e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

STOCHASTIC EPIDEMIC MODELLING

Aparna Shree Sivanandam*1

*1Student, Department Of Computer Science And Engineering , College Of Engineering Guindy, Anna University, Chennai, Tamil Nadu, India.

ABSTRACT

This project focusses on depicting an epidemic stochastically using parameters whose values are input by the user. According to this model when the population is subjected to an epidemic, it gets divided into three categories- Susceptible to the disease, infected by the disease and Recovered from the disease. Once a person recovers from an infection, he can't serve as a potential carrier. Each person in the population is a node, consisting of three components-index position, infected or not, duration of carrying the infection, is susceptible initially. Every node can pass on the infection to his neighbors and the no. varies based on his exposure. This is modelled using 3 different neighbor distributions to give a more realistic result-uniform distribution over user-input range, normal distribution over user-input mean and skew normal distribution over user-input mean. There exists a certain probability 'p' for transmitting the infection. Those neighbors who hold the probability 'p' when come in contact with a carrier, get infected. The extents of susceptibility for any node to catch the infection is generated randomly as a uniform distribution, this ensures incorporation of disparity of susceptibility. Each person retains the infection for a certain period Tsim. We simulate this epidemic on the user input population for a series of simulations-Nsim input by the user. We plot the rate of infection spread or no. infected vs no. of simulations. This way we will be able to predict when the peak heightens and how long it takes for the curve to flatten. We can vary the parameters and observe the changes produced on the curve.

Keywords: Stochastic, Susceptible, Infected, Recovered, Neighbor Distribution.

I. INTRODUCTION

A simple stochastic epidemic model is defined and exact and asymptotic (relying on a large community) properties are presented. The purpose of modelling is illustrated by studying effects of contact, exposure and also in terms of inference procedures for important parameters, such as the basic reproduction number and the contact ratio of each individual and the varying susceptibilities of each individual. The model is based on several generalizations towards realism.

Background

Network epidemiology has become a core framework for investigating the role of human contact patterns in the spreading of infectious diseases. Network epidemiology represents the contact structure as a network of nodes (individuals) in contact with others and the disease as a compartmental model (where individuals are assigned states with respect to the disease and follow certain transition rules between the states). The nodes propagate the epidemic across a network of neighbours. In the context of an epidemic, it is important that we monitor spread of the disease inorder to take appropriate caution and ensure the spread is contained. Modeling can fill gaps in the decision making process by using available data to provide quantitative estimates of outbreak trajectories. Effective reduction of the spread of infectious diseases can be achieved through collaboration between the modeling community and public health policy community. We need to suitably model the epidemic spread based on all the features affecting its spread. The model designed here helps model any epidemic, where the person after recovery will not be susceptible again. If we feed in the population, the neighbour distribution for each node, the probability of spreading infection, the time for which an individual retains the infection and the period you wish to monitor over. The model, augments the input parameters by incorparating differences in susceptibility of each node, or probability of getting infected. On the basis of the simulation, the model produces a plot of No. of people infected Vs. No. of simulations. This stochastically represents how the infection spreads, when the peak occurs and when the curve begins to flatten. The entire framework is designed using Python and its several libraries provided for data analysis and visualisation.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

Various Types of Modelling Used:

The model was initially developed to have a discrete number of neighbors for each node(phase 1) and the probability of getting infected for each node is distributed uniformly(continuously). As we developed the model (phase 2, appendix B) we generate a random no. of neighbors for each node (continuous random variables) which are either normally distributed, uniformly distributed, or, skew normally distributed to give a more accurate representation like reality.

Assumptions:

The model is constructed on the basis of the following assumptions:

- The population is assumed to remain constant throughout observation.
- Any demographic process - birth, death and migration, is considered as not existent during the period of observation
- The neighbors of a node are constant.
- The entirety of the population is susceptible to the infection.
- The infected individual retains the infection for a period Tsim input by the user, and as long as he retains the infection, he can continue to spread it to his neighbors.
- The individual who recovers from the infection cannot be infected again.
- Not all neighbors of an infected node get infected, only the ones which have a probability of getting infected <= the probability of infection spread.

Rationale

The rationale behind this study and modelling is the need to predict the spread of a disease and determine approximately condition under which the situation gets better or worse. The need for modelling epidemics heightened during this global pandemic. Modelling the spread, observing peaks and falls are important in issuing orders such as curfew so as to ensure the spread remains contained. This model also gives the expected number of infections and recoveries for a set of parameters, so this predicts concisely what range of values the parameters must be kept at so as to contain the breakout.

Statement of the Problems

In the wake of a pandemic, it feels extremely important to know how much of the population the virus has taken over. Observing the cases rise in our locality forces us to remain cautious. Demarcating areas with a larger number of infections as containment zones prevents contact with these high-risk locations. Seeing recoveries tells us if the current strategy to combat the epidemic is working as intended or needs improvement. Modelling an epidemic is crucial to determining how to handle the outbreak. Predicting approximately the spread, when it rises to its peak, how long it tentatively continues to spread severely, helps us to determine measures to be undertaken to control spread. There are several parameters that affect the spread of the disease and it varies from person to person. The immunity, exposure and hence susceptibility of infection varies. In order to stochastically model such a dynamic outbreak we have to approximately showcase the disparities in susceptibilities and spread. Thus, several parameters have to be considered.

Hence the problem consisted of:

- Generating nodes representing the members of the population in a manner such that, when infected they can be distinguished from the rest of the uninfected population.
- The nodes have to be designed such that we can identify if they have already been infected and have recovered from the infection so that even if they come in contact with a carrier, they won't be infected again.
- The nodes should have different number of neighbors to precisely show the differences in exposure and hence the number of people they come in contact with.
- The neighbors of an infected node are not all equally probable to get infected to depict practically the factors of variability such as immunity and precautions taken by neighboring nodes
- The neighbors should be distributed equally around the infected node and not remain concentrated to to one side. This will make the propagation non-linear and more realistic.
- Through the communicable period, an infected node must continue to infect neighbors.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

- Each infected node must retain infection only for the communicable period after which he has to be marked as recovered.
- We need to repeat the simulation several times and take average of the values of infection spread each time so as to get a reliable representation.
- We need to plot the No. infected vs. The simulation to show curve hitting the peak and eventually flattening.
- Produce a visualization to show how the model simulates the spread of the disease.

