Updated Graph Neural Network to Predict Patient Zero(s)

Aditya Sasanur
asasanur@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Ninaad Lakshman
ninaadlakshman@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

ACM Reference Format:
Aditya Sasanur and Ninaad Lakshman. 2023. Updated Graph Neural Network to Predict Patient Zero(s). In *Proceedings of CSE 8803 (CSE 8803 EPI).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the midst of an outbreak, the imperative task of identifying the outbreak's root cause becomes increasingly pronounced. This endeavor carries significant implications, as pinpointing the origin of the outbreak empowers researchers and scientists to delve into the underlying factors responsible for the initial emergence of the disease. Such insights are invaluable for not only mitigating the ongoing outbreak but also for implementing strategies to contain and prevent future outbreaks from occurring. However, the quest to locate a patient zero, or potentially multiple patient zeros, becomes considerably more complex and intricate as the outbreak unfolds. The complexity arises from the diverse and intricate network of connections through which the disease can propagate.

2 RESPONSE TO MILESTONE COMMENT

We looked back at our original literature review and identified gaps that can be filled by reading through additional research papers. We analyzed new methods of graph neural networks in identifying a patient zero and in other analogous tasks with social media and also looked into more algorithmic approaches. We read through 15 references total per the comment left in the milestone. From the comments, we also realized that our initial code-printed graph neural network architecture lacked clarity in understanding how our architecture was structured, as it was just a list of text. Instead, we created a new graphic that visualizes the different components of our graph neural network and how they interact. We felt this improved the clarity of the graphic. Finally, in terms of utilizing datasets from previous works, we were not able to come across the exact graphs that were used to test and train the models. These testing/training sets however were created synthetically like other resesarch papers on this topic. We have similarly also created synthetic datasets with the specifications that are mentioned later in this paper.

3 PROBLEM

From a computational standpoint, the challenge of identifying a patient zero poses considerable difficulties. As the outbreak progresses, tracing the precise source or sources of the infection becomes computationally expensive and often impractical. This is primarily due to the stochastic and unpredictable nature of disease spread within a dynamic population and network. Consequently, the formidable challenge at hand revolves around not only determining the number of initial patient zeros but also accurately identifying their identities amidst the intricate web of transmission pathways. This multifaceted challenge underscores the need for innovative approaches, such as the one we are pursuing, which harness advanced technologies like graph neural networks (GNNs) to navigate the complexity of disease dynamics and patient zero identification.

4 LITERATURE REVIEW

4.1 Patient 0 localization and foundations

Shelke and Attar's [9] paper on source detection in social networks was a foundational way for our team to understand what factors should be considered when finding a patient zero. They mention several factors from network structure such as its topology to diffusion models used (such as epidemiological models) to centrality measures. For locating patient zeros, these factors are the foundation for future source localization methods. This paper is helpful in listing out what avenues one could pursue when generating a source localization estimator (such as whether to identify a single source or multiple sources). Shelke and Attar do not propose a novel method but rather test out different approaches.

There have been multiple novel methods used for identifying and predicting a patient zero. One of the foundational papers by Shah and Zaman [8] looks into finding the source of a “rumor” in a network (analogous to the source of a disease). To minimize estimation error in their rumor spreading model, they leverage the maximum likelihood estimator. In this case, they use the underlying assumption that the best source is one that maximizes data likelihood in an SI model. This is a strength of the paper, as the algorithm itself is quite explainable in comparison to other neural network counterparts. This method works efficiently in k-regular trees, but one of the underlying weaknesses is that it is computationally inefficient in general graphs (without converting the graph to a tree). This is because computing the summation of the likelihoods of all possible permutations is expensive (which for a regular tree is not the case). However, in realistic scenarios, it is often the case where we are dealing with networks that follow a general graph pattern. Approaches to change general graphs into trees (minimum spanning tree and BFS) would work in formatting this network in a tree, making the task more efficient.

CSE 8803 EPI, September 2023, Atlanta, GA, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The problem of identifying a patient zero has parallels in social media. Finding the source of rumors in social networks is task analogous to identifying a patient zero, though the data may be different. There has been considerable effort placed in identifying the source of rumors in social meida. One of the earlier papers by Chierichetti, Lattanzi, and Panconesi [1] uses an algorithm called PUSH-PULL strategy to spread information and can spread information through all nodes in log squared time complexity. This study is specifically for Preferential Attachment (PA) networks which contain subgraphs that are hard for rumor spreading. These PA networks can model social networks, and their algorithm has applications in social media rumor detection.

