ESTIMATING EPIDEMIC INFECTION SPREAD USING GRAPH NEURAL NETWORKS

ARIF AHMAD[1] 190110010
EESHAAN JAIN[2] 19D070022
TUSHAR NANDY[3] 190020125

ABSTRACT

Epidemic forecasting provides an opportunity to predict geographic disease spread as well as case counts to better inform public health interventions when outbreaks occur. In this project, we firstly explore the work already done in the domain of epidemics, using both classical machine learning and deep learning. Further, we demonstrate the spread of epidemics using the famous SIR model on random graphs generated by the Erdős–Rényi-Gilbert, Watts-Strogatz and Barabási–Albert models. Finally, we construct a graph neural network followed by a fully connected network to predict the category (safe/infected/recovered) a person might be in during an epidemic.

1 INTRODUCTION

The analysis of networks and their dynamics, form a class of problems that can be studied best if modeled as a system whose state assumes values on a graph rather than in Euclidean space [1]. One such problem in network dynamics is modeling and predicting the spread of an epidemic in a fixed population [2]. Describing the network as a graph with each node representing a subject or group of subjects, and the edges of the graphs as contacts between the subjects is an effective way of modeling the problem as a graph. Simple rules are used for describing the spread of the disease on the network.

Several works have dealt with the much more common problem of extracting robust statistics on the total amount of subjects being infected, recovered, hospitalized, etc (listed in [1]). However, as observed in the recent epidemic of Covid-19, it is often desirable to know the state of the epidemic on a node level, for active intervention to prevent generation of clusters, implement isolation policies and develop feedback strategies. Also, it is often impractical or even impossible, to know and monitor the state of each node (person). Hence, developing algorithmic strategies to infer about the state of the population from limited number of measures is extremely desirable.

[1, 2, 3] *Department of Electrical Engineering, IIT Bombay*

2 BACKGROUND INFORMATION AND RELATED WORK

2.1 Random Graphs

2.1.1 *Erdős–Rényi-Gilbert model*

The model, also known as $G(N, p)$ model was introduced in [3], and in this, a graph is constructed containing N nodes by connecting each pair of the N nodes with probability p. Although the generation of such graphs is simple and often proves to be powerful, there are two drawbacks which doesn't allow them to represent real-world graphs:

① They are void of local clustering and triadic closure giving them a low clustering coefficient.

② The degree distribution of $G(N, p)$ converges to a Poisson distribution and not the power-law which is observed in real-world networks.

2.1.2 *Watts-Strogatz Model*

The WS model (introduced in [4]) is an extension of the ER model motivated by the *Small World Property* and *High Clustering*. The model (also called the small-world model) is a compromise between the regular lattice and a random network, and thus solves ①. But since it is an extension of the ER model, it predicts a Poisson-like bounded degree distribution and hence still fails to solve ②. Say that we want a N node graph with K as the average degree.

1. Choose $N > K > 1$ and β such that $0 \leqslant \beta \leqslant 1$
2. Construct a regular ring lattice with each node connected to $K/2$ nodes on each side
3. For each node i, for every right-edge ξ, rewire it with a node chosen uniformly at random from the $N-1$ nodes with probability β

Algorithm 1: Generation of WS graphs

2.1.3 *Barabási–Albert model*

It is seen that usually random graph models use a fixed N, but real-world networks can have growth of nodes. The BA model [5, Chapter 5] utilizes the notion of growth and preferential attachment to generate random graphs in the following manner:

1. Start with m_0 nodes, between which the edges are chosen arbitrarily (but each node has at least one edge)
2. At each step, add a new node with $m \leqslant m_0$ edges to previously present nodes
3. The probability π_k that an edge of the new node connected to node τ already present in the graph is $\pi_k = \bar{k}_\tau$ where $\bar{\cdot}$ denotes that we normalize the degree

Algorithm 2: Generation of BA graphs

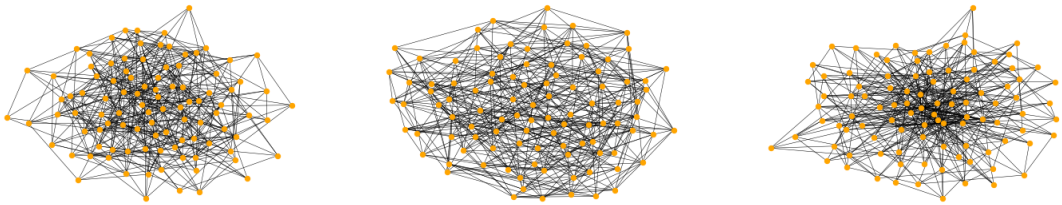
The BA random graph model generates scale-free networks and hence solves ②, but fails to solve ①.

