

CS 575 Project 1

Riley Sinema

February 10, 2026

Abstract

This report analyzes the propagation of a contagion using the SEIR model across various network topologies, including complete, lattice, scale-free, and real-world (Infect-Dublin) graphs. I configured a custom infectiousness setting to experiment how the contagion spread is affected by the lower probability of infectiousness on contact and a longer infectiousness period. I learned that the spread of the contagion in scale-free networks are sensitive to that probability.

1 Introduction

Simulating the spread of a contagion across different networks provides a unique perspective on how topology influences global dynamics—a process I found particularly insightful given the real-world context of recent global health events. Although the simulation parameters were selected for comparative analysis rather than clinical precision, the project served as a valuable exercise in modeling epidemic propagation.

I simulated the spread of a pathogen using the SEIR model across several network topologies generated with the NetworkX Python library. By varying configurations of infectiousness and connectivity, I analyzed how structural properties affect transmission. This report outlines the chosen experimental conditions, proposes hypotheses based on network metrics, and discusses the results of the resulting simulations.

All figures for the experiments are found in the Appendix. Code for this project is on my GitHub.¹

2 Experiment Conditions

2.1 Scale-Free Network Parameters

For the graphs created by the Barabasi-Albert algorithm, the value of m that I used was 2. A value of 2 makes the resulting graph more dense than a value of $m=1$. Using $m=1$ makes the network more ‘tree-like’ [1]. This would cause the contagion to spread much more slowly.

2.2 Parameters Used in Experiments

Table 1 shows the parameters that I chose for the other experiments. I changed the infection probability to 0.06 and the infectious duration distribution to have a mean of 4.5.

¹<https://github.com/rsinema/seir-sim>

Table 1: Contagion Parameters for Experimenting

| Exposed Duration | Infectious Duration | Infectiousness |
|------------------|---------------------|---|
| $\mu = 1$ | $\sigma = 1$ | $\mu = 4.5$ $\sigma = 0.105$ $p_{1c} = 0.06$ $\beta = -0.00503$ |

3 Hypotheses

Based on the structural metrics of the generated topologies, I propose the following hypotheses regarding the propagation of the contagion under both the default and custom parameter configurations:

3.1 Complete Graph

I hypothesize that the contagion will spread to all nodes in both parameter settings. Because the graph is perfectly dense (which is why it is a complete graph), every node acts as a potential “super-spreader” with a direct edge to every other agent. Even in the custom configuration where infectiousness may be lower, the complete density of the graph makes the virus perfectly efficient at spreading, leaving zero uninfected individuals.

3.2 Lattice Graph

For the lattice graph, I hypothesize that the first parameter configuration will result in a broad but slow spread, while the custom configuration will leave a significant portion of the population uninfected. With a density of only 0.036 and a maximum degree capped at 4, the lattice lacks hubs necessary for rapid transmission. The absence of high-degree nodes means propagation is limited to local neighborhoods, significantly increasing the time-to-peak.

3.3 Scale-Free (Barabasi-Albert) Networks

The scale-free graphs will spread the contagion more efficiently than the lattice due to the preferential attachment mechanism. The presence of hubs (nodes with high degree) will act as catalysts for rapid transmission. However, because of the power-law distribution, many low-degree nodes are only connected to the rest of the network with a few edges. I hypothesize that in the custom (less infectious) configuration, these eccentric nodes will often remain uninfected, and the time-to-steady-state will increase.

3.4 Infect-Dublin Graph

I hypothesize that the Infect-Dublin network will show spreading patterns similar to the scale-free networks, with slightly better efficiency. The scale-free network (410 node configuration) has an average degree of approximately 4, while the Infect-Dublin network also has a average degree of approximately 13.5. The density of this graph is also much higher (relatively) than the scale-free network, with a density of approximately 0.0329 compared to approximately 0.0097.

4 Discussion

The simulation data largely supports the initial hypotheses regarding topological influence on propagation. As shown in Table 2, the Complete graph serves as the upper bound for efficiency, reaching

peak infection in under 10 time steps. In contrast, the Lattice graph's high diameter (18) and low average degree (3.6) resulted in the slowest propagation and the highest susceptibility to changes in contagion parameters.