4.2 Algorithmic approaches

Another approach involves a modified dynamic message-passing algorithm by Lokhov [4] and other researchers. They derive this dynamic message-passing (DMP) algorithm from a pre-existing belief propagation (BP) method, which estimates marginal probability distribution in a variety of problems. Using this DMP method, they were able to compute a patient zero with high probability (in comparison to other methods such as distance and Jordan centralities). There are some noted weaknesses such as if the graph has self-loops and that there could be better ways to approximate their likelihood outside of mean-field-like approximation. Outside of these minor weaknesses, this is a valid and accurate way to compute patient zeros outside of neural networks as well. We would ideally like to find an even more accurate way to compute patient zero in our graph neural network approach.

Menin and Bauch [5] suggest using generalized simulated annealing algorithms (GSA) to identify the source of an outbreak. Generalized simulated annealing algorithms are used in linear programming contexts that may be non-convex (with potentially multiple local optima); these algorithms involve exploring an entire search space and accept/reject based on a probability derived from an acceptance temperature and the change in the old guess solution with the new guess solution. This unique approach in leveraging GSA to identify patient zero(s) means we essentially treat this problem as an optimization problem with an objective function derived from variations of Susceptible Infected Recovered Susceptible. Menin and Bauch found good accuracy when information about infection status is available in at least 10% of nodes, which means this algorithm can perform under considerable uncertainty (but not more uncertainty than this 10% approximate threshold).

There have also been algorithmic approaches that consider resource limitations on the effectiveness of contract tracing, which frames patient zero identification in a more realistic context. Waniek and other researchers [13] deep dive into the investigating the best possible parameter tuning, which essentially minimizes the amount of contact tracing while maintaining patient zero identification accuracy, on diffusion models. They propose tracing parameters that can be used to balance contact tracing budget with patient zero identification. They test varying tracing parameters on different types of graphs, such as Barabasí-Albert networks, and found that increasing the budget for contract tracing beyond a certain calculation does not greatly increase patient zero identification (essentially diminishing returns).

4.3 Graph neural networks

We looked at Zhou and others' [15] paper to provide a clear idea on how graph neural networks work and are applied. Their paper dives into graph representation learning and provides guidelines on GNN design based on the task at hand and structure of the graph input. Graph neural networks do have broad reaching applications within epidemiology.

Papers like Song and others' [10] publication on COVID-19 infections with graph neural networks to predict influence of current infected patients on future infections. This helped establish the general structure of graph neural networks and step-by-step implementation to handle graph-structure data from a COVID-19 epidemiological standpoint. Their GNN-specific approach, including both graph convolutional networks and graph attention networks, did have success with capturing relational information between nodes which helped its accuracy. Their research provides a clear baseline and functionality of epidemiological graph neural networks in the case of forecasting. Their paper focused on whether each infected node had an effect on each subsequent round of infection. However, it did not delve into the opposite case: localizing the infection source by finding a patient zero.

Shah and other researchers [7] have a paper that learn a single patient zero through a graph neural network. They seek to model relationships within the graph training on labeled data which normally use a graph as input. In this case, the paper uses a graph neural network to identify the source of the contagion dynamics of this graph, modeling a contagion using Susceptible-Infected-Recovered (SIR) and Susceptible-Exposed-Infected-Recovered (SEIR). Compared to dynamic message-passing, they state the algorithmic time complexity of a trained GNN in the inference stage has complexity $O(N^2L)$ where N is the number of nodes (with max complexity of N^2 per layer) and L layers. This is polynomial in time, and they state that this time complexity results in a computational speed that's 50-100 times faster than dynamic message-passing.

Their GNN architecture follows a modern GNN design with linear, convolutional, batch normalization, and leaky-ReLu activation layers. They found a high accuracy in the first two weeks of spread with accuracy degrading by 50% after the two week period, and their efficiency exceeded classical methods by over 100 times. Additionally, they specify that they approach the theoretical upper bound in patient zero detection accuracy, but this accuracy only applies to identifying single patient zeros. However, due to the use of a graph neural network, the patterns and computations the network generates are not very explainable compared to dynamic message-passing and other algorithm-based approaches. Furthermore, this paper assumes a single source, whereas additional changes could be made to the neural network to identify multiple patient zeros. In fact, this was an extension they mention in their conclusion.

