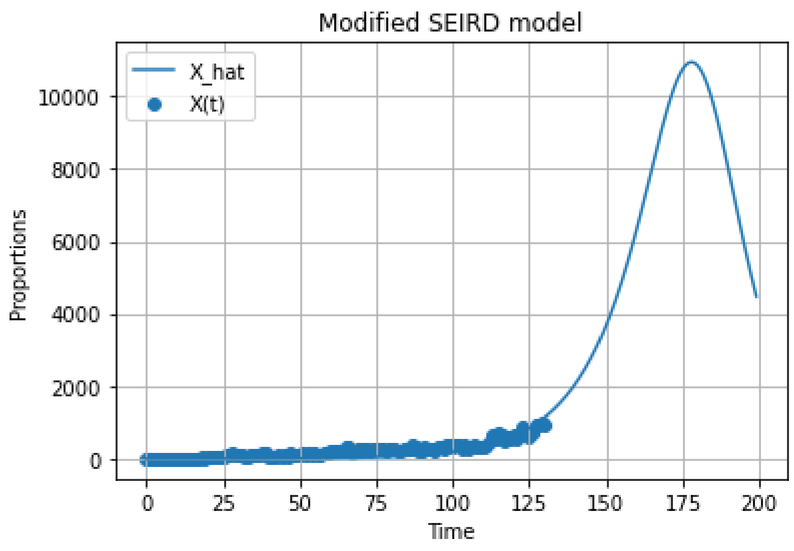
The authorities will have option to check how changing the lockdown status affects the spread of the virus. We have a colors variable which will contain the corresponding unlock percentage.
For each color the model will predict the spread of the virus for the next 50 days. The working behind it is similar to the second change we made earlier except 'mm[t]' could be different for each day but here we are taking the same lockdown status for the next 50 days.

The above first two arguments are being supported by the following results:

Predictions on Delhi (with original lockdown conditions and migration around 5%):



Predictions on Rajasthan (with original lockdown conditions and migration ~5%):



(Note: In case of no lockdown condition and fully lockdown conditions, it is highly unjustified to train the model on original data, because the original data is a result of lockdown and we should abstain fitting our no lockdown model on the lockdown conditions data-set)**

Comparison:

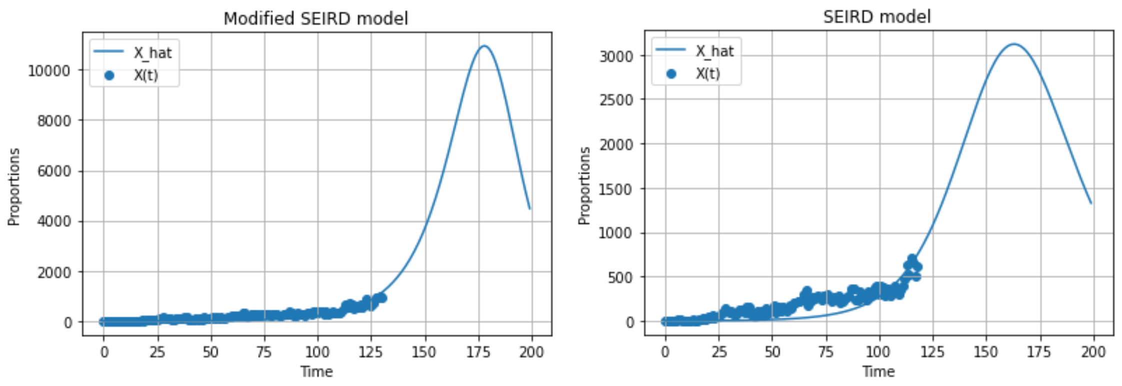


Fig: comparison in the case of Rajasthan

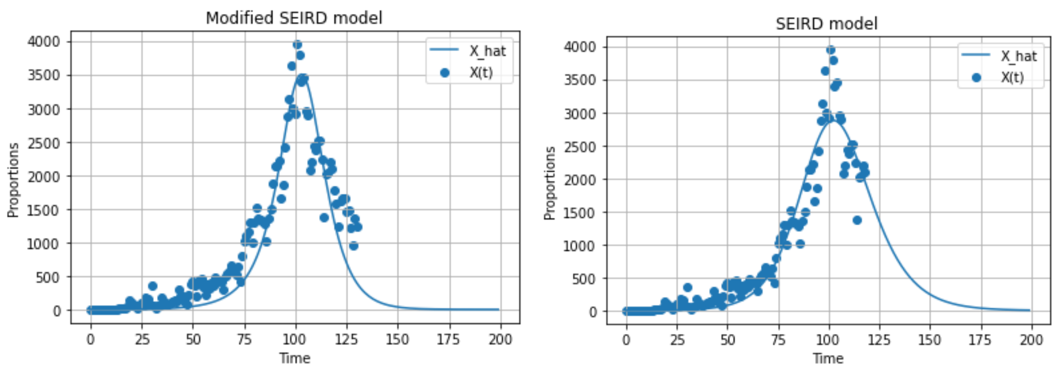


Fig: comparison in the case of Delhi

The difference of the effects of Migration, movement is clear by the comparisons between the two models.

Future predictions as a function of lockdown:

For Hyderabad:

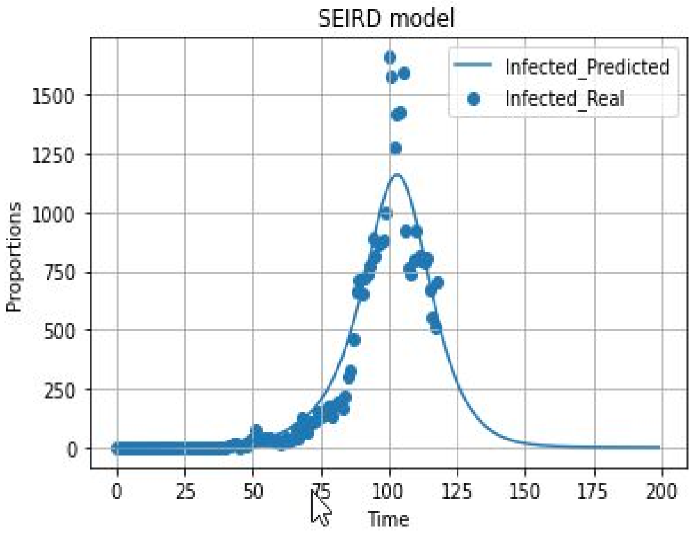


Fig. Predictictions if current measures continue

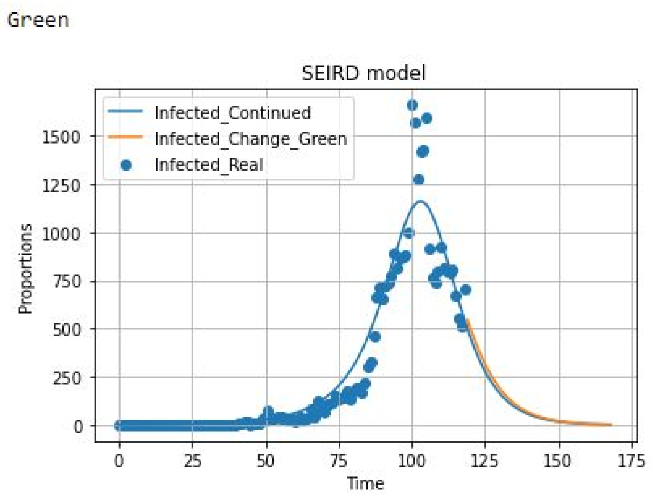
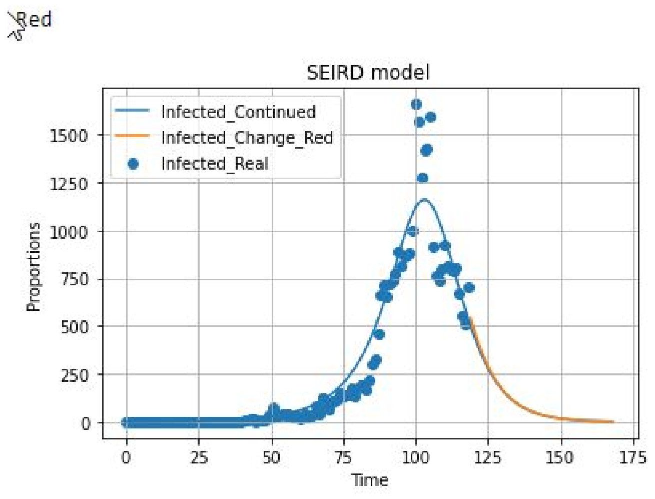
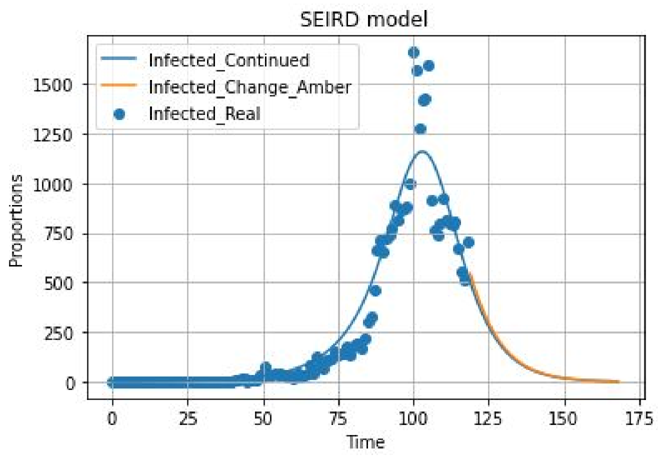
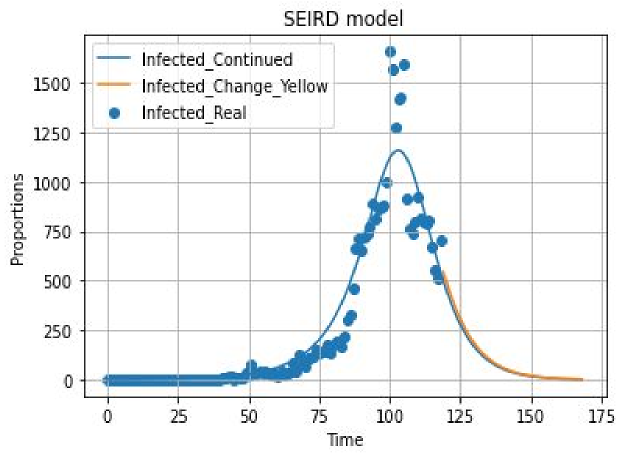
	

Fig. Prediction on changing lockdown to Green	Fig. Prediction on changing lockdown to Red

Amber 	Yellow 
Fig. Prediction on changing lockdown to Amber	Fig. Prediction on changing lockdown to Yellow