2.2 The SIR Model

Our work is based around epidemic dynamics modeled using the Kermack-McKendrick theory [6], and this splits the population surveyed under an epidemic into three categories: **S** for Susceptible, **I** for Infected and **R** for recovered. We treat the epidemic as a continuous-time Markov Chain [7] giving the following transitions

$$(S, I, R) \xrightarrow{\beta SI} (S-1, I+1, R) \qquad (S, I, R) \xrightarrow{\gamma I} (S, I-1, R+1) \tag{1}$$

A parameter of importance is the reproduction number R_0. It is defined as the product of the contact rate (β) and latent period (γ^{-1}). An epidemic will take off only if $R_0 \geqslant 1$ and it should be noted that the



(a) ER graph with $p = 0.1$ **(b)** WS graph with $K = 10$, $\beta = 0.6$ **(c)** BA graph with $M = 5$

Figure 1: Sample 100 node random graphs generated from the three models described in 2.1

spreading dynamics is often seen to depend mainly on R_0. After surveying and following [8] we can say that $R_0 \approx 2$ is a reasonable choice for epidemic spreading.
For such a contagion process on a graph $\mathcal{G} = (V, E)$ with adjacency matrix $\mathbf{A}$, the SIR dynamics are specified in [9]. Let S_i, I_i and R_i be the average probabilities of node i being in each of the respective states (and $R_i + S_i + I_i = 1$).

$$\frac{dS_i}{dt} = -\beta \sum_j A_{ij} I_j S_i \qquad \frac{dR_i}{dt} = \gamma I_i \qquad \frac{dI_i}{dt} = -\left(\frac{dS_i}{dt} + \frac{dR_i}{dt}\right) \tag{2}$$

When we are in early stages of the infection $S_i \approx 1$ and the infection spread can be modeled approximately in the following form [10]:

$$I_i(t) \approx \sum_j \exp[t(\beta \mathbf{A} - \gamma \mathbb{I})]_{ij} I_j(0) \approx \exp[(\beta\lambda_1 - \gamma)t]\left(v^{(1)} I(0)\right) v_i^{(0)} \tag{3}$$

where λ_1 is the largest eigenvalue of $\mathbf{A}$ and $v^{(1)}$ is the corresponding eigenvector.
Note: The SIR model is one of the simpler models to describe epidemic dynamics over networks. It doesn't account for exposure, birth, quarantine, vaccination, birth or re-susceptibility. In spite of that, the model is powerful and is widely used in modeling epidemics. Some alternatives to the model can be SIS, SIRD, SIRV, SEIR, MSEIRS etc.

2.3 Graph Neural Networks

Networks or graphs have emerged as useful models to represent complex interconnected systems and data defined on them. It has been shown that graph neural network based deep learning methods have shown better performance over a wide variety of tasks such as node-classification [11] and link-prediction [12].

2.3.1 Message Passing

The idea behind GNNs is that we can couple the input node's signal with propagation of information from the node's neighbors to better inform the future hidden state of the original input. The message-passing framework has been described in [13], and operate on undirected graphs with node features x_v and edge features e_{vw}. Consider differentiable functions M_t (message function) and U_t (update function). During each step, we update the hidden state and the messages passed as follows:

$$m_v^{t+1} = \sum_{w \in \mathcal{N}(u)} M_t(h_v^t, h_w^t, e_{vw}) \tag{4}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{5}$$