Ru and other researchers [6] also used GNNs for the purpose of inferring a patient zero; however, they specifically looked into temporal networks, whose links become active only at certain points in time. Their model leverages current states of populations to generate the probabilities of each individual being a source. They develop a backtracking network designed to generate an inverse mapping from final state to initial state, where we can identify which individual was the source. Their dataset focused on temporal networks, and they found that their GNN-based method performed well even in situations where not all the information was provided.

Song, Huang, and Lu's paper [11] on social media rumor detection also leverages GNNs; however, they opt for a specific type of graph neural network named Out-In-Degree Graph Convolutional Networks. Essentially, to accommodate for variance in the number of relations that forward information and the number of relations being forwarded to, they divide the network into an in-degree graph and an out-degree graph. They also add a weighted concatenation, a fully-connected layer, and optimizations for the model architecture. Their model can generalize to a dataset much better than other models due to the flexibility of splitting networks into an in-degree and out-degree graph.

There have been additional efforts to revise GNN structure to fit a rumor detection problem. Xu and other researchers [14] created a novel method known as Hierarchically Aggregated Graph Neural Networks (HAGNN) to learn from text-level granularity in social media applications. Their dataset contains text content, and they want to develop high-level rumor predictions based on this low granularity text representations. Their GNN model also contains a document graph, which is essentially a graph representation where the nodes are source tweets and the edges are co-occurrence relations between words. Using this novel structure, they found a high degree of accuracy in real-world social media datasets that rivaled current methods.

Including attention-based mechanisms into GNNs to better predict patient zero(s) has also been experimented with. Cui and Shang [2] propose an attention and graph-based neural network model in their research paper on social media rumor detection. They suggest that current GNN-based rumor detection approaches fail to consider external knowledge when developing representations. This external knowledge in their paper is created using entity mentions within posts, which are semantic-level knowledge in the GNN. Adding this knowledge-level representations of posts was shown, in many cases, to outperform many current methods in 4 real-world datasets derived from Twitter, PHEME, and Politifact. While we do not have text content necessarily in identifying patient zero(s) as we are looking at individuals rather than posts, it does show that knowledge-level representations in GNNs are another way to reconstruct them for specific tasks.

4.4 Multiple source detection model

Wang and other researchers [12] have created a propagation model, named label propagation-based source identification (LPSI), for the specific case of identifying multiple sources without knowing the underlying propagation model. This was done algorithmically in both an iterative and convergence fashion, which both yielded high accuracy. This paper explained that the convergence method for LPSI did suffer from a large time complexity but the iterative method was faster than other counterparts.

4.5 Variational autoencoder

Ling and others found variational autoencoders [3] to have a great use in graph diffusion problems such as finding the source. After training their source localization variational autoencoder (SL-VAE), they plan on predicting an optimal diffusion source given the observation. This SL-VAE is an ensemble of multiple information propagation models, including a graph neural network. Essentially, their SL-VAE leverages a GNN among others coupled with their decoder and encoder, which are hallmarks of a variational autoencoder. They tested their SL-VAE performance using Susceptible-Infected (SI) and SIR models. For SI and SIR, even though the immunity of individual nodes brought more randomness to the localization task, they noticed that SL-VAE has the best performance in comparison to other models. This SL-VAE also has the ability to predict multiple sources as their predicted optimal diffusion source variable can hold more than one source. This SL-VAE approach has a clear advantage of accuracy but does suffer from an additional layer of complexity on top of graph neural networks and therefore less explainability.

5 PROPOSED METHOD

5.1 Intuition

Our method represents a significant advancement in modeling disease outbreaks by addressing several limitations present in current approaches. One key innovation lies in our recognition that the assumption of a single Patient Zero (P0) may not accurately reflect the complexity of real-world scenarios. In many instances, outbreaks may be initiated by multiple individuals simultaneously, and these multiple P0s could subsequently disperse to various locations before the disease begins to spread. Unlike existing models that focus on predicting a single P0, our approach extends the scope to accommodate the possibility of multiple initial cases, providing a more realistic representation of outbreak dynamics.

Furthermore, our model surpasses state-of-the-art techniques by incorporating snapshots of disease progression into its framework. Conventional approaches often rely on predicting the initial P0 based on the final state of the outbreak, a challenging task given the limited information available at the outset. In contrast, our model leverages snapshots captured at various stages of the outbreak, allowing it to track the disease's evolution over time. By considering the dynamic nature of the outbreak, our model gains insights into where and how the infection is spreading. This temporal information enables our model to refine its predictions and identify not only the location but potentially multiple P0s with higher accuracy.

