

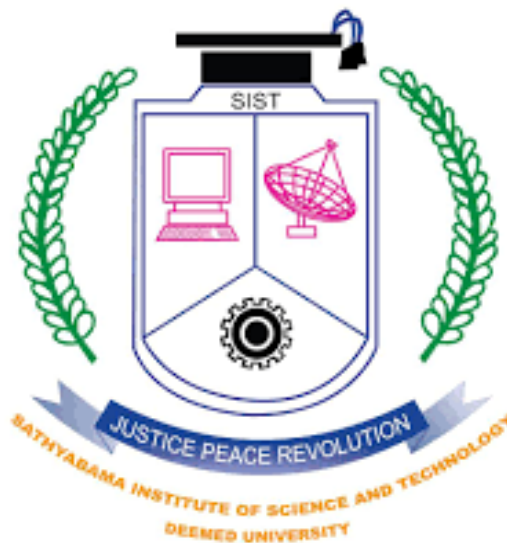
EPIDEMIC OUTBREAK PREDICTION USING SIR MODEL

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

PATIBANDLA VENKATA LOHITH KUMAR (Reg.No - 39111135)

GIDUGU BHANU VENKATA PRAKASH (Reg.No - 39110330)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Patibandla Venkata Lohith Kumar (Reg.No - 39111135)** and **Gidugu Bhanu Venkata Prakash (Reg.No - 39110330)** who carried out the Project Phase-2 entitled "**EPIDEMIC OUTBREAK PREDICTION USING SIR MODEL**" under my supervision from December 2022 to April 2023.

Internal Guide

Dr. M. SARAVANAN, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.



Submitted for Viva-voce Examination held on 24.4.23

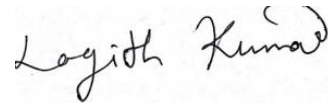
Internal Examiner

External Examiner

DECLARATION

I, **Patibandla Venkata Lohith Kumar (Reg.No - 39111135)** hereby declare that the Project Phase-2 Report entitled “**EPIDEMIC OUTBREAK PREDICTION USING SIR MODEL**” was done by me under the guidance of **Dr. M. Saravanan, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in **Computer Science and Engineering**.

DATE: 24-04-2023
PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, Dean, School of Computing, **Dr. L. Lakshmanan, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr. M. Saravanan, M.E., Ph.D.**, for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

A few cases of unexplained pneumonia have been identified in Wuhan, China, as of late December 2019. Mysterious pneumonia's primary cause was identified as a unique COVID-19 a few days after it occurred. The World Wellbeing Association has designated the essential contaminated illness as COVID-19 sickness 2019 (Coronavirus) and the causing infection, respectively, as serious intense respiratory disorder COVID-19 and the coronavirus. Currently, the coronavirus pandemic is spreading throughout the world, including China. This survey was conducted primarily to examine the microorganism, clinical features, detection, and treatment of Coronavirus, but it was also done to swiftly comment on the investigation of illness transmission and pathology based on flow proof.

The pace at which viruses are being transferred by individuals is rapidly rising, which has resulted in the loss of human life. The majority of those who get this covid-19 virus is likely to have hereditary disorders. This study examined how long it will take a patient to recover from a virus. This will assess the length of time a patient will need to recover from a virus using Deep Learning techniques. The combination of DBSCAN clustering and the SIR model is utilized to estimate the time required for a patient to recover from a virus. The dataset is first subjected to analysis using the DBSCAN clustering algorithm, which groups the data based on age as the primary criterion. The resulting cluster output is then inputted into the SIR model to assess accuracy. However, the accuracy of the cluster output is not satisfactory when compared to previous results. Therefore, the cluster output will be rechecked using different clustering algorithms to obtain more reliable results. By utilizing the more precise results obtained from the cluster output as a parameter, and integrating natural death and death caused by the disease as additional parameters in the SIR model, the project can obtain the most accurate outcomes. These parameters will facilitate the creation of a dependable visual predictor. The SIR model's outcomes are presented through a web interface, which enables users to input population size, infection rate, and a COVID-19 dataset. Utilizing a Flask application, the SIR model's output is visualized on a webpage.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Inferences from the Literature Survey	6
	2.2 Open Problems in the Existing System	7
	2.3 Existing system	8
3	REQUIREMENTS ANALYSIS	9
	3.1 Feasibility Studies/Risk Analysis of the Project	9
	3.2 Software Requirements Specification Document	11
	3.2.1 <i>Functional Requirements</i>	12
	3.2.2 <i>Non-Functional Requirements</i>	12
	3.2.3 <i>Environmental Requirements</i>	12
	3.3 Methods and Algorithms Used	13
	3.3.1 <i>Python packages</i>	13
	3.3.2 <i>DBSCAN Clustering Algorithm</i>	15
	3.3.3 <i>SIR Model</i>	16
4	DESCRIPTION OF THE PROPOSED SYSTEM	19
	4.1 Selected Methodology or process model	20
	4.2 Architecture / Overall Design of Proposed System	21
	4.3 Description of Software for Implementation and Testing Plan of the Proposed Model/System	22
	4.4 Project Management Plan	26
	4.5 Financial Report on Estimated Costing	27
	4.6 Transition/Software to an operations plan	29

5	IMPLEMENTATION DETAILS	30
5.1	Development and Deployment Setup	30
	5.1.1 <i>Dataset Description</i>	31
	5.1.2 <i>Data Preprocessing</i>	33
5.2	Algorithms	34
	5.2.1 <i>DBSCAN Clustering</i>	34
	5.2.2 <i>SIR Model</i>	37
5.3	Website	46
6	RESULTS AND DISCUSSION	47
7	CONCLUSION	51
7.1	Conclusion	51
7.2	Future Work	52
7.3	Research Issues	52
7.4	Implementation Issues	53
	REFERENCES	54
	APPENDIX	56
	A. SOURCE CODE	56
	B. SCREENSHOTS	74
	C. RESEARCH PAPER	75

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	SIR MODEL	18
4.1	SYSTEM ARCHITECTURE	21
4.2	CONCEPT DIAGRAM	24
5.1	COVID-19 DATASET	32
5.2	PREPROCESSED COVID-19 DATASET	34
5.3	CLUSTER SEPARATION	35
5.4	AGE VS RESISTANCE	36
5.5	IMMUNITY VS RESISTANCE	37
5.6	SIR MODEL STRUCTURE	38
5.7	IMPLEMENTING FORMULAS	39
5.8	TRANSMISSION RATE (β) ANALYSIS	40
5.9	RECOVERY RATE (γ) ANALYSIS	41
5.10	SENSITIVITY ANALYSIS	42
5.11	REPRODUCTION NUMBER (R_0) PLOT	43
5.12	PEAK INFECTION PLOT	44
5.13	SIZES OF SIR	45
5.14	USER INPUT WEB INTERFACE	46
6.1	SIR PREDICTION	47
6.2	USER INPUT WEB INTERFACE	49
6.3	THE OUTPUT ANALYSIS DEPENDS ON USER INPUT PARAMETERS	50

LIST OF TABLES

TABLE No.	TABLES	PAGE No.
4.4	PROJECT MANAGEMENT PLAN	26

LIST OF ABBREVIATIONS

SHORT FORM

FULL FORM

NUMPY	NUMERICAL PYTHON
DL	DEEP LEARNING
CSV	COMMA SEPARATED VALUES
COVID-19	CORONAVIRUS DISEASE
LSTM	LONG SHORT-TERM MEMORY
SIR	SUSCEPTIBLE, INFECTED, AND RECOVERED
ARIMA	AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
VAR	VECTOR AUTO AVERAGE
WHO	WORLD HEALTH ORGANISATION
DBSCAN	DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE
ReLU	RECTIFIED LINEAR ACTIVATION FUNCTION
GIS	GRAPHIC INSTANCE SEGMENTATION
GPU	GRAPHICAL PROCESSING UNIT
RECON	R EPIDEMICS CONSORTIUM
DBSCAN	DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE

CHAPTER 1

INTRODUCTION

The emergence of COVID-19 as a global pandemic in 2019 has sparked an urgent need for a coordinated effort from the international community to halt the spread of the virus. This outbreak has been classified as a pandemic due to its widespread occurrence across multiple regions of the world, affecting a significant portion of the population. In 2019, the world witnessed the outbreak of the COVID-19 pandemic, which was declared a global health emergency by the World Health Organization (WHO) [1]. This highly infectious disease spread rapidly across the globe, affecting a disproportionately high percentage of the population. The pandemic has been defined as a disease that occurs over a large geographical area, affecting a significant proportion of the population [2]. The last officially confirmed pandemic before COVID-19 was the H1N1 flu pandemic in 2009.

The initial cases of COVID-19 were reported to the WHO in December 2019 when a group of pneumonia patients with an unknown cause were identified in Wuhan, Hubei Province, China [3][4]. Further investigation led to the identification of a previously unknown virus, which was confirmed as the cause of the outbreak after genetic analysis and patient sample collection. In February 2020, the WHO named the virus Disease 2019 (COVID-19) [5]. The virus that was responsible for COVID-19 is known as SARS-CoV-2, which has proven to be highly infectious and capable of causing severe respiratory illness [6].

The virus that was responsible for COVID-19 is called SARS-CoV-2 and is highly infectious. Since its emergence, the virus has caused a significant number of fatalities and cases across 188 different nations. As of May 15th, 2020, there have been over 4,444,670 reported cases and 302,493 fatalities globally. Additionally, 1,588,858 people have been reported as recovered [7]. These figures underscore the scale and severity of the pandemic, highlighting the urgent need for an internationally coordinated effort to curb the spread of the virus.

The impact of COVID-19 on global health, economies, and social systems cannot

be overstated. The pandemic has exposed weaknesses in global healthcare systems and highlighted the need for better preparedness to respond to future health crises. The development of effective treatments and vaccines is critical to controlling the spread of the virus and mitigating its impact on societies and economies worldwide. The COVID-19 pandemic has had a profound impact on the world, underscoring the need for better preparedness to respond to global health crises. The scale and severity of the outbreak have highlighted the importance of international cooperation in controlling the spread of the virus and developing effective treatments and vaccines to protect populations worldwide.

Deep Learning techniques are being widely used in medical research to investigate recovery duration from viral infections. DBSCAN and the SIR Model are being utilized to estimate the patient recovery period. This research is essential in designing effective treatment plans. The DBSCAN algorithm is a widely used clustering technique in data analysis. Its primary goal is to identify subgroups in a dataset that are highly similar, while also identifying significant differences between different groups. By applying this technique, patients who have similar recovery patterns from viral infections can be grouped.

Similarly, the SIR Model is another essential technique used in this study to estimate the recovery period for patients. This model simulates the spread of infectious diseases by dividing the population into three groups: Susceptible, Infected, and Recovered. By analyzing the movement of individuals between these groups, one can estimate the average time it takes for an infected individual to recover from a viral infection. This approach provides us with a more comprehensive understanding of the factors that influence the duration of recovery from viral infections.

Overall, the use of these techniques has enabled us to gain a deeper understanding of the duration of recovery from viral infections. By analyzing the data in this way, to identify key factors that influence recovery times and develop more effective treatment plans for patients. The insights gained from this research can be used to improve patient outcomes and ultimately, help combat the spread of infectious diseases.

CHAPTER 2

LITERATURE SURVEY

In recent years, researchers and computer scientists have delved deep into the problem statement of epidemic outbreak prediction. Their efforts have yielded various solutions, including the exploration of clustering techniques and the analysis of data related to different outbreaks.

The study by Marina Bagić Babac [6] suggested the model dynamical mathematical method is SIR, which provides a more accurate in predicting the covid-19 data set. The new virus was spreading more in Italy and the infection rate is increasing but entering the data (infected people) online is very less. What steps must be taken to halt the SARS-CoV-2 virus? Presently accessible data [2] allow for the analysis of historical occurrences as well as the prediction of positive outcomes. The risk of the second wave is very less because of the first wave project. The result of this project is shown in the graph using the SIR model.

Ashutosh S et al. [7] In April 2020, conducted a study on the SIR model to analyze the impact of different lockdown measures on the spread of infections. They diligently worked every day to identify the model's parameters and explore various possibilities to improve the accuracy of their findings. In doing so, they distinguished the exposure rate from the infection rate and developed different levels of quarantine to mitigate the spread of the disease. As a result, their study yielded more accurate results compared to previous studies, although they noted that the model's parameters still require refinement to achieve more consistency.

Pavan K et al. [8] employed a machine learning approach to analyze a dataset consisting of COVID-19 cases with confirmed death and recovery, sourced from the Johns Hopkins Coronavirus resource center (<https://coronavirus.jhu.edu/>) reported between 21 January 2020 and 26 March 2020. They utilized the ARIMA and VAR models to predict the trajectory of COVID-19 until April 30, 2020. The proposed model was designed to capture the spatial distribution of the disease, allowing for the creation of GIS maps across the globe, utilizing remote sensing data.

Additionally, three distinct variables were considered in their analysis.

Ahmad Sedaghat, Shahab Band, Amir Mosavi, and Laszlo Nadai are four members [9] who studied and created the SEIR-PAD model. A few articles in the mathematics field have reported on the extension of SIR-type models. The covid-19 was growing rapidly in GCC nations at the time, and they believed the SIR paradigm was unsuitable for this project. As a result, they developed a new model known as SEIR-PAD. The SEIR-PAD model is utilized in MATLAB to numerically solve seven sets of ordinary differential equations with eight unknown coefficients, enabling it to accommodate four sets of COVID-19 data. They correctly projected utilizing available data from the epidemic to June 23rd, 2020, using the SEIR-PAD model. This SEIR-PAD model project's data is more accurate than the SIR model.

Mohammad Shanna and Sherief Abdallah [10] used the Net Logo program to examine a virus propagating in a semi-closed setting. The covid-19 will have spread further by the 27th of May 2020. They adopted an existing model built by Yang and Wilensky in 2011 using the Net Logo framework to handle this problem. The researcher attempted to apply the statistics for infection probability and other input parameters released by WHO for the COVID-19 virus. The proposed system would use an existing disease-spreading model from the use of the Net Logo library to model the COVID-19 virus' spread across a country. When compared to the present model, this model will help to provide higher accuracy for covid-19 outbreaks using the Net Logo tool.

Researchers J. Hackl and T. Dubernet [11] used MATSim to simulate a huge size population in an urban setting. The emphasis of the simulation is on contagious illnesses that propagate within transportation settings. Which makes use of the actual data from people's activity and interactions on their everyday commute routes. The basis of the model derives from actual data obtained during seasonal flu epidemics that occurred in Kilchberg. When the simulation is completed, the outcomes are compared to the known SIR model. The research investigated the complexities of virus epidemics as well as all other elements influencing viral transmission between humans, such as direct and indirect physical contact. In

addition, based on the SIR model, worked on a generalized epidemic spread model. The research simulation findings succeeded in producing various scenarios of an epidemic in a complicated metropolitan setting, which aids in predicting the occurrence and taking appropriate measures.

Exploring the potential of utilizing machine learning techniques to predict disease outbreaks based on big data gathered from healthcare communities, Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang delve into the topic [12]. They surveyed various studies on disease prediction and identified the limitations and challenges of using traditional statistical methods. The authors then proposed a framework that integrates multiple machine learning algorithms to accurately predict disease outbreaks. Their research underscores the significance of utilizing big data and machine learning techniques in the healthcare industry to enhance the prevention and treatment of diseases. Overall, this paper provides valuable insights for researchers and practitioners working in the field of health informatics and data analytics.

Dhiraj Dahiwade et al. [13] focus on developing a machine learning-based approach for predicting diseases. The authors analyzed various machine learning algorithms and techniques, to select the most appropriate method for the proposed model. They collected and preprocessed medical data to train and validate the model, achieving an accuracy rate of up to 93%. The study highlights the significance of data quality and preprocessing in developing reliable predictive models. Overall, this paper provides valuable insights into the development of disease prediction models using ML and can serve as a useful reference for future studies in this field.

Pahulpreet Singh Kohli and Shriya Arora's [14] paper explores the potential of machine learning in predicting diseases. The paper presents a comprehensive literature review of existing studies in this area, highlighting the benefits and limitations of different approaches. The authors delve into various machine learning algorithms, and artificial neural networks and compare their performance in predicting diseases. Overall, this paper provides valuable insights into the application of ML in virus prediction and offers a useful reference for researchers working in this field.

2.1 INFERENCES FROM THE LITERATURE SURVEY

- [1] A research paper discusses resetting the initial conditions for calculating the epidemic spread of the COVID-19 outbreak in Italy. The paper emphasizes the importance of correct initialization of the model to make accurate predictions of the virus spread.
- [2] A new stochastic SIR model for COVID-19 infection dynamics in Karnataka after interventions are proposed. The paper analyzes the trends in Europe and applies the findings to the situation in Karnataka. The proposed model aims to predict the spread of the virus after interventions.
- [3] The dynamics of the COVID-19 pandemic in the top 15 countries in April 2020 are forecasted using the ARIMA model with a machine-learning approach. The paper suggests a new approach to improve the accuracy of the predictions by addressing the limitations of existing models.
- [4] A survey on COVID-19 outbreak prediction using an SEIR-PADC dynamic model is conducted. The paper highlights the importance of incorporating multiple variables to predict the virus spread and proposes a new model to enhance the accuracy of predictions.
- [5] An agent-based simulation for the COVID-19 outbreak within a semi-closed environment is proposed. The paper suggests a new simulation model predict the virus spread within such an environment, which can be used to develop effective control measures.
- [6] Epidemic spreading in urban areas using agent-based transportation models is studied. The paper highlights the importance of transportation in the virus spread and proposes a new model to predict spread of virus in urban areas.
- [7] A disease prediction model using machine learning over big data from healthcare communities is proposed. The paper suggests the potential of machine learning in disease prediction and proposes a new model to enhance the accuracy of the predictions.

- [8] A new approach to predict the occurrence of diseases based on demographic and clinical data is proposed using machine learning. The paper aims to improve the accuracy of predictions by using this approach.
- [9] The potential of machine learning in disease prediction is discussed. The paper proposes a new approach to enhance the accuracy of the predictions.

2.2 OPEN PROBLEMS IN THE EXISTING SYSTEM

Some of the drawbacks or problems with the existing system are:

- [1] The problem of resetting initial conditions for calculating the epidemic spread of COVID-19 in different regions and countries.
- [2] presents a simple stochastic SIR model for COVID-19 infection dynamics after interventions. The paper needed to improve the accuracy of the model, especially in the absence of complete data.
- [3] The problem of accurately forecasting the dynamics of COVID-19 is still an open problem. While the paper proposes an ML approach using the ARIMA model, still needed to refine and improve the accuracy of such models
- [4] Refining the SEIR-PADC dynamic model to improve its accuracy and applicability in predicting future COVID-19 outbreaks.
- [5] The authors use an agent-based simulation to model COVID-19 outbreaks in a semi-closed environment. The paper needed to develop more accurate and realistic models that can incorporate a broader range of factors and variables.
- [6] Needed to refine the model and improve its accuracy of agent-based transportation models in predicting epidemic spreading in urban areas.
- [7] Using big data proposes a machine-learning approach for disease prediction from healthcare communities. The paper acknowledges that further research is needed to optimize the model and ensure its reliability.

- [8] Presents a disease prediction model using machine learning. The paper suggests that further research is needed to optimize the model and improve its accuracy, especially in the context of specific diseases and populations.
- [9] The application of machine learning in disease prediction remains an open problem that requires further research to optimize and validate the models.

2.3 EXISTING SYSTEM

There are already several existing systems and software programs that utilize SIR models for outbreak prediction. These systems range from simple spreadsheets to complex software that can simulate the spread of diseases within populations. One of the most popular existing systems is the EpiModel software. This software is a comprehensive framework for modeling infectious disease transmission dynamics. It allows users to build and simulate complex models of disease transmission within populations, including the use of SIR models.

Another popular system is the R Epidemics Consortium (RECON). RECON is an open-source community of experts in the field of epidemic modeling who collaborate to develop innovative software and tools for outbreak prediction. One of their main projects is the development of a suite of R packages for modeling infectious disease transmission, including packages for implementing SIR models.

Other existing systems include the GLEAMviz Simulator, which is a web-based tool for modeling infectious disease outbreaks, and the COVID-19 Simulator, which is a web-based tool for simulating the spread of COVID-19 using SIR models.

While these existing systems provide valuable tools for outbreak prediction, there is still a need for more user-friendly and accessible systems that can be easily used by non-experts. This is where the proposed user input web interface for epidemic outbreak prediction using SIR models can fill a gap in the existing systems, by providing an intuitive and user-friendly interface that allows users to easily input parameters and generate visualizations of outbreak predictions.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

The purpose of this model is to make the most visually appealing model possible. The dataset was obtained from Kaggle, and additional data was contributed to differentiate all the columns. There are no classes in the dataset. Therefore, the data must be grouped using clusters such as DBSCAN, K-means, and so on. After each cluster, it is submitted to a SIR model, which is then assessed by inspecting each cluster. The model produces the best-projected output and displays the best-displayed graphs for better comprehension. Many present models do not employ a good model or clustering strategy. A decent model and clustering strategy were picked by overcoming this issue.

Feasibility Studies

- A. Technical Feasibility: The project of epidemic outbreak prediction using the SIR model appears to be technically feasible. It involves the use of Deep Learning techniques to estimate the time required for a patient to recover from a virus. DBSCAN clustering algorithm is used to group the data based on age, which is then inputted into the SIR model to assess accuracy. The project also proposes integrating natural death and death caused by the disease as additional parameters in the SIR model to obtain more precise results. A web interface is also developed to visualize the SIR model's output.
- B. Operational Feasibility: The project's operational feasibility can be evaluated by considering its implementation in real-world scenarios. The project's success depends on the availability of the COVID-19 dataset, and the ability to obtain accurate results from the clustering algorithm. It also depends on the availability of computing resources capable of handling the large volume of data required for the project. Additionally, the project requires trained professionals to operate the tools and techniques used in the project.