Objectives of the Research

Overall objective

The main objective of this model is to predict epidemic spread largely close to actuality by incorporating all features of variability and all parameters that contribute to the spread. The model aims to stochastically simulate infection spread over multiple runs to show the distribution of the infected in population over the period of observation with the contributing factors as the basis of infection spread. The final major outcome of the model is to produce a plot showing the curve depicting infection spread over the observation period.

Additional Objective

The model also aims at providing an animation to effectively visualize how the model implements the spread.

Scope

The present work focusses mainly on the spread of a simple disease where the entire population is susceptible. The factors controlling the spread are probability of infection spread, the number of neighbors for each node and the probability of each neighboring node to get infected. Each node gets infected and then recovers after the communicable period. The varying severity of the disease in each individual considering their existing health condition, age and medical history hasn't been accounted for. The model doesn't consider demographic factors such as birth, migration and death. The mobility of a node is not accounted for, the model can't be used in a case where a nodes travels and comes on contact with a new set of neighbors each time.

II. METHODOLOGY

The model is compartmental. The nodes transition between 3 stages: susceptible, infected and recovered.

All the parameters that control the spread of the disease are user input. The parameters taken into consideration are:

1. Total Population-N
2. The neighbor distribution
3. Probability of infection-p
4. Time taken to recover-Tsim
5. No. of simulations-Nsim
6. No. of nodes affected initially.

Creation of Nodes:

Based on the user input, we created an environment of N nodes that are susceptible to the epidemic. We implemented these nodes as a circular list to avoid any edge effect and for efficient distribution of neighbors across the population. The nodes are designed to hold 3 tuples:

- Index position
- Infected or not (indicated using 1 and 0 respectively)
- Time the node has retained the infection.

```
#creating population-circular list of tuples
while i <N:
population[i]=[i+1,0,0]
i+=1
```

Figure 1: Code for Population: list of nodes

- The index position is used for locating the neighbors in the population.
- All the nodes are initially modelled as [index position, 0,0].

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

- If the node gets infected, the second tuple is made 1 to indicate it as a carrier. The carrier node passes infection on to his neighbors based on their exposure and probability of getting infected.
- The carrier node is appended to a list infected []
- From the time when the node gets infected, with each simulation, its third tuple or the communicable period is incremented to show the period the node has been infected.
- When the value is equal to Tsim, the second tuple is made 0 again to show the node has recovered.

```
if(population[i][1]==1):
population[i][2]+=1#incrementing Tsim by 1
```

Figure 2: Code for Incrementing period of being infected

Infecting initial set of nodes

Based on the user input of the number to be infected initially, we randomly generate indices to be infected. These nodes are infected before the start of the simulation. These nodes act as initial carriers that spread the disease and are dispersed randomly across the environment under observation.

```
#generating first set of affected individuals
arr=np.random.randint(1,N,ini)
print(arr)
for i in arr:
infected.append(i)
j=1
while j<=N:
if(j==i):
population[j-1][1]=1
j=j+1
```

Figure 3: Code for creating initial set of infected nodes

We generate random numbers using a function randint under the random module of package NumPy. We set the second tuple of the nodes corresponding to generated indices to mark them as infected.

We store all the index positions of infected nodes in a list 'infected'. This is to keep track of all infections and also ensures, once infected the node isn't infected again.

Generating neighbors for each node

The model focusses on generating neighbors in a manner such that it approximately shows the real disparity in exposure or contact. While some individuals, say shopkeepers or delivery agents, come in contact with large number of people. On the other hand, a person living alone and isolating himself wouldn't have much exposure.

This variation in exposure is modelled using randomly generated values distributed continuously across a range specified by the user.

There are 3 continuous distributions that the user can choose from:

1. Uniform distribution
2. Normal distribution
3. Skew normal distribution

Each node is assigned a different continuous random variable distributed across any range and distribution chosen by the user.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

```
print("Specify your choice for neighbour distribution:\n1.Uniform Distribution\n 2.Normal Distribution\n 3.Skew Normal Distribution\n")
choice=int(input())
if(choice==3):#Skew Normal Distribution
print("Enter the mean  of neighbours for each node:")
m=float(input())
neighbours=sn.rvs(4,m,size=N)
if(choice==2):#Normal Distribution
print("Enter the mean of neighbours for each node:")
m=float(input())
neighbours=nrm.rvs(m,size=N)
if(choice==1):#UniformDistribution
print("Enter the lower bound of range:")
a=int(input())
print("Enter the upper bound of range:")
b=int(input())
b=b-a
neighbours=uni.rvs(a,b,size=N)
```

Figure 4: Code for Generating neighbors across different distributions

The neighbors generated for each node are stored in a list 'neighbours[]',where the neighbors for a particular node can be found at corresponding index position.

This is implemented using Python's package Scipy.stats and modules such as skewnorm (skew normal distribution), norm (normal distribution), uniform (uniform distribution) and the function rvs ( ) to generate the random variables.

Spreading infection to neighbors

Not all neighbors are equally likely to get infected after coming in contact with the infected node. In an attempt to model differences in immunity, medical history and hence, susceptibility to the infection, we generate uniformly distributed probabilities for each node denoting their ability to resist the infection. The threshold for resistance is the probability of infection, 'p', input by the user. Those neighboring nodes with generated values less than or equal to the user input value of probability of spread are infected thereafter.

```
if(population[i][1]==1):
                n=int(neighbours[i])
if(n<0):
                    n==0
                prob=np.random.rand(n)#array for generated probabilities of getting infected for each neighbour.
                pos=0
                flag=0
for j in prob:
                    pos+=1#pos represents relative position of the neighbour
if(j<=p):#checking which probabilities are lesser than user input p
                        flag=1
break
```

Figure 5: Code for Determining which neighbors get infected

Here we randomly generate the probabilities that are uniformly distributed using the rand ( ) function under random module under NumPy.