In essence, our approach offers a more nuanced and adaptable solution to outbreak modeling, accommodating scenarios with multiple initial cases and utilizing temporal snapshots to enhance predictive capabilities. By addressing these challenges, our model stands out as a robust and realistic tool for understanding and forecasting the complexities of disease outbreaks.

5.2 Approach

To predict the identities of patient zero or zeros we plan on utilizing a Graph Neural Network to make node-level predictions. We created a GNN model to do so.

We created a Graph Convolutional Neural Network with 3 hidden layers, each of which had 16 nodes. We also added ReLUs to provide additional nonlinear transformations and Dropouts to improve the generalizability of our GCN. We arrived this architecture by looking at examples, abiding by general best practices when setting up GNNs, and testing out varying structures.

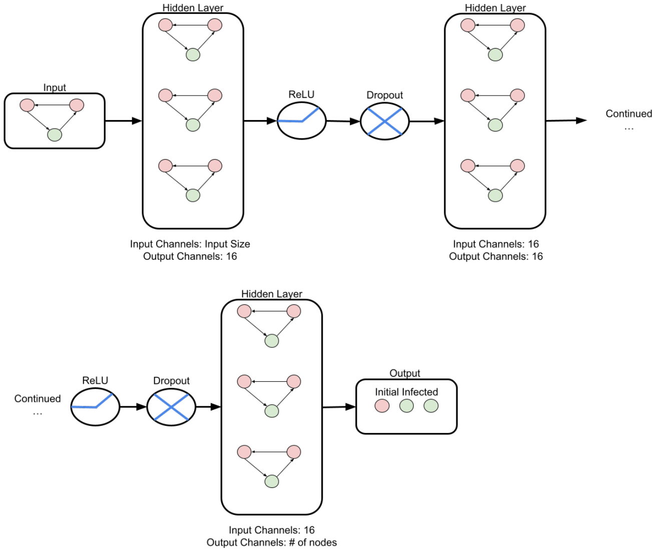

Figure 1: GCN Architecture

The first part of our experiment involved training our GNN on graphs with a certain timestep, and each network had one labeled patient zero. We used this as a baseline to compare; outputting the identity of exactly one patient 0 at a specific timestep leaves less variability for the GNN to learn. Additionally, this baseline allowed experimentation with the GNN architecture.

To build on this, we altered our training set to not just predict the identity of a single patient 0 but instead multiple patient 0s in a graph network in a certain timestep. We generated networks with multiple patient 0s and fed them as inputs to our GNN.

Subsequently, we supplemented the GNN with multiple snapshots of the network outbreak at different timesteps. In theory, this additional information would provide the GNN with more details about the dynamics of how the infection spreads, which would result in better predictions.

After performing all the aforementioned experiments, we will generate visualizations across a different number of patient zeros, timesteps and snapshots.

5.3 Data Collection

To acquire the data required for both training and testing, our approach involves a series of steps aimed at generating a diverse and comprehensive dataset with inspiration from the data collection process described in Shah's 2020 study [7]. To create the actual network graphs, we employed the Python package called NetworkX. This versatile library enabled us to generate random networks, compute various network properties, and visualize the resulting networks. Leveraging NetworkX ensured the diversity and complexity of the generated graphs.

As for the epidemiological snapshots, we have created our own data sets through the use of SI model simulations and have started with the randomly generated graphs with 50 nodes and 100 edges. In terms of the features, we have provided the infection status per node.

Once the networks are constructed, our next step is to conduct multiple simulations on each of them. During these simulations, we will systematically vary the values of key infection parameters, including β and γ. This variation will provide us with a set of infection dynamics, covering a range of scenarios and scenarios.

After configuring the graph structures and setting the parameters, we randomly designated a set of nodes as patient zero(s) to initiate the disease outbreak. The propagation of the outbreak will be tracked over a variable time span, T, ranging from 5 to 20 time steps. This flexibility in time step selection ensures that we capture a broad spectrum of outbreak durations and behaviors.

Once the outbreak propagation simulations are completed, we will have generated all the requisite data for training and testing our model. The input data for the model will just be snapshot(s) generated from a synthetic dataset. This comprehensive dataset of 10,000 graphs (based on estimates for necessary datapoints needed for training) allowed our model to learn and generalize from a wide array of infection scenarios.

