2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP) | 978-1-6654-4290-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/AISP53593.2022.9760527

# Graph Convolutional Networks for Predicting State-wise Pandemic Incidence in India

Siddharth Sriraman
*Dept. of Computer Science and Engineering*
*SSN College of Engineering*
Kalavakkam, India
siddharth18150@cse.ssn.edu.in

Manjunathan R*
*Dept. of Electronics and Communication Engineering*
*SSN College of Engineering*
Kalavakkam, India
manjunathan19057@ece.ssn.edu.in

Nethraa Sivakumar*
*Dept. of Electronics and Communication Engineering*
*SSN College of Engineering*
Kalavakkam, India
nethraa18096@ece.ssn.edu.in

Pooja S*
*Dept. of Electronics and Communication Engineering*
*SSN College of Engineering*
Kalavakkam, India
pooja18112@ece.ssn.edu.in

Nikhil Viswanath*
*Dept. of Electronics and Communication Engineering*
*SSN College of Engineering*
Kalavakkam, India
nikhil18098@ece.ssn.edu.in

***Abstract*—In this paper, we analyze the performance of graph convolutional networks (GCNs) in predicting COVID-19 incidence in states and union territories (UTs) in India as a semi-supervised learning task. By training the model with data from a small number of states whose incidence is known, we analyze the accuracy in predicting incidence levels in the remaining states and UTs in India. We explore the effect of pre-existing factors such as foreign visitor count, senior citizen population and population density of states in predicting spread. To show the robustness of this model, we introduce a novel method to choose states for training that reduces bias through random sampling in five regions that cover India's geography. We show that GCNs, on average, produce a 9% improvement in accuracy over the best performing non-graph-based model and discuss if the results are feasible for use in a real-world scenario.**

***Index Terms*—Graph convolutional network, semi-supervised learning, pandemic incidence**

## I. Introduction

Coronavirus (COVID-19) is a disease that first appeared in China in December 2019. Over 190 million cases have been documented over the world as of July 2021 and 31.1 million cases just in India, with a fatality rate of over 2% (of all closed cases). This rapid pandemic spread is a worldwide issue and a severe threat to public health and the global economy. Countries restricted social interaction as a preventive measure, through isolation and quarantine, to prevent the sickness from spreading. Early response and strategic decisions have had a massive impact in controlling the spread of the virus.

Machine learning techniques have the ability to model various aspects of a pandemic by collecting relevant data and training a model with it to make inferences in the future. However, careful analysis of what the model learns from the data, understanding if the predictions are worthwhile and studying the interpretability of the model in real-world scenarios is of paramount importance. This study focuses on classifying the incidence of COVID-19 in Indian states at the start of the pandemic into three levels (low, medium and high). We aim to study if accurate knowledge of incidence levels collected from a small number of states is sufficient to predict incidence in remaining states using a semi-supervised learning technique. We incorporate pre-pandemic census data (such as senior citizen population, foreign visitor count, health index etc.) of states and connectivity between states by modeling the states of India as a graph since connectivity between states is an important factor influencing spread. Graph convolutional networks [1] have become increasingly popular recently due to their semi-supervised learning ability, finding applications in domains with large amounts of unstructured data, such as text data. We analyze the performance of graph convolutional networks and traditional non-graph-based models, discuss advantages, drawbacks and conclude if the results are meaningful for real-world use.

## II. Related Work

A literature review showed that while the intersection of graph neural networks and COVID-19 spread has been studied, the main focus has been on dynamically using current COVID-19 confirmed case counts to forecast case counts in

* These authors contributed equally.

the future. Spatio-temporal graph neural network is used in [2] to predict how the case count changes on a daily basis by incorporating aggregated mobility data at a county level. The authors show a reduction in root mean square logarithmic error and absolute Pearson correlation improvement compared to baseline models. However, training the model required a dataset of 100 days of confirmed case counts. Spatio-Temporal Attention Network (STAN) for predicting long and short-term case counts in [3] show improved performance compared to conventional epidemiological models in predicting confirmed case counts and up to 87% reduction in mean squared error compared to the best baseline. STAN encodes relationships between states using geographical proximity and similarity in demographics, taking into hospitalization and ICU visits for 80 days. While these fully supervised models are useful in aiding decision-making and policies in later stages, with limited data availability in the initial stages, a semi-supervised approach can help in providing quick estimates on the spread before data for more complex time-series models become available. Reference [4] which introduces a cloud-based approach for predicting cases using prior case counts, emphasizes that adding demographic indicators such as population density, age distribution can lead to more accurate models.