Distributing neighbors around the node

The model is designed to largely mimic reality. The neighboring nodes don't occur to either side, rather around the node expressing contact with the carrier. This is modelled by having half the neighbors to one side and the other half to the other side.

The list is largely observed as circular to avoid any edge effect, i.e., Nth node's immediate neighbors must include 1 and vice versa.

By distributing neighbors around the infected and having determined which neighbor gets infected, we use the relative index position to locate and infect the node. Before we infect the node, we check to ensure it is not

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

already infected. This is done by scanning 'infected[ ]' as all nodes can only be infected once. We append the infected index position to the list 'infected [ ]'.

```
if(found==0):#if any node has immunity/susceptibility below threshold
                        infected.append(((m-pos)%N))#to avoid edge effect we treat list circularly. we add to list 'infected' to keep track of all infections
                        population[(m-pos)%N][1]=1
                        before=1#showing neighbour's position is before the current carrier
else:#case for neighbours succeeding considered node
                    m=i
                    j=0
                    found=0

for j in infected:
if(j==(((m+(n-pos))%N))):
                            found=1
break

if(found==0):
                        infected.append(((m+(n-pos))%N))
                        population[((m+(n-pos))%N)][1]=1
                        after=1#showing neighbour's position is after the current carrier
```

Figure 6: Code for Infecting neighbors

Recovery of infected node

The model tracks the population for any node that has remained infected for the period greater than or equal to user input communicable period Tsim. If any such node is found, it is changed to its original state [index,0,0].

The question that arises now is how can a recovered node be distinguished from a susceptible node? The answer to the question is scanning infected[ ]. The list not only keeps track of all the infected individuals but also gives us a list of all the nodes already infected. Before infecting a node, we must check for its absence in infected [ ] showing its still susceptible.

```
for i in population:
if(population[i][2]>Tsim):
                    population[i][1]=0
```

Figure 7: Code for Recovery of infected node

Counting the number of infections

The major outcome of modelling is a plot to show the no. of infections in each simulation. The curve obtained will help us to predict the disease spread. so as to achieve this, after every simulation on the population we need to keep track of the number of infections generated.

This is done by counting the number of elements in the list infected [ ] at the end of each simulation.

```
#count variable
        c=0
        x=0
while x<N:#counting the no.of infected individuals in the population
if(population[x][1]==1):
                c+=1
            x+=1
        count.append(c)
```

Figure 8: Code for counting the number of infections in each simulation

We store the count of infections generated in each simulation into a list called count [ ]. The index of the list element corresponds to the simulation its produced. We finally have to plot the list elements vs. their indices.

Avoiding small-number effect

The law of small numbers explains the Judgmental bias which occurs when it is assumed that the characteristics of a sample population can be estimated from a small number of observations or sample data.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

Therefore, while studying any survey, the length of data should be given an important consideration as the probability of it being relatable is directly correlated to the length of the sample.

To overcome small-number effect we run the code several times and take the average of the values obtained for no. of infections obtained in each simulation and plot the average no. of infections/ simulations.

```
i=0
while i<10:
    j=0
    infected=[]
    count=infection(N,neighbours,Tsim,Nsim,p1,ini,population)
while j<Nsim:
        s1[j]=s1[j]+count[j]
        j=j+1
    i=i+1
s1=np.array(s1)
s1=s1/10
```

Figure 9: Code for eliminating small no. effect

So as to plot the number of infections against each simulation, we have to change the list to an array, this is done using NumPy.

III. MODELING AND ANALYSIS

We divide the population into 3 categories:

SUSCEPTIBLE(S): The portion of the population that is prone to catching the infection

INFECTED(I): The portion of the susceptible that has caught the infection

RECOVERED(R): The portion of the infected that recovers from the infection and never catches it again.

We know the total population remains a constant, say N

Table 1: Equation for population

N=S+I+R

Now, let us consider the rate of susceptibility, infection and recovery to analyze the effect of the epidemic. To do the same, we know:

The infection remains for say, 2 weeks. So, 14 days from the start, the people infected initially recover. That is,1/14, or a fraction "b" of the infected people recover every day since the infection.

The infection spread depends on the number of people who are susceptible or prone and also the probability of catching the infection or a fraction "a". It also depends on the number of nodes an infected person comes in contact with- the contact ratio "c".

The spread of infection can be calculated as shown in table 2

Table 2: Equation for spread of infection

a × contact ratio × susceptible× infected

because every time a susceptible individual comes in contact with an infected individual, he also catches the infection with a probability a.

The number of susceptible decreases as the no. of infected increase. So rate of susceptibility is the negative of infection spread.

Table 3: Equation for calculating infection spread

$\frac{dS}{dt} = -a \times c \times S \times I$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

Table 4: Rate of Infection

$$amount\ of\ infection\ spread - the\ amount\ of\ recovery\ of\ infected\ \frac{dI}{dt} = a \times c \times S \times I - b \times I$$

Table 5: Rate of recovery

$$fraction\ of\ recovery \times the\ no.\ of\ infected\ \frac{dR}{dt} = b \times I$$

Table 6: Rate of Infection

$$\frac{dI}{dt} = a \times c \times S \times I - b \times I$$
$$= (acS - b)I$$

Table 7: Conditions that control the epidemic

Condition:	Effect on spread of epidemic	Effect on rate of infection spread (I')
S > b/ac	EPIDEMIC SPREADS	I' > 0
S < b/ac	EPIDEMIC STOPS SPREADING	I' < 0

The threshold here is b/ac, i.e., if we started with fewer than b/a no. of susceptible, then the epidemic would have never spread.

Our objective is to minimize the disease spread and this can be done in the following ways:

- We need to minimize the transmission coefficient ac
- We need to reduce no. of susceptible
- We need to increase b.