For Medchal Malkajgiri District

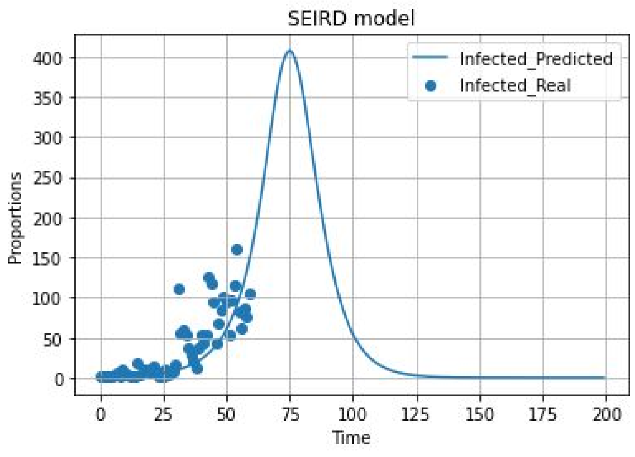


Fig. Predictictions if current measures continue

Red

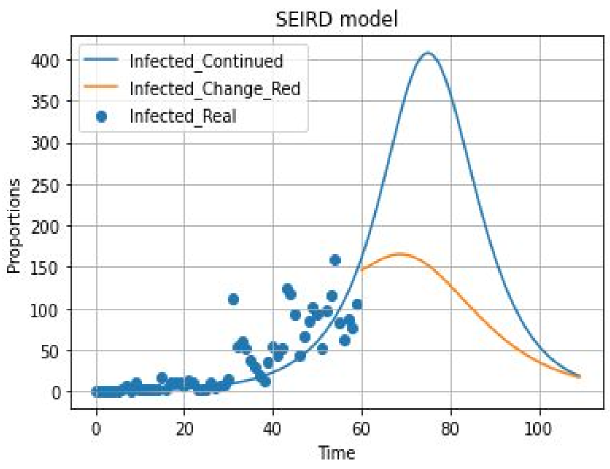


Fig. Prediction on changing lockdown to Red

Amber

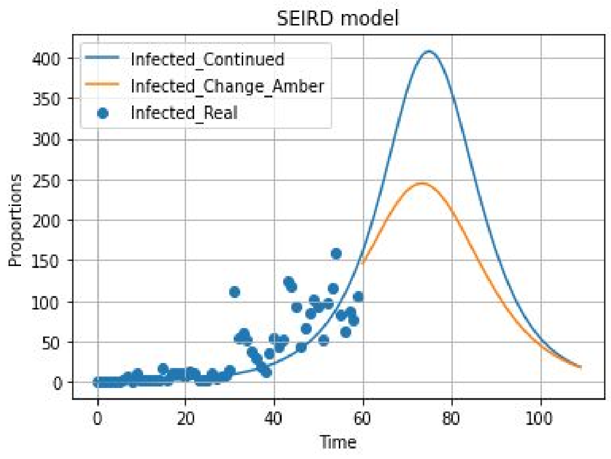


Fig. Prediction on changing lockdown to Amber

Yellow

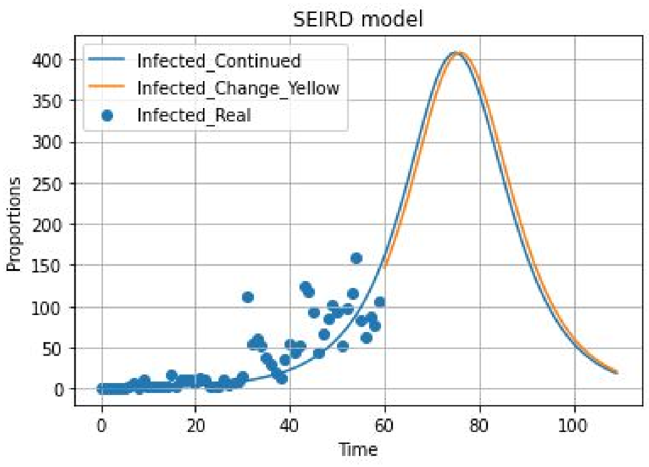


Fig. Prediction on changing lockdown to Yellow

Green

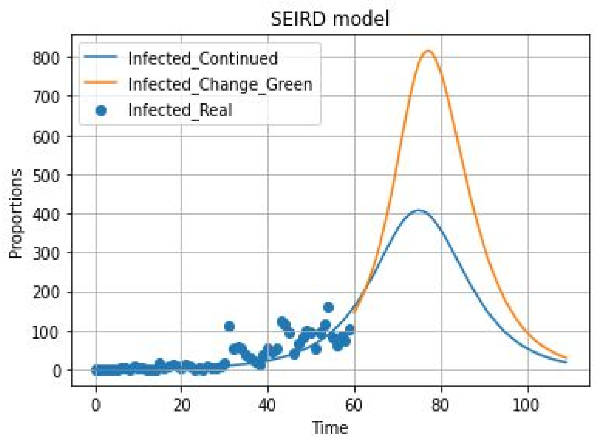


Fig. Prediction on changing lockdown to Green