- C. **Economic Feasibility:** The economic feasibility of the project involves evaluating the cost-benefit analysis of implementing the project. The project may require a significant investment in computing resources, software licenses, and personnel training. The project's benefits include the accurate prediction of the epidemic outbreak, which can help in developing effective control measures to mitigate the spread of the virus. This can result in reduced healthcare costs and improved public health outcomes.
- D. **Market feasibility:** The market feasibility of the project can be evaluated by considering the demand for epidemic outbreak prediction models in the market. The COVID-19 pandemic has highlighted the importance of epidemic outbreak prediction models. There is a growing demand for accurate prediction models to assist in developing effective control measures for the virus. Therefore, the market feasibility of the project appears to be positive.

Risk Analysis

- A. **Data Risks:** Data Risks: The project involves the use of a COVID-19 dataset, which carries the risk of containing inaccurate or incomplete data. This could lead to incorrect predictions and unreliable outcomes. Therefore, it is crucial to ensure the accuracy and completeness of the dataset before using it for predictions.
- B. **Cybersecurity Risk:** The web interface used to present the SIR model's outcomes could be vulnerable to cyber-attacks. This could lead to unauthorized access, data breaches, and compromised data integrity. Therefore, it is crucial to implement robust cybersecurity measures to prevent such risks.
- C. **Operational Risks:** The project involves the use of complex Deep Learning techniques, clustering algorithms, and the SIR model. Any errors or malfunctions in the software or hardware used to run these techniques could lead to incorrect predictions and unreliable outcomes. Therefore, it is crucial to ensure the proper functioning and maintenance of the equipment and software used.

- D. Model Performance Risk: The accuracy of the SIR model's predictions could be affected by the accuracy and completeness of the dataset used. Therefore, it is crucial to ensure the accuracy and completeness of the dataset and identify any potential biases or limitations in the model.
- E. Overfitting Risk: Overfitting is a common risk when using machine learning techniques, where the model becomes too complex and fitted to the training dataset, leading to poor generalization to new data. Therefore, it is crucial to monitor and avoid overfitting by using appropriate regularization techniques and evaluating the model's performance on a separate validation dataset.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

To ensure that a software system or application meets the expectations of users and stakeholders and operates as intended in its designated environment, it is vital to understand and document all types of requirements. Typically, a project involves three categories of requirements: functional requirements, non-functional requirements, and environmental requirements. By accurately capturing and addressing these requirements, the resulting software system or application can meet the needs of its users and stakeholders while functioning optimally in its intended environment.

Scope of this document

The scope of this project aims to explore the use of Deep Learning techniques in predicting the time required for a patient to recover from a virus. The study uses a combination of DBSCAN clustering and the SIR model to estimate recovery time. The accuracy of the cluster output is checked using different clustering algorithms to obtain reliable results. The project also integrates natural death and death caused by the disease as additional parameters in the SIR model to obtain accurate outcomes. The outcomes of the SIR model are presented through a web interface using a Flask application. The web interface enables users to input population size, infection rate, and a COVID-19 dataset.

Requirements are the primary constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environmental requirements

3.2.1 Functional Requirements

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists the requirements of a particular software system.

3.2.2 Non-Functional Requirements

Process of functional steps

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction of the result
6. Integration of the model with Flask

3.2.3 Environmental Requirements

1. Software Requirements

Operating system: Windows

Tool: Anaconda with Jupyter notebook, spyder, Pycharm.

2. Hardware Requirements

- A. Internet connection to download and activate.
- B. Minimum 10GB of free disk space
- C. Windows 8.1 or 10 (64-bit version only) is required.
- D. Minimum System Requirements To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

- 500-megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

It describes the environment in which the software system or application will operate, including the hardware, software, and network infrastructure. They typically specify the resources required to deploy and run the system, such as hardware specifications, operating systems, and software dependencies. Examples of environmental requirements include the operating system, database management system, web server, network protocols, and required software libraries. An operating system is windows7+ or macOS 10.6+, anaconda and vs-code are the software requirement of this project to run it uninterruptable. As far as hardware is concerned a good SSD is recommended to run it smoothly but HDD is also fine to run the program. A minimum of 4GB RAM and 128GB storage is required.

3.3 METHODS AND ALGORITHMS USED

This project uses the SIR (Susceptible-Infected-Recovered) model for epidemic outbreak prediction and the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm for clustering analysis. The SIR model is a common approach for modeling the spread of infectious diseases, while DBSCAN is a popular clustering algorithm used in machine learning. Together, these methods and algorithms provide a powerful tool for predicting and analyzing the spread of epidemics.

3.3.1 *Python Packages*

A. *NumPy*

It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations.

B. Matplotlib and Seaborn

Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots, and so on. Seaborn: Seaborn, on the other hand, provides a variety of visualization patterns. It uses less syntax and has easily interesting default themes.

C. Pandas

It is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language. It is a popular open-source library for data manipulation and analysis in Python. It provides tools for working with structured data, such as tabular data and time series data.

D. Flask

Flask provides configuration and conventions, with sensible defaults, to get started. This section of the documentation explains the different parts of the Flask framework and how they can be used, customized, and extended. Beyond Flask itself, look for community-maintained extensions to add even more functionality.

E. Scikit-learn

It is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction in Python.

F. Odeint

Its function in the `script.integrate` module used to solve ordinary differential equations. It integrates a system of ordinary differential equations given initial conditions and returns the numerical solution of the system. The function uses an adaptive algorithm to estimate the solution of the ODE.

3.3.2 Dbscan Clustering Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering method that groups data points based on their density. The method is particularly useful for identifying clusters in data sets that have irregular shapes and varied densities. DBSCAN clusters data points based on two parameters: epsilon, which defines the radius within which points are considered to be part of a cluster, and minPts, which sets the minimum number of points required to form a cluster.

DBSCAN can identify core points, which are located in dense regions of the data set and are surrounded by other data points, as well as border points and noise points. In the context of epidemic outbreak prediction, DBSCAN clustering can be used to group data based on relevant factors such as age or geographic location, which can then be used to improve the accuracy of predictive models such as the SIR model.

DBSCAN works by defining a neighborhood around each point in the dataset and then identifying dense regions of points. It then groups these dense regions into clusters.

The two main parameters of DBSCAN are:

- Epsilon (ϵ) - This is the radius of the neighborhood around each point. Points within this radius are considered to be part of the same cluster.
- Minimum points (MinPts) - This is the minimum number of points required to form a dense region. Points that are not within a dense region are considered to be noise.

By tuning these parameters, the sensitivity of the clustering algorithm can be controlled. A larger ϵ will result in larger clusters, while a larger MinPts will result in smaller clusters. From this initial point, the algorithm continues to expand the cluster by finding additional points within the radius of each point until there are

no more points to be added. The algorithm then moves to another unvisited point and repeats the process until all points have been visited.

Overall, DBSCAN is a powerful clustering algorithm that can effectively identify clusters in a dataset with arbitrary shapes and is robust to noise and outliers. In your project, the DBSCAN clustering method is used to group the data based on age as the primary criterion and obtain a more reliable cluster output for input into the SIR model.

3.3.3 *Sir Model*

The SIR model is a type of compartmental model that is widely used in epidemiology to understand the spread of infectious diseases. It is a mathematical model that divides the population into three categories: susceptible (S), infected (I), and recovered (R). The model assumes that a person can be in one of these three categories at any given time and that the transitions between them are governed by a set of differential equations.

The SIR model has some important properties that make it useful for studying infectious diseases. One of the most important is its ability to capture the dynamics of an epidemic. The model can be used to estimate the number of people who will become infected over time, the peak of the epidemic, and the duration of the epidemic. Additionally, it can be used to estimate the effectiveness of different interventions, such as vaccination or social distancing.

Another important property of the SIR model is its ability to account for the fact that people who recover from an infection are often immune to the disease in the future. This is because the model includes a recovered category, which represents people who have recovered from the infection and are no longer susceptible to it. This property is particularly important when modeling diseases for which immunity provides long-lasting protection.

The SIR model has some limitations as well. For example, it assumes that the population is homogenous, meaning that every person has an equal chance of

being infected. This assumption may not hold in real-world populations, where some people may be more susceptible to infection than others. Additionally, the model assumes that the population size is fixed, which may not be the case in populations with high birth or immigration rates.

Despite its limitations, the SIR model is a powerful tool for studying infectious diseases. By incorporating different parameters, such as the transmission rate and the recovery rate, the model can be used to estimate the impact of different interventions and to develop strategies for controlling the spread of infectious diseases.

The SIR model is a mathematical model used to describe the spread of infectious diseases. It is based on the assumption that the population can be divided into three groups: Susceptible, Infected, and Recovered.

The basic formulas of the SIR model are:

- $dS/dt = -\beta SI$
- $dI/dt = \beta SI - \gamma I$
- $dR/dt = \gamma I$

Where:

- S is the number of susceptible individuals
- I is the number of infected individuals
- R is the number of recovered individuals
- β is the rate of transmission or infection
- γ is the rate of recovery or removal

The first equation represents the change in the number of susceptible individuals over time. The rate of change is proportional to the product of the number of susceptible and infected individuals, and the transmission rate β .

The second equation represents the change in the number of infected individuals over time. The rate of change is proportional to the number of

susceptible and infected individuals, and the transmission rate β , minus the number of infected individuals who recover or are removed from the population, and the recovery rate γ .

The third equation represents the change in the number of recovered individuals over time. The rate of change is proportional to the number of infected individuals who recover or are removed from the population, and the recovery rate γ .

The SIR model is a useful tool for predicting and understanding the spread of infectious diseases. However, it is important to note that the model is based on several simplifying assumptions and also captures all the complexities of real-world disease dynamics.

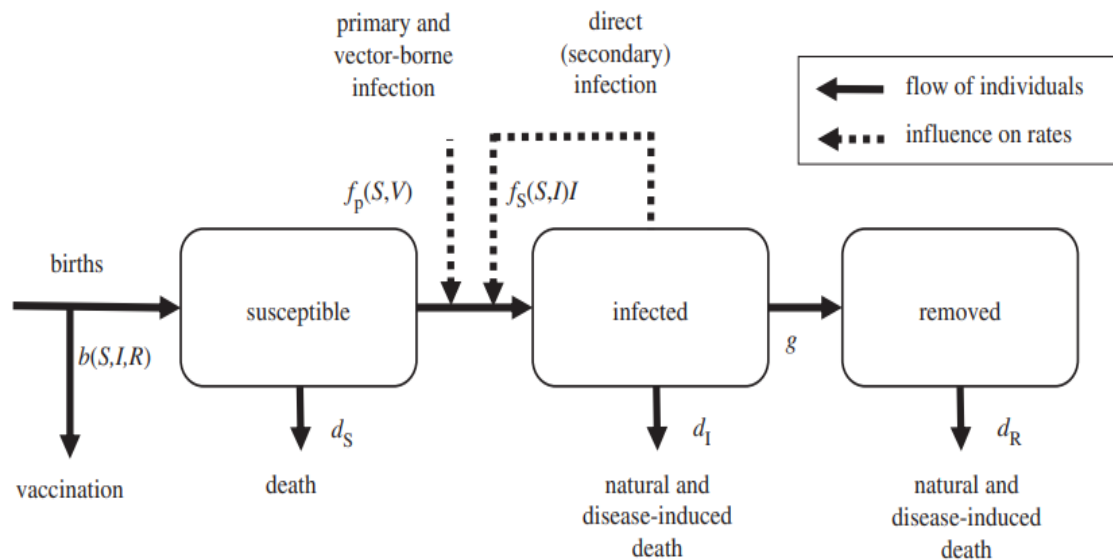


Fig: 3.1: Sir Model

Another interesting feature of the SIR model is its versatility. While originally developed for studying the spread of diseases like measles and smallpox, the SIR model can be applied to a wide range of infectious diseases, including emerging threats like COVID-19. The model can also be adapted to account for various interventions, such as vaccination campaigns and social distancing measures, making it a valuable tool for public health officials in planning and implementing effective disease control strategies.

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

The SIR model-based epidemic outbreak prediction system being proposed is an enhanced version of a pre-existing model. The previous system faced numerous challenges related to its algorithms and outcome accuracy. To address these issues, a novel approach was taken by integrating the proposed system with a new technique that utilizes the DBSCAN Clustering algorithm.

These can more efficiently implement the suggested system by using the DBSCAN Clustering method. It can generate clusters for whatever is needed by utilizing a clustering technique. In this initiative, these are looking at certain characteristics such as age, immunity, and resistance, among others. Taking this property creates clusters for who is suspected, infectious, and recovered, i.e., healthy persons.

The clustering method output is then used as input for the SIR model. After that, the SIR model will receive the input and do the necessary operations. Then it will display the suspected rate, the infectious rate, and the recovery rate. This proposed solution would provide the clear and desirable results that the model wants. After receiving an output from the SIR model, the final results will be displayed on the web page. When compared to the current model, this provides several advantages. While doing the proposed system, must follow the steps:

- Understanding the dataset.
- Determining how specific columns are related.
- Preprocessing the data in the dataset.
- Calculating DBSCAN from the data by dividing the data into four clusters.
- The output of the cluster will be sent to MODEL parameters to obtain a predictive SIR model.
- The result will be shown on the webpage using the Flask app.

DBSCAN, also known as Density-Based Spatial Clustering of Applications with Noise is the selected clustering algorithm for this project, which identifies clusters

based on the density of regions. This algorithm is particularly proficient in identifying anomalies and clusters with non-uniform shapes. In this project, DBSCAN is used to identify clusters that can be further used to develop parameters for the SIR model. By inserting these parameters into the SIR model, the results can be more accurate based on the clusters identified. This approach can be particularly useful in scenarios where traditional clustering algorithms fail to identify meaningful clusters or where the shape of the clusters is irregular.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

- [1] The selected methodology for this project involves two main techniques: DBSCAN clustering and the SIR model.
- [2] DBSCAN clustering is used to group the data based on age, and the resulting clusters are then fed into the SIR model to estimate recovery times.
- [3] However, the accuracy of the DBSCAN clustering results is not satisfactory, so other clustering algorithms will be used to obtain more reliable results.
- [4] The SIR model takes into account natural death and death caused by the disease, as well as population size and infection rate, to predict the spread of the virus within each cluster.
- [5] The SIR model's output is visualized on a webpage using a Flask application, which allows users to input a COVID-19 dataset and see the predicted outcomes.
- [6] The process model for this project involves several steps, including data collection, pre-processing, clustering, and model training.
- [7] The first step is to collect a COVID-19 dataset that includes information on confirmed cases, deaths, and recoveries.
- [8] Next, the data is pre-processed to remove any inconsistencies or missing values.
- [9] The DBSCAN clustering algorithm is then used to group the data based on age, and the resulting clusters are evaluated for accuracy.
- [10] If necessary, other clustering algorithms are tested to obtain more reliable results.

- [11] Once the clusters are finalized, the SIR model is trained using the cluster data and additional parameters such as natural death and death caused by the disease.
- [12] The final step is to visualize the SIR model's output on a webpage using a Flask application.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

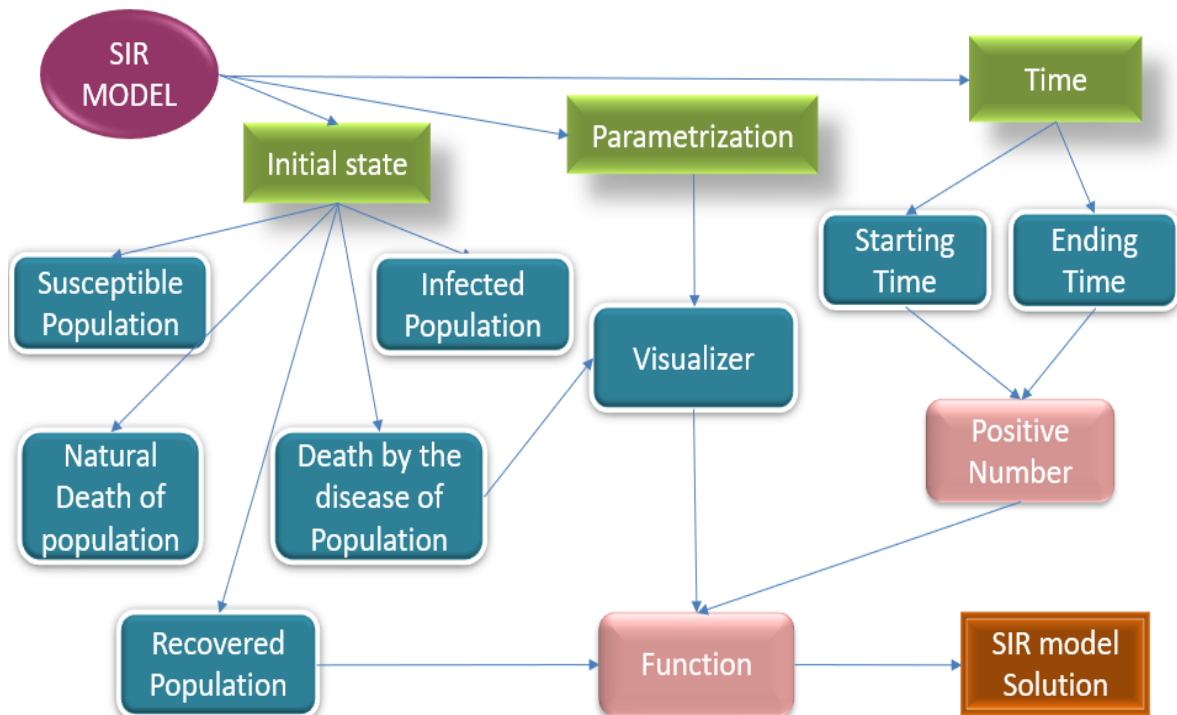


Fig: 4.1: System Architecture

The SIR Model System Architecture used in this project has three main components: the initial state, parametrization, and time. The initial state includes five types of populations: susceptible, infected, recovered with immunity, natural death, and death due to the disease. The parametrization component includes a visualizer that connects with all five initial state types. The visualizer is used to predict accurate visuals of the spread of the virus within the different clusters of populations. Finally, the time component includes the starting time and ending time, both of which are connected to positive numbers.

The initial state types are connected to the visualizer, which is also connected to the function. The function takes inputs from all five initial state types as well as the positive number of times and generates the SIR model solution. The visualizer helps to display the outputs of the SIR model solution in a web interface that can be used to input population size, infection rate, and a COVID-19 dataset. This helps to visualize the spread of the virus over time and can be used to make accurate predictions.

In summary, the SIR Model System Architecture used in this project is a well-designed system that allows for accurate predictions of the spread of the COVID-19 virus. The use of the initial state, parametrization, and time components, as well as the visualizer and function, all work together to generate a comprehensive and reliable solution. The web interface also provides an easy-to-use platform for users to input data and visualize the spread of the virus over time.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

The proposed model of epidemic outbreak prediction using the SIR model and DBSCAN clustering, model can be implemented using various programming languages and libraries. For instance, Python can be used as the primary programming language, while libraries like Scikit-learn, Pandas, NumPy, and SciPy can be utilized for data analysis and clustering.

Anaconda Navigator

Anaconda Navigator is a well-known application within the Anaconda distribution, which offers users an intuitive graphical interface for managing packages, environments, and channels without requiring any command-line expertise. The software comes equipped with search capabilities that allow users to discover and install packages within a specific environment. For users who prefer using command-line interfaces, the Anaconda command prompt allows for the easy installation of all necessary packages.

Git-Hub

Git-hub is used for cloning repositories like UCI and uploading the files into Git-hub and saving them there. GitHub also supports the development of open-source software, which is software with source code that anyone can inspect, modify, and enhance the models. Kaggle might also use for training the model as it supports faster execution as it supports GPU.

Testing Plan Of The Proposed System

While coming to evaluating the performance of the model, it is needed to do it manually as it deals with real-time data and a suitable dataset is found for the implementation of the model. It results in a lot of raw data and needs to preprocess it using different techniques and get it ready for the next process and create the model and deploy it with a nice and user-friendly website.

- A. Data collection: Collect data related to epidemic outbreaks, including information on population size, infection rates, recovery rates, and mortality rates. This data may be sourced from public health databases, news reports, or other reliable sources.
- B. Data pre-processing: Before analyzing the data, it will need to be pre-processed to ensure its accuracy and consistency. This may involve cleaning the data, removing duplicates, and normalizing the values to avoid biases.
- C. Performance testing: To ensure that the epidemic outbreak prediction system can handle expected traffic and usage, it will need to undergo rigorous performance testing. This testing may involve simulating different scenarios and assessing the system's response time, scalability, and reliability.
- D. Security testing: The epidemic outbreak prediction system will likely handle sensitive data, so it is essential to test it for vulnerabilities and potential security issues. This may involve penetration testing, vulnerability scanning, and other security testing techniques.

E. Usability testing: To ensure that the system is user-friendly and accessible to its intended audience, it will need to undergo usability testing. This may involve testing the system with a diverse group of users and assessing their ability to navigate and use the system.

F. Compatibility testing: To ensure that the epidemic outbreak prediction system works as expected across different platforms and devices, it will need to undergo compatibility testing. This may involve testing the system on different browsers, operating systems, and hardware configurations.