Table 3 shows that the scale-free structure is highly sensitive to the reduced infectiousness; the number of uninfected nodes in the 410-node configuration rose from 16.4 to 94.6, suggesting that low-degree eccentric nodes are protected when the contagion is less infectious. Finally, the Infect-Dublin graph proved to be the most efficient non-complete graph, even with the slower infectiousness configuration the average non-infected nodes was only 15.8 which is the smallest value for the second configuration of parameters.

Table 2: Comparative Network Metrics and Viral Propagation Results

| Network | Max Deg. | Avg Deg. | Diam. | Rad. | Dens. | T-Peak (t) | Peak Infections | T-Steady (t) | Uninf. (N) |
|------------------------------|-------------|-------------|-------|------|-------|---------------|-----------------|-----------------|---------------|
| Complete | 99.0 | 99.0 | 1.0 | 1.0 | 1.000 | 9.5 | 83.5 | 45.2 | 0.0 |
| Lattice | 4.0 | 3.6 | 18 | 10.0 | 0.036 | 21.7 | 33.9 | 70.9 | 3.1 |
| Scale-Free (100) ($m = 2$) | 27.1 | 3.92 | 5.5 | 3.2 | 0.040 | 18.7 | 48.2 | 64.0 | 4.0 |
| Scale-Free (410) ($m = 2$) | 52.1 | 3.98 | 6.9 | 4.0 | 0.010 | 18.1 | 178.7 | 78.1 | 16.4 |
| Infect-Dublin | 50.0 | 13.5 | 9.0 | 6.0 | 0.033 | 13.1 | 255.6 | 78.3 | 3.8 |

Average metrics across 10 runs for each graph. *T-Peak* is to time-to-peak infections; *T-Steady* is to time-to-steady-state.

Table 3: Viral Propagation Results of Custom Parameter Configuration

| Network | T-Peak (t) | Peak Infections | T-Steady (t) | Uninf. (N) |
|------------------------------|---------------|-----------------|-----------------|---------------|
| Complete | 10.3 | 81.7 | 55.4 | 0.0 |
| Lattice | 17.2 | 19.2 | 79.8 | 32.9 |
| Scale-Free (100) ($m = 2$) | 19.2 | 27.9 | 72.2 | 25.2 |
| Scale-Free (410) ($m = 2$) | 21.2 | 108.5 | 82.4 | 94.6 |
| Infect-Dublin | 15.1 | 206.2 | 70.9 | 15.8 |

Average metrics across 10 runs for each graph. *T-Peak*: time-to-peak infections; *T-Steady*: time-to-steady-state.

5 Future Work

For this project, I wish that we had some data from BYU to simulate the spread of a contagion through a student network. I think that it would be so interesting to apply this to a real-world network that is localized to our university. Another thing that I think would be interesting to study is the impact on setting specific nodes to start with the contagion in the scale-free and Infect-Dublin networks, and how fast the contagion would spread if it started on low-degree vs. high-degree nodes.

References

- [1] Wikipedia contributors. Barabási-albert model — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model#Degree_

distribution, 2026.

A AI Disclosure

I wrote most of the code for this project. I used Gemini and Claude to help me with drawing the graphs, creating the animations, and creating the plots. Ever since AI tools have become effective at coding, I have basically lost the ability to write with matplotlib from scratch.