IV. RESULTS AND DISCUSSION

The outcome of this project is the plot that shows infection spread as function of number of simulations. This plot is essential for analysis and predicting how the epidemic grows over the population.

```
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

1
Enter the lower bound of range:
3
Enter the upper bound of range:
10
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability1 of infection:
0.55
Enter the no. initially infected:
5
```

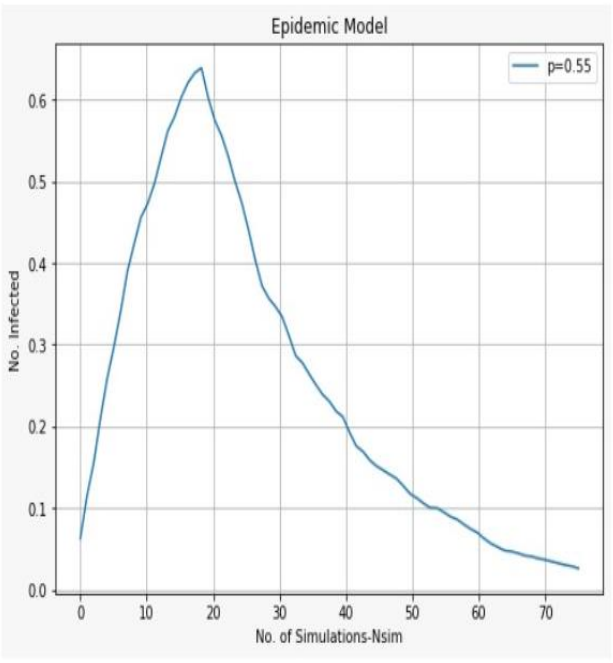

Figure 10: Final plot obtained

Effect of varying the probability of infection:

We observe the changes in no. of infections per simulation by varying probability of infection spread between high and low values and observe the changes produced in the curve.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

```
EFFECT OF PROBABILITY
Enter the Population:
1000
Specify your choice for neighbour dist
ribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

1
Enter the lower bound of range:
3
Enter the upper bound of range:
10
Enter the time for each individual rem
ains diseased:

20
Enter no. of simulations:
75
Enter the Probability1 of infection:
0.01
Enter the Probability2 of infection:
0.43
Enter the Probability3 of infection:
0.879

Enter the no. initially infected:
10
```

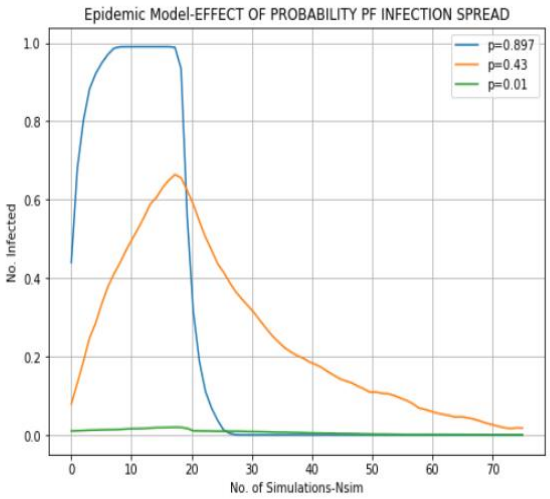

Figure 11: Effect of change in probability (when neighbors are distributed uniformly)

```
EFFECT OF PROBABILITY
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

2
Enter the mean no. of neighbours for each node:
5
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability1 of infection:
0.01
Enter the Probability2 of infection:
0.43
Enter the Probability3 of infection:
0.879
Enter the no. initially infected:
5

Enter the no. initially infected:
10
```

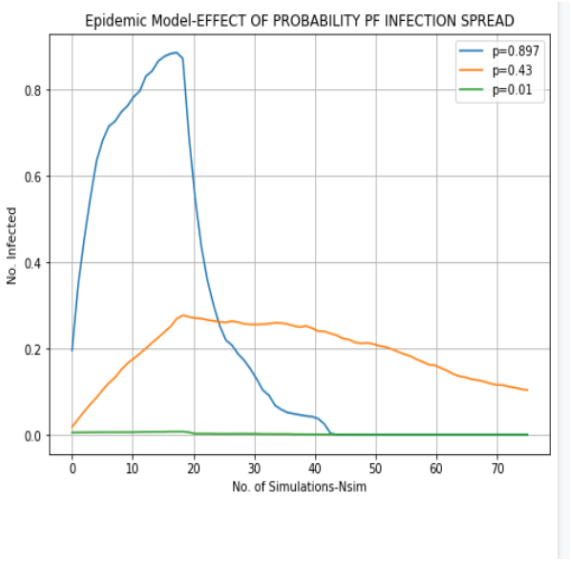

Figure 12: Effect of change in probability (when neighbors are distributed normally)

```
EFFECT OF PROBABILITY
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

3
Enter the mean no. of neighbours for each node:
5
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability1 of infection:
0.01
Enter the Probability2 of infection:
0.43
Enter the Probability3 of infection:
0.879

Enter the no. initially infected:
10
```

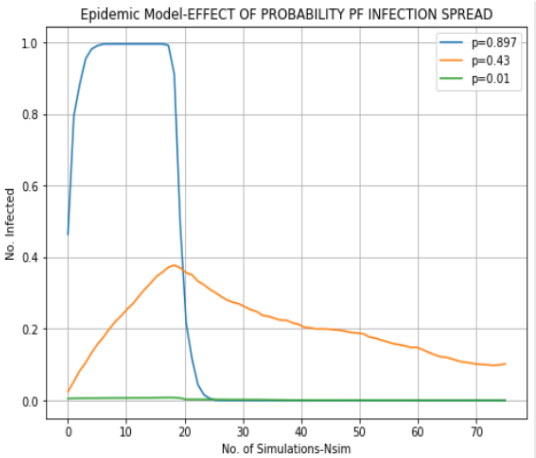

Figure 13: Effect of change in probability (when neighbors are distributed skew normally)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

Table 8: Relation between probability and spread

$Probability\ of\ infection\ spread \propto no.of\ infections/simulation$

Effect of varying the ranges or means of neighbor distribution

The model produces neighbors for each node as a random variable distributed uniformly, or normally, or skew normally. These distributions are governed by lower and upper bounds (in the case of uniform distribution) and mean (in the case of normal and skew normal distributions).