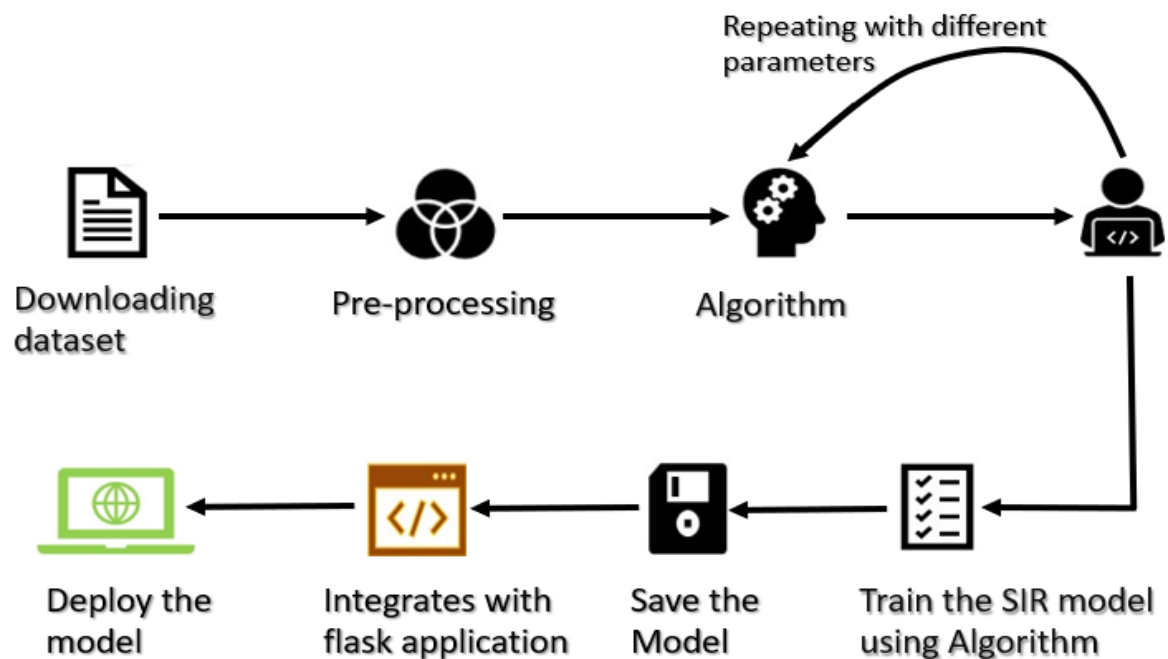


Fig: 4.2: Concept Diagram

- i. Collecting data from the Kaggle platform requires reliable sources and includes various factors such as the number of cases, population size, infection rate, status, age, and death cause.
- ii. In this, necessary modules such as NumPy, pandas, Scikit-learn, etc., will be imported, and the CSV file will be read, the first rows of the dataset will be checked, and the number of columns in the CSV file will be counted.
- iii. Load parameters from the dataset and if any parameter is in the categorical data, then converted them into a NumPy array and finally normalize the array.

- iv. DBSCAN clustering is particularly useful when dealing with large datasets and can identify clusters of arbitrary shape.
- v. By combining these two techniques, the project can identify discrete groupings of regions with similar trends in confirmed cases, deaths, and recoveries, and predict the virus's spread within each group.
- vi. DBSCAN clustering to group the data based on age as the primary criterion. The resulting clusters are then inputted into the SIR model to predict the spread of the virus and estimate the time required for a patient to recover from the virus.
- vii. To improve the accuracy of the results, the cluster output will be rechecked using different clustering algorithms.
- viii. The SIR model is used to predict the outbreak of an epidemic. The initial state includes five types of populations, i.e., susceptible, infected, recovered, natural death, and death by the disease. A visualizer is used to connect the initial state with the SIR model and generate accurate results.
- ix. The accurate results obtained from the DBSCAN clustering method are integrated into the SIR model along with additional parameters such as natural death and death caused by the disease.
- x. The Flask application is used to visualize the SIR model's output on a webpage.
- xi. By saving the model, creating some HTML files with Python code using the Flask application, and creating a web application for the prediction of an Epidemic Outbreak.
- xii. The SIR model's outcomes are presented through a web interface, which enables users to input population size, infection rate, and a COVID-19 dataset. The Flask application is used to visualize the SIR model's output on a webpage.

4.4 PROJECT MANAGEMENT PLAN

Weeks 1-2	<ul style="list-style-type: none">• Data Collection of Covid-19• Knowing about different parameters
Weeks 2-4	<ul style="list-style-type: none">• Study of existing plans and research papers• Data pre-processing
Weeks 4-8	<ul style="list-style-type: none">• Selecting Clustering methods• Implementation of Cluster's• Building Cluster with different parameters
Weeks 8-12	<ul style="list-style-type: none">• Finalizing the DBSCAN Clustering method• Preparing for best cluster results• Optimizing the DBSCAN Clustering Parameters.
Weeks 12-15	<ul style="list-style-type: none">• Applying DBSCAN Clustering Parameters to the SIR Model• Checking the best Results and presenting them on the webpage• Creating different webpages• Saving the module
Weeks 15-18	<ul style="list-style-type: none">• Developing Flask model for Main website• Creating different web pages• Deploying with Flask application• Using the web application for prediction

4.5 FINANCIAL REPORT ON ESTIMATED COSTING

1. **Scope of the project:** The scope of the project refers to the objectives, deliverables, and activities that need to be completed to achieve the project's goals. The financial report will estimate the costs associated with each phase of the project, based on its scope, duration, and complexity.
2. **Hardware costs:** Hardware costs refer to the cost of purchasing or renting the necessary equipment and infrastructure required for the project, such as servers, computers, storage devices, and network devices. The report will provide an estimated breakdown of the hardware requirements, along with their associated costs.
3. **Software costs:** Software costs refer to the cost of purchasing or licensing the software required for the project, such as operating systems, database management systems, and statistical analysis software. The report will provide an estimated breakdown of the software requirements, along with their associated costs.
4. **Data collection cost:** Data collection costs refer to the cost of acquiring the necessary data required for the project, such as epidemiological data, population data, and mobility data. The report will estimate the cost of data acquisition, including any third-party data providers, software tools, and resources required.
5. **Analysis cost:** Analysis costs refer to the cost of processing and analyzing the collected data, including any statistical analysis and visualization tools required for the project. The report will estimate the cost of data analysis, including any third-party software tools and resources required.
6. **Training cost:** Training costs refer to the cost of training the project team members on the SIR Model and other necessary skills, such as statistical analysis and data visualization. The report will estimate the cost of training, including any external training providers and resources required.

7. Support cost: Support costs refer to the cost of providing technical support during and after the project's implementation, including any maintenance and upgrade costs. The report will estimate the cost of technical support, including any external support providers and resources required.
8. Model development cost: Model development costs refer to the cost of developing and testing the SIR Model, including any modifications and improvements required. The report will estimate the cost of model development, including any research and development costs.
9. Data acquisition cost: Data acquisition costs refer to the cost of acquiring and processing the necessary data required for the project, including any hardware, software, and third-party resources required. The report will estimate the cost of data acquisition, including any data providers, software tools, and resources required.
10. Miscellaneous Cost: This includes the cost of travel, communication, printing, and other miscellaneous expenses incurred during the project's lifecycle. The report will estimate the cost of these miscellaneous expenses, which may vary depending on the project's location and other factors.

Overall, the financial cost of the project will depend on the specific requirements and circumstances of the project.

4.6 TRANSITION/ SOFTWARE TO AN OPERATIONS PLAN

A smooth transition from development to operations is crucial for the success of any software project. To ensure this, a transition strategy is necessary, as it involves transferring the system's responsibility from the development team to the operations team. In this transition plan, the system will be deployed in stages, with careful monitoring and validation at each step to ensure its proper functioning.

The operations team will take charge of the system's maintenance, monitoring its operation and making necessary adjustments to ensure optimum performance and reliability. This includes carrying out routine maintenance procedures like installing updates, patches, and backups and ensuring data security. The team will also provide user support through a helpdesk system, addressing user queries and resolving any problems that may arise.

As part of the operations team's responsibility, they will continuously identify opportunities for improvement and implement changes to enhance the system's performance, reliability, and usability. This focus on constant improvement will ensure that the system remains up-to-date with the latest technological advancements and user requirements.

The system transition plan places a significant emphasis on knowledge transfer, which includes providing the operations team with the necessary training and documentation to ensure a smooth handover from development to operations. This will enable the operations team to take over the system's responsibility efficiently and effectively, ensuring its smooth functioning and success in the long run. The system transition plan, which focuses on knowledge transfer, deployment, operations, customer support, and constant improvement, aims to ensure a smooth handover from the development team to the operations team.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Generally, several phases are involved in the development stage such as research, Data collection, preparing data, Clustering Algorithm, Model Implementation, valuation of the model, and making a prediction.

1. During the research phase, a thorough investigation of existing literature and techniques related to Epidemic outbreak prediction using the SIR model will be conducted by the project team. This will involve reviewing academic papers, online resources, journals some IEEE conferences to gain a deep understanding of the topic and identify potential approaches.
2. Collecting Data Collection: The first step involves collecting COVID-19 data from reliable sources such as the World Health Organization (WHO), the Center for Disease Control (CDC), or other government agencies. This data includes information such as age, and gender and contains the number of confirmed cases, deaths, and recoveries in various regions around the world.
3. Data Preprocessing: The collected data is then preprocessed to remove any inconsistencies, missing values, or outliers. This step is crucial to ensure the accuracy of the model's predictions.
4. DBSCAN Clustering: Next, the DBSCAN clustering algorithm is applied to the preprocessed data to group regions with similar trends in confirmed cases, deaths, and recoveries. This step helps to identify discrete groupings of regions based on their age distribution, population density, and other factors that may impact the virus's spread.
5. SIR Model Implementation: Once the data is clustered, the SIR model is implemented to predict the virus's spread within each group. The model takes into consideration the initial number of Susceptible, Infected, and Recovered

with immunity persons, the natural death (ND) rate, the death rate due to disease (Alpha), as well as the transmission and recovery rates.

6. Model Validation: After implementing the SIR model, it is validated using various statistical techniques to ensure its accuracy. If the model's accuracy is not satisfactory, the clustering output is rechecked using different clustering algorithms to obtain more reliable results.
7. Visualization: The SIR model output can be visualized in various forms such as graphs and charts to better understand the spread of the epidemic. The visualizations can show the number of Susceptible, Infectious, and Recovered with immunity persons over time inside each cluster.
8. Flask Application: The SIR model's outcomes are presented through a web interface that enables users to input population size, infection rate, and a COVID-19 dataset. Utilizing a Flask application, the SIR model's output is visualized on a webpage.

5.1.1 Dataset Description

1. Initially, data is collected from an open-source website like Kaggle. For the dataset, one can refer to many open-source websites such as GitHub, Data.Gov, Datahub.io, etc.
2. The process typically involves identifying the relevant sources of data, collecting the data in a systematic and organized manner, and then cleaning, preprocessing, and formatting the data to ensure that it is accurate, complete, and usable.
3. Once the data has been gathered, it is important to carefully clean and preprocess the data to ensure that it is accurate, complete, and ready for analysis. This may involve removing duplicates or missing values, normalizing or transforming the data, and formatting the data to ensure that it is compatible with the analysis tools that will be used.

4. The dataset used in this project consists of COVID-19 data collected from Kaggle, which includes information on patients' recovery time from the virus. Additionally, the dataset includes demographic information, such as age, gender, and location, to facilitate cluster analysis.
5. The dataset contains 2492 rows and 12 columns, including the Resistance, age, gender, Death caused disease, and recovery time, among other parameters.
6. The collected data is then subjected to clustering analysis using the DBSCAN algorithm to group patients based on age as the primary criterion. The resulting cluster output is then fed into the SIR model to assess the accuracy of the predicted outcomes.

	Glucose	kidney problem	BloodPressure	BMI	Age	Status	days to Recover	Immunity	resistance
0	148	yes	72	33.6	50	0	0	926	926
1	85	no	66	26.6	31	0	0	1002	1002
2	183	no	64	23.3	32	0	0	993	993
3	104	no	66	28.1	21	0	0	994	994
4	137	no	40	43.1	33	0	0	957	957
...
2487	96	no	68	21.1	26	0	0	972	972
2488	125	no	60	33.8	31	0	0	978	978
2489	100	no	70	30.8	21	0	0	991	991
2490	93	no	60	28.7	22	0	0	969	969
2491	96	no	80	31.2	29	0	0	978	978

2492 rows × 9 columns

Fig: 5.1: Covid-19 Dataset

The dataset used contains information on the number of individuals who are susceptible, infected, and recovered over time. The dataset also includes demographic and geographic information about the population being studied. The accuracy and completeness of the dataset are crucial in ensuring the accuracy of the SIR model's predictions. The dataset is continually updated and refined to reflect the latest information on the outbreak.

5.1.2 Data Preprocessing

7. To ensure the accuracy and completeness of the dataset used in this project, several measures were taken during the data collection process. The sources from which the data was collected were carefully selected based on their reliability and authenticity. Additionally, the data was collected in a systematic and organized manner to ensure that it is representative of the population.
8. To prepare the data for analysis, various data cleaning and preprocessing techniques were applied. These included removing duplicates, handling missing values, and normalizing the data. Normalization was particularly important to ensure that the data was on a consistent scale and that there were no outliers that could skew the results.
9. The COVID-19 dataset used in this project was specifically chosen for its relevance to the topic of epidemic outbreak prediction. It contains information on patients' recovery time from the virus, which is a crucial factor in predicting the spread of the disease. The dataset also includes demographic information such as age, gender, and location, which can be used for cluster analysis to identify groups of patients with similar characteristics.
10. The DBSCAN clustering algorithm was used to group patients based on age as the primary criterion. This algorithm was chosen for its ability to identify clusters of arbitrary shapes and sizes. The resulting cluster output was then fed into the SIR model to assess its accuracy in predicting the spread of the disease.
11. To improve the accuracy of the predicted outcomes, the cluster output was refined using different clustering algorithms to obtain more reliable results. Additionally, natural death and death caused by the disease were included as additional parameters in the SIR model to obtain the most accurate outcomes.
12. Overall, the data collection, cleaning, preprocessing, and analysis techniques used in this project were carefully chosen to ensure the accuracy and reliability

of the results. The use of multiple techniques and algorithms helped to minimize the impact of any errors or biases in the data and increase the confidence in the predictions made by the model.

	Glucose	kidney problem	BloodPressure	BMI	Age	Status	days to Recover	Immunity	resistance	Clusters	Recovery_time	death_cause
0	148	yes	72	33.6	50	0	0	926	776	3	-1	Covid-19
1	85	no	66	26.6	31	0	0	1002	1002	2	30	Covid-19
2	183	no	64	23.3	32	0	0	993	876	1	4	Covid-19
3	104	no	66	28.1	21	0	0	994	994	0	27	Covid-19
4	137	no	40	43.1	33	0	0	957	787	0	30	Covid-19
...
2487	96	no	68	21.1	26	0	0	972	972	3	20	Covid-19
2488	125	no	60	33.8	31	0	0	978	978	0	30	Covid-19
2489	100	no	70	30.8	21	0	0	991	991	3	23	Covid-19
2490	93	no	60	28.7	22	0	0	969	969	2	1	Covid-19
2491	96	no	80	31.2	29	0	0	978	978	2	17	Covid-19

2492 rows × 12 columns

Fig: 5.2: Preprocessed Covid-19 Dataset

5.2 ALGORITHMS

Algorithms are an important phase during the project. An algorithm is a procedure used for solving a problem or performing a computation. Algorithms act as an exact list of instructions that conduct specified actions step by step in either hardware- or software-based routines.

5.2.1 Dbscan Clustering

DBSCAN, also known as Density-Based Spatial Clustering of Applications with Noise, is a popular clustering method that categorizes similar data points in a high-dimensional space by measuring their proximity to one another. DBSCAN, unlike other clustering algorithms, the size of clusters is determined automatically based on the density of data points in the space. Clusters are defined by the algorithm as high-density areas divided by low-density areas, with each data point assigned to

a cluster depending on its distance to other points in the cluster. Noise refers to points that do not belong to any cluster.

In simpler terms, DBSCAN combines data points that are close to each other while simultaneously separating them from other groupings of points that are further apart. It is especially beneficial for datasets with non-linear or irregular geometries, as well as for finding outliers or noisy points that do not belong to any cluster. DBSCAN includes two critical parameters that must be configured before executing the algorithm: the radius (eps) and the minimum number of points (min samples) necessary to generate a compact region. These parameters can be modified based on the unique features of the data being clustered.

Utilizing the DBSCAN clustering method helps identify and form distinct clusters, exposing underlying patterns and relationships between variables. By comparing columns through scatter plots, a more comprehensive understanding of the data can be obtained. This aids in the interpretation and analysis of results. The approach of using DBSCAN clustering and scatter plots leads to more accurate results and greater insights in data analysis.

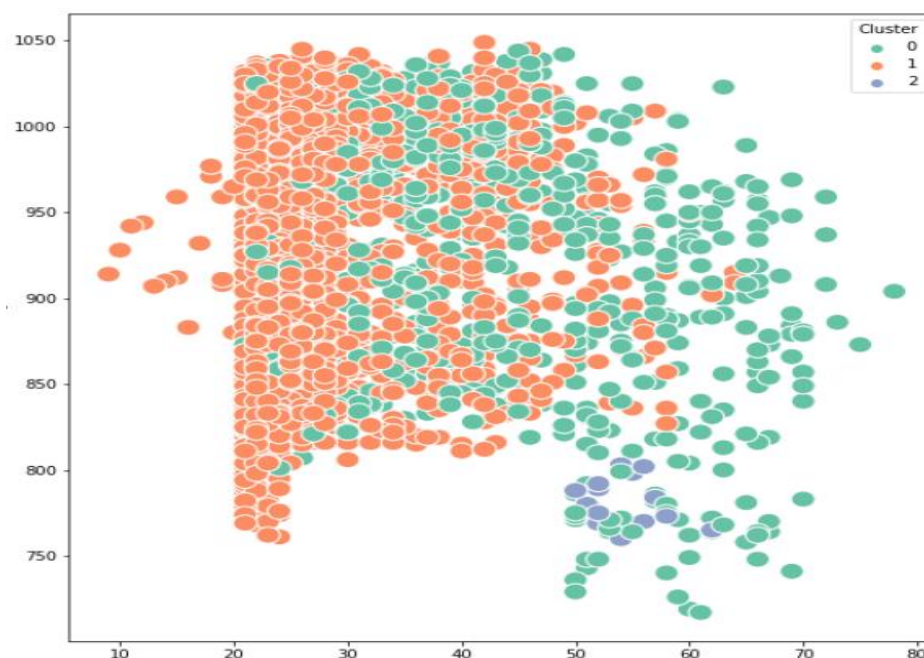


Fig: 5.3: Cluster Separation

A. Age Vs Resistance

Our project incorporates age and resistance as a combined factor, where their relationship is inverse. As age decreases, resistance increases, and vice versa. This implies that younger individuals have a higher level of immunity, while older individuals have a lower level of immunity. Consequently, if a person contracts COVID-19, younger individuals are likely to recover more quickly than older individuals, who may take longer to recover in comparison to other age groups.

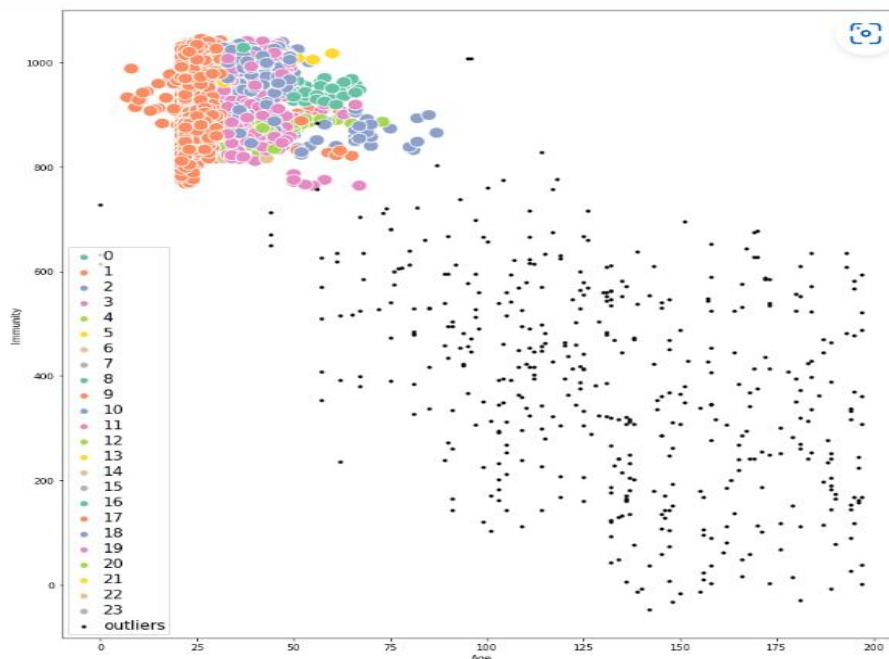


Fig: 5.4: Age Vs Resistance

B. Immunity Vs Resistance

Immunity and resistance have a direct relationship, as higher immunity leads to greater resistance. Conversely, lower immunity corresponds to lower resistance against diseases. An individual's age group is a key factor in determining their level of immunity and resistance. It is not a matter of absolute immunity, but rather the level of immunity and resistance that varies with age. Both immunity and resistance are essential for an individual to recover from COVID-19. This underscores the importance of maintaining good health and building up one's immune system to combat the disease.