5.4 GNN Model

In order to develop a GNN for rumor centrality, we had to understand the fundamentals of GNNs first. For our most basic use case, we want our GNN, given a graph as an input with information about its nodes, edges, and infected/not-infected at a snapshot, to output the source of the infection, a typically node-specific context. We will update the inputs later on (including more than one snapshot for example), but for our first iteration prior to the midpoint, this was our goal.

$$h_i^{t+1} = f(h_i^t W + \sum_{j \in \text{Neighbors of } i} \frac{1}{c_{ij}} h_j^t U) \tag{1}$$

Let's consider h to be an information vector for node i. It encodes information across layers to generate better representations across iterations. This is essentially the backbone behind how a GNN, or in our initial prototype, how a graph convolutional network (GCN) "learns". The weight matrix W are the weights that we let the neural network: it essentially gives the neural network the freedom to decide which patterns are more/less important from our past information vector.

We also want to embed information from a node's neighbors to form our new information vector. For each neighbor, we also have a U vector that is learned by the neural network and normalize by c_{ij}. Similar to W, the U vector is used for us to selectively choose what's important and not important to us for a neighbor. Additionally, the normalization value c_{ij} allows us to weight neighbors differently, allowing the neural network to pick out important information. We then aggregate all these information using the summation.

It is important to note that we do not have an ordering for the neighbors, which is why our GCN is permutation-invariant. This is an important concept because permutation invariance allow us to preserve graph symmetry, which is important because we want to learn node-context about an infection source node regardless of the order of the nodes. We do not want the order of the nodes to affect our neural network's output.

With each iteration, we can update our representation, which will eventually lead us to node-context about the source of the infection. This update formula sets the foundation for a simple GCN. There are more complexities we will definitely add (such as additional layers and pooling).

GNNs differ from purely algorithmic approaches, such as Shah and Zama's rumor centrality algorithm [8] using maximum likelihood estimation, in that they essentially try to learn and improve representations in order to find rumors rather than have a fixed mathematical calculation. Due to this, different hyperparameters will affect the accuracy of GNNs, which is why we are going take special care in choosing them (likely through grid search).

6 EXPERIMENT

6.1 Questions

Here are some of the questions we wanted to answer:

- Can our GNN be a viable option to infer more than one patient zero prediction?
- Does including intermediate snapshots improve the accuracy of our GNN?
- How does increasing the days affect the GNN accuracy? We can infer that a snapshot at a later day may cause patient zero inference to be tougher, but how does it work out in practice?
- Would we require a more complex GCN architecture to make very difficult patient zero inferences?

6.2 Experiment Design and Testbed

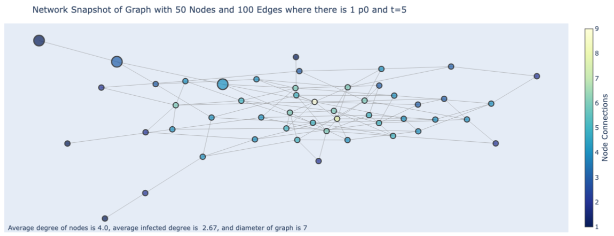
Network Snapshot of Graph with 50 Nodes and 100 Edges where there is 1 p0 and t=5
Average degree of nodes is 4.0, average infected degree is 2.67, and diameter of graph is 7
Node Connections
9
8
7
6
5
4
3
2
1

Figure 2: Example Graph (Larger Circles are Infected)

To test the effects of multiple snapshots, multiple patient 0s, and time horizon, we decided to run experiments on every combination of the 3 variables. For multiple snapshots, we experimented with the following:

- 1 Snapshot = Snapshot of Final infections
- 2 Snapshots = 1 Snapshot halfway through outbreak progress and 1 snapshot of the final infections
- 3 Snapshots = Evenly spaced snapshots at increasing intervals of $t_f/3$ where t_f is the number of days we are expanding the outbreak to

For days we let the outbreak expand to, we will be using 5, 10, and 20 days.
For number of Patient 0s, we will be using 1, 2, and 3 Patient 0s.