Reference [5] is a preliminary work on studying incidence levels of COVID-19 using graph convolutional networks in Mexican states, inviting researchers to review the usefulness of graph-based models for this task. It uses population density as the only feature for each state and borders as edges between states. Reference [6] concluded that traffic in airports and high age groups correlate with confirmed cases. Our contributions in this paper are as follows: Model the incidence of COVID-19 in Indian states through a semi-supervised lens, integrating relevant features like foreign visitor count, senior citizen population percentages, rural population percentages in states etc., introduce a novel method to choose the states to get ground truth incidence levels from and finally discuss the interpretability of the model.

## III. Proposed Approach

To classify each state into one of the three incidence levels, feature data for each state and an adjacency matrix to describe the borders between them are fed as input data. This approach assumes that ground truth incidence levels for a few states are present. A graph convolutional network is trained with this limited data, which uses its semi-supervised learning capability, to make predictions on remaining states.

### A. Semi-supervised learning

Semi-supervised learning is a method that combines aspects of supervised and unsupervised learning. It involves a dataset containing a small amount of labeled data to train with and a large amount of unlabeled data to classify. This is often the case with real-world data, as the cost and time required to manually label data may be infeasible. Semi-supervised learning, like unsupervised learning, works by exploiting the underlying structure in the distribution of the data and at the same time incorporates labeled data. The simplest relationship observed would be when training examples that are closer to each other in the feature space tend to share the same label. In our approach, the relationship between data points is modeled using a graph $G(V, E)$, where $V$ is the set of nodes and $E$ is the set of edges that connect the nodes. In our case, nodes are in the graph modeled as states in India and edges are borders between states. The two islands (Andaman and Nicobar and Lakshadweep) are omitted as they make the graph disconnected. The resulting graph has 34 nodes (28 states, 6 union territories) and 134 edges.

### B. Graph Convolutional Network (GCN)

Graph neural networks (GNN) are a class of deep learning models that accept graph data as input. Since there is an abundance of graph data in the real world, GNNs have found a variety of applications ranging from recommender systems in social networks [7] to biological applications like predicting protein interfaces [8]. Graphs inherently encode underlying relationships between nodes through edges that connect them. GNNs can be used to classify nodes in a graph into discrete classes, classify graphs themselves into discrete classes or generate embeddings to be used in other downstream tasks. Semi-supervised learning for node classification can be applied in this paradigm through graph convolutional networks. This aspect of a GCN has been widely studied, finding applications ranging from predicting user geolocation [9] to predicting parking availability in cities [10]. GCNs assume that connected nodes in the graph tend to share the same label.

GCNs take as input an $N \times D$ matrix $X$ that describes the *D*-dimensional features for each of the *N* nodes and an $N \times N$ adjacency matrix *A* to specify the edges between the nodes. A graph convolutional layer applies a transformation on the input node features to produce a $N \times F$ matrix. At each layer, for a given node, this forward step involves a weight transformation applied on the current node representation and a component that aggregates the representation of its neighbouring nodes. Since graphs have no explicit node ordering, this aggregation function has to be invariant to permutation (e.g. sum) and this process is called message passing, which facilitates semi-supervised learning. This can be represented as:

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)}) \tag{1}$$

Where $H^l$ is the node representation at layer $l$ (a $N \times D$ matrix), $A$ is the $N \times N$ adjacency matrix and $W$ is the common $D \times F$ weight matrix at layer $l$. $\sigma(\cdot)$ is the activation function applied on the matrix multiplication result to obtain the $N \times F$ matrix node representation for layer $l + 1$. At the input layer, the node representation is the training data, $H^0 = X$. The weights are shared across all nodes in a layer. The propagation step has to take into account the node's existing

representation, in addition to it's neighbours representation. To do this, the adjacency matrix is modified as $\hat{A} = A + I$, to include self-loops. Each node has varying degree, hence the effect of message passing on nodes has to be normalized, this is done using the diagonal node degree matrix $\hat{D}$ as follows:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \quad (2)$$

Finally, the output for each node can be compared with the expected label through a loss function that can be minimized with backpropagation. In our case, each state is a node to be classified into three categories: low, medium and high incidence of COVID-19, hence it is a three-class node classification task. Incidence levels for only six of the 34 states and union territories (UTs) are used for training and semi-supervised learning is used to predict the incidence level of the other 28 states/UTs. Six states/UTs are chosen for training to ensure there are 2 training examples for each class.