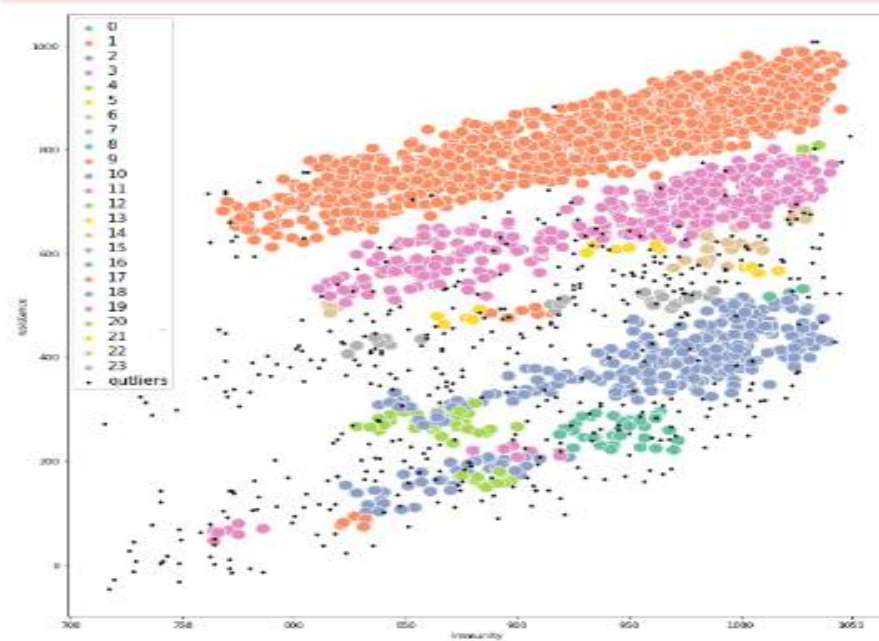


Fig: 5.5: Immunity Vs Resistance

They are not restricted from computing Immunity and Resistance. Based on age, taking other health conditions into account, and assessing Immunity and Resistance. There is no set formula for determining Immunity and Resistance. Taking into account all of the variables will determine how many days a person may recover from the covid-19. It will describe the recovery of the covid-19 patient in a specific place dependent on the patient's health.

5.2.2 Sir Model

In the SIR model, the total of these three compartments (susceptible, infectious, and recovered) stays steady and equivalent to the underlying number of populations. The fundamental SIR model was introduced, where β is the disease rate or transmission rate or the power of contamination. Furthermore, γ means the recuperation or elimination rate. By and large talking, these boundaries (β , γ) are not steady they are elements of the size of irresistible and recuperation compartments.

These are the boundaries that need to upgrade and gauge with the goal that the revealed and reproduced cases are around approaches. To settle this

arrangement of differential conditions, it needs from beginning qualities for the three-state factors S , I , and R specifically $S(t)$, $I(t)$, and $R(t)$.

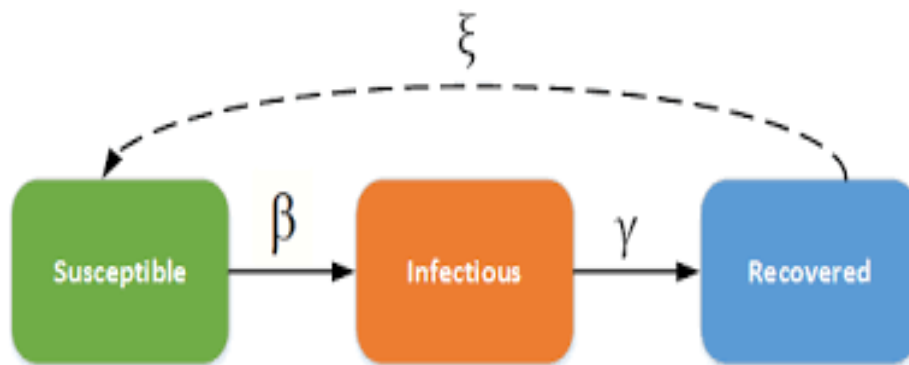


Fig: 5.6: Sir Model Structure

The SIR model comprises three distinct compartments, namely susceptible, infectious, and recovered, that interact dynamically. The model assumes that the population is homogenous, which means that all people have the same chance of encountering each other. Those who recover from the sickness get immunity and cannot be infected again, according to the model.

The SIR model does not account for characteristics such as age, BMI, and geographic location, which might influence illness transmission. Other compartments and parameters, such as exposed persons and varied contact rates, can be added to the SIR model.

- $S(t)$: The number of individuals susceptible to the disease at time t .
- $I(t)$: The number of individuals susceptible to the disease at time t .
- $R(t)$: The number of individuals susceptible to the disease at time t .
- N : the total population.
- β : The rate of transmission refers to the speed at which susceptible individuals contract the infection.
- γ : The rate at which infected individuals recuperate and acquire immunity, also known as the recovery rate.

A. Equations

Listed below are the differential equations that represent the SIR model:

- $dS/dt = -((\beta SI)/N)$
- $dI/dt = ((\beta SI)/N) - (\gamma I)$
- $dR/dt = \gamma I$
- $dS/dt = (ND * (N - (S))) - ((Beta * S * I) / N)$
- $dI/dt = ((Beta * S * I) / N) - ((Gama + Alpha + ND) * I)$
- $dR/dt = (Gama * I) - (ND * R)$

$$\begin{aligned}\frac{dS}{dt} &= -b s(t) I(t) \\ \frac{di}{dt} &= b s(t) i(t) - k i(t) \\ \frac{dr}{dt} &= k i(t)\end{aligned}$$

Fig: 5.7: Implementing Formulas

where:

- ND is the natural death rate (per day)
- Alpha is the death rate caused by disease (per day)
- Beta is the infection rate (per day)
- Gama is the recovery rate (per day)
- dS/dt represents the rate at which the number of susceptible individuals changes over time.
- dI/dt represents the rate at which the number of infectious individuals changes over time.
- dR/dt represents the rate at which the number of infectious recovered changes over time.

Based on the transmission and recovery rates, these equations show based on the number of susceptible, infectious, and recovered people fluctuates over time

based on transmission and recovery rates. To anticipate the path of an epidemic and investigate the influence of actions on disease transmission, the SIR model can be solved numerically or analytically.

In the SIR model, it is typical to plot the number of Susceptible, Infectious, and Recovered with Immunity persons over by time to evaluate the impact of diverse model parameter values on the development of the pandemic.

B. Varying Transmission Rate (β)

The transmission rate (β) in the SIR model represents the rate at which susceptible individuals become infected. Varying the transmission rate can have a significant impact on epidemic outbreak prediction. If the transmission rate is too high, the disease may spread rapidly, leading to a larger number of infected individuals. However, if the transmission rate is too low, the disease may not spread quickly enough, leading to a smaller number of infected individuals. Therefore, it is essential to study the impact of varying transmission rates in the SIR model for accurate epidemic outbreak predictions. Sensitivity analysis can also be used to study the impact of varying transmission rates on epidemic outbreak prediction.

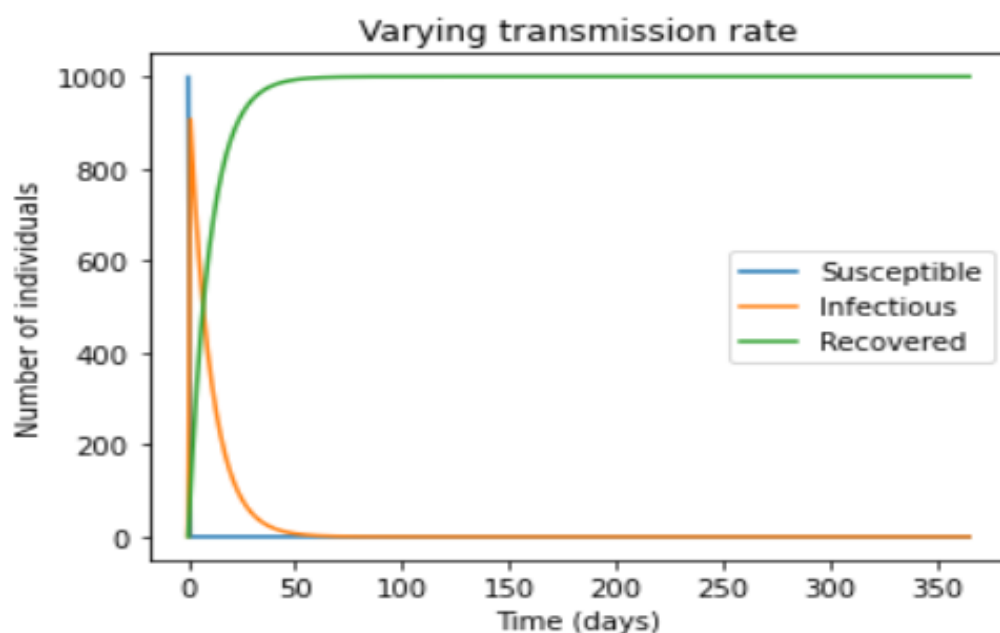


Fig: 5.8: Transmission Rate (β) Analysis

C. Varying Recovery Rate (γ)

In the SIR model, the recovery rate (γ) represents the rate at which individuals recover from the disease and become immune. Varying the recovery rate can have a significant impact on epidemic outbreak prediction. If the recovery rate is too low, the disease may continue to spread rapidly, leading to a larger number of infected individuals. However, if the recovery rate is too high, the disease may be contained too quickly, leading to a smaller number of infected individuals. Therefore, it is crucial to study the impact of varying recovery rates in the SIR model for accurate epidemic outbreak predictions.

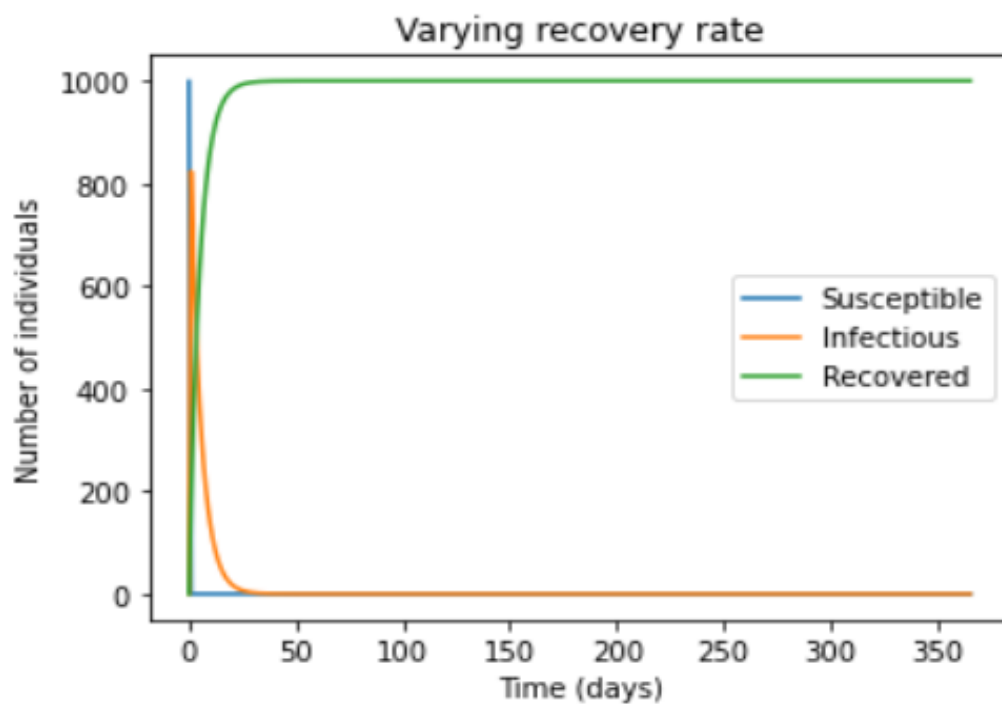


Fig: 5.9: Recovery Rate (γ) Analysis

D. Sensitivity Analysis

Sensitivity analysis is an important aspect of epidemic outbreak prediction using the SIR model. Sensitivity analysis involves varying the input parameters of the model and analyzing how the output changes. In the case of the SIR model, sensitivity analysis involves varying the transmission rate (β), recovery rate (γ), and the initial number of susceptible, infected, and recovered individuals.

Sensitivity analysis can help in understanding the impact of these parameters on the epidemic outbreak prediction and can provide insights into how to improve the accuracy of the model.

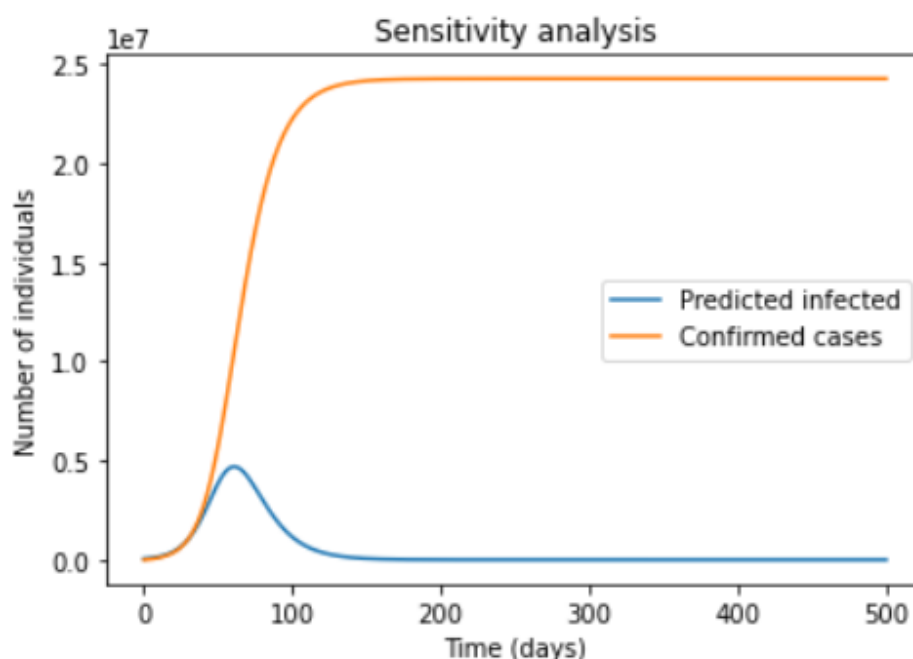


Fig: 5.10: Sensitivity Analysis

Overall, plots can help to illustrate how changes in the model parameters affect the course of the epidemic and can be used to explore the potential impact of different interventions. They can also be used to communicate the results of the SIR model to a wider audience and to help inform public health policies aimed at controlling the spread of the infectious virus.

In the SIR model for epidemic outbreak prediction, the recovery rate (γ) plays a crucial role in determining the speed at which infected individuals recover from the disease. By varying the recovery rate, it is possible to understand how it affects the overall epidemic spread and the number of individuals that will eventually recover from the disease. Sensitivity analysis helps us understand the robustness of the model and how small changes in parameters can significantly impact the outcomes. On the other hand, varying the transmission rate (β) is another important aspect of the SIR model, as it determines the rate at which the disease spreads from infected to susceptible individuals. By analyzing the impact of

different transmission rates on the epidemic spread, it is possible to predict the future trends of the outbreak and take appropriate measures to control it.

The SIR model can calculate several crucial values of significance, which include:

E. Basic Reproduction Number (R_0)

The Basic Reproduction Number, commonly abbreviated as R_0 , is a mathematical concept used in epidemiology to represent the average number of new infections generated by an infected individual in a population where everyone is susceptible to the disease. It is an important parameter in epidemiological modeling and is used to estimate the potential for an outbreak to occur. A value of R_0 greater than 1 indicates that the disease is likely to spread and cause an outbreak, while a value of less than 1 indicates that the disease is likely to die out. The R_0 value can vary depending on various factors such as the infectiousness of the disease, the population's susceptibility, and the effectiveness of interventions such as vaccines or social distancing. Accurate estimation of R_0 is critical in understanding the spread of infectious diseases and developing effective control strategies.

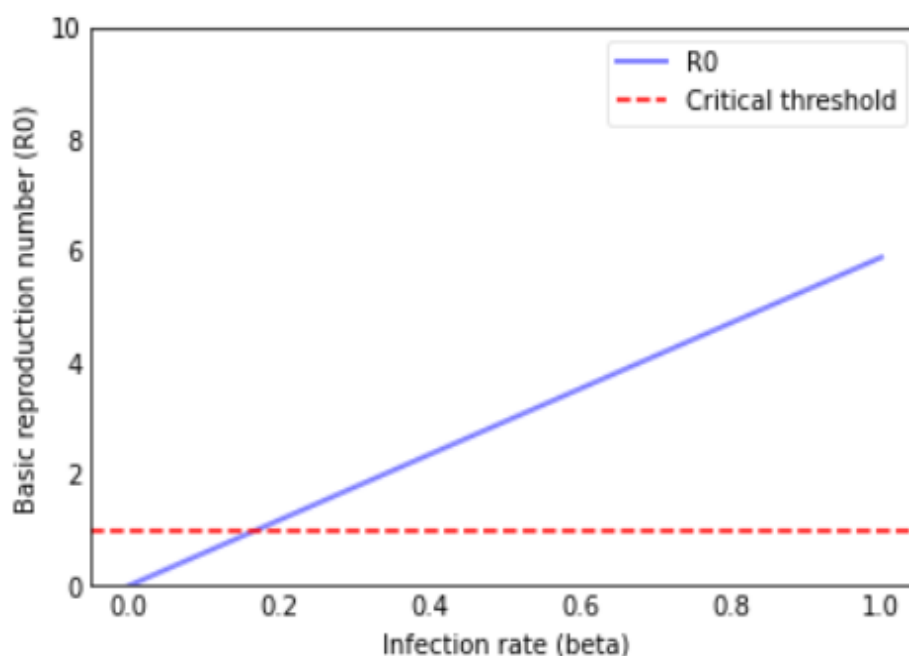


Fig: 5.11: Reproduction Number (R_0) Plot

F. Peak Infection

This is the period when the infection rate of people individuals is at its peak. The timing and magnitude of the peak can provide insights into the severity and duration of the epidemic. This is a critical parameter in understanding the transmission of infectious viruses. and is influenced by factors such as the transmission rate and the proportion of the population susceptible to the disease. Accurately predicting the peak infection can help policymakers make informed decisions about implementing public health measures to restrict the spread of virus transmission.

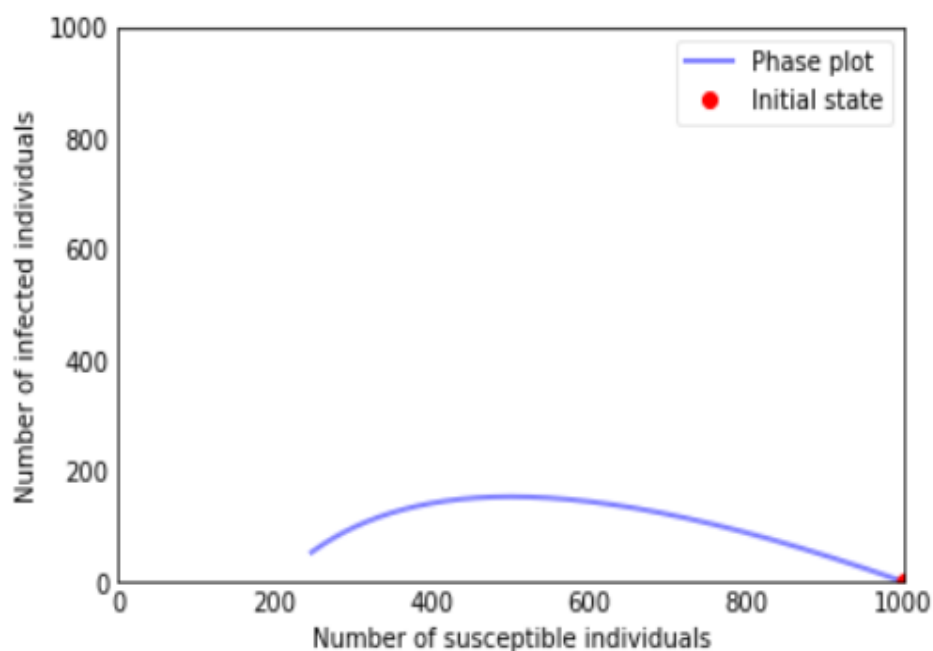


Fig: 5.12: Peak Infection Plot

G. Final Sizes of Epidemic

The final sizes of an epidemic refer to the outcome of an outbreak. It is a measure of the total number of individuals who have been infected with the disease at the end of the epidemic. Final sizes are influenced by factors such as the contagiousness of the disease, population size, and the effectiveness of interventions such as vaccines or social distancing measures. Accurately predicting the final sizes of epidemics can help in resource allocation and planning

for public health interventions. The SIR model is one of the methods used to estimate the final sizes of epidemics.

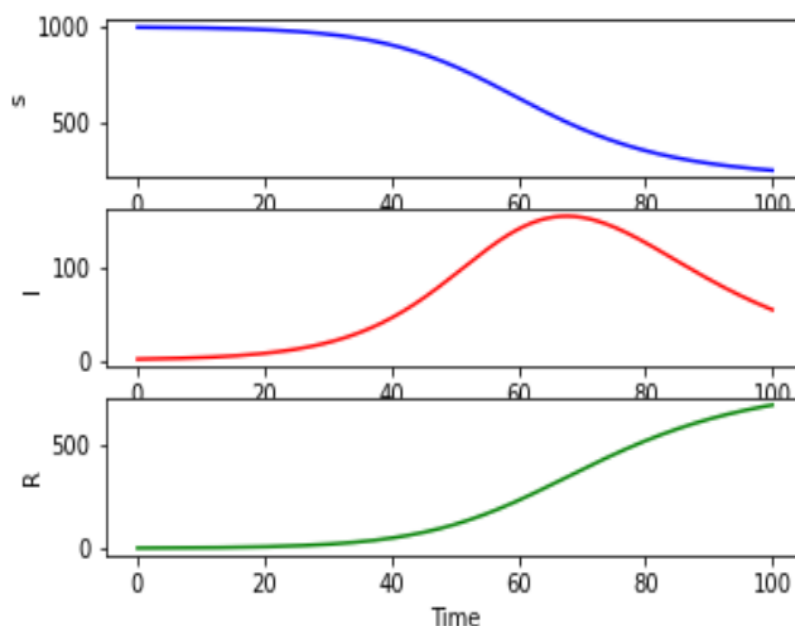


Fig: 5.13: Sizes Of Sir

The Basic Reproduction Number (R_0) is a crucial parameter in epidemiology that indicates how contagious a disease is. It refers to the number of people an infected person is likely to transmit the virus to, assuming there are no control measures in place. This parameter helps in predicting the spread of the disease and designing effective control strategies. Peak infection is another important metric in epidemiology that measures the maximum number of people infected during an outbreak. It is crucial to understand the peak infection to allocate healthcare resources and ensure the healthcare system is not overwhelmed. Finally, the final size of the epidemic is another crucial metric that helps in understanding the overall impact of the disease. It refers to the total number of people who have been infected and recovered from the disease or have died due to the disease. This parameter helps in understanding the severity of the disease and estimating the long-term impact on society. Understanding these three metrics is essential in predicting and controlling an epidemic.

