Epidemic Spread Simulation (SIR Model)

A Project Component for
Simulation and Modelling (UCS751) by:

1. Yash Rohilla   102203830
2. Kashish Agarwal   102203999
3. Vanshika   102383067

Under the guidance of
Dr.Sanjeev Rao

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Department of Computer Science and Engineering

November 2025

Thapar Institute of Engineering and Technology, Patiala

Contents

1   Introduction

1.1   Background

Epidemic spread modeling is an essential area of study in simulation and computational epidemiology. It allows researchers to understand and predict how infectious diseases spread through populations under varying conditions. After the COVID-19 pandemic, such models gained significant attention as they help visualize the impact of preventive measures such as vaccination, quarantine, or lockdowns. The SIR Model—short for Susceptible, Infected, and Recovered—is one of the most widely used mathematical models to study disease transmission. It divides the population into three groups and uses transition equations to simulate how individuals move between these states over time.

1.2   Introduction to Problem Statement

Traditional epidemiological models often rely on mathematical equations and lack visual intuition. This project aims to develop a visual and interactive simulation of disease spread using the SIR model. The simulation represents individuals as moving dots within a bounded space, where infected individuals can transmit the disease to nearby susceptible ones. Over time, infected individuals either recover or remain susceptible depending on model parameters. The simulation aims to demonstrate the impact of interventions like vaccination rates or movement restrictions (lockdowns) on the infection curve and population health.

2   Literature Review

Reference	Technique	Key Insights	Limitations
[1]Kermack & McKendrick (1927)	Mathematical SIR Model	Introduced the foundational SIR framework dividing population into Susceptible, Infected, and Recovered compartments.	Assumes homogeneous mixing and ignores spatial movement.
[2] Hethcote (2000)	Differential Equation Modeling	Provided detailed mathematical analysis of infectious disease spread and reproduction numbers.	Difficult to visualize and lacks stochastic or real-world dynamics.
[3] Ferguson et al. (2006)	Agent-Based Simulation	Modeled epidemic spread with individual agents to capture realistic interactions.	Computationally intensive and requires parameter tuning.
[4] Del Valle et al. (2010)	Network-Based Epidemic Model	Represented population as nodes in a network to simulate contact-based transmission.	Complex for large populations; data dependency high.

3 Research Gaps

- Focus primarily on mathematical accuracy, not visual understanding.

- Lack interactivity and real-time visualization.

- Do not incorporate control measures like vaccination or lockdown simulation. This project bridges these gaps by providing a visual, intuitive, and parameter-driven simulation that can effectively demonstrate epidemiological dynamics to students and researchers.

4 Problem Formulation

- Represent a population of individuals (as moving particles).

- Randomly assign infection to a subset of the population.

- Define transmission probability when infected and susceptible individuals come into contact.

- Include a recovery rate after a fixed number of time steps.

- Allow optional interventions such as vaccination percentage or lockdown restrictions.

5 Objectives

- To implement the SIR model using Python for visual epidemic simulation.

- To demonstrate disease spread dynamics through animation.

- To observe the impact of lockdowns and vaccination levels on infection curves.

- To provide an educational and analytical tool for understanding epidemic control strategies.

6 Methodology

6.1 Algorithms

The SIR model uses three key states:

- **S (Susceptible):** Individuals who can catch the disease.

- **I (Infected):** Individuals currently carrying and transmitting the disease.

- **R (Recovered):** Individuals who are immune and no longer transmit the disease.

Algorithm Steps:

- Initialize a population of N individuals with random positions.

- Randomly infect a small portion (e.g., 5%) of the population.

- For each time step:

  - Move each individual randomly within the bounded area.
  - If a susceptible individual comes within a threshold distance of an infected one, infection occurs with probability p_{infect}.
  - Infected individuals recover after a set duration (t_{recover}).

- Continuously update and visualize the counts of S, I, and R individuals.

- Plot the **infection curve** using Matplotlib over time.

- Implement optional controls for lockdown (restricted movement) or vaccination (initial immunity).

Tools Used:

- Python for simulation logic.

- Matplotlib for real-time infection curve plotting.

- Pygame for animation of individuals as moving dots.

7 Datasets

No external dataset is required since the simulation generates its own synthetic data. However, the system records internal variables such as:

- Number of susceptible, infected, and recovered individuals per time step.

- Infection and recovery rates.

- Effects of varying parameters (infection probability, movement speed, vaccination rate, etc.).

8 Results and Discussion

The simulation successfully visualized the spread of infection in a closed environment. Key observations include:

- With no intervention, the infection spreads rapidly, peaking early before most individuals recover.

- Implementing lockdown reduces contact frequency, flattening the infection curve.