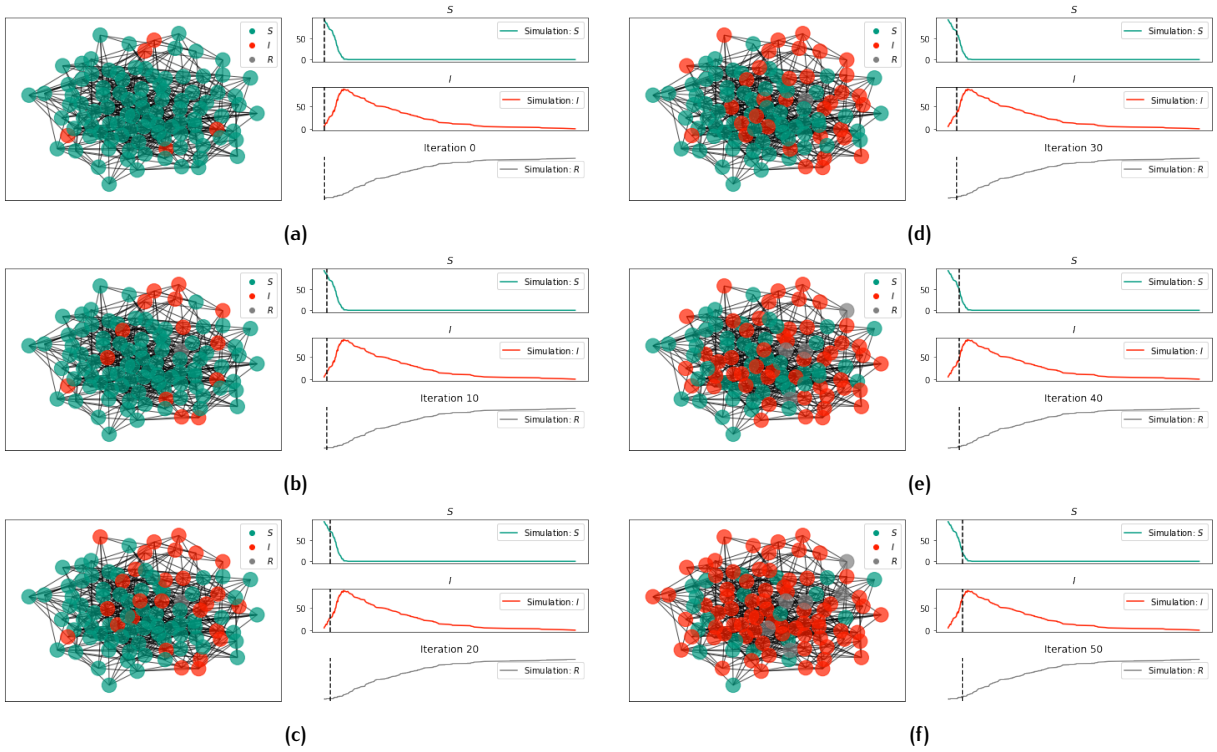


Figure 2: SIR dynamics over 50 days on a 100 node ER graph starting with 5 infected nodes

2.3.2 *Graph Convolutional Network*

The Graph Convolutional Network outlined in [11] employs the symmetric-normalized aggregation and the self-loop update approach. The message-passing function in this case has the corresponding functions as

$$M_t(h_v^t, h_w^t) = \frac{A_{vw} h_w^t}{\sqrt{k_v k_w}} \qquad U_v^t(h_v^t, m_v^{t+1}) = \mathrm{ReLU}(W^t m_v^{t+1}) \tag{6}$$

where k_i denotes the degree of node i.

2.4 Previous Work

The utility of graphs in studying epidemics has been realized in various forms. The following sections shed some light on previous research that have used graphs in different ways to minimize the spread of a disease.

2.4.1 *Identification of super-spreaders*

By virtue of their locations, each node in a network commands different influence over the rest of the network. In [14], the authors aim to assess the importance of each node in an epidemic by calculating the exact outbreak size Ω. The features of the nodes are a combination of different centrality measures, which are used to train a supervised learning model to predict Ω for every node. The study is conducted over all isomorphic graphs of $|V| \leqslant 10$.

2.4.2 Forecasting Cases

Spatio-temporal graphs are a kind of graph that model connections between nodes as a function of both time and space. By representing the counties in USA as nodes in a network, the authors of [15] aim to forecast infections in a network by incorporating mobility data for intra-region and inter-region travel. The network is as a multi-graph where nodes represent intra-region mobility, spatial edges represent inter-region connectivity and temporal edges represent node features through time. The goal is to predict the increase in the number of cases per county (represented by a node) using GNNs.