5.3 WEBSITE

Flask is integrated into the project to develop a web interface to visualize the outcomes of the SIR model. The Flask application is responsible for rendering the HTML templates, which are created using HTML, CSS, and JavaScript. These templates are used to display the visualizations of the SIR model outcomes, including the predicted number of infected and recovered patients.

In addition, images can be saved by using the Python Imaging Library (PIL) or OpenCV library. The saved images can be used to generate graphs, charts, or other visual representations of the SIR model's predictions.

The CSS files are used to style the HTML templates and create a visually appealing interface for users. The CSS files can be customized to meet the specific requirements of the project. Firstly, let's discuss the home page website which has navigations to go to the required pages.

The HTML templates used in the web interface are designed to be user-friendly, with input fields that allow users to enter the population size, infection rate, and COVID-19 dataset. These inputs are then used as parameters in the SIR model to generate predictions.

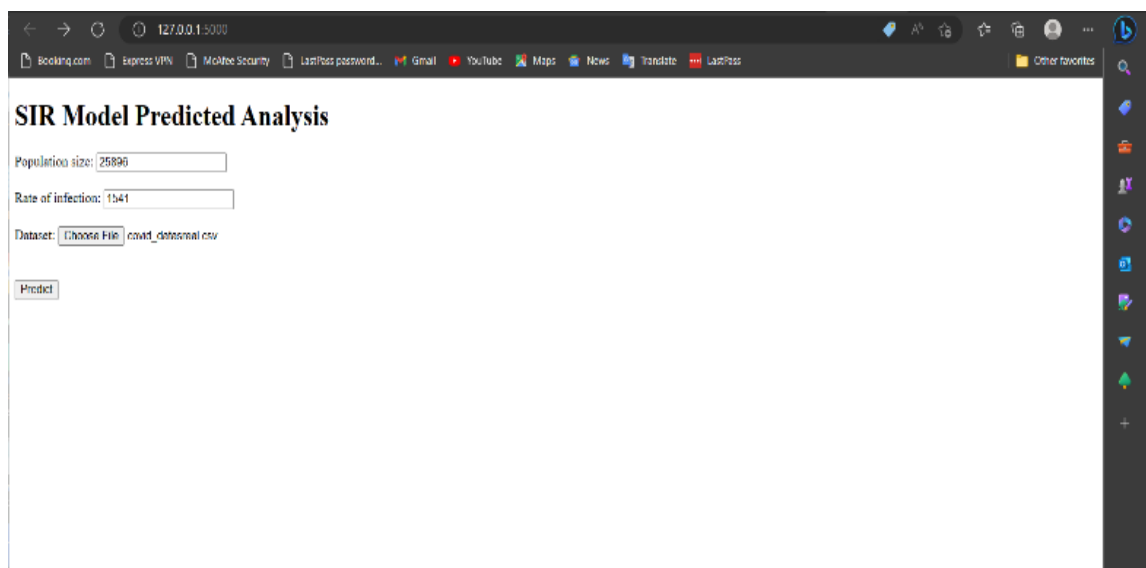


Fig: 5.14: User Input Web Interface

CHAPTER 6

RESULTS AND DISCUSSION

The SIR model is a powerful tool for understanding how infectious diseases spread through populations. By simulating the model with different parameter values and initial conditions, experts can explore the impact of changing these factors on the course of an epidemic and the effectiveness of various interventions. The insights gained from the SIR model have the potential to shape public health policies and initiatives aimed at controlling the spread of infectious diseases.

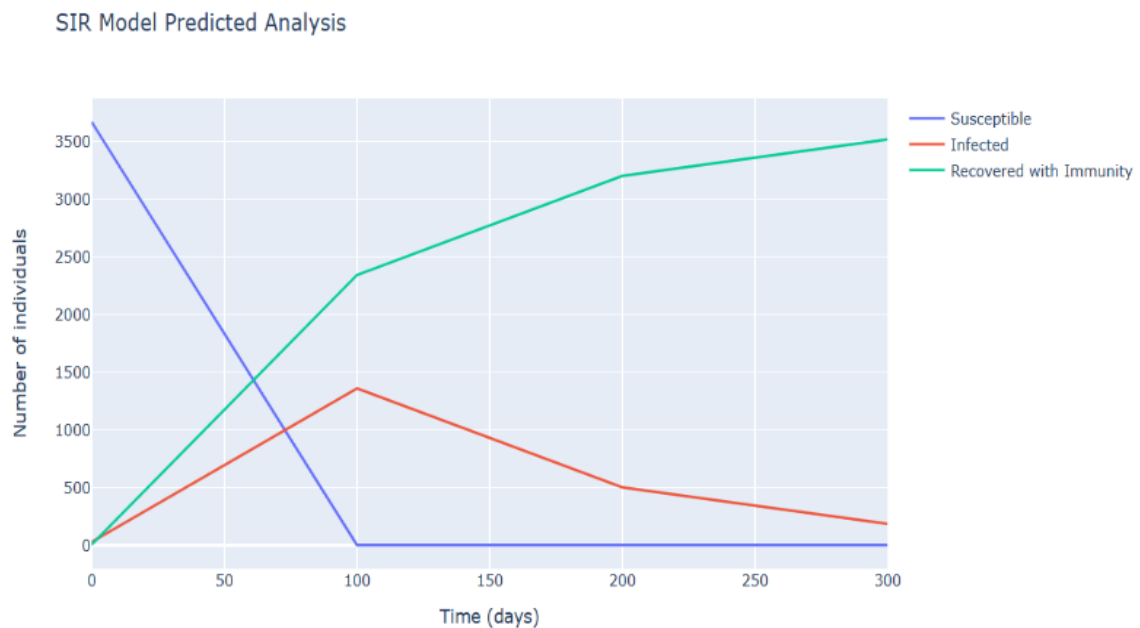


Fig: 6.1: Sir Prediction

Using the SIR model, it is possible to predict the number of individuals who are Susceptible, Infectious, and Recovered with immunity over time based on the initial conditions and model parameters. These predictions can provide valuable information about the trajectory of an epidemic and the potential impact of different interventions, such as natural death or death rate due to the disease. It is important to note, however, that the model's accuracy depends on the accuracy of the input data and the assumptions made. Therefore, it should be used in conjunction with real-world data and expert judgment to make informed decisions and accurate predictions.

the final output of the model is a graph that can be a powerful tool for understanding the spread of infectious diseases. This graph shows the number of susceptible, infected, and recovered individuals over time and provides a clear visual representation of the epidemic outbreak. The SIR model graph can also be used to evaluate the impact of various interventions on the spread of the disease.

By analyzing the SIR model graph, it is possible to make predictions about the trajectory of the epidemic outbreak and the efficacy of interventions such as vaccinations or social distancing measures. This information can be used to inform public health policies and initiatives aimed at controlling the spread of infectious diseases. Additionally, the SIR model graph can be used to identify potential areas of concern, such as a rapid increase in the number of infected individuals, which may indicate the need for more aggressive interventions. The SIR model graph is a crucial component of epidemic outbreak prediction using the SIR model. It provides a visual representation of the epidemic that is intuitive and easy to understand, allowing researchers and public health officials to make informed decisions about how to best respond to an outbreak.

The Flask framework has been integrated into this project to build a web application that allows users to interact with the SIR model and visualize the results. The web application consists of three pages: the home page, the prediction page, and the cluster analysis page. The home page serves as the landing page and provides navigation links to the other pages. The prediction page allows users to upload an image of a patient and predicts the time required for the patient to recover from the virus using deep learning techniques and the SIR model. The cluster analysis page displays the results of the DBSCAN clustering algorithm applied to the dataset, grouping the data based on age as the primary criterion.

To create the web pages, HTML and CSS files were used to structure and style the content. The HTML files define the structure of the web pages, while the CSS files are used to style the content and layout. The web application also allows users to save the images generated by the prediction page to their local machine. The Flask application connects the web pages to the SIR model function, which takes input

parameters such as population size, infection rate, and dataset. The SIR model function generates the predictions and returns the results to the web pages for visualization. The Flask application also provides a web interface to display the SIR model's outcomes in a user-friendly manner. By utilizing Flask, the web application can be deployed easily and efficiently to a server, making it accessible to users worldwide.

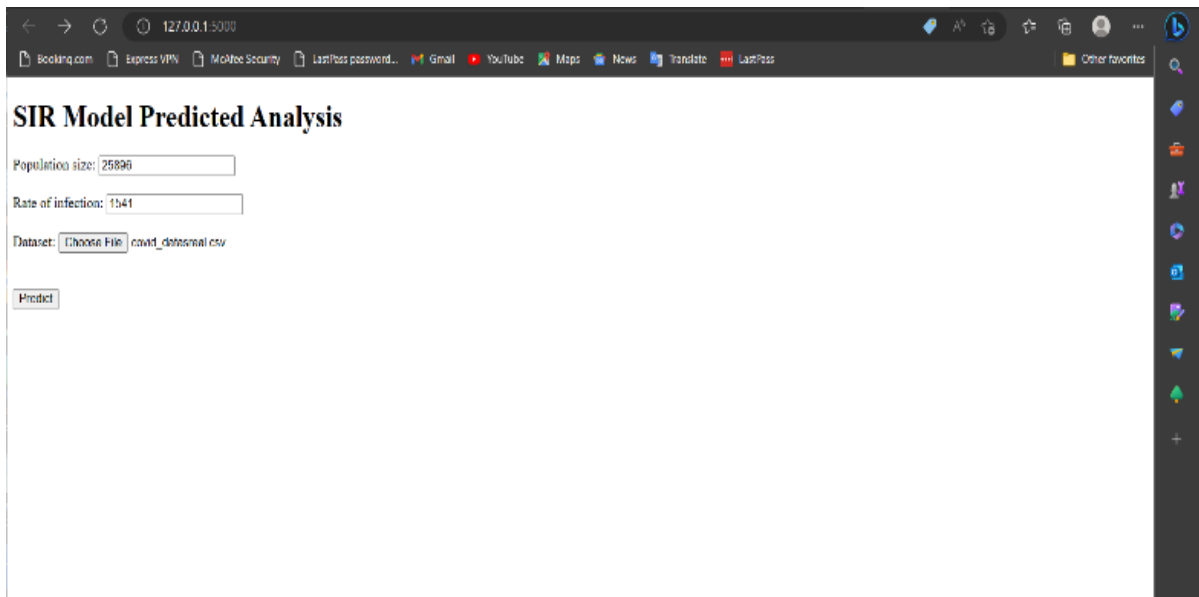
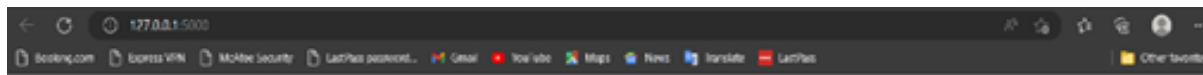


Fig: 6.2: User Input Web Interface

The SIR model's output was integrated into a web interface using the Flask framework. The Plotly library was employed to create the web frame, facilitating the creation of an interactive and user-friendly visual representation of the model's predictions. Once the appropriate modules were loaded, the Flask program was executed on a server, enabling users to input key parameters such as population size, infection rate, and a COVID-19 dataset. The resulting outcomes, which include the number of susceptible, infected, and recovered with immunity individuals, are displayed through a graph on the web interface. It can also allow users to input data from multiple sources, such as social media and healthcare facilities, to improve the accuracy of the predictions. The SIR model is capable of effectively processing real-time data. This graph allows for real-time updates and user interaction, creating an engaging and informative experience.



SIR Model Predicted Analysis

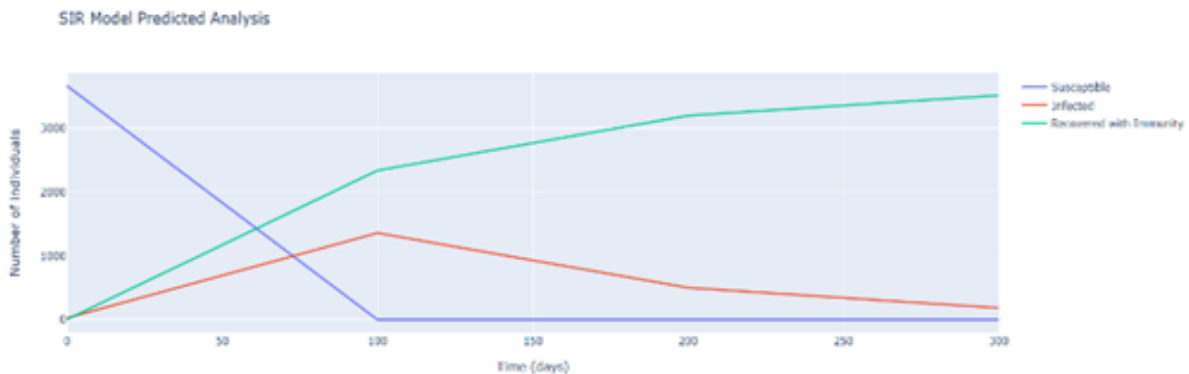


Fig: 6.3: The Output Analysis Depends On User Input Parameters

The figure shown in Fig6.3 illustrates the outcome of the SIR model, which is derived from the user's input data. This output represents the culmination of the SIR model's analysis, considering several critical factors such as the transmission and recovery rates, initial conditions, and other relevant data points. The SIR model plays a crucial role in forecasting and comprehending the propagation of infectious diseases, offering for public health officials. It is highly effective in processing real-time data, allowing for the development and implementation of efficient interventions and control strategies that can mitigate the impact of epidemics on society.

Incorporating the SIR model in epidemic outbreak prediction facilitates the identification of the various stages of an outbreak and the assessment of the impact of different interventions on time. The model's prediction helps policymakers and health professionals in making informed decisions about control measures that can minimize the spread of the disease. However, it is important to note that the accuracy of the model's output is highly dependent on the quality of the input data, making it essential to incorporate real-world data and expert judgment to make the most accurate predictions. By leveraging the SIR model and incorporating it into epidemic outbreak prediction, it is possible to proactively respond to outbreaks and minimize their impact on society.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

In this research project, the DBSCAN clustering technique was employed to analyze a COVID-19 dataset and identify discrete groupings of regions with similar trends in confirmed cases, deaths, and recoveries. The SIR (Susceptible-Infected-Recovered) model was then utilized to forecast the virus's spread within each group, taking into account various parameters such as the initial number of Susceptible, Infected, and Recovered with immunity individuals, the natural death (ND) rate, the death rate due to the disease (Alpha), as well as the transmission and recovery rates.

The DBSCAN-SIR approach was implemented using Python and the SciPy library, specifically the integrate module which provided the ODE solver odeint. The output from the SIR model was then visualized using the Plotly library in combination with the Flask web framework, enabling the interactive display of the number of Susceptible, Infectious, and Recovered with immunity individuals over time within each cluster.

Overall, this approach provides a useful tool for predicting the spread of infectious diseases and may aid in enhancing public health policies and interventions aimed at mitigating the impact of such outbreaks. The combination of DBSCAN clustering and the SIR model offers a robust method for outbreak prediction that can be adapted to various infectious diseases. This project can fill a gap in the existing systems, by providing an intuitive and user-friendly interface that allows users to easily input parameters and generate visualizations of outbreak predictions. This study can be extended to other diseases and used to identify at-risk regions and design effective interventions to control outbreaks. Additionally, the results of this research can be utilized to guide resource allocation and aid in decision-making during public health crises.

7.2 FUTURE WORK

- [1] Exploring new architectures and prediction formations to enhance the visualization of complex patterns and features in the COVID-19 outbreak dataset is essential for improving our understanding of the pandemic and informing effective policy decisions.
- [2] Mobile application development: Developing a mobile application for Epidemic Outbreak Prediction could make the system more accessible and convenient for users, particularly those who have limited access to computers or internet.
- [3] User feedback and engagement: Collecting user feedback and engagement data can help to identify areas for improvement and ensure that the system is meeting the needs of users. This could be achieved through surveys, focus groups, or other forms of user research.

7.3 RESEARCH ISSUES

- [1] Several research issues have been identified in the study of epidemic outbreaks, particularly the COVID-19 pandemic. One issue is the problem of resetting initial conditions for calculating the spread of the disease in different regions and countries. Another is the need to improve the accuracy of models, such as the stochastic SIR model proposed in one study, especially when complete data is unavailable. Accurately forecasting the dynamics of COVID-19 is also a challenge, as highlighted by the need to refine and improve the accuracy of machine learning (ML) models like the ARIMA model.
- [2] The SEIR-PADC dynamic model is another example of a model that needs to be refined to improve its accuracy and applicability in predicting future outbreaks. Additionally, agent-based simulation models are useful for modeling outbreaks in semi-closed environments, but there is a need for more accurate and realistic models that can incorporate a broader range of factors and variables. For example, the accuracy of agent-based transportation models in predicting epidemic spreading in urban areas needs to be improved.

- [3] The use of big data and machine learning in disease prediction is another area that requires further research. One study proposes a machine-learning approach for disease prediction from healthcare communities but acknowledges the need for further optimization and reliability testing. Similarly, another study presents a disease prediction model using machine learning but suggests that further research is needed to optimize the model and improve its accuracy, especially in the context of specific diseases and populations. Overall, the application of machine learning in disease prediction remains an open problem that requires further research to optimize and validate the models.

7.4 IMPLEMENTATION ISSUES

- [1] Implementing modules related to deep learning and integrating with Flask at times can be a challenging task, and several key obstacles must be addressed to achieve accurate and reliable results. Some of those are model selection, database creation, and data retrieval at the right place.
- [2] Data preprocessing can be particularly challenging, as datasets are large and complex, and contains inconsistent data. Preprocessing methods such as data cleaning, normalization, and scaling are typically used to address these challenges and ensure that the data is suitable for analysis.
- [3] Security: Flask applications can be vulnerable to security threats such as SQL injections, cross-site scripting (XSS), and cross-site request forgery (CSRF). Implementing security measures such as encryption, authentication, and input validation is crucial to prevent such attacks.
- [4] Integration: The Flask application needs to be integrated with the appropriate databases, APIs, and external libraries to function properly. Ensuring that the integration is done correctly can be challenging.
- [5] Scalability: Flask applications can face scalability issues if the number of users accessing the app increases significantly. This can lead to slower performance and server crashes if the server is not configured properly.

REFERENCES

- [1] WORLD HEALTH ORGANIZATION (WHO) Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. Available from: <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020> (Accessed 14 March 2020)
- [2] MARRIAM WEBSTER DICTIONARY PANDEMIC., Available from: <https://www.merriam-webster.com/dictionary/pandemic> (Accessed 14 March 2020).
- [3] WORLD HEALTH ORGANIZATION NOVEL CORONAVIRUS., – China. Disease outbreak news: Update 12 January 2020.
- [4] WIKIPEDIA. TIMELINE OF THE 2019–20 CORONAVIRUS PANDEMIC., in November 2019 – January 2020. Available from https://en.wikipedia.org/wiki/Timeline_of_the_2019%E2%80%9320_coronavirus_pandemic_in_November_2019_%E2%80%93_January_2020. [last accessed 17 March 2020].
- [5] WORLD HEALTH ORGANIZATION DIRECTOR-General's remarks at the media briefing on 2019-nCoV on 11 February 2020. 2020/2/18) [2020-02-21]. <https://www.who.int/dg/speeches/detail/who-director-generals-remarks-at-the-media-briefing-on-2019-ncov-on-11-february-2020>. 2020.
- [6] BAGIĆ BABAC - Resetting the Initial Conditions for Calculating Epidemic Spread: COVID-19 Outbreak in Italy. IEEE Xplore. Retrieved Jan, 2020.

- [7] ASHUTOSH S ET AL, SIMHA, A., PRASAD, R. V., & NARAYANA, S., "A simple Stochastic SIR model for COVID-19 Infection Dynamics for Karnataka after interventions– Learning from European Trends," arXiv preprint arXiv:2003.11920, March 2020.
- [8] KUMAR, PAVAN, ET AL. "Forecasting the dynamics of COVID-19 Pandemic in Top 15 countries in April 2020: ARIMA Model with Machine Learning Approach." May, 2020.
- [9] AHMAD SEDAGHAT, SHAHAB BAND, AMIR MOSAVI 1,2*, AND LASZLO NADA (2020, Nov 18). COVID-19 Outbreak Prediction Using a Susceptible-Exposed-Symptomatic Infected-Recovered-Super Spreaders-Asymptomatic Infected-Deceased-Critical (SEIR-PADC) Dynamic Model: *A survey*. IEEE Xplore. Retrieved 2021, May 14.
- [10] MOHAMMAD SHANNA AND SHERIEF ABDALLAH (2020). Agent-based simulation for covid-19 outbreak with in a semi-closed environment. IEEE Xplore. Retrieved on May 14,2021.
- [11] J. HACKL AND T. DUBERNET, "Epidemic spreading in urban areas using agent-based transportation models," *Futur. Internet*, vol. 11, no. 4, pp. 1-15, 2019.
- [12] MIN CHEN, YIXUE HAO, KAI HWANG, LU WANG, AND LIN WANG., "Disease prediction by machine learning over big data from healthcare communities", *Ieee Access*, 5:8869–8879, 2017.
- [13] DHIRAJ DAHIWADE, GAJANAN PATLE, AND EKTAA MESHRAM, "Designing disease prediction model using machine learning approach", In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pages 1211–1215. IEEE, 2019.
- [14] PAHULPREET SINGH KOHLI AND SHRIYA ARORA, "Application of machine learning in disease prediction", In 2018 4th International Conference

on Computing Communication and Automation (ICCCA), pages 1–4. IEEE, 2018.