- Introducing vaccination significantly decreases the infection peak and accelerates herd immunity.
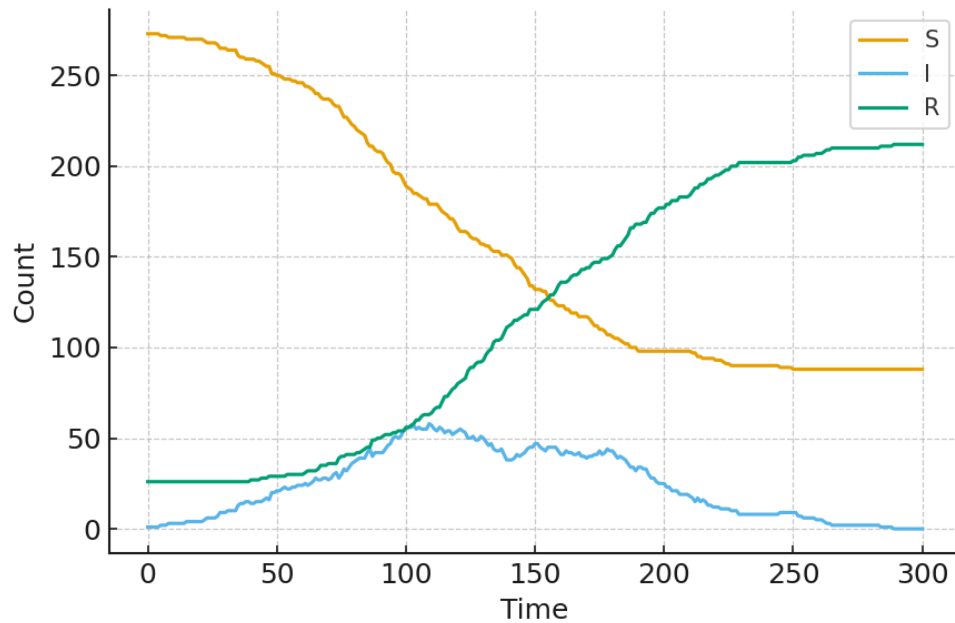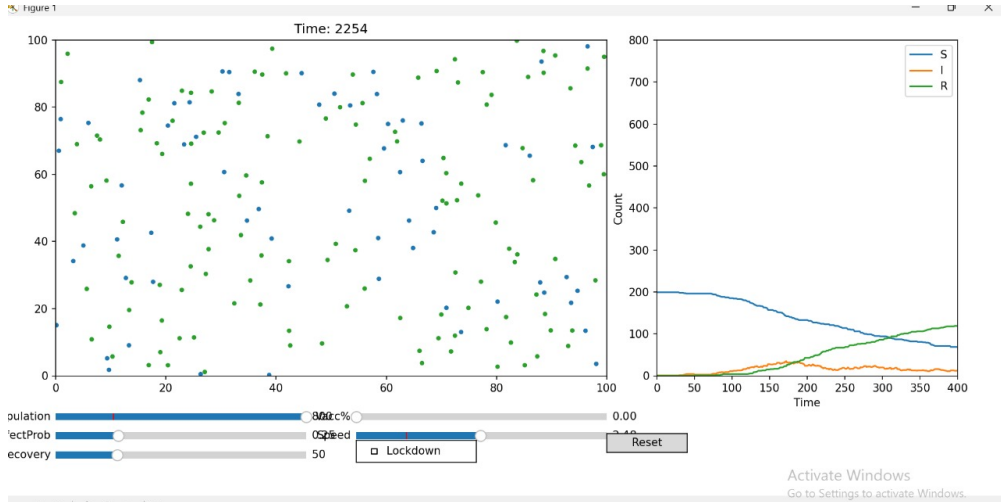
9   Conclusion

This project demonstrates how computational simulation can effectively model and visualize epidemic spread. Using the SIR framework, it provides insights into disease transmission and recovery patterns. The interactive and visual nature of the project makes it a valuable

educational tool, highlighting the significance of interventions like vaccination and lockdowns in managing outbreaks.

10   References

1. Kermack, W.O., & McKendrick, A.G. (1927). A Contribution to the Mathematical Theory of Epidemics. Proceedings of the Royal Society A.

2. Hethcote, H.W. (2000). The Mathematics of Infectious Diseases. SIAM Review.

3. Python Documentation: https://docs.python.org/

4. Matplotlib Documentation: https://matplotlib.org/

11   Future Work

1. Incorporate real-world datasets for parameter tuning.

2. Introduce additional compartments (e.g., SEIR model with Exposed class).

3. Add graph-based social networks to represent realistic contact structures.

4. evelop an interactive UI to allow users to modify parameters dynamically.