### C. Data for each state and union territory

This task is static as it aims to find a pattern of spread in the initial incidence of the pandemic in Indian states and union territories when information about the spread is not well known. Hence, the node features chosen do not involve any information pertaining to the pandemic spread itself, rather they involve demographic conditions prior to the pandemic. Subsets of the following features were tested:

- **Foreign visitor count**: Number of foreign visitors in the year 2019 in each state/UT [11].
- **Urban population percentage**: Fraction of urban population compared to total population in each state/UT (2011) [12] [13].
- **Rural population percentage**: Fraction of rural population compared to total population in each state/UT (2011) [12] [13].
- **Population density**: Ratio of population to area of each state/UT (2011) [12] [13].
- **Health index** of each state/UT recorded in 2015-16 [14].
- **Senior citizen population**: Fraction of population of citizens aged 60 and above in each state/UT (2011) [15].

Edges between states were modeled as shared borders between them. Another approach we tested was to consider the number of common highways between states to be weighted edges [16]. We note that modeling the effects of lockdowns and travel restrictions which limit the spread can lead to more accurate results. We encourage future research to integrate these effects into graph models.

### D. Choice of the six states

Choosing which of the six states to obtain the ground truth data from is important, as the training set size is very small. To do this, India is split into five regions as shown in Figure 1. As the map legend specifies, one or two states are chosen from each region. Intuitively, choosing states that are known to have more impact on the spread would lead to better results, but to test the robustness of the model, states are sampled randomly from these five regions. This is done to test if the model performs decently regardless of which of the six states are chosen, as long as they cover all regions of India's geography.

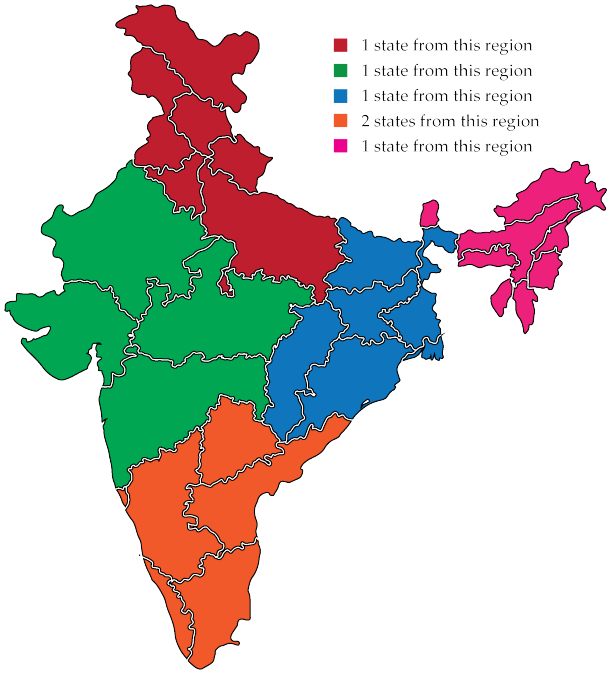


Fig. 1. Regions in India.

### E. Hyperparameters and model architecture

The graph convolutional network architecture consists of two graph convolution layers of 16 neurons each, activated with the Leaky ReLU [17] activation function. This is followed by a linear layer that transforms the 16-dimensional output (of each graph node) into 3 dimensions, which is passed into a softmax activation function to obtain a probability distribution over the classes for each node. The weights of this linear layer are shared across all 34 nodes. The architecture is shown in Figure 2. Paired with negative log loss, this architecture gave the best results. Since validation accuracy would not be known in a real-world scenario (as the class labels for all states may not be known), training cannot be stopped based on validation data performance, instead, we stop it once the training accuracy reaches 100%. While other optimization methods like RMSprop were tested, Adam [18] with a learning rate of 0.01 converged the fastest. The hyperparameters used are tabulated in Table I. The model was implemented using Deep Graph Library [19] in Python.