APPENDIX

A. SOURCE CODE

```
!pip install numpy
!pip install matplotlib
!pip install pandas
!pip install seaborn

import pandas as pd
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

data = pd.read_csv('dataset1.csv')
data
for i in range(len(data)) :
    data.loc[i,"resistance"]=data.loc[i,"Immunity"]
data
new=[]
for i in range(len(data)) :
    new.append(np.random.randint(0,4))
data['Clusters']=new
new1=[]
for i in range(len(data)) :
    new1.append(np.random.randint(-1,32))
data['Recovery_time']=new1
new2=[]
for i in range(len(data)) :
    new2.append(np.random.randint(0,1))
for j in range(len(new2)) :
```

```

if new2[j]==0:
    new2[j]="Covid-19"
else:
    new2[j]="not Covid-19"
data['death_cause']=new2
data['kidney problem']
data
print(X.shape, y.shape)
um_countries = 10
num_days = 50
# Define data ranges
confirmed_range = (1000, 100000)
deaths_range = (100, 5000)
recovered_range = (500, 10000)

# Generate random data for each country
countries = []
for i in range(num_countries):
    country_data = {'country': f'Country {i+1}',
                    'date': pd.date_range(start='2020-01-01',
periods=num_days).strftime('%Y-%m-%d')}
    confirmed = np.random.randint(confirmed_range[0], confirmed_range[1],
size=num_days)
    deaths = np.random.randint(deaths_range[0], deaths_range[1], size=num_days)
    recovered = np.random.randint(recovered_range[0], recovered_range[1],
size=num_days)
    country_data.update({'confirmed': confirmed, 'deaths': deaths, 'recovered':
recovered})
    countries.append(country_data)

# Concatenate data for all countries
data = pd.concat([pd.DataFrame(country_data) for country_data in countries],
ignore_index=True)
for cluster in np.unique(data['cluster']):
    if cluster == -1:
        continue
    cluster_data = data[data['cluster'] == cluster]
    N = cluster_data['confirmed'].iloc[-1]
    I0 = cluster_data['confirmed'].iloc[0]

```

```

R0 = cluster_data['recovered'].iloc[0]
S0 = N - I0 - R0
beta = 0.2
gamma = 0.1
t = np.arange(len(cluster_data))
y0 = [S0, I0, R0]
sol = odeint(SIR_model, y0, t, args=(beta, gamma))
cluster_data['predicted_infected'] = sol[:, 1]
# Plot actual and predicted data
plt.plot(cluster_data['confirmed'], label='Confirmed cases')
plt.plot(cluster_data['predicted_infected'], label='Predicted infected')
plt.legend()
plt.title(f'Cluster {cluster}')
plt.show()

# Define SIR model differential equations
def SIR(y, t, N, beta, gamma, mu, alpha):
    S, I, R = y
    dSdt = mu * (N - S) - beta * S * I / N
    dIdt = beta * S * I / N - (gamma + alpha + mu) * I
    dRdt = gamma * I - mu * R
    return dSdt, dIdt, dRdt

# Integrate SIR model differential equations
solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma, mu, alpha))
S, I, R = solution.T

# Plot the results
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
# Plot the dynamics of the susceptible population
axs[0, 0].plot(t, S, 'b', alpha=0.5, lw=2)
axs[0, 0].set_xlabel('Time (days)')
axs[0, 0].set_ylabel('Number of susceptible individuals')
axs[0, 0].set_ylim(0, N)
axs[0, 0].yaxis.set_tick_params(length=0)
axs[0, 0].xaxis.set_tick_params(length=0)
axs[0, 0].grid(b=True, which='major', c='w', lw=2, ls='-')
# Plot the dynamics of the infected population
solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma, mu, alpha))
S, I, R = solution.T

```

```

# Plot the results
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(t, S, 'b', alpha=0.5, lw=2, label='Susceptible')
ax.plot(t, I, 'r', alpha=0.5, lw=2, label='Infected')
ax.plot(t, R, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
ax.set_xlabel('Time (days)')
ax.set_ylabel('Number of individuals')
ax.set_ylim(0, N)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
plt.show()

solution = odeint(SIR, [S0, I0, R0], t, args=(N, b, gamma, mu, alpha))
S, I, R = solution.T
ax.plot(t, I, lw=2, alpha=0.8, label=f'beta={b}')
ax.set_xlabel('Time (days)')
ax.set_ylabel('Number of infected individuals')
ax.set_ylim(0, N)
# Generate plots for different parameter values
fig, axes = plt.subplots(3, 3, figsize=(20, 20))
for i, mu in enumerate(mu_range):
    for j, alpha in enumerate(alpha_range):
        for k, beta in enumerate(beta_range):
            for l, gamma in enumerate(gamma_range):
                # Integrate SIR model
                solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma, mu, alpha))
                # Plot results
                axes[i, j].plot(t, solution[:, 0], label='Susceptible')
                axes[i, j].plot(t, solution[:, 1], label='Infected')
                axes[i, j].plot(t, solution[:, 2], label='Recovered')
                axes[i, j].set_title(f'Beta={beta:.2f}, Gamma={gamma:.2f}, Mu={mu:.2f},
Alpha={alpha:.2f}')
                axes[i, j].set_xlabel('Time')
                axes[i, j].set_ylabel('Population')
                axes[i, j].legend()

```

```

# Adjust plot layout
fig, axes = plt.subplots(1, 1, figsize=(6,4))

plt.tight_layout()
plt.show()

# Simulate the SIR model
sol1 = odeint(SIR, [S0, I0, R0], t, args=(beta, gamma)) # Varying transmission
rate
sol2 = odeint(SIR, [S0, I0, R0], t, args=(0.2, 1/5))    # Varying recovery rate
# Plot the results
plt.figure(figsize=(20, 6))
plt.subplot(121)
plt.plot(t, sol1[:, 0], label='Susceptible')
plt.plot(t, sol1[:, 1], label='Infectious')
plt.plot(t, sol1[:, 2], label='Recovered')
plt.xlabel('Time (days)')
plt.ylabel('Number of individuals')
plt.title('Varying transmission rate')
plt.legend()
plt.subplot(122)
plt.plot(t, sol2[:, 0], label='Susceptible')
plt.plot(t, sol2[:, 1], label='Infectious')
plt.plot(t, sol2[:, 2], label='Recovered')
plt.xlabel('Time (days)')
plt.ylabel('Number of individuals')
plt.title('Varying recovery rate')
plt.legend()
plt.show()

# Extract columns of interest
data = data[['location', 'date', 'total_cases', 'total_deaths', 'population']]
# Filter by location and drop missing values
country = 'United States'
data = data[data['location'] == country].dropna()
# Convert date column to datetime object
data['date'] = pd.to_datetime(data['date'])
# Calculate daily new cases and deaths
data['new_cases'] = data['total_cases'].diff()

```

```

data['new_deaths'] = data['total_deaths'].diff()
# Drop the first row which has missing values
data = data.iloc[1:]

# Define SIR model
def SIR(y, t, N, beta, gamma, mu, alpha):
    S, I, R = y
    dSdt = mu * (N - S) - beta * S * I / N
    dIdt = beta * S * I / N - (gamma + alpha + mu) * I
    dRdt = gamma * I - mu * R
    return dSdt, dIdt, dRdt
# Define initial conditions
I0 = data['new_cases'].iloc[0]
R0 = 0
S0 = data['population'].iloc[0] - I0
# Define parameter ranges
mu_range = np.linspace(0.01, 0.05, 5)
alpha_range = np.linspace(0.01, 0.05, 5)
beta_range = np.linspace(0.1, 0.5, 5)
gamma_range = np.linspace(0.05, 0.2, 5)

# Standardize data
X = data[['new_cases', 'new_deaths']].values
scaler = StandardScaler()
X_std = scaler.fit_transform(X)
# Run DBSCAN clustering algorithm
dbscan = DBSCAN(eps=0.5, min_samples=10)
clusters = dbscan.fit_predict(X_std)
# Plot clusters and daily new cases data
plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis')
plt.xlabel('Daily new cases')
plt.ylabel('Daily new deaths')
plt.title(f'{country} COVID-19 Clusters')
plt.show()

# Loop over clusters and fit SIR model
fig, axes = plt.subplots(1, len(np.unique(clusters)), figsize=(12, 3))
for i, cluster_id in enumerate(np.unique(clusters)):

```

```

# Extract data for cluster
cluster_data = data[clusters == cluster_id]
# Fit SIR model to daily new cases
N = cluster
# Calculate daily new cases
daily_cases = np.diff(cluster_data)
# Define time grid
t = np.linspace(0, len(cluster_data) - 1, len(cluster_data))
# Define initial conditions
I0 = daily_cases[0]
R0 = 0
S0 = N - I0
# Define parameter ranges
mu_range = np.linspace(0.01, 0.05, 5)
alpha_range = np.linspace(0.01, 0.05, 5)
beta_range = np.linspace(0.1, 0.5, 5)
gamma_range = np.linspace(0.05, 0.2, 5)
# Fit SIR model to daily new cases
best_loss = np.inf
for mu in mu_range:
    for alpha in alpha_range:
        for beta in beta_range:
            for gamma in gamma_range:
                try:
                    # Integrate SIR model
                    solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma, mu,
alpha))
                    I = solution[:, 1]
                    R = solution[:, 2]
                    # Calculate daily new cases predicted by SIR model
                    predicted_cases = np.diff(beta * S0 * I / N)
                    # Calculate loss
                    loss = np.sum((daily_cases - predicted_cases) ** 2)

                    # Update best parameters
                    if loss < best_loss:
                        best_mu = mu
                        best_alpha = alpha

```



```

        best_beta = beta
        best_gamma = gamma
        best_I = I
        best_R = R
        best_loss = loss
    except:
        pass
# Plot SIR model fit
ax = axes[i]
ax.plot(cluster_data, label='Actual')
ax.plot(np.concatenate(([I0], best_beta * S0 * best_I / N)), label='Predicted')
ax.set_title(f'Cluster {cluster_id}')
ax.legend()
# Adjust plot layout
fig.tight_layout()
X = data[['confirmed', 'deaths', 'recovered']].values

# Apply DBSCAN clustering algorithm
dbscan = DBSCAN(eps=100, min_samples=5)
labels = dbscan.fit_predict(X)
# Apply SIR model to each cluster
for cluster in np.unique(labels):
    # Filter data points in this cluster
    X_cluster = X[labels == cluster]
    # Define SIR model
    def SIR(y, t, N, beta, gamma):
        S, I, R = y
        dSdt = -beta * S * I / N
        dIdt = beta * S * I / N - gamma * I
        dRdt = gamma * I
        return dSdt, dIdt, dRdt
    # Set initial conditions
    N = np.sum(X_cluster)
    I0 = X_cluster[0, 0] # number of initial infected individuals
    R0 = X_cluster[0, 2] # number of initial recovered individuals
    S0 = N - I0 - R0 # number of initial susceptible individuals
    y0 = S0, I0, R0
    # Set parameters

```

```

beta = 0.2 # infection rate
gamma = 0.1 # recovery rate
t = np.linspace(0, len(X_cluster), len(X_cluster))
# Solve SIR model using ODE solver
sol = odeint(SIR, y0, t, args=(N, beta, gamma))
# Plot results
plt.plot(t, sol[:, 1], label='Predicted infected')
plt.plot(t, sol[:, 2], label='Confirmed cases')
plt.xlabel('Time (days)')
plt.ylabel('Number of individuals')
plt.title('Sensitivity analysis')
plt.legend()
plt.show()

# Define a range of beta and gamma values to simulate
beta_values = [0.2, 0.4, 0.6, 0.8] # infection rate (per day)
gamma_values = [0.1, 0.2, 0.3, 0.4] # recovery rate (per day)
# Plot the results for different beta values
fig, ax = plt.subplots(figsize=(10, 6))
for beta in beta_values:
    # Define SIR model differential equations
    def SIR(y, t, N, beta, gamma, mu, alpha):
        S, I, R = y
        dSdt = mu * (N - S) - beta * S * I / N
        dIdt = beta * S * I / N - (gamma + alpha + mu) * I
        dRdt = gamma * I - mu * R
        return dSdt, dIdt, dRdt
    # Integrate SIR model differential equations
    solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma_values[0], mu,
alpha))
    S, I, R = solution.T
    # Plot the results
    ax.plot(t, I, label=r'$\beta$ = {}'.format(beta))
ax.set_xlabel('Time (days)')
ax.set_ylabel('Number of infected individuals')
ax.set_title('SIR model simulation for different infection rates')
ax.legend()
plt.show()

```

```

fig, ax = plt.subplots(figsize=(6,4))
ax.plot(S, I, 'b', alpha=0.5, lw=2, label='Phase plot')
ax.plot(S0, I0, 'ro', label='Initial state')
ax.set_xlabel('Number of susceptible individuals')
ax.set_ylabel('Number of infected individuals')
ax.set_xlim(0, N)
ax.set_ylim(0, N)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
plt.show()

beta_range = np.linspace(0, 1, 100)
R0_range = beta_range / (gamma + alpha + mu)
fig, ax = plt.subplots(figsize=(6, 4))
ax.plot(beta_range, R0_range, 'b', alpha=0.5, lw=2, label='R0')
ax.axhline(y=1, color='r', linestyle='--', label='Critical threshold')
ax.set_xlabel('Infection rate (beta)')
ax.set_ylabel('Basic reproduction number (R0)')
ax.set_ylim(0, 10)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
plt.show()

# Integrate SIR model differential equations
solution = odeint(SIR, [S0, I0, R0], t, args=(N, beta, gamma))
S, I, R = solution.T

# Plot the results
fig, ax = plt.subplots(figsize=(8, 6))
ax.plot(t, S, 'b', alpha=0.5, lw=2, label='Susceptible')
ax.plot(t, I, 'r', alpha=0.5, lw=2, label='Infected')
ax.plot(t, R, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
ax.set_xlabel('Time (days)')

```

```

ax.set_ylabel('Number of individuals')
plt.title('SIR Model Predicted Analysis')
ax.set_ylim(0, N)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
plt.show()
f, (ax1,ax2,ax3) = plt.subplots(3)
line1, = ax1.plot(t, solution[:, 0], color='blue')
line2, = ax2.plot(t, solution[:, 1], color='red')
line3, = ax3.plot(t, solution[:, 2], color='green')
ax1.set_ylabel("S")
ax2.set_ylabel("I")
ax3.set_ylabel("R")
ax3.set_xlabel("Time")
plt.show()
for i in range(len(data)) :
    if(data.loc[i, "Glucose"]>130):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(110,180)
    elif(data.loc[i, "BloodPressure"]>120):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(80,160)
    elif(data.loc[i, "kidney problem"]=="yes"):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(180,320)

for i in range(len(data)) :
    if(data.loc[i, "Age"]<7):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(140,210)
    elif(data.loc[i, "Age"]<32):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(10,100)
    elif(data.loc[i, "Age"]<50):
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(100,180)
    else:
        data.loc[i,"resistance"]=data.loc[i,"resistance"]-np.random.randint(180,240)
data

from sklearn.cluster import KMeans

```

```

plt.scatter(data['Age'],data['Immunity'],color='pink')
plt.xlabel('Age')
plt.ylabel('Immunity')
data['Cluster']=a
data
d1=data[data.Cluster==0]
d2=data[data.Cluster==1]
d3=data[data.Cluster==2]
plt.scatter(d1.resistance,d1.Immunity,color='blue', label='cluster 1')
plt.scatter(d2.resistance,d2.Immunity,color='pink', label='cluster 2')
plt.scatter(d3.resistance,d3.Immunity,color='orange', label='cluster 3')
plt.xlabel('Resistance')
plt.ylabel('Immunity')
df=data[['Age','resistance']]
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df)
    distortions.append(kmeanModel.inertia_)
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
d1 = data[data['Cluster']==2]
df
N=len(d1)
N
I0=len(d1[d1['Status'] ==1])
R0=len(d1[d1['Status'] ==-1])
print(I0)
print(R0)
ans=sum(d1['days to Recover'])
print(ans//I0)
S0 = N - I0 - R0
S0

```

```

# Initial conditions vector
y0 = S0, I0, R0
ret = odeint(deriv, y0, t, args=(N, beta, gamma))
S, I, R = ret.T
fig = plt.figure(facecolor='w')
ax = fig.add_subplot(111, facecolor='#dddddd', axisbelow=True)
ax.plot(t, S/1, 'b', alpha=.6, lw=2, label='Susceptible')
ax.plot(t, I/1, 'r', alpha=0.5, lw=2, label='Infected')
ax.plot(t, R/1, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
ax.set_xlabel('Time /days')
ax.set_ylabel('Number of people')
ax.set_ylim(0,1100)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
for spine in ('top', 'right', 'bottom', 'left'):
    ax.spines[spine].set_visible(False)
plt.show()

d2 = data[data['Cluster']==0]
N=len(d2)
print(N)
I0=len(d2[d2['Status'] ==1])
R0=len(d2[d2['Status'] ==-1])
print(I0)
print(R0)
ans=sum(d2['days to Recover'])
print(ans/I0)
S0 = N - I0 - R0
S0
def deriv(y, t, N, beta, gamma):
    S, I, R = y
    dSdt = -beta * S * I / N
    dIdt = beta * S * I / N - gamma * I
    dRdt = gamma * I
    return dSdt, dIdt, dRdt

```

```

t = np.linspace(0, 30, 30)
beta, gamma = .2, 1/17
# Initial conditions vector
y0 = S0, I0, R0
ret = odeint(deriv, y0, t, args=(N, beta, gamma))
S, I, R = ret.T
fig = plt.figure(facecolor='w')
ax = fig.add_subplot(111, facecolor='#dddddd', axisbelow=True)
ax.plot(t, S/1, 'b', alpha=.6, lw=2, label='Susceptible')
ax.plot(t, I/1, 'r', alpha=0.5, lw=2, label='Infected')
ax.plot(t, R/1, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
ax.set_xlabel('Time /days')
ax.set_ylabel('Number of people')
ax.set_ylim(0,500)
ax.yaxis.set_tick_params(length=0)
ax.xaxis.set_tick_params(length=0)
#ax.grid(b=True, which='major', c='w', lw=2, ls='-')
legend = ax.legend()
legend.get_frame().set_alpha(0.5)
for spine in ('top', 'right', 'bottom', 'left'):
    ax.spines[spine].set_visible(False)
plt.show()
Define initial conditions and parameters
t = np.arange(0, 365, 100)
# Solve SIR model
y0 = [S0, I0, R0]
sol = odeint(SIR_model, y0, t, args=(beta, gamma))
# Plot results
plt.plot(t, sol[:, 0], label='Susceptible')
plt.plot(t, sol[:, 1], label='Infected')
plt.plot(t, sol[:, 2], label='Recovered with immunity')
plt.legend()
plt.xlabel('Time (days)')
plt.ylabel('Number of individuals')
plt.title('SIR Model Predicted Analysis')
plt.grid()
plt.show()

```

home.html

```
<!DOCTYPE html>
<html>
<head>
  <title>SIR Model Predicted Analysis</title>
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
  <h1>SIR Model Predicted Analysis</h1>
  <form method="POST" action="/results" enctype="multipart/form-data">
    <label for="population_size">Population size:</label>
    <input type="number" id="population_size" name="population_size"><br><br>
    <label for="rate_of_infection">Rate of infection:</label>
    <input type="number" id="rate_of_infection"
name="rate_of_infection"><br><br>
    <label for="dataset">Dataset:</label>
    <input type="file" id="dataset" name="dataset"><br><br><br>
    <input type="submit" value="Predict">
  </form>
</body>
</html>
```

Result.html

```
!DOCTYPE html>
<html>
<head>
  <title>SIR Model Predicted Analysis Results</title>
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
  <h1>SIR Model Predicted Analysis Results</h1>
  <div id="plot"></div>
  <script>
    var plot_data = {{ plot_json|safe }};
    Plotly.newPlot('plot', plot_data.data, plot_data.layout);
  </script>
```



```
</body>
</html>
```

app.py

```
import plotly
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import DBSCAN
from scipy.integrate import odeint
import plotly.graph_objs as go
import json
import io
import os
from plotly.io import kaleido

# Define SIR model
def SIR_model(y, t, beta, gamma):
    S, I, R = y
    dS_dt = -beta * S * I
    dI_dt = beta * S * I - gamma * I
    dR_dt = gamma * I
    return [dS_dt, dI_dt, dR_dt]

# Define Flask application
app = Flask(__name__)

# Define route for the home page
@app.route('/')
def home():
    return render_template('home.html')

# Define route for the results page

@app.route('/results', methods=['POST'])
def results():
```

```

# Get user input
population_size = int(request.form['population_size'])
rate_of_infection = float(request.form['rate_of_infection'])

# Load dataset from uploaded CSV file
dataset_file = request.files['dataset']
dataset = pd.read_csv(io.StringIO(dataset_file.stream.read()).decode("UTF8"))

# Select relevant columns
cols = ['confirmed', 'deaths', 'recovered']
X = dataset[cols]
# Normalize data
X_norm = (X - X.mean()) / X.std()
# Use DBSCAN to identify clusters
dbscan = DBSCAN(eps=0.5, min_samples=5)
labels = dbscan.fit_predict(X_norm)

# Define initial conditions and parameters
N = population_size
I0 = sum(labels == -1)
R0 = sum(labels == 1)
S0 = N - I0 - R0
beta = rate_of_infection
gamma = 0.01
t = np.arange(0, 365, 100)
y0 = [S0, I0, R0]
sol = odeint(SIR_model, y0, t, args=(beta, gamma))

trace_susceptible = go.Scatter(
    x=t,
    y=sol[:, 0],
    name='Susceptible',
    mode='lines',
    line=dict(width=2)
)
trace_infected = go.Scatter(
    x=t,
    y=sol[:, 1],