B Figures

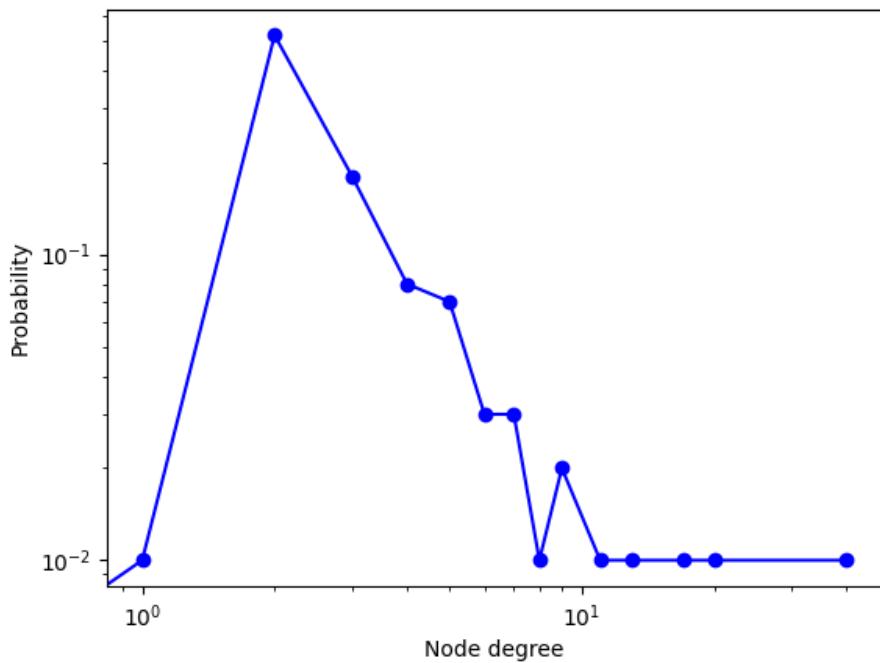


Figure 1: Scale-free (100 node) network degree distribution

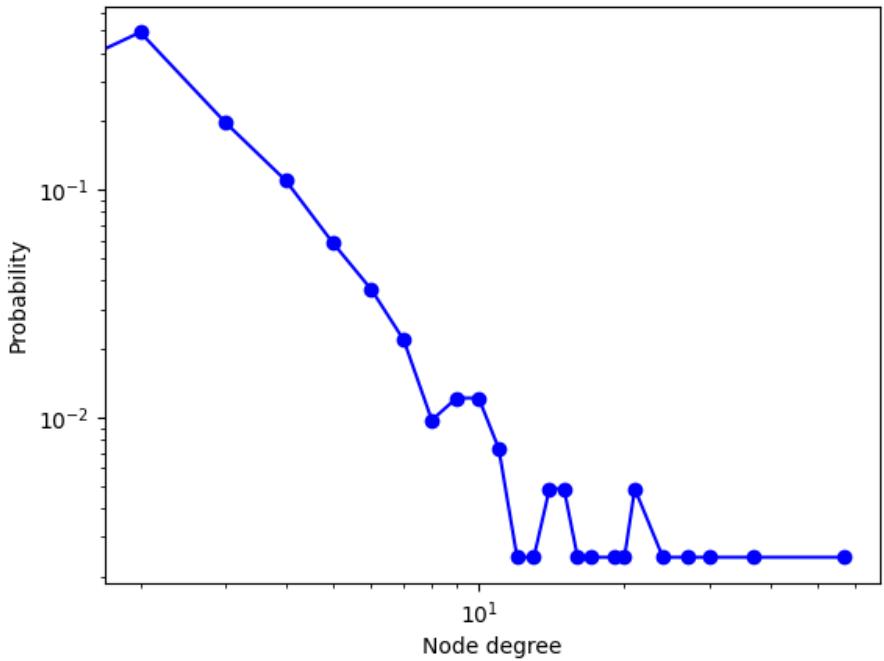


Figure 2: Scale-free (410 node) network degree distribution