## IV. Results

Since the states chosen for training in the selection process vary each time, evaluating the model performance was challenging. As the training data set size is very small, a significant variance was observed in the convergence for each training run. To obtain a general perspective on the model performance, 100 trials were run and the maximum

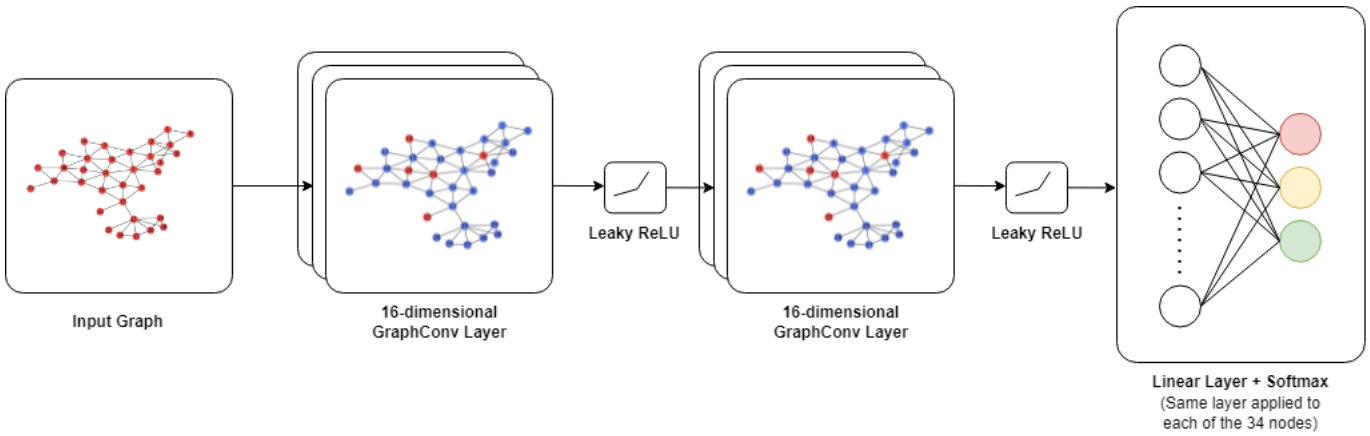


Fig. 2. GCN model architecture.

TABLE I
HYPERPARAMETER VALUES

| Hyperparameter name | Value |
|---|---|
| Optimizer | Adam |
| Learning rate | 0.01 |
| Final layer activation | Softmax |
| Maximum epochs | 200 |
| Loss function | Negative log loss |

TABLE II
RESULTS WITH DIFFERENT MODELS OVER 100 RUNS

| Model | Max accuracy | Average accuracy |
|---|---|---|
| Support Vector Machine | 67.6% | 47.8% |
| kNN (k=5) | 50% | 33.4% |
| Random Forest Classifier | 76.4% | 59% |
| **Graph Convolutional Net** | **85.3%** | **67.7%** |

and average accuracy over all 34 states/UTs were tabulated in Table II. The best accuracy achieved with this model was 85.3% with training states/UTs as Jammu and Kashmir, Rajasthan, Kerala, Puducherry, Jharkhand and Tripura. Since the range of each feature varied widely, the features were normalized to have zero mean and unit variance. To generate labels, state-wise confirmed cases on 12th June 2020 were taken and split into three parts at the 33rd and 66th percentile to generate labels for each state/UT. This ensured a balanced split between the three classes, leading to a 33% random baseline accuracy. Using foreign visitor count and population density as node features gave the best results, showing that tightly populated regions with a higher number of foreign visitors tend to be affected worse. Senior citizen percentages between states/UTs did not vary much, hence it did not have a significant impact on the results. The traditional non-graph-based models tested were Support Vector Machine (SVM), K-Nearest Neighbours (with 5 nearest neighbours) and Random Forest Classifier (with 32 estimators). As GCN is a deep learning model trained with a small dataset, the variance shown between multiple runs that used the same training data was more than the non-graph-based models. Feeding node2vec [20] embeddings for each node to the non-graph-based models to encode graph data and using the number of common national highways as weighted edges between states did not improve performance. This signifies that in order to incorporate edge features effectively, complex mobility data between states is required. Results shown here are with shared borders between states as edges. On average, we observed that GCNs are more robust in capturing the spread compared to non-graph-based models using the same data.