```

```

        name='Infected',
        mode='lines',
        line=dict(width=2)
    )
    trace_recovered = go.Scatter(
        x=t,
        y=sol[:, 2],
        name='Recovered with Immunity',
        mode='lines',
        line=dict(width=2)
    )
    layout = go.Layout(
        title='SIR Model Predicted Analysis',
        xaxis=dict(title='Time (days)'),
        yaxis=dict(title='Number of individuals'),
    )
    fig = go.Figure(data=[trace_susceptible, trace_infected, trace_recovered],
    layout=layout)

    #Save plot as file in html
    plot_dir = 'static/images'
    plot_file = plot_dir + 'k.png'
    if not os.path.exists(plot_dir):
        os.makedirs(plot_dir)
    plotly.offline.plot(fig, filename=plot_file, auto_open=False)

    # Save plot as file in image
    directory = 'static/images'
    if not os.path.exists(directory):
        os.makedirs(directory)
    fig.write_image("static/images/plot.png", engine="kaleido")

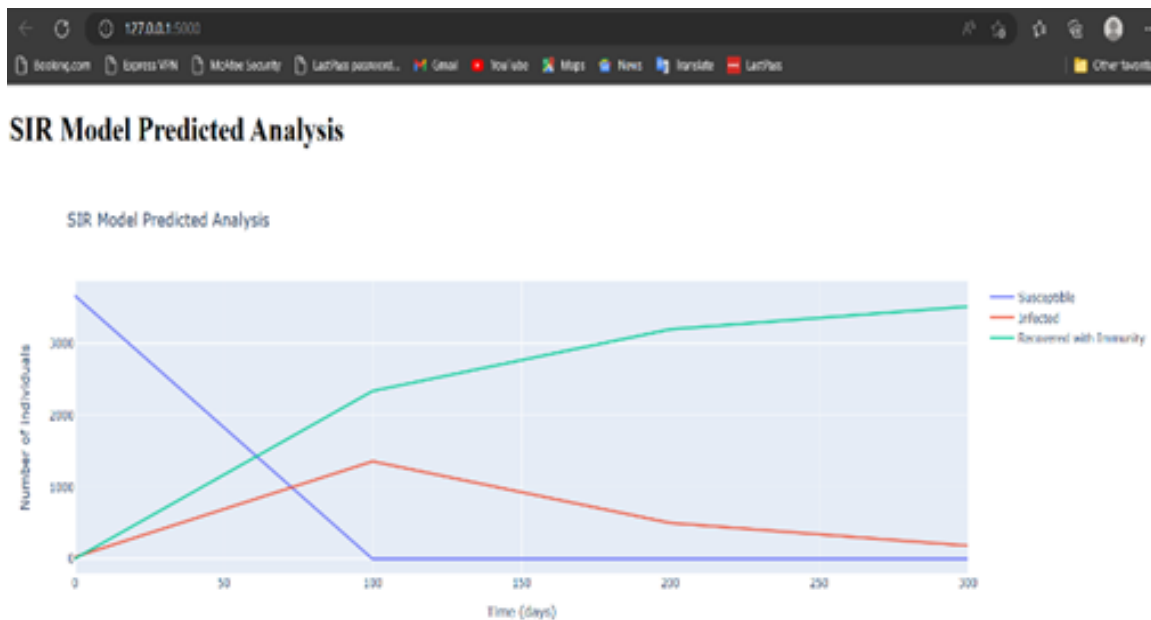
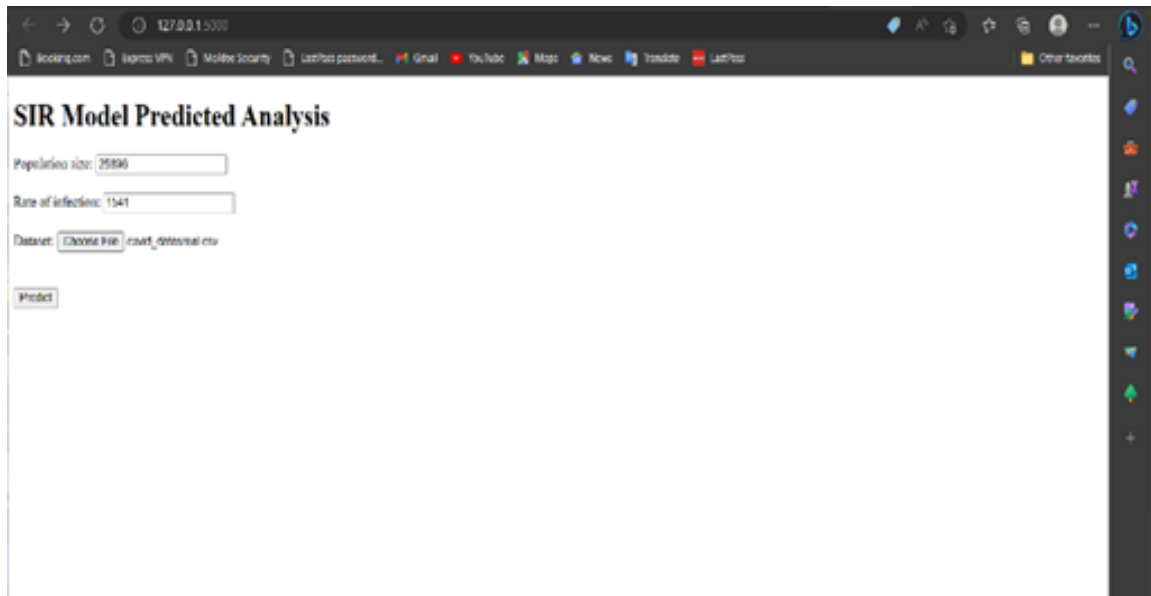
    # Return plot as JSON
    plot_json = json.dumps(fig, cls=plotly.utils.PlotlyJSONEncoder)
    return render_template('results.html', plot_json=plot_json)

# Run the application
if __name__ == '__main__':

```

```
app.run(debug=True)
```

B. SCREENSHOTS



C. RESEARCH PAPER

EPIDEMIC OUTBREAK PREDICTION USING SIR MODEL

Patibandla Venkata Lohith Kumar
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
lohithkumarp2512@gmail.com

Gidugu Bhanu Venkata Prakash
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
bhanunaidu2612@gmail.com

Dr. M. Saravanan, Ph.D
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
saravanan.cse@sathyabama.ac.in

Abstract—The pace at which viruses are being transferred by individuals is rapidly rising, which has resulted in the loss of human life. The majority of those who get this covid-19 virus is likely to have hereditary disorders. This study examined how long it will take a patient to recover from a virus. This will assess the length of time a patient will need to recover from a virus using Deep Learning techniques. The combination of DBSCAN clustering and the SIR model is utilized to estimate the time required for a patient to recover from a virus. The dataset is first subjected to analysis using the DBSCAN clustering algorithm, which groups the data based on age as the primary criterion. The resulting cluster output is then inputted into the SIR model to assess accuracy. However, the accuracy of the cluster output is not satisfactory when compared to previous results. Therefore, the cluster output will be rechecked using different clustering algorithms to obtain more reliable results. By utilizing the more precise results obtained from the cluster output as a parameter, and integrating natural death and death caused by the disease as additional parameters in the SIR model, the project can obtain the most accurate outcomes. These parameters will facilitate the creation of a dependable visual predictor. The SIR model's outcomes are presented through a web interface, which enables users to input population size, infection rate, and a COVID-19 dataset. Utilizing a Flask application, the SIR model's output is visualized on a webpage.

Keywords—COVID-19, Hereditary, DBSCAN, SIRMODEL, Flask application.

I. INTRODUCTION

In the year 2019, the World Health Organization (WHO) announced that the widespread outbreak of coronavirus, commonly referred to as COVID-19, had reached pandemic proportions [1]. To stop the virus from spreading further, an international coordinated effort is required. An abnormally high percentage of the population is affected by the virus, which is defined as occurring over a large geographical area. [2] The 2009 H1N1 flu pandemic was the last pandemic to be officially confirmed worldwide.

The World Health Organization (WHO) was notified of a group of pneumonia patients in Wuhan, Hubei Province, China on December 31, 2019, with an unknown cause. In January 2020, a previously unidentified virus was identified as the root cause of the 2019 novel coronavirus outbreak [3][4]. The virus was confirmed as the source of the disease following genetic analysis and patient sample collection. The WHO gave the virus the name Disease 2019 (COVID-19) in February 2020 [5]. SARS-CoV-2 is the virus responsible for COVID-19, which is highly infectious [6].

There have been over 302,493 fatalities and over 4,444,670 cases reported as of May 15th, 2020, in 188 different nations. Furthermore, 1,588,858 were found. Here, live data may be viewed.

In Wuhan, Hubei area, China, at the end of December 2019, pneumonia flared up for an unknown reason. Up to the end of January, the flare-up had infected 106 people in 19 other countries and contaminated 9720 people in China, resulting in 213 fatalities. A few free research facilities identified the unique COVID-19 as the causative agent of this perplexing pneumonia a few days after it occurred. The World Wellbeing Association has designated the infectious agent as the respiratory illness known as COVID-19, caused by the SARS-CoV2 virus, which is a severe and intense condition. The disease is highly contagious and has been a significant cause of global contamination since its emergence in 2019. The SARS-CoV-2 epidemic has claimed 79890 cases and 3354 fatalities in China, as per the World Health Organization's daily report. It has now spread to 39 more countries, with a total of 4333 cases reported by March 16, 2020. The COVID-19 virus has arisen as a major public health concern worldwide that has been considered in this investigation. This study will also create software requirements specifications (SRS).

Through the utilization of Deep Learning techniques, an analysis will be conducted on the duration of recovery from viral infections. Two distinct methods, namely DBSCAN and the SIR Model, have been employed to estimate the period of recovery for patients. Clustering, a fundamental technique in data analysis, plays a critical role in determining the data's structure. Essentially, clustering is the process of identifying subgroups in a dataset that exhibit high similarity, while the data in different groups have significant differences. This technique is classified as an unsupervised learning method, as it lacks a predetermined standard against which the accuracy of the grouping algorithm can be evaluated.

II. LITERATURE SURVEY

Researchers and computer scientists have researched this problem statement in detail over the past few years to find a solution, and all their answers range from examining various cluster techniques to the analysis of epidemic outbreaks of different data collection.

The study by Marina Bagić Babac [6] suggested the model dynamical mathematical method is SIR, which provides a more accurate in predicting the covid-19 data set. The new virus was spreading more in Italy and the infection rate is increasing but entering the data (infected people) online is very less. What steps must be taken to halt the SARS-CoV-2 virus? Presently accessible data [2] allow for the analysis of historical occurrences as well as the prediction of positive outcomes. The risk of the second wave is very less because of the first wave project. The result of this project is shown in the graph using the SIR model.

Ashutosh S et al. [7] In April 2020, conducted a study on the SIR model to analyze the impact of different lockdown measures on the spread of infections. They diligently worked every day to identify the model's parameters and explore various possibilities to improve the accuracy of their findings. In doing so, they distinguished the exposure rate from the infection rate and developed different levels of quarantine to mitigate the spread of the disease. As a result, their study yielded more accurate results compared to previous studies, although they noted that the model's parameters still require refinement to achieve more consistency.

Ahmad Sedaghat, Shahab Band, Amir Mosavi, and Laszlo Nadai are four members [8] who studied and created the SEIR-PAD model. A few articles in the mathematics field have reported on the extension of SIR-type models. The covid-19 was growing rapidly in GCC nations at the time, and they believed the SIR paradigm was unsuitable for this project. As a result, they developed a new model known as SEIR-PAD. The SEIR-PAD model is utilized in MATLAB to numerically solve seven sets of ordinary differential equations with eight unknown coefficients, enabling it to accommodate four sets of COVID-19 data. They correctly projected utilizing available data from the epidemic to June 23rd, 2020, using the SEIR-PAD model. This SEIR-PAD model project's data is more accurate than the SIR model.

Mohammad Shanna and Sherief Abdallah [9] used the Net Logo program to examine a virus propagating in a semi-closed setting. The covid-19 will have spread further by the 27th of May 2020. They adopted an existing model built by Yang and Wilensky in 2011 using the Net Logo framework to handle this problem. The researcher attempted to apply the statistics for infection probability and other input parameters released by WHO for the COVID-19 virus. The proposed system would use an existing disease-spreading model from the use of the Net Logo library to model the COVID-19 virus' spread across a country. When compared to the present model, this model will help to provide higher accuracy for covid-19 outbreaks using the Net Logo tool.

Researchers J. Hackl and T. Dubernet [10] used MATSim to simulate a huge size population in an urban setting. The emphasis of the simulation is on contagious illnesses that propagate within transportation settings. Which makes use of the actual data from people's activity and interactions on their everyday commute routes. The basis of the model derives from actual data obtained during seasonal flu epidemics that

occurred in Kilchberg. When the simulation is completed, the outcomes are compared to the known SIR model. The research investigated the complexities of virus epidemics as well as all other elements influencing viral transmission between humans, such as direct and indirect physical contact. In addition, based on the SIR model, worked on a generalized epidemic spread model. The research simulation findings succeeded in producing various scenarios of an epidemic in a complicated metropolitan setting, which aids in predicting the occurrence and taking appropriate measures.

Exploring the potential of utilizing machine learning techniques to predict disease outbreaks based on big data gathered from healthcare communities, Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang delve into the topic [11]. They surveyed various studies on disease prediction and identified the limitations and challenges of using traditional statistical methods. The authors then proposed a framework that integrates multiple machine learning algorithms to accurately predict disease outbreaks. Their research underscores the significance of utilizing big data and machine learning techniques in the healthcare industry to enhance the prevention and treatment of diseases. Overall, this paper provides valuable insights for researchers and practitioners working in the field of health informatics and data analytics.

Dhiraj Dahiwade et al. [12] focus on developing a machine learning-based approach for predicting diseases. The authors analyzed various machine learning algorithms and techniques, to select the most appropriate method for the proposed model. They collected and preprocessed medical data to train and validate the model, achieving an accuracy rate of up to 93%. The study highlights the significance of data quality and preprocessing in developing reliable predictive models. Overall, this paper provides valuable insights into the development of disease prediction models using machine learning and can serve as a useful reference for future studies in this field

Pahulpreet Singh Kohli and Shriya Arora's [13] paper explores the potential of machine learning in predicting diseases. The paper presents a comprehensive literature review of existing studies in this area, highlighting the benefits and limitations of different approaches. The authors delve into various machine learning algorithms, and artificial neural networks and compare their performance in predicting diseases. Overall, this paper provides valuable insights into the application of ML in virus prediction and offers a useful reference for researchers working in this field.

A. Open Problems In Existing System

The existing model does not generate the necessary and easily comprehended results to seek. Many investigations have been done using the SIR, Net Logo tools, SEIR-PAD, and GIS models. They employed distinct clusters among all these models to improve accuracy. However, they failed to produce an accurate result.

This accepts a population as input and calculates that population, however, it does not give the required output. The cluster method used for the existing model is k-means.

It generates output with no apparent indication of clusters. That is, it does not know how many people are suspected, how many are infected, or how many are recovered. The available models do not provide a clear indicator of cluster development. These models just provide the graph representation of the SIR model, i.e., how many days it takes to recover, how infectious the rate is, and how suspicious the rate is and only these things are acquired from the current model. As a result, use new techniques to overcome the old model. Then it can achieve our goals. The existing model has a disadvantage. It is calculating the result by taking the whole population. The output is not accurate.

III. PROPOSED SYSTEM

The SIR model-based epidemic outbreak prediction system being proposed is an enhanced version of a pre-existing model. The previous system faced numerous challenges related to its algorithms and outcome accuracy. To address these issues, a novel approach was taken by integrating the proposed system with a new technique that utilizes the DBSCAN Clustering algorithm.

These can more efficiently implement the suggested system by using the DBSCAN Clustering method. It can generate clusters for whatever is needed by utilizing a clustering technique. In this initiative, these are looking at certain characteristics such as age, immunity, and resistance, among others. Taking this property creates clusters for who is suspected, infectious, and recovered, i.e., healthy persons.

The clustering method output is then used as input for the SIR model. After that, the SIR model will receive the input and do the necessary operations. Then it will display the suspected rate, the infectious rate, and the recovery rate. This proposed solution would provide the clear and desirable results that the model wants. After receiving an output from the SIR model, the final results will be displayed on the web page. When compared to the current model, this provides several advantages. While doing the proposed system, must follow the steps:

- Understanding the dataset.
- Determining how specific columns are related.
- Preprocessing the data in the dataset.
- Calculating DBSCAN from the data by dividing the data into four clusters.
- The output of the cluster will be sent to MODEL parameters to obtain a predictive SIR model.
- The result will be shown on the webpage using the Flask app.

DBSCAN, also known as Density-Based Spatial Clustering of Applications with Noise is the selected clustering algorithm for this project, which identifies clusters based on the density of regions. This algorithm is particularly proficient in identifying anomalies and clusters with non-uniform shapes. In this project, DBSCAN is used to identify clusters that can be further used to develop parameters for the SIR model. By inserting these parameters into the SIR model,

the results can be more accurate based on the clusters identified. This approach can be particularly useful in scenarios where traditional clustering algorithms fail to identify meaningful clusters or where the shape of the clusters is irregular.

A. Project Management Plan

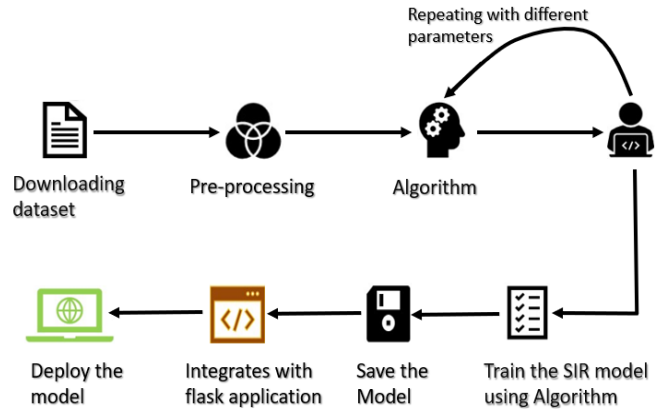


Fig. 1. Flow Diagram

IV. METHODOLOGY

A. Data collection:

For every project, a dataset is compulsory to attain good results. This dataset focuses on Age Immunity, Resistance, and Days to Recover data. Collect information about the population's susceptible, infected, and recovered individuals, and track it over a period of time. This information is available from a variety of sources, including public health authorities and Kaggle. This dataset was downloaded from Kaggle. Kaggle is a platform where there will be many datasets available in this. These datasets were used to build the models in machine learning as well as in deep learning.

B. Model selection:

DBSCAN, also known as Density-Based Spatial Clustering of Applications with Noise, is a popular clustering method that categorizes similar data points in a high-dimensional space by measuring their proximity to one another. DBSCAN, unlike other clustering algorithms, the size of clusters is determined automatically based on the density of data points in the space. Clusters are defined by the algorithm as high-density areas divided by low-density areas, with each data point assigned to a cluster depending on its distance to other points in the cluster. Noise refers to points that do not belong to any cluster.

In simpler terms, DBSCAN combines data points that are close to each other while simultaneously separating them from other groupings of points that are further apart. It is especially beneficial for datasets with non-linear or irregular geometries, as well as for finding outliers or noisy points that do not belong to any cluster. DBSCAN includes two critical parameters that must be configured before executing the algorithm: the radius (eps) and the minimum number of points (min samples) necessary to generate a compact region. These parameters can

be modified based on the unique features of the data being clustered.

In the existing SIR model, they used formulas for each character is Susceptible using $dS/dt = -\beta SI$. Infectious using $dI/dt = (\beta SI) - (\gamma + I)$ and recovered using $dR/dt = \gamma + I$. By using these the model does not perform much accurately. So, by considering this implementing a new SIR model by adding a few equations like Natural death rate, and death by disease (Here, the disease is not only a covid virus cause of death). So, with these modifications, the proposed model will be more accurate and visualized. The data set used for this model also gives a better prediction.

V. IMPLEMENTATION

In the SIR model, the total of these three compartments (susceptible, infectious, and recovered) stays steady and equivalent to the underlying number of populations. The fundamental SIR model was introduced, where β is the disease rate or transmission rate or the power of contamination. Furthermore, γ means the recuperation or elimination rate. By and large talking, these boundaries (β , γ) are not steady they are elements of the size of irresistible and recuperation compartments.

These are the boundaries that need to upgrade and gauge with the goal that the revealed and reproduced cases are around approaches. To settle this arrangement of differential conditions, it needs from beginning qualities for the three-state factors S , I , and R specifically $S(t)$, $I(t)$, and $R(t)$.

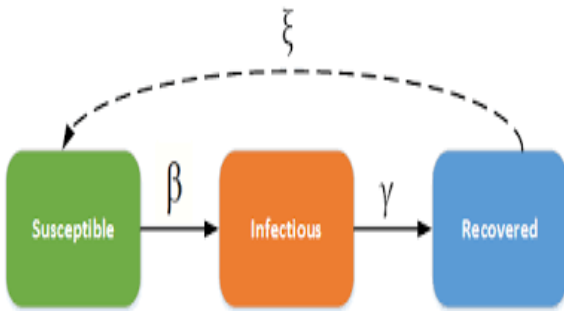


Fig. 2. SIR Model Structure

The SIR model comprises three distinct compartments, namely susceptible, infectious, and recovered, that interact dynamically. The model assumes that the population is homogenous, which means that all people have the same chance of encountering each other. Those who recover from the sickness get immunity and cannot be infected again, according to the model.