By changing the governing factor, i.e., lower and upper bounds or mean of these distributions, we can in turn change the range across which the number of neighbors are randomly generated for each node and observe how it affects then spread.

```
EFFECT OF DISTRIBUTION OF NEIGHBOURS-VARYING RANGE
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

1
Enter the lower bound 1 of range:
1
Enter the upper bound 1 of range:
4
Enter the lower bound 2 of range:
6
Enter the upper bound 2 of range:
10
Enter the lower bound 3 of range:
7
Enter the upper bound 3 of range:
17
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability of infection:
0.55

Enter the no. initially infected:
10
```

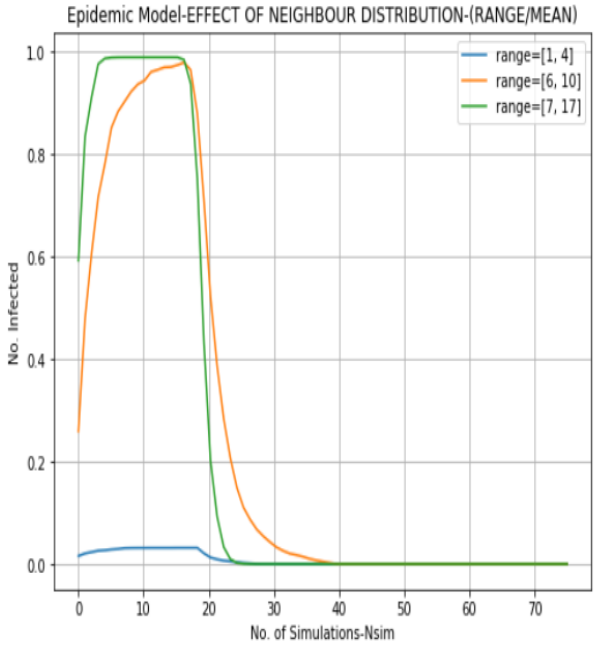

Figure 14: Effect of change in range of neighbor distribution (when neighbors are distributed uniformly)

```
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

2
Enter the mean no.1 of neighbours for each node:
1
Enter the mean no. 2 of neighbours for each node:
5.54
Enter the mean no. 3 of neighbours for each node:
9.56
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability of infection:
0.55

Enter the no. initially infected:
10
```

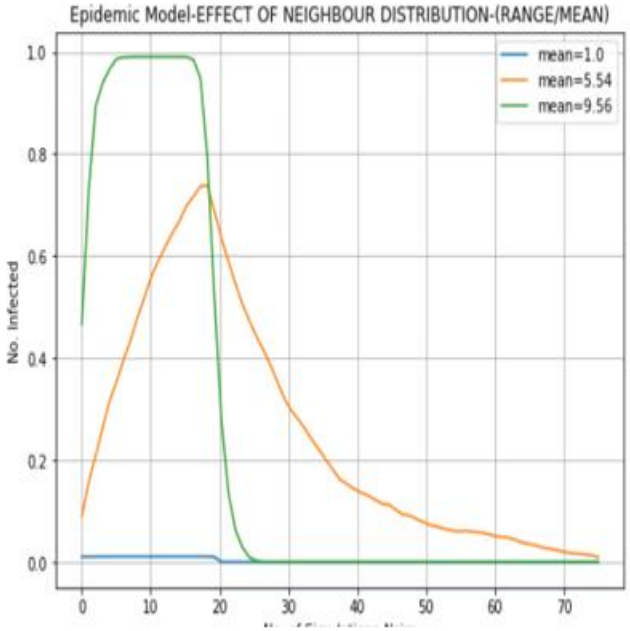

Figure 15: Effect of change in range of neighbor distribution (when neighbors are distributed normally)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

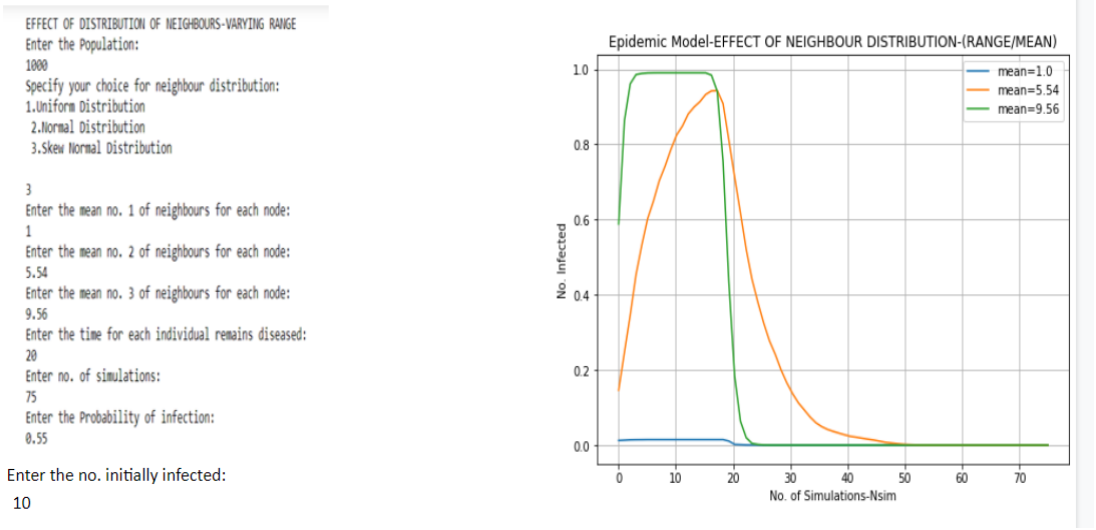

Figure 16: Effect of change in range of neighbor distribution (when neighbors are distributed skew normally)