2.4.3 Controlling the Spread over a Network

Minimizing the spread of a disease under the constraints of limited resources is synonymous to the problem of controlling a partially-observed dynamic process on a graph by a limited number of interventions. In [16], the optimization goal is to limit the spread of the disease by intervening and testing a few nodes at a time. The approach is to use GNNs to predict the nodes which will maximize the spread of the disease, and use RL to rank and test the top k node. The RL framework and GNNs function in an alternate manner at even and odd time-steps. Two GNNs are used to collectively score the nodes. One network scores according to local diffusion model while the other is used for long-range information propagation. The score of a node is affected both by propagation dynamics and by information available to the agent. The agent sequentially applies the suggested action and logs the (state, action) tuple in an experience replay buffer. The training of the model is based on the Proximal Policy Optimization loss term.

2.4.4 Contact Tracing and Source Identification

The identification of early transmission chains and the reconstruction of the possible paths of diffusion of the virus can be the difference between stopping an outbreak in its infancy and letting an epidemic unfold and affect a large share of a population. In [10], the authors aim to trace the source of a disease using GNNs, motivated by the fact that the contagion dynamics (1) are a special case of Reaction-Diffusion (RD) processes on graphs which is structurally equivalent to GNNs.

3 DATA AND METHODOLOGY

3.1 Generation of Synthetic Data

One of the challenges faced was the availability of datasets. We were unable to find a dataset that catered to our task on an individual level, and thus we generated synthetic data controlled by the SIR dynamics. Firstly, two sets of graphs were generated: first containing 100 nodes with an average degree of 10, and the second containing 500 nodes with an average degree of 30. The first set has 80 graphs generated from each of the models: ER, WS and BA. The second set has 256 graphs generated from each of the models mentioned earlier.
The generation of random graphs has been done using NetworkX [17], and the SIR dynamics on the graph have been simulated using NDlib [18]. For the models, the reproduction number $R_0 = 2$ with $\beta = 0.01$ and $\gamma = 0.005$. The simulation for all graphs has started with 5 infected nodes. The graphs have been saved in the GEXF format with the simulation information containing $S = 0, I = 1, R = 2$ label for 90 days saved in the CSV format. A sample of the simulation is shown in Fig 2.

3.2 Network Architecture and Details of the Training Process

Since this is a node classification problem, our network consists of two parts. The first part consists of 2 layers of GCN convolution which map the 3-dimension initial feature space to a 64-dimension embedding space ($3 \rightarrow 32 \rightarrow 64$). Each of these convolution layers are followed by ReLU activation and 30% dropout for

regularization. The second part consists of two fully connected layers with linear activation for classifying the nodes. The network has three output channels, one for each class, and we pass these through log-softmax function.
The model was coded entirely using PyTorch Geometric and trained using the Adam optimizer available in PyTorch with learning rate 0.0002. We used batch size = 32 for training both networks, i.e. 100 nodes and 500 nodes.

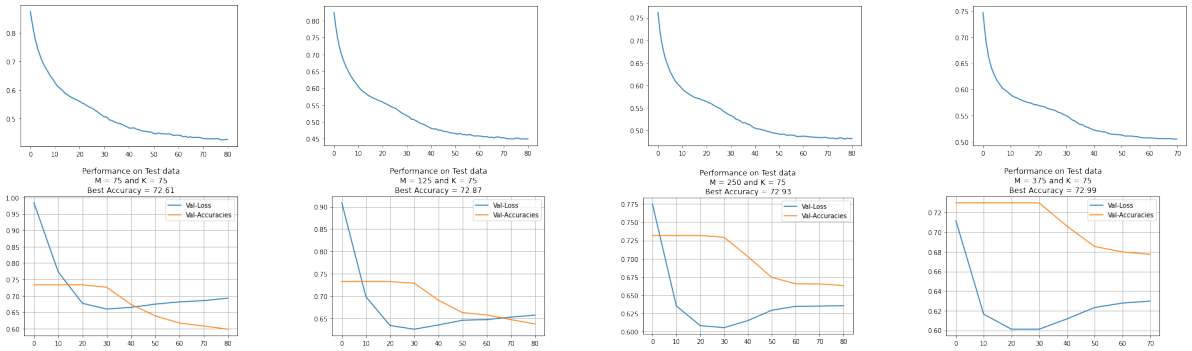
3.3 Accuracy vs Information at Final State