## V. CONCLUSION AND FUTURE WORK

The goal of this paper was to understand what type of pre-existing data supports semi-supervised learning in graphs for pandemic spread, while existing work focuses more on fully supervised learning with GNNs using pandemic-specific training data. GCNs improved performance by 9% compared to the best performing non-graph-based model. This work is a preliminary study to generate estimates for spread with publicly available data. In addition to the pre-pandemic factors we tested, the spread is affected by factors that are pandemic-specific as well, such as usage of masks, lockdowns etc. While at this stage, directly using this in a real-world scenario might not be feasible, we show that GCNs have the ability to model incidence well with any subset of training data, laying a framework for building more complex models in the semi-supervised paradigm to analyze incidence. This can also be extended to work with a global graph to predict country-to-country spread based on travel. While mobility data between states collected through mobile phones could give a clearer representation of connectivity between states, obtaining such data is difficult due to privacy reasons. Future work to expand on this could focus on improved encoding of edges between states, obtaining data that could correlate more with the incidence, such as temperature and economic development level [21] and obtaining a larger graph by approaching this task from a finer district-level view.

## REFERENCES


[1] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[2] 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," *MLG workshop @ KDD'2020, epiDAMIK workshop @ KDD'2020*, 2020. [Online]. Available: https://arxiv.org/abs/2007.03113

[3] J. Gao, R. Sharma, C. Qian, L. M. Glass, J. Spaeder, J. Romberg, J. Sun, and C. Xiao, "STAN: Spatio-Temporal Attention Network for Pandemic Prediction Using Real World Evidence," 2020.

[4] S. Tuli, S. Tuli, R. Tuli, and S. S. Gill, "Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing," *Internet of Things*, vol. 11, p. 100222, Sep. 2020. [Online]. Available: https://doi.org/10.1016/j.iot.2020.100222

[5] J. M. N. Duarte. (2020) Graph convolutional nets for classifying COVID-19 incidence on states. [Online]. Available: https://towardsdatascience.com/graph-convolutional-nets-for-classifying-covid-19-incidence-on-states-3a8c20ebac2b

[6] S. Roy and P. Ghosh, "Factors affecting COVID-19 infected and death rates inform lockdown-related policymaking," *PLOS ONE*, vol. 15, no. 10, pp. 1–18, 10 2020. [Online]. Available: https://doi.org/10.1371/journal.pone.0241165

[7] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation," *CoRR*, vol. abs/1902.07243, 2019. [Online]. Available: http://arxiv.org/abs/1902.07243

[8] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein Interface Prediction Using Graph Convolutional Networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6533–6542.

[9] A. Rahimi, T. Cohn, and T. Baldwin, "Semi-supervised User Geolocation via Graph Convolutional Networks," *CoRR*, vol. abs/1804.08049, 2018. [Online]. Available: http://arxiv.org/abs/1804.08049

[10] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-Supervised Hierarchical Recurrent Graph Neural Network for City-Wide Parking Availability Prediction," *CoRR*, vol. abs/1911.10516, 2019. [Online]. Available: http://arxiv.org/abs/1911.10516

[11] Ministry of Tourism, Government of India. State and union territory-wise domestic and foreign tourist visits during 2018, 2019. [Online]. Available: https://tourism.gov.in/sites/default/files/2020-08/Figures.pdf

[12] Office of the Registrar General and Census Commissioner, Government of India. (2011) Population and decadal change by residence. [Online]. Available: www.censusindia.gov.in/2011census/PCA/PCA_Highlights/pca_highlights_file/India/Chapter-1.pdf

[13] Directorate of Economics and Statistics, Government of Telangana. (2015) Statistical Year Book. [Online]. Available: www.telangana.gov.in/PDFDocuments/Statistical%20Year%20Book%202015.pdf

[14] National Institute for Transforming India, Government of India. (2016) Health Performance: NITI Aayog. [Online]. Available: http://social.niti.gov.in/hlt-ranking

[15] Ministry of Statistics and Programme Implementation, Government of India. (2016) Elderly in India. [Online]. Available: mospi.nic.in/sites/default/files/publication_reports/ElderlyinIndia_2016.pdf

[16] Ministry of Road Transport and Highways, Government of India. (2016) Basic road statistics of India 2015-16. [Online]. Available: https://morth.gov.in/sites/default/files/File3100.pdf

[17] A. L. Maas, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," 2013.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[19] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks," 2020.

[20] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," 2016.

[21] W. Cao, C. Chen, M. Li, R. Nie, Q. Lu, D. Song, S. Li, T. Yang, Y. Liu, B. Du, and X. Wang, "Important factors affecting COVID-19 transmission and fatality in metropolises," *Public health*, vol. 190, 2021.