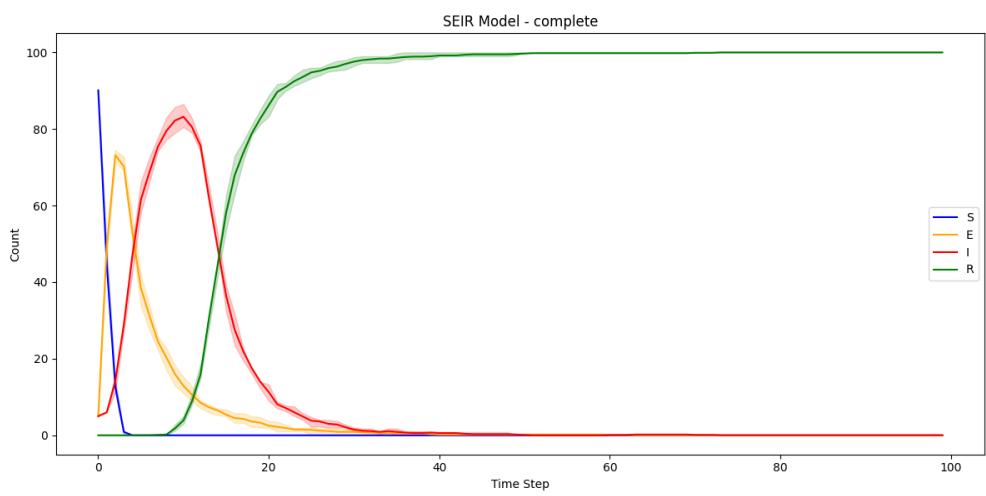


Figure 3: Complete network time series

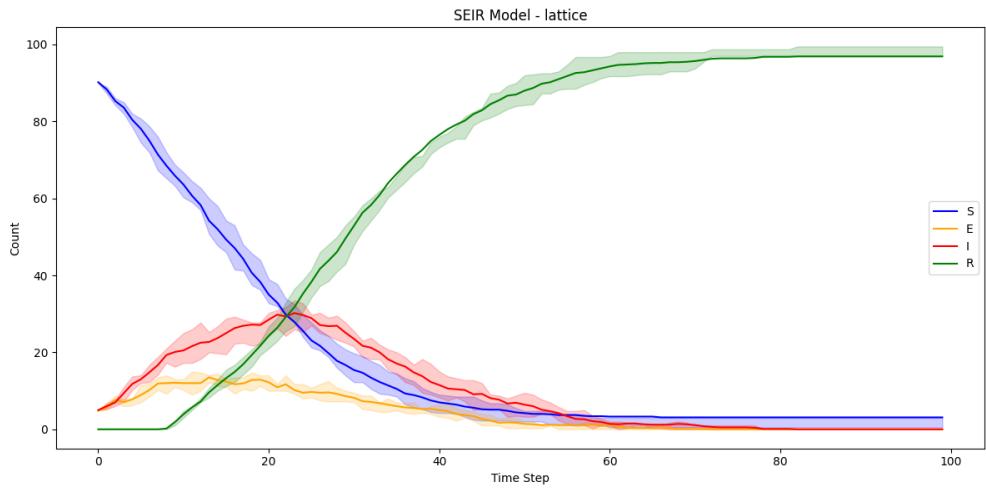


Figure 4: Lattice network time series

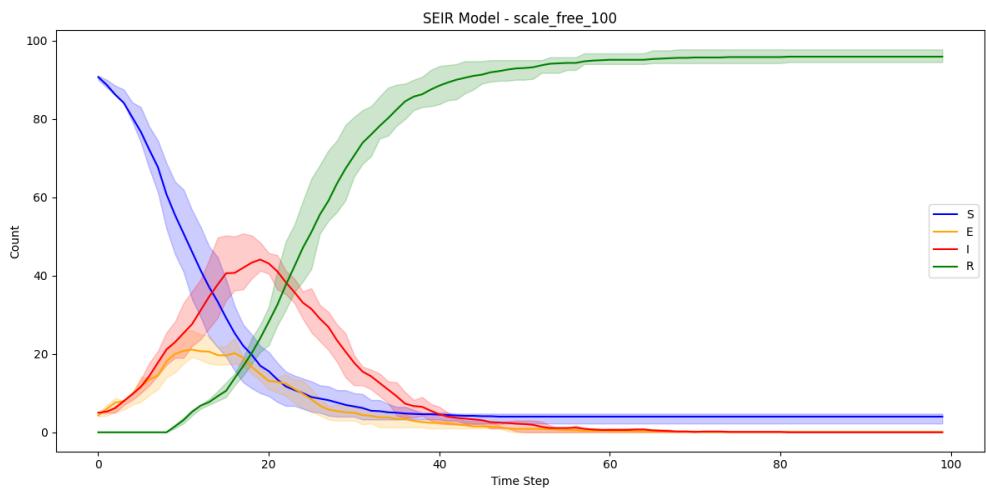


Figure 5: Scale-free (100 node) time series