As described previously, the output of the GNN for each node is a 3-dimensional vector with each component representing a score for S, I and R. The state of the node is declared by taking argmax over these scores. The training data is confined to the m known nodes (assuming m in percentage) while the accuracy is calculated over the remaining (100 - m)% of nodes. In effect, the total percentage of nodes with correctly known states is $m + accuracy \times (100 - m)$.

4 EXPERIMENTS AND RESULTS

4.1 WS Model

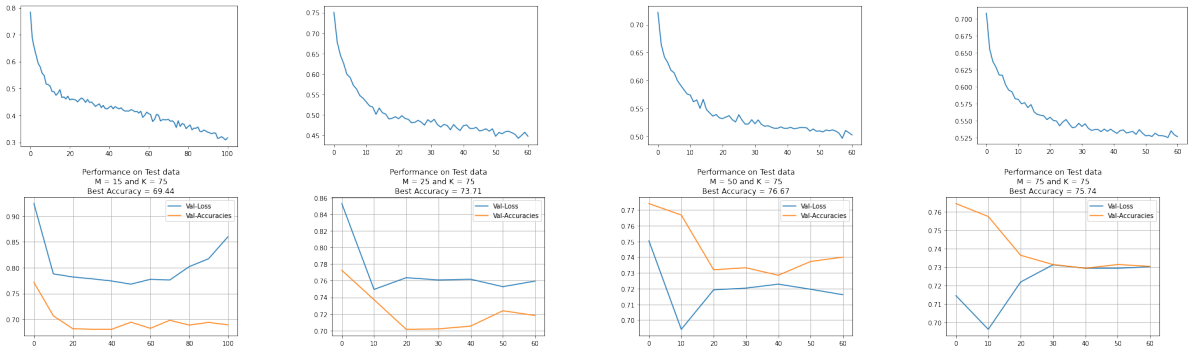
4.1.1 100 Nodes



(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 3: Loss and Accuracy on the WS Random Model with 100 Nodes

4.1.2 500 Nodes

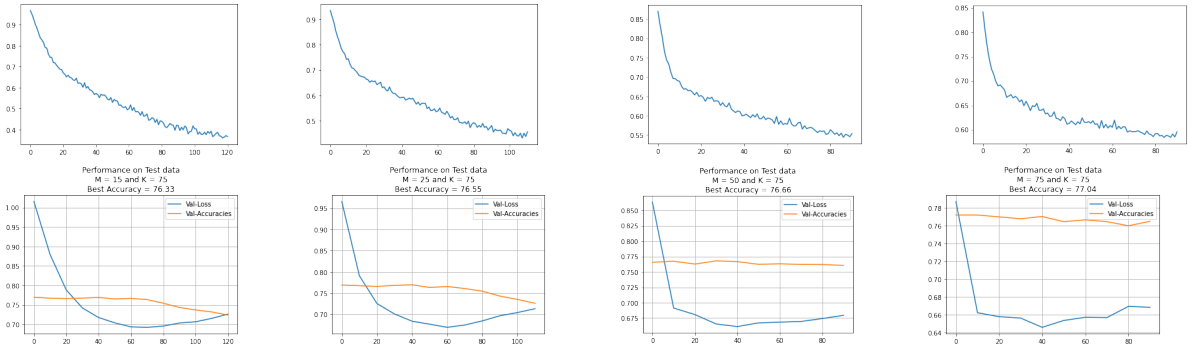


(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 4: Loss and Accuracy on the WS Random Model with 500 Nodes