Table 9: Relation between range and spread

$mean\ or\ range of neighbour\ distribution \propto no.of\ infections/simulation$

Effect of varying number of nodes infected initially

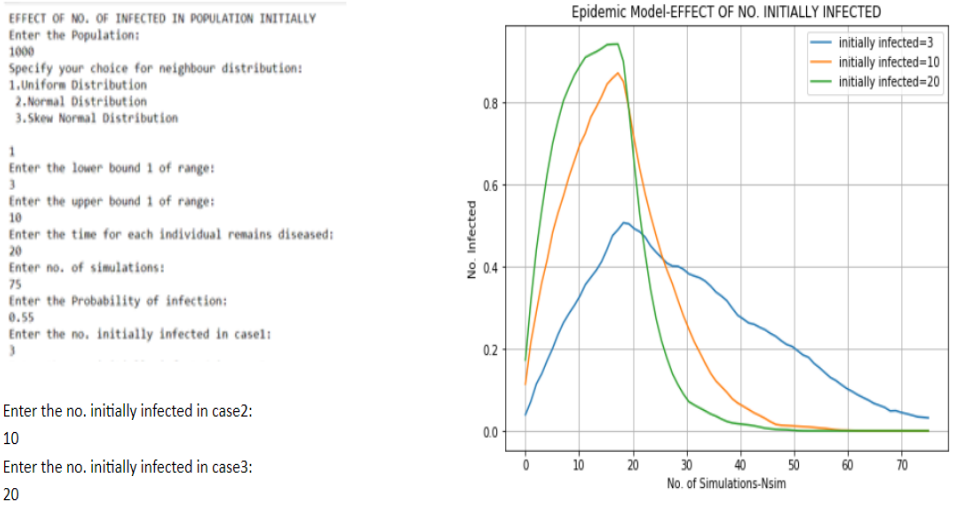

Figure 17: Effect of change in number of nodes affected initially (when neighbors are distributed uniformly)

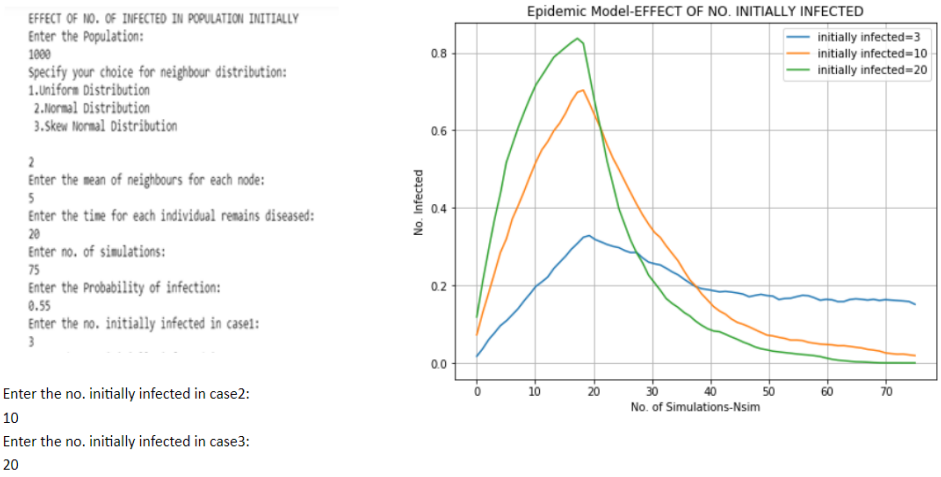

Figure 18: Effect of change in number of nodes affected initially (when neighbors are distributed normally)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

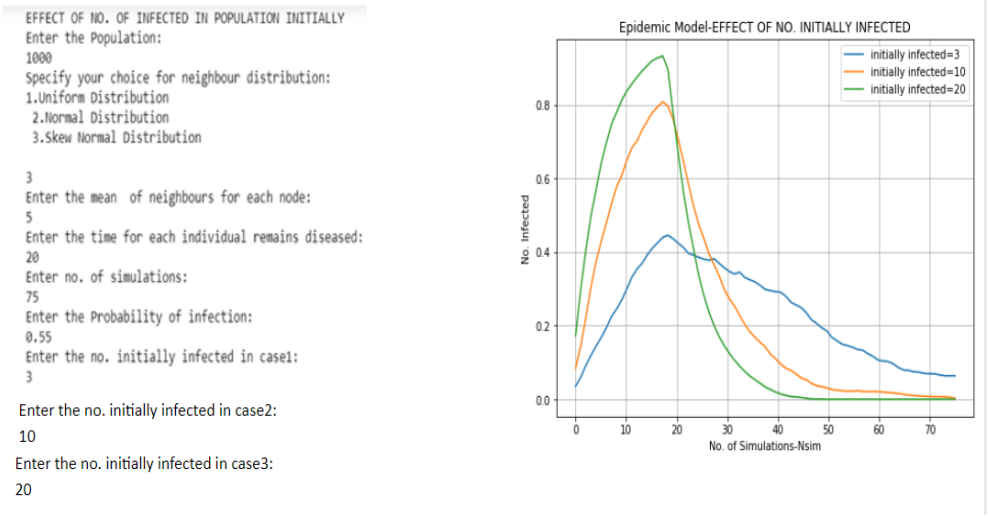

Figure 19: Effect of change in no. of nodes affected initially (when neighbors are distributed skew normally)

Table 10: Relation between number of nodes initially infected and spread

$No. of\ nodes\ infected\ initially \propto no. of\ infections/simulation$

Effect of varying communicable period

Communicable period is the time for which a node remains infected and spreads the epidemic. By changing this value between low and high points we can study the effect that time taken to recover has on the spread.