Some variables that we kept constant throughout the experiments were the following:

- $\beta = 0.15$
- $\gamma = 0$
- Number of Nodes = 50
- Number of Edges = 100
- Size of training set = 10000 graphs
- Training:Testing Split = 0.7:0.3

6.3 Results

Accuracy	Day 5	Day 10	Day 20
1 Snapshot	26.73%	13.47%	16.60%
2 Snapshots	33.57%	23.73%	11.60%
3 Snapshots	30.43%	22.17%	15.53%

Figure 3: Accuracy Table w/ 1 Patient Zero

Accuracy	Day 5	Day 10	Day 20
1 Snapshot	22.08%	15.08%	6.13%
2 Snapshots	19.93%	18.53%	12.88%
3 Snapshots	23.23%	19.92%	13.95%

Figure 4: Accuracy Table w/ 2 Patient Zeros

Accuracy	Day 5	Day 10	Day 20
1 Snapshot	19.92%	12.04%	8.13%
2 Snapshots	17.97%	20.93%	13.23%
3 Snapshots	18.76%	16.87%	14.38%

Figure 5: Accuracy Table w/ 3 Patient Zeros

Throughout these experiments, we have given partial accuracy points to the models that can predict some of the patient 0s but not necessarily all. By monitoring the correlation between days first, we can notice one of the first trends. As the snapshot day increases, the overall accuracy decreases. Intuitively, this makes sense as an earlier snapshot gives more focused information about the initial infected node(s). The snapshot for Day 5 is likely to be more centered around the initial infected node(s) versus higher day snapshots, making it easier for the GNN to locate patient zero(s). In other words, increased infections (coming from a longer elapsed period) brings about more possibilities for potential patient zero(s).

Now, looking at the trend as snapshots increase, snapshots seem to have a negligible effect on accuracy for Day 5 results but have a positive effect for Day 10 and an even larger positive effect for Day 15. Honing on Day 5 results, this negligible effect may be because the advantage of having snapshots before Day 5, which is already pretty close to the original graph with patient zero(s), is minimal. The GNN likely does not find the earlier snapshots to be much more useful than the current, lending to minimal change in accuracy. However, for having the Day 5 intermediate snapshot along with the Day 10 snapshot for instance, compared to just Day

10 snapshot, can prove to be quite useful; an Day 5 intermediate snapshot can lend quality information to the GNN.

Taking a look at the effect of changing the number of patient 0s has produced surprising results. Throughout the experiments between 1, 2, and 3 patient 0s, we have seen a slight drop off in the accuracy of predicting the identities of these P0s. We were expecting to see a bigger drop off in accuracy values as the amount of p0s increased since this would lead to a greater complexity for the GCN to calculate. One reason for the results we have gathered could be due to the limited time horizon. Since the outbreak only progressed for a relatively short period, not many of the individual spreads of each P0 overlapped. This most likely lead to an easier computation for the GCN. In cases where the outbreak progressed for much longer, this is where we could see diminishing accuracy values.

In terms of general observations, these accuracies were what we expected given our architecture. In comparison to other efforts, notably Shah and other researchers' GNN to predict a single patient zero [7], our model underperforms. We surmise this is due to the vast difference in complexity between the models. For example, our model has 3 GNN layers whereas their model has 100, and while our model only is trained on 11 epochs, their model is trained on 150 epochs. Unfortunately, increasing our architecture and training to match their size would have made it difficult for us to generate results as the compute resources to train a model with a similar configuration would have expensive. Therefore, we decided on a simpler architecture and less epochs for this experiment.

7 CONCLUSION AND DISCUSSION

Our overarching goal for this semester was to develop a pioneering graph neural network architecture that possesses the unique capability to take into account two critical factors: multiple snapshots at different time steps at which the disease outbreak has progressed and the number of patient zeros involved. This innovative approach represents a significant advancement over previous research efforts, which predominantly focused on predicting a single patient zero while neglecting temporal information. By considering both multiple timesteps and the number of initial infections, our model aims to revolutionize our ability to pinpoint the identities of the patient zeros accurately.

The incorporation of timestep information is particularly groundbreaking as it allows us to track the evolution of outbreaks over time. This temporal awareness is crucial for modeling and understanding the dynamics of multi-source pandemics comprehensively. By capturing the temporal dimension, our model can provide insights into the spread of diseases, enabling us to intervene more effectively and proactively in the early stages of an outbreak, thereby mitigating their potential to escalate into large-scale crises.