4.2 ER Model

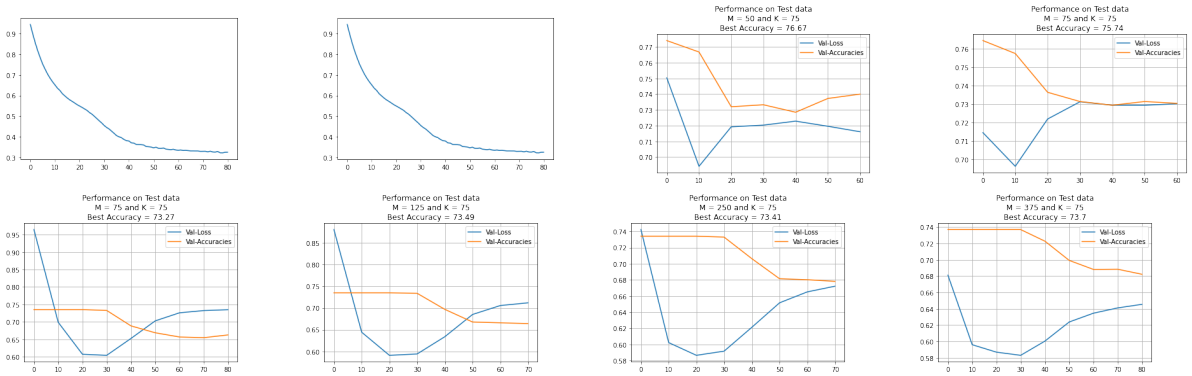
4.2.1 100 Nodes



(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 5: Loss and Accuracy on the ER Random Model with 100 Nodes

4.2.2 500 Nodes

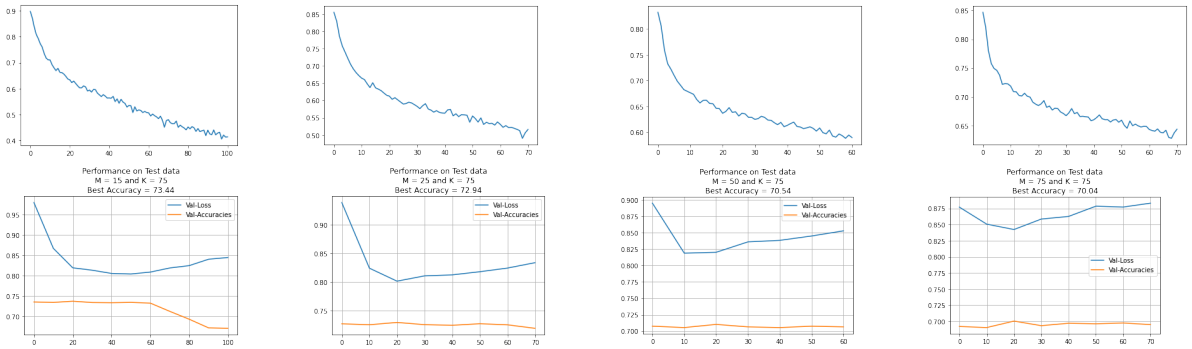


(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 6: Loss and Accuracy on the ER Random Model with 500 Nodes

4.3 BA Model

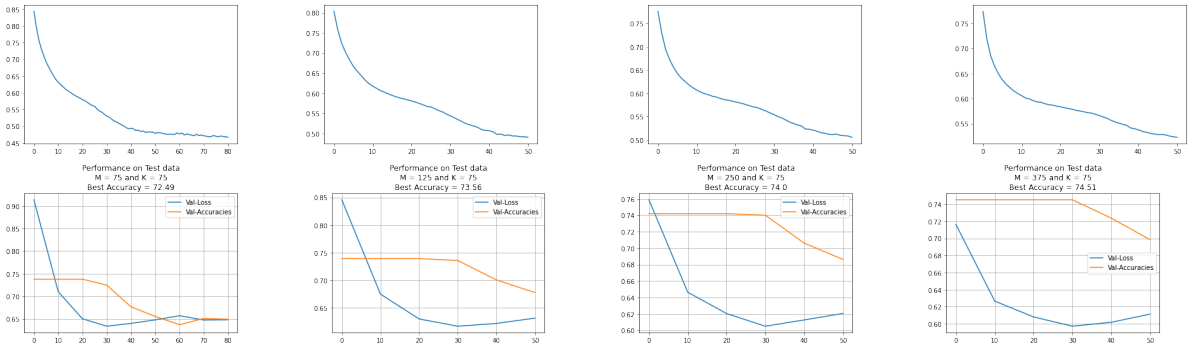
4.3.1 100 Nodes



(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 7: Loss and Accuracy on the BA Random Model with 100 Nodes