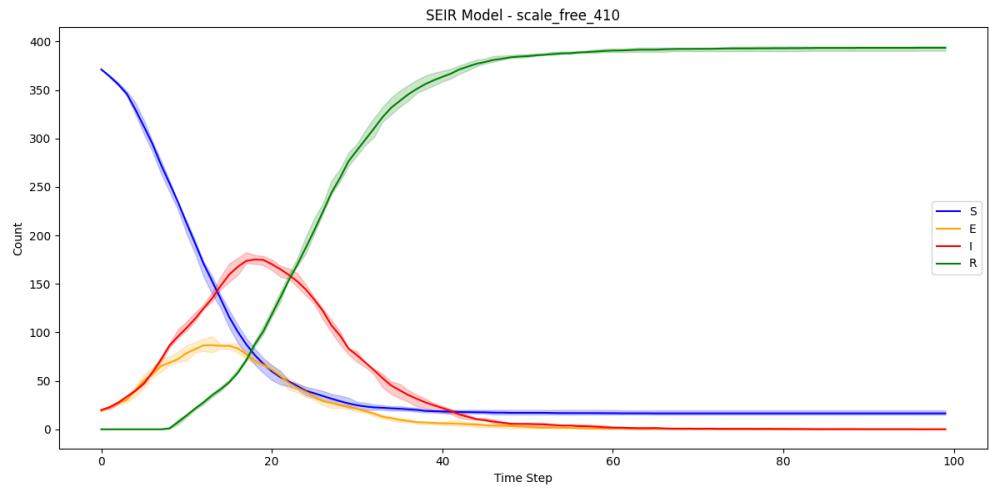


Figure 6: Scale-free (410 node) time series

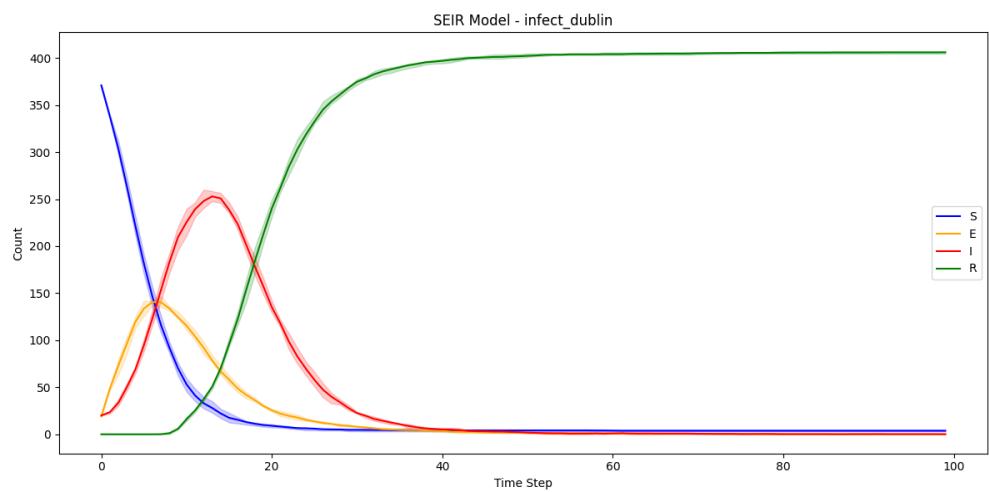


Figure 7: Infect-Dublin network time series

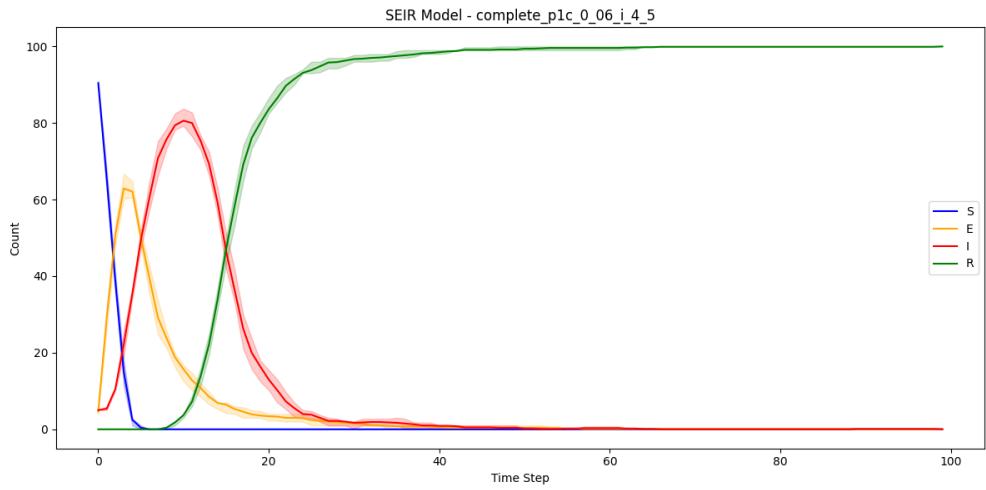