Moreover, our novel GNN architecture holds great promise in shedding light on network cascades and the prevention of future multi-source outbreaks. By accurately identifying the sources of infections at different time steps, we gain a deeper understanding of how diseases propagate through networks and how to disrupt these pathways. This knowledge can be harnessed for devising more effective prevention and containment strategies, ultimately contributing to enhanced public health preparedness and response.

In summary, our semester-long endeavor aims to advance the field of disease outbreak modeling and prediction by developing a cutting-edge GNN architecture that considers both time and the number of initial infections. This innovative approach has the potential to transform our ability to tackle multi-source pandemics, understand network cascades, and bolster our capacity to prevent future outbreaks.

In the future, possible avenues for further experiments would be to consider deeper time steps to model the real world better and test the limitations of the graph convolutional network structure. In tandem with this, testing more intricate model architecture can lead to significant improvements in accuracy. Finally, running these experiments on different types of graphs (such as BA or ER graphs) could provide insight into the diversity of applications of GCN patient zero prediction.

ACKNOWLEDGMENTS

To Dr. Aditya Prakash for his guidance in the research.

REFERENCES

[1] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2009. Rumor Spreading in Social Networks. In *Automata, Languages and Programming*. International Colloquium on Automata, Languages, and Programming, 375–386.

[2] Wei Cui and Mingsheng Shang. 2023. KAGN:knowledge-powered attention and graph convolutional networks for social media rumor detection. *Journal of Big Data* (2023). https://doi.org/10.1186/s40537-023-00725-4

[3] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 2022. Source Localization of Graph Diffusion via Variational Autoencoders for Graph Inverse Problems. (Aug. 2022). https://doi.org/10.1145/3534678.3539288

[4] Andrey Y. Lokhov, Marc Mezard, Hiroki Ohta, and Lenka Zdeborova. 2014. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E* 90, 1 (July 2014). https://doi.org/10.48550

[5] Olavo H. Menin and Chris T. Bauch. 2018. Solving the patient zero inverse problem by using generalized simulated annealing. *ScienceDirect* (Jan. 2018). https://doi.org/10.1016/j.physa.2017.08.077

[6] Xiaolei Ru, Jack Murdoch Moore, Xin-Ya Zhang, Yeting Zeng, and Gang Yan. 2023. Inferring Patient Zero on Temporal Networks via Graph Neural Networks. In *Proceedings of the Thirty-Seventh Conference on Association for the Advancement of Artificial Intelligence (AAAI)*. 9632–9640.

[7] Chintan Shah, Nima Dehmamy, Nicola Perra, Matteo Chinazzi, Albert-Laszlo Barabasi, Alessandro Vespignani, and Rose Yu. 2020. Finding Patient Zero: Learning Contagion Source with Graph Neural Networks. (June 2020). https://doi.org/10.48550

[8] Devavrat Shah and Tauhid Zaman. 2009. Rumors in a Network: Who's the Culprit? (Sept. 2009). https://doi.org/10.48550

[9] Sushila Shelke and Vahida Attar. 2019. Source detection of rumor in social network – A review. 9 (Jan. 2019). https://doi.org/10.1016

[10] Kyungwoo Song, Hojun Park, Junggu Lee, Arim Kim, and Jaehun Jung. 2023. COVID-19 infection inference with graph neural networks. (2023). https://doi.org/10.1038/s41598-023-38314-3

[11] Shihui Song, Yafan Huang, and Hongwei Lu. 2021. Rumor Detection on Social Media with Out-In-Degree Graph Convolutional Networks. (Oct. 2021). https://doi.org/10.1109/SMC52423.2021.9659106

[12] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. 2018. Multiple Source Detection without Knowing the Underlying Propagation Model. (Dec. 2018). https://doi.org/10.48550/arXiv.1812.08434

[13] Marcin Waniek, Petter Holme, Katayoun Farrahi, Rémi Emonet, Manuel Cebrian, and Talal Rahwan. 2022. Trading contact tracing efficiency for finding patient zero. *PubMed* (Dec. 2022). https://doi.org/10.1038/s41598-022-26892-7

[14] Shouzhi Xu, Xiaodi Liu, Kai Ma, Fangmin Dong, Basheer Riskhan, Shunzhi Xiang, and Changsong Bing. 2022. Rumor detection on social media using hierarchically aggregated feature via graph neural networks. (May 2022). https://doi.org/10.1007/s10489-022-03592-3

[15] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2017. Graph Neural Networks: A Review of Methods and Applications. (Feb. 2017). https://doi.org/10.1609/aaai.v31i1.10477