4.3.2 500 Nodes



(a) 15% of the nodes **(b)** 25% of the nodes **(c)** 50% of the nodes **(d)** 75% of the nodes

Figure 8: Loss and Accuracy on the BA Random Model with 500 Nodes

5 CONCLUSION AND FURTHER WORK

In this paper, we have highlighted the advantage of GCNs in correctly identifying the states of a large number of unknown nodes from a small subset of monitored population, and reinforced the worth of GCNs in semi-supervised learning problems. We also present the use of GNNs as a model-agnostic framework in epidemiology which can be used on graphs generated by different processes and their capability to scale to a large population. In the following tables we present comparisons of different values of monitored population (m) and the corresponding percentage of correctly identified nodes, averaged over the three generation models.

We have only experimented with random graphs and the SIR model. There are many more complex epidemic models available to be tested that account for vaccination, quarantine, death, birth etc. GNNs can be tweaked to work upon those too. Also, the data generated is not reminiscent of real data, and we can train models on epidemic data, of which the most commonly available one currently is of COVID-19.

% of population monitored (m)	Accuracy on Unknown Nodes	Total Known Nodes
15	0.7413	78.01
25	0.7412	80.59
50	0.7338	86.69
75	0.7336	93.34

Table 1: Percentage of Correctly Identified Nodes for 100 nodes

% of population monitored (m)	Accuracy on Unknown Nodes	Total Known Nodes
15	0.7173	75.97
25	0.7359	80.19
50	0.7338	87.35
75	0.7336	93.66

Table 2: Percentage of Correctly Identified Nodes for 500 nodes

REFERENCES

[1] A. Tomy, M. Razzanelli, F. Di Lauro, D. Rus, and C. Della Santina, "Estimating the state of epidemics spreading with graph neural networks," *arXiv preprint arXiv:2105.05060*, 2021.

[2] I. Z. Kiss, J. C. Miller, P. L. Simon *et al.*, "Mathematics of epidemics on networks," *Cham: Springer*, vol. 598, 2017.

[3] E. N. Gilbert, "Random graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[4] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[5] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.

[6] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[7] P. L. S. Istvan Z. Kiss, Joel C. Miller, *Mathematics of Epidemics on Network*. Springer, 2017, vol. 46.

[8] N. Ferguson, D. Cummings, C. Fraser, J. Cajka, P. Cooley, and S. Burke, "Strategies for mitigating an influenza pandemic," *Nature*, vol. 442, pp. 448–52, 08 2006.

[9] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[10] C. Shah, N. Dehmamy, N. Perra, M. Chinazzi, A.-L. Barabási, A. Vespignani, and R. Yu, "Finding patient zero: Learning contagion source with graph neural networks," *arXiv preprint arXiv:2006.11913*, 2020.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[12] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 5165–5175, 2018.

[13] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[14] D. Bucur and P. Holme, "Beyond ranking nodes: Predicting epidemic outbreak sizes by network centralities," *PLOS Computational Biology*, vol. 16, no. 7, p. e1008052, Jul 2020. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1008052

[15] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion, "Examining COVID-19 forecasting using spatio-temporal graph neural networks," *CoRR*, vol. abs/2007.03113, 2020. [Online]. Available: https://arxiv.org/abs/2007.03113

[16] E. A. Meirom, H. Maron, S. Mannor, and G. Chechik, "How to stop epidemics: Controlling graph dynamics with reinforcement learning and graph neural networks," *CoRR*, vol. abs/2010.05313, 2020. [Online]. Available: https://arxiv.org/abs/2010.05313

[17] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[18] G. Rossetti, L. Milli, S. Rinzivillo, A. Sîrbu, D. Pedreschi, and F. Giannotti, "Ndlib: a python library to model and analyze diffusion processes over complex networks," *International Journal of Data Science and Analytics*, vol. 5, no. 1, pp. 61–79, 2018.