Figure 8: Complete network time series for custom parameters

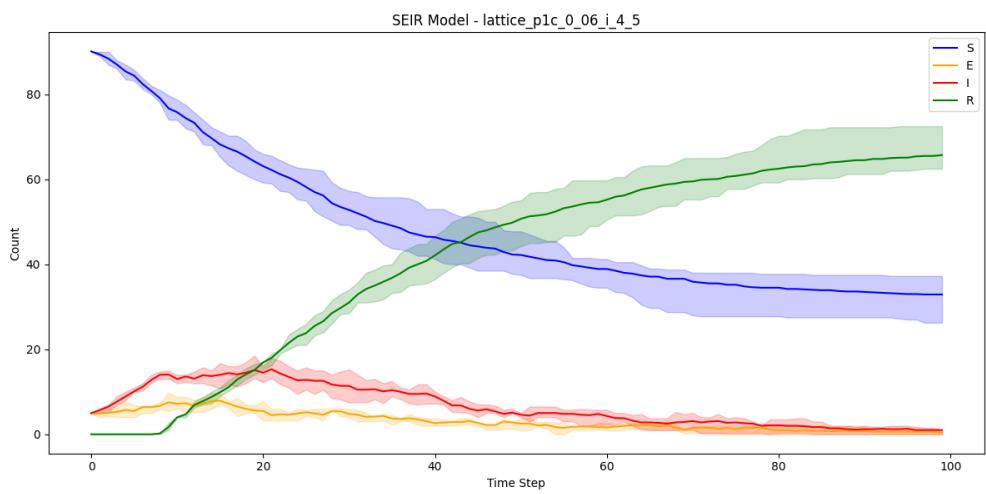


Figure 9: Lattice network time series for custom parameters

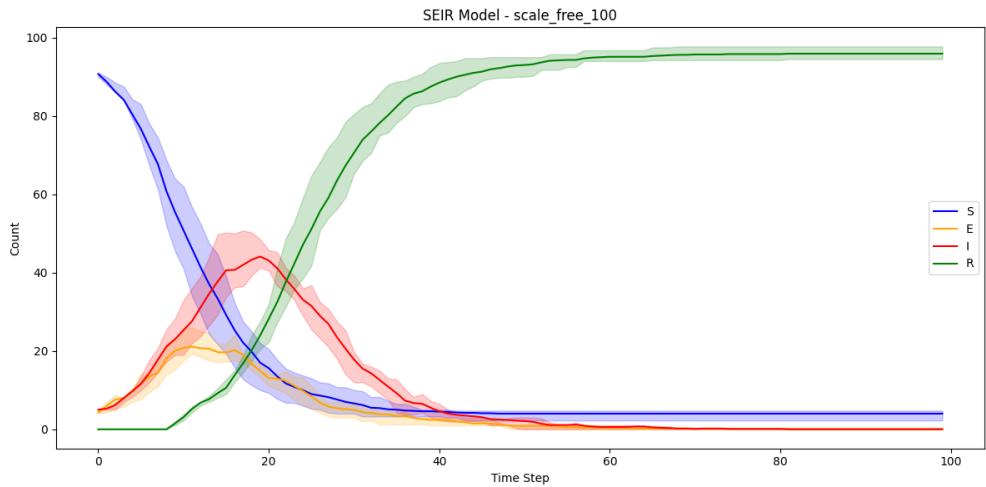


Figure 10: Scale-free (100 node) time series for custom parameters