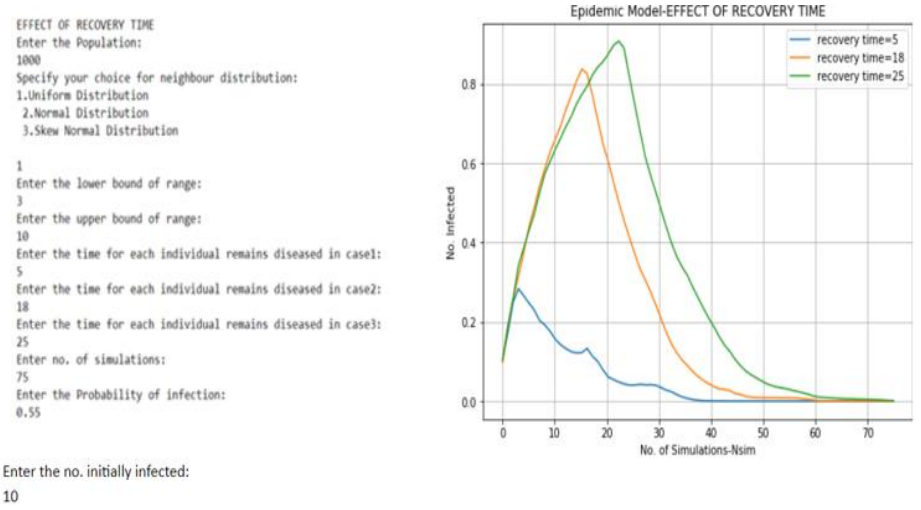

Figure 20: Effect of change in time taken for recovery (when neighbors are distributed uniformly)

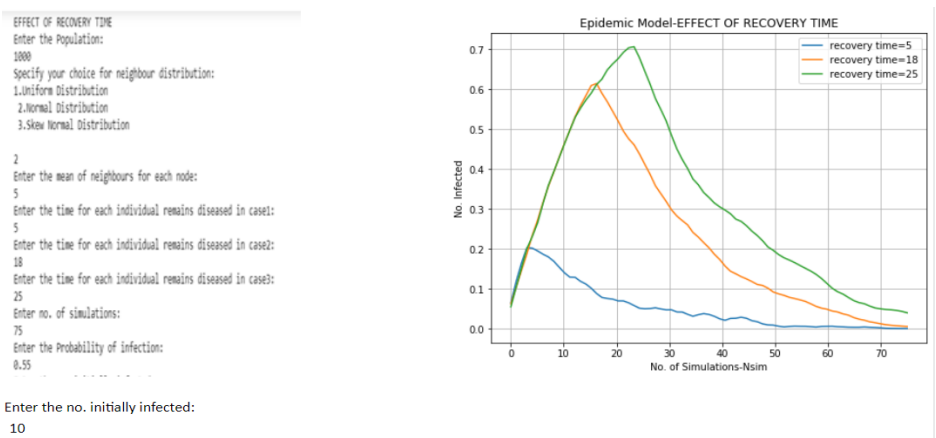

Figure 21: Effect of change in time taken for recovery (when neighbors are distributed normally)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

```
EFFECT OF RECOVERY TIME
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

3
Enter the mean  of neighbours for each node:
5
Enter the time for each individual remains diseased in case1:
5
Enter the time for each individual remains diseased in case2:
18
Enter the time for each individual remains diseased in case3:
25
Enter no. of simulations:
75
Enter the Probability of infection:
0.55
Enter the no. initially infected:
10
```

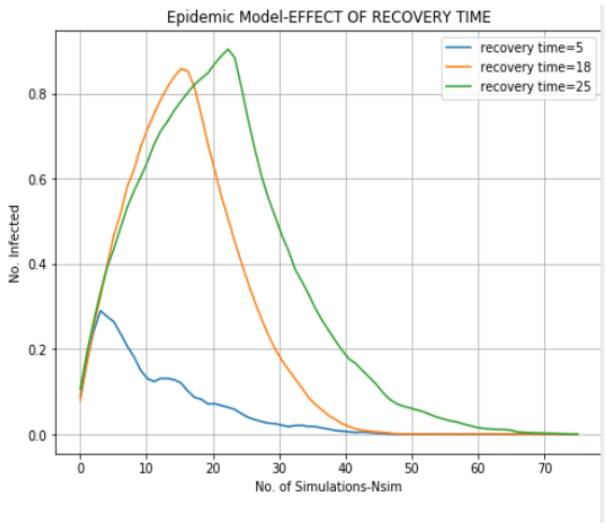

Figure 22: Effect of change in time taken for recovery (when neighbors are distributed skew normally)

Table 11: Relation between time taken for recovery and spread

$Time\ taken\ to\ recover \propto no.of\ infections/simulation$

Effect of different distributions of neighbors

As an analytical measure, we also try and compare the outcome for the different neighbor distributions over roughly the same range to see how the same disease parameters leads to different outcomes.

```
STUDYING THE EFFECT OF TYPE OF DISTRIBUTION
Enter the Population:
1000
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

1
Enter the lower bound of range:
3
Enter the upper bound of range:
10
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

2
Enter the mean of neighbours for each node:
6.5
Specify your choice for neighbour distribution:
1.Uniform Distribution
 2.Normal Distribution
 3.Skew Normal Distribution

3
Enter the mean of neighbours for each node:
6.5
Enter the time for each individual remains diseased:
20
Enter no. of simulations:
75
Enter the Probability of infection:
0.55
Enter the no. initially infected :
10
```

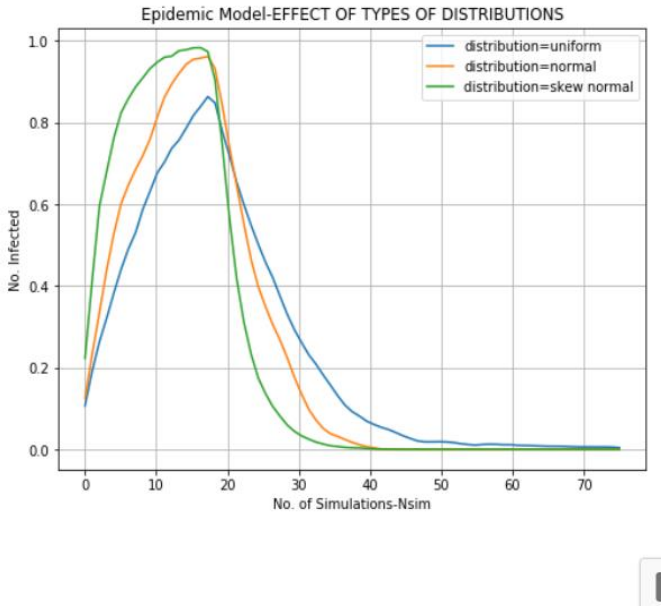

Figure 23: Effect of type of neighbor distribution

V. CONCLUSION

Summary

An agent based stochastic model has been created using Python programming. The framework based on several parameters largely contributing to the spread is operated to obtain results close to real.