The SIR model does not account for characteristics such as age, BMI, and geographic location, which might influence illness transmission. Other compartments and parameters, such as exposed persons and varied contact rates, can be added to the SIR model.

- $S(t)$: The number of individuals susceptible to the disease at time t .
- $I(t)$: The number of individuals susceptible to the disease at time t .
- $R(t)$: The number of individuals susceptible to the disease at time t .
- N : the total population.
- β : The rate of transmission refers to the speed at which susceptible individuals contract the infection.
- γ : The rate at which infected individuals recuperate and acquire immunity, also known as the recovery rate.

A. Equations:

Listed below are the differential equations that represent the SIR model:

- $dS/dt = -((\beta SI)/N)$
- $dI/dt = ((\beta SI)/N) - (\gamma I)$
- $dR/dt = \gamma I$
- $dS/dt = (ND * (N - (S))) - ((Beta * S * I) / N)$
- $dI/dt = ((Beta * S * I) / N) - ((Gama + Alpha + ND) * I)$
- $dR/dt = (Gama * I) - (ND * R)$

$$\frac{dS}{dt} = -b s(t) I(t)$$

$$\frac{di}{dt} = b s(t) i(t) - k i(t)$$

$$\frac{dr}{dt} = k i(t)$$

where:

- ND is the natural death rate (per day)
- α is the death rate caused by disease (per day)
- β is the infection rate (per day)
- γ is the recovery rate (per day)
- dS/dt represents the rate at which the number of susceptible individuals changes over time.
- dI/dt represents the rate at which the number of infectious individuals changes over time.
- dR/dt represents the rate at which the number of infectious recovered changes over time.

Based on the transmission and recovery rates, these equations show based on the number of susceptible, infectious, and recovered people fluctuates over time based on transmission and recovery rates. To anticipate the path of an epidemic

and investigate the influence of actions on disease transmission, the SIR model can be solved numerically or analytically.

B. Overall Design of The Proposed System

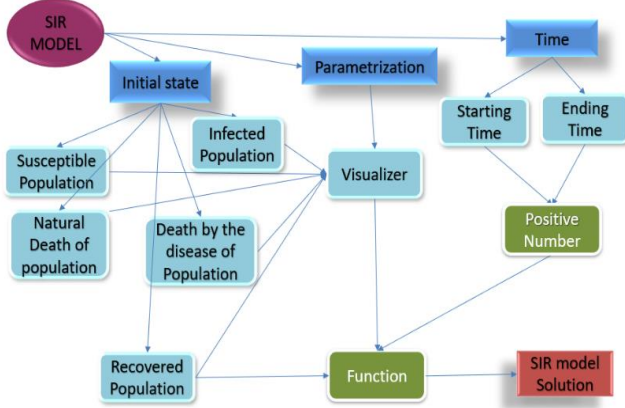


Fig. 3. System Architecture

VI. RESULTS AND DISCUSSION

Utilizing the DBSCAN clustering method helps identify and form distinct clusters, exposing underlying patterns and relationships between variables. By comparing columns through scatter plots, a more comprehensive understanding of the data can be obtained. This aids in the interpretation and analysis of results. The approach of using DBSCAN clustering and scatter plots leads to more accurate results and greater insights in data analysis.

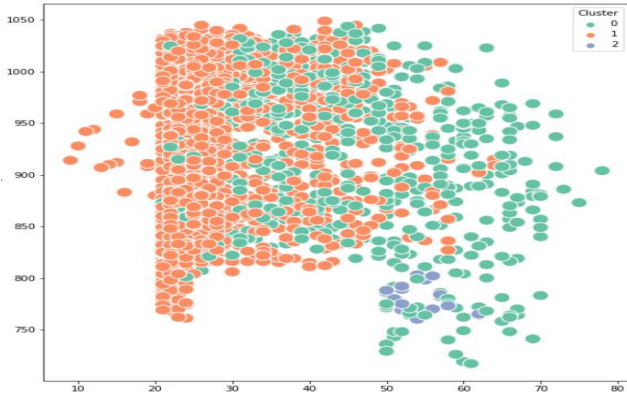


Fig. 4. Cluster Separation

C. Age vs Resistance:

By using age and resistance as a single component in our project. In this case, age and resistance are inversely related to one another. Because if age is less, immunity is more (or if age is greater, immunity is greater). In this scenario, if a person is affected by covid-19, if he is younger, he will recover quickly; alternatively, if he is older, he will take longer to recover as compared to other age groups.

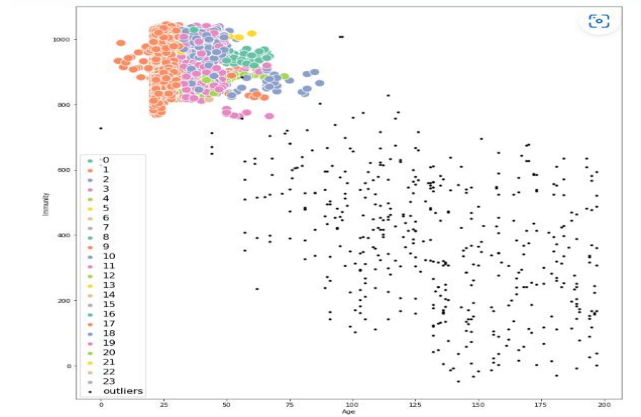


Fig. 5. Age vs Resistance

D. Immunity vs Resistance:

Immunity and resistance are proportionate to one another. Because a person's resistance increases with his immunity. Otherwise, a person with lower immunity will have lower resistance. Immunity and resistance are completely reliant on an individual's age group. That age range merely determines whether the individual has immunity and resistance. To recover from the covid-19, both immunity and resistance are required.



Fig. 6. Immunity vs Resistance

They are not restricted from computing Immunity and Resistance. Based on age, taking other health conditions into account, and assessing Immunity and Resistance. There is no set formula for determining Immunity and Resistance. Taking into account all of the variables will determine how many days a person may recover from the covid-19. It will describe the recovery of the covid-19 patient in a specific place dependent on the patient's health.

Using a covid-19 dataset, clusters are formed using the DBSCAN algorithm, and the distance between clusters is calculated. Framing clusters between Parameters such as Age, Resistance, Immunity, BMI, Status, and Illness for better understanding to depict in the sir model. The sir model output is created with graph time (days) and the number of persons based on the cluster formation analysis.

In the SIR model, it is typical to plot the number of Susceptible, Infectious, and Recovered with Immunity persons over by time to evaluate the impact of diverse model parameter values on the development of the pandemic.

C. Varying Transmission Rate (β):

A commonly used plot in the SIR model examines the impact of different transmission rates on the epidemic's progression. The plot involves adjusting the transmission rate (β) maintaining the recovery rate (γ) and the initial conditions fixed. It displays that for every β value, the figures representing the count of vulnerable, infected, and recovered persons will vary over time. This graphic aids in illustrating how changes in the transmission rate affect the epidemic's timing, severity, and overall size.

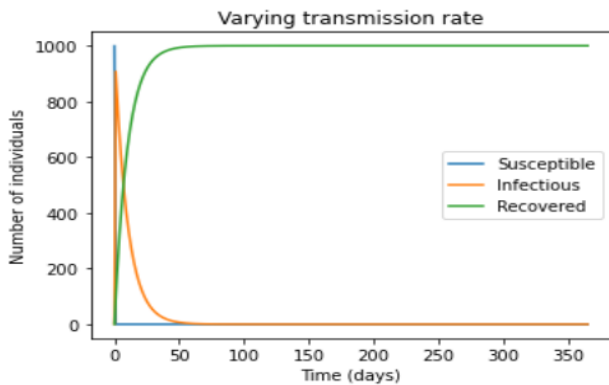


Fig. 7. Transmission Rate (β) Analysis

D. Varying Recovery Rate (γ):

Another commonly used plot in the SIR model involves comparing the impact of various recovery rates on the epidemic's progression. The plot involves adjusting the recovery rate (γ) maintaining the transmission rate (β) and initial conditions fixed. For each value of (γ), it displays the count of susceptible, infected, and recovered individuals over time. This plot can help illustrate how changes in the recovery rate affect the epidemic's duration and scope.

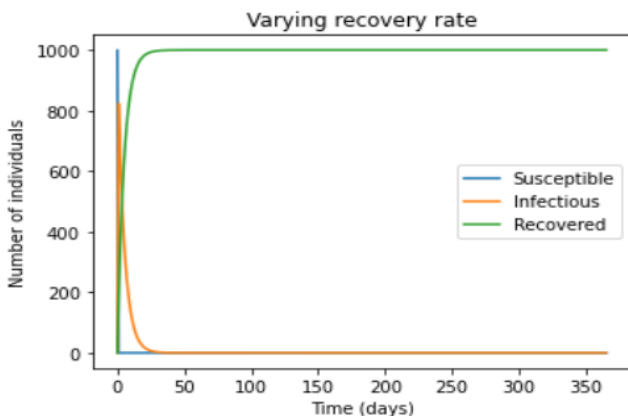


Fig. 8. Recovery Rate (γ) Analysis

E. Sensitivity Analysis:

A sensitivity analysis plot can be used to examine the effect of small changes in the model parameters on the outcome of the epidemic. In this plot, the model parameters are varied slightly around their baseline values, and the plotted outcome displays the variation in the number of infected individuals. The plot can facilitate the identification of the key parameters driving the epidemic's trajectory, and provide valuable insights for designing interventions that aim to curb the disease's spread.

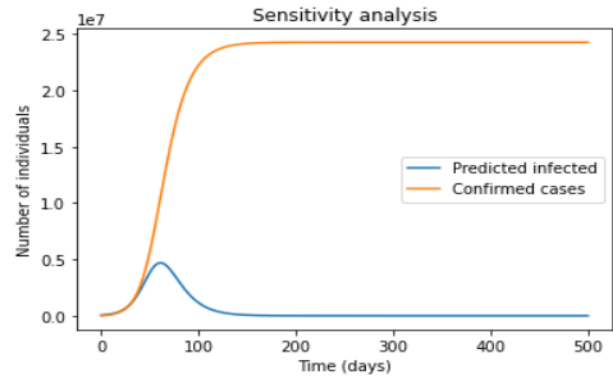


Fig. 9. Sensitivity Analysis

Overall, plots can help to illustrate how changes in the model parameters affect the course of the epidemic and can be used to explore the potential impact of different interventions. They can also be used to communicate the results of the SIR model to a wider audience and to help inform public health policies aimed at controlling spread of the infectious virus.

The SIR model may also be used to compute various important values of interest, such as:

F. Basic Reproduction Number (R_0):

The contagiousness of a disease can be determined by a metric known as R_0 , which represents the average number of secondary infections caused by an individual infected with a disease in a population where everyone is susceptible. R_0 is calculated by dividing the transmission rate (β) by the recovery rate (γ). If R_0 is greater than one, the disease will likely propagate throughout the population, but if it is less than one, the disease will eventually fade away.

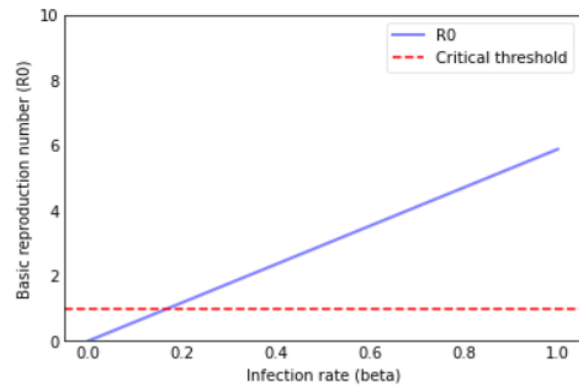


Fig. 10. Reproduction Number (R_0) Plot

G. Peak Infection:

This is the period when the infection rate of people individuals is at its peak. The timing and magnitude of the peak can provide insights into the severity and duration of the epidemic. This is a critical parameter in understanding the transmission of infectious viruses, and is influenced by factors such as the transmission rate and the proportion of the population susceptible to the disease. Accurately predicting the peak infection can help policymakers make informed decisions about implementing public health measures to restrict the spread of virus transmission.

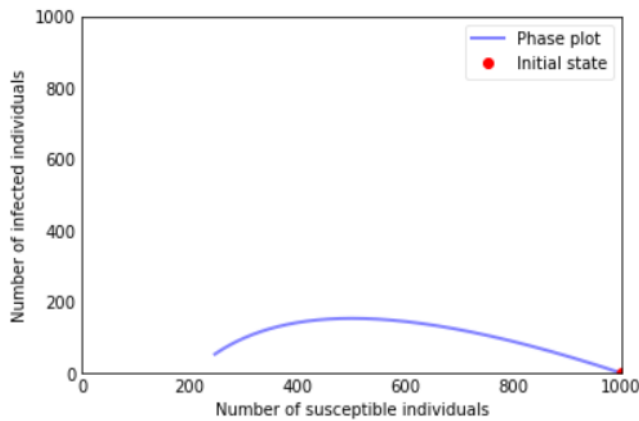


Fig. 11. Peak Infection Plot

H. Final Sizes of Epidemic:

This represents the overall count of people who contract the infection throughout the epidemic. The final size is influenced by factors such as the initial conditions, the transmission and recovery rates, and any interventions that are implemented.

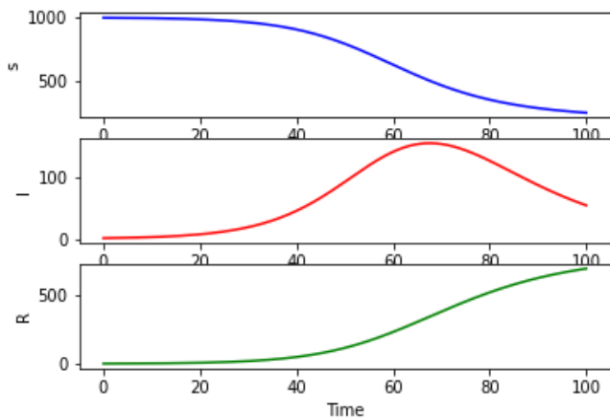


Fig. 12. Sizes of SIR

Through the utilization of various parameter values and initial conditions in simulating the SIR model, experts can investigate the impact of altering these factors on the development of the epidemic and the efficacy of various interventions. The insights gathered from the SIR model hold the potential to influence public health policies and initiatives focused on curtailing the spread of infectious diseases.

SIR Model Predicted Analysis

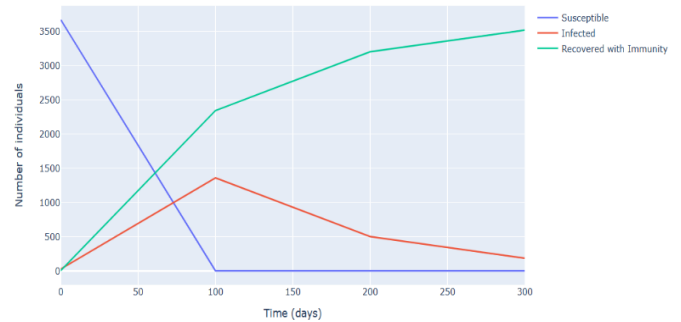


Fig. 13. SIR Prediction

Considering the baseline conditions of the epidemic and the model parameters, the SIR model predicts the number of Susceptible, Infectious, and Recovered with immunity persons over time. This prediction can be used to acquire insights into the path of an epidemic and to assess the possible impact of various interventions, such as natural death or death rate due to disease. However, it is important to note that the model is only as good as its assumptions and data inputs, and should be used in conjunction with real-world data and expert judgment to make accurate predictions.

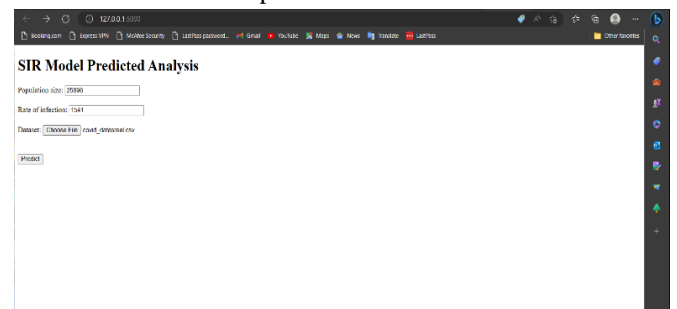
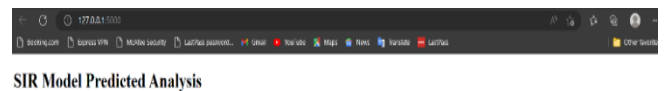


Fig. 14. User Input Web Interface

The SIR model output was pushed into the web interface using the Flask framework. The Plotly library comes in handy while creating the web frame. The flask program runs on a server after loading the appropriate modules, with the results shown in a web interface. The web interface allows users to input the population size, rate of infection, and a COVID-19 dataset. The resulting outcomes, including the number of susceptible, infected, and recovered with immunity individuals, are graphed for an interactive user experience.



SIR Model Predicted Analysis

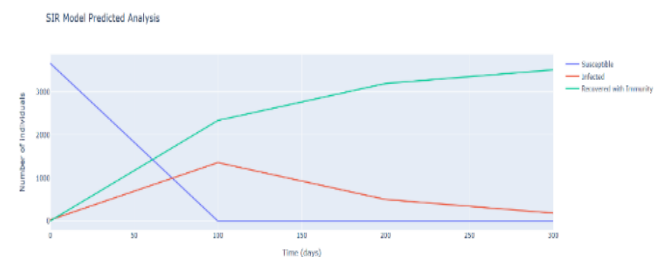


Fig. 15. The analysis of the output is based on the user input parameters.

The above Fig15 displays the ultimate prediction of the SIR model based on the user-provided input data. It represents the culmination of the model's analysis, considering various factors such as transmission and recovery rates, initial conditions, and other relevant data points. The SIR model, plays a crucial role in predicting and understanding the spread of infectious diseases, providing vital information for policymakers and public health officials. The SIR model is capable of effectively processing real-time data. By leveraging this model, it is possible to develop and implement effective interventions and control strategies that can help minimize the impact of epidemics on society.

VII. CONCLUSION

In this research, the DBSCAN clustering technique was applied to a COVID-19 dataset to find discrete groupings of regions with similar trends in confirmed cases, deaths, and recoveries. The SIR model was then used to anticipate the virus's spread within each group, taking into consideration the initial number of Susceptible, Infected, and Recovered with immunity persons, the natural death (ND) rate, the death rate due to disease (Alpha), as well as the transmission and recovery rates. The SIR model output was shown on a web page using Plotly and Flask app, this allows us to examine how the number of Susceptible, Infectious, and Recovered with immunity persons over time inside each cluster.

Overall, this approach can fill a gap in the existing systems, by providing an intuitive and user-friendly interface that allows users to easily input parameters and generate visualizations of outbreak predictions provides a valuable instrument for forecasting the spread of infectious illnesses and may assist improve public health policies and initiatives targeted at reducing the effect of such outbreaks.

VIII. REFERENCES

- [1] World Health Organization. WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. Available from: <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19--11-march-2020> (Accessed 14 March 2020)
- [2] Merriam-Webster Dictionary. Pandemic. Available from: <https://www.merriam-webster.com/dictionary/pandemic> (Accessed 14 March 2020)
- [3] World Health Organization. Novel Coronavirus – China. Disease outbreak news: Update 12 January 2020.
- [4] Wikipedia. Timeline of the 2019–20 coronavirus pandemic in November 2019 – January 2020. Available from [https://en.wikipedia.org/wiki/Time-](https://en.wikipedia.org/wiki/Time-line_of_the_2019%E2%80%9320_coronavirus_pandemic_in_November_2019_%E2%80%93_January_2020)
- [5] World Health Organization. Director-General's remarks at the media briefing on 2019-nCoV on 11 February 2020. 2020/2/18) [2020-02-21]. <https://www.who.int/dg/speeches/detail/who-director-general-remarks-at-the-media-briefing-on-2019-ncov-on-11-february-2020>. [last accessed 17 March 2020].
- [6] Marina Bagić Babac - Resetting the Initial Conditions for Calculating Epidemic Spread: COVID-19 Outbreak in Italy. IEEE Xplore. Retrieved Jan 2020.
- [7] Ashutosh S et al, Simha, A., Prasad, R. V., & Narayana, S., "A simple Stochastic SIR model for COVID-19 Infection Dynamics for Karnataka after interventions– Learning from European Trends," arXiv preprint arXiv:2003.11920, March 2020.
- [8] Ahmad Sedaghat, Shahab Band, Amir Mosavi 1,2*, and Laszlo Nada (2020, Nov 18). COVID-19 Outbreak Prediction Using a Susceptible-Exposed-Symptomatic Infected-Recovered-Super Spreaders-Asymptomatic Infected-Deceased-Critical (SEIR-PADC) Dynamic Model: A survey. IEEE Xplore. Retrieved 2021, May 14.
- [9] Mohammad Shanna and Sherief Abdallah (2020). Agent-based simulation for covid-19 outbreak within a semi-closed environment. IEEE Xplore. Retrieved on May 14, 2021.
- [10] J. Hackl and T. Dubernet, "Epidemic spreading in urban areas using agent-based transportation models," *Futur. Internet*, vol. 11, no. 4, pp. 1-15, 2019.
- [11] Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang., "Disease prediction by machine learning over big data from healthcare communities", *Ieee Access*, 5:8869–8879, 2017.
- [12] Dhiraj Dahiwade, Gajanan Patle, and Ektaa Meshram, "Designing disease prediction model using machine learning approach", In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pages 1211–1215. IEEE, 2019.
- [13] Pahulpreet Singh Kohli and Shriya Arora, "Application of machine learning in disease prediction", In 2018 4th International Conference on Computing Communication and Automation (ICCCA), pages 1–4. IEEE, 2018.