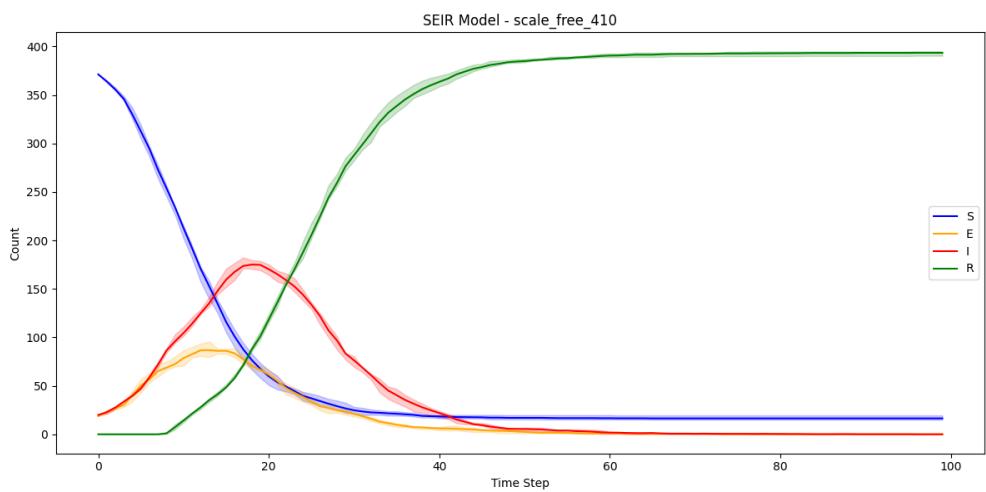


Figure 11: Scale-free (410 node) time series for custom parameters

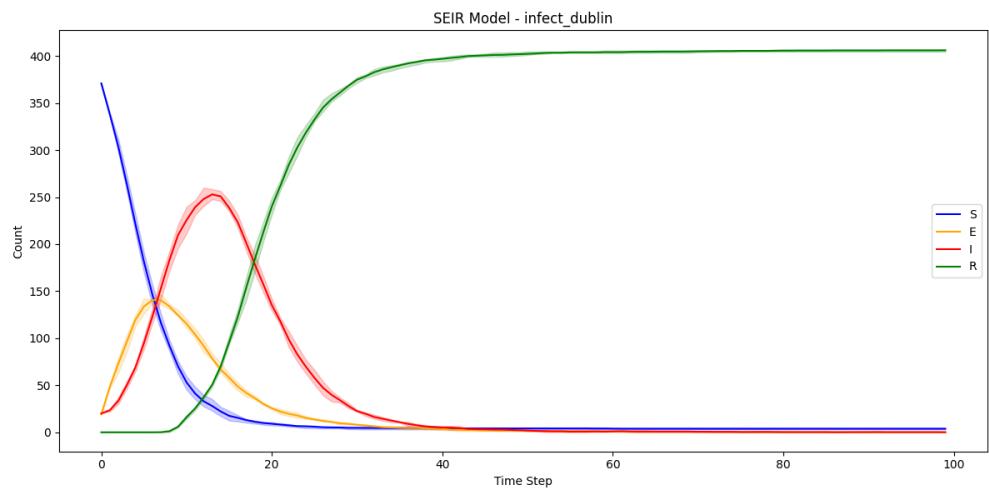


Figure 12: Infect-Dublin network time series for custom parameters