The purpose of this model is to predict the spread of the disease by passing parameters such as population, number initially infected, time an individual remains infected, probability of the infection to spread and time for which we are observing the population. To add to the portrayed realism of the model, it attempts to mimic disparity in exposure, immunity and susceptibility to get infected.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

This way, the results the model provides are stochastic- approximate but reliable. The plot of number of infections for each simulation shows us the conditions under which the spread is excessive and when it begins to get contained.

Evaluation

Shortcomings

- The model doesn't classify the infection as severe and mild, it generalizes the infection and considers time for recovery as a constant.
- The model doesn't account for mobility of nodes where they come in contact with new set of nodes, the neighbors of each node remain constantly
- The model infects neighbors in the vicinity of the infected node, so it may have to infect a more susceptible node whose index < index of current node, to spread its infection, we iterate the whole process of spreading infection from the lower index value. As a consequence, the process is time consuming.

Advantages

- The model is simple to analyze
- The model considers most parameters important to determining the extent of spread.
- The model accounts for all disparities we observe in a real population such as exposure and susceptibility.
- The model is driven by network between nodes and is agent-driven. It's not a deterministic model framed using differential equations. As a consequence, the model can adapt to various scenarios and can easily be modified to add features.

Concluding remarks

Python with its potential support for data science provided a fairly simple environment to build a model that predicts epidemic spread by considering all parameters that affect the break out on the population under observation. The model is designed to simulate an environment largely like real and predict concisely how the disease spreads. The model produces a plot that shows approximately the peaks and flattening of the curve. The model can be used to analyze the effect of all the epidemic parameters and to examine effectiveness of measures to contain the spread.

ACKNOWLEDGEMENTS

The authors can acknowledge professor, friend or family member who help in research work in this section. First and foremost, I would thank God for bestowing upon me this opportunity, good health and an environment supportive enough to complete this fellowship.

I wish to sincerely and whole-heartedly thank Dr. Manoj Varma, Center for Nano Science and Engineering, Indian Institute of Science, Bangalore for agreeing to guide me remotely. I am extremely grateful for his continuous support and guidance through the course of the fellowship. He equipped me with a structured plan to follow in order to build the model and had frequent checks on my progress. His feedback, suggestions and ideas were what made the project what it is. I am glad that this topic was chosen as the basis of the fellowship as it has opened newer windows for exploration. His knowledge, expertise, promptness in identifying a problem and quickness in offering a solution inspires me.

I would wish to convey my heartiest thanks to Indian Academy of Sciences for offering the Summer Research Fellowship, which has been a great opportunity for me to venture newer fields and learn so many new concepts. I would especially thank the academy for letting us continue t I extent my gratitude to the Coordinator of Science Education Program, Mr. C.S. Ravi Kumar for his constant help. Sincere thanks for being so accessible and patiently responding to our innumerable concerns, thereby easing our way through the fellowship. I would like to thank Mr. M.R.N. Murthy, Chairman, Joint Science Education Panel for allowing all summer fellows continue the fellowships from the comfort of our homes. the provision to work remotely alone made this dream come true.

I would like to thank my friends' and family's unparalleled support and love that drove me to complete this project. Their motivation and positive feedback brightened my spirits and helped me have a delightful learning experience.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

VI. REFERENCES

[1] The need for data science in epidemic modelling: Comment on: "Mathematical models to characterize early epidemic growth: A Review" by Gerardo Chowell et al.

[2] Outbreak analytics: a developing data science for informing the response to emerging pathogens Jonathan A. Polonsky,Amrish Baidjoe,Zhian N. Kamvar,Anne Cori,Kara Durski,W. John Edmunds,Rosalind M. Eggo,Sebastian Funk,Laurent Kaiser,Patrick Keating,Olivier le Polain de Waroux,Michael Marks,Paula Moraga,Oliver Morgan,Pierre Nouvellet,Ruwan Ratnayake,Chrissy H. Roberts,Jimmy Whitworth and Thibaut Jombart

[3] Model versions and fast algorithms for network epidemiology Petter Holme.

[4] Mathematics of Epidemics on Networks, From Exact to Approximate Models-Appendix A.Authors: István Z. Kiss, Joel C. Miller, Péter L. Simon

[5] An agent-based approach for modeling dynamics of contagious disease spread- Liliana Perez [1], Suzana Dragicevic

[6] Ashlynn R. Daughton, Nicholas Generous, Reid Priedhorsky, Alina Deshpande, 2017, Erratum: Corrigendum: An approach to and web-based tool for infectious disease outbreak intervention analysis, Scientific Reports, vol. 7, no. 1

[7] Human mobility patterns predict divergent epidemic dynamics among cities, Proceedings of the Royal Society B: Biological Sciences, vol. 280, no. 1766, pp. 20130763 Stephen P. Ellner, 2013

[8] https://tradebrains.in/law-of-small-numbers/#:~:text=The%20law%20of%20small%20numbers%20explains%20the%20Judgmental%20bias%20which,of%20observations%20or%20sample%20data.

[9] https://numpy.org/

[10] https://matplotlib.org/

[11] https://www.scipy.org/

[12] https://github.com/kiteco/python-youtube-code/tree/master/virus-spreading-visualization

[13] https://www.youtube.com/embed/WyfpwxW2ze4?rel=0&enablejsapi=1