Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Covid-19 Detection Using Data Analysis, Graph, GCN/GAT Models

tanmay sharma

ts825362@gmail.com

Babu Banarasi Das University

Article

Keywords:

Posted Date: November 20th, 2026

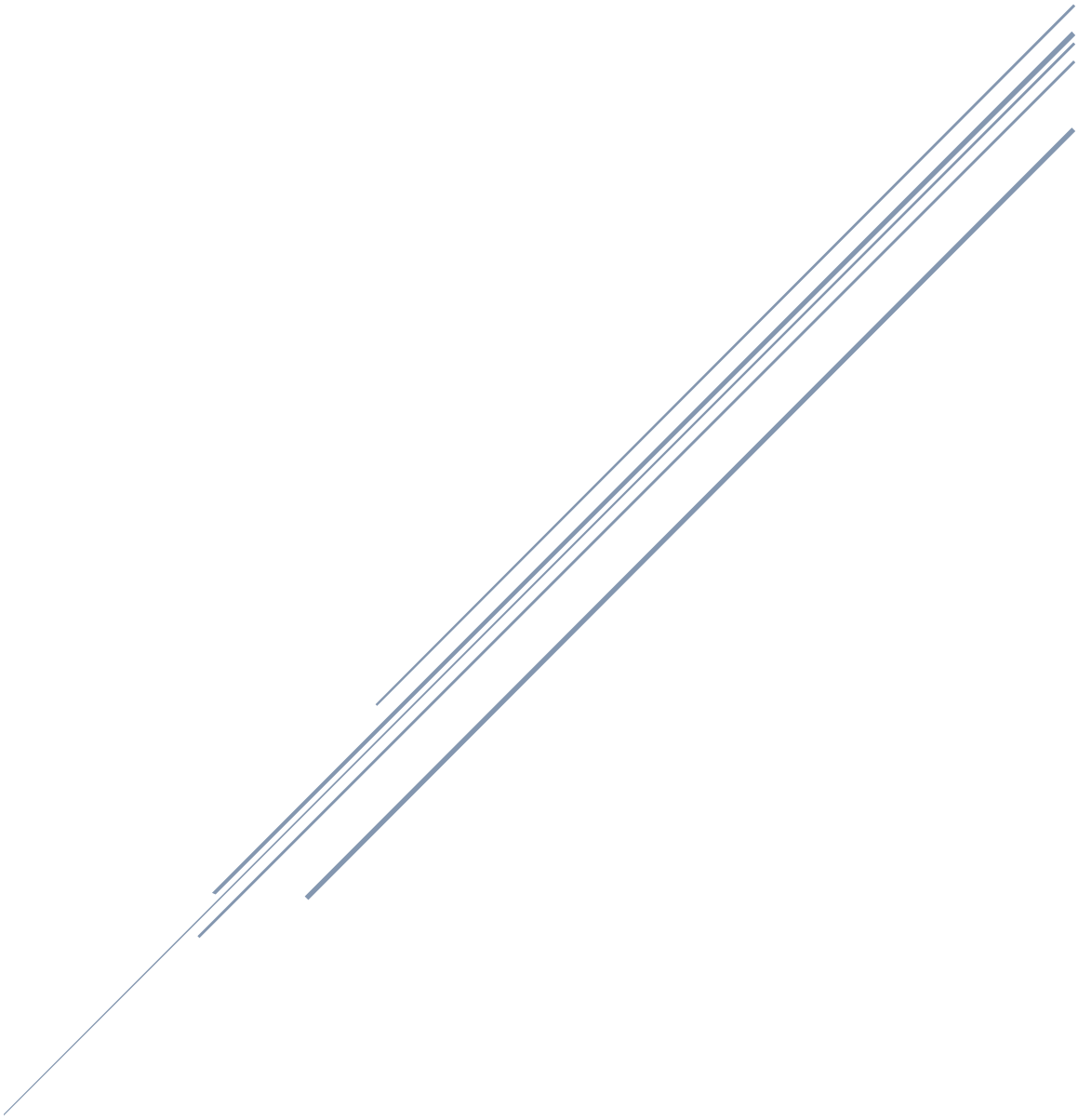
DOI: https://doi.org/10.21203/rs.3.rs-11132417/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

COVID-19 DETECTION USING DATA ANALYSIS, GRAPH, GCN/GAT MODELS

Graph Neural Networks



Authors: Tanmay Sharma, Email: ts825362@gmail.com

Covid-19 Detection Using Data Analysis, Graph, GCN/GAT Models

ts825362@gmail.comTanmay Sharma1
1Department of Computer Application, BBD University, Lucknow, Uttar Pradesh, India
*Corresponding author name and email address: Tanmay Sharma - ts825362@gmail.com

***Abstract*— Covid-19 has been a very deadly virus that directly attacks human lungs and over a year in 2020 it has Caused a devastating impact on human lives, including social interaction, economic status.**

I. INTRODUCTION

Although, it has been under control after middle of the year 2021, Humans are still being diagnosed with Covid-19. Till now many Researchers have published their research on many areas and used machine learning algorithms (examples: RF (Random Forest), Support Vector Machine, Naïve bayes and shown promising results with RF-94.75%, NB-95.53%, SVM-96.12%). But we cannot depend on it completely as we need more accurate and closer to 100% results even though ML algorithms have played a crucial role in detecting covid-19. Graph analysis in the field of computer science had also been used for covid -19 monitoring and chest x-rays. (Das et al., 2023; Rahman et al., 2021; Woodward et al., 2025)

II. LITERATURE REVIEW

With the help of generative AI and increasing data analytics field, we could help countries reduce the increasing cases of Covid-19, via live monitoring of confirmed cases all of the world, eventually reducing human efforts.

Combining graph analysis and data analytics could provide better and significantly higher results. Integrating both of these theories together and it has the potential for improved predictive performance. AI-based Covid-19 detection systems primarily utilize medical imaging data such as chest X-rays (CXR), computed tomography (CT) scans, clinical laboratory data, cough sounds, and epidemiological datasets(Huang et al., 2021). [2]

Data analytics methods have also been applied for forecasting infection spread, patient risk analysis, and healthcare decision support. Recent studies demonstrate that AI models can achieve high accuracy, sensitivity, and specificity in identifying COVID-19 cases, thereby assisting healthcare professionals in clinical diagnosis and disease management.

Integrating data analytics, graph analysis and AI It will Help with infection propagation, healthcare decision making, infection monitoring, symptoms propagation instead of just checking Covid-19 detection, hence opening doors to graph neural networks (GAT/GCN) and hybrid models (RF + Graph Metrics, SVM + Graph Metrics), which might provide better results.

GNN's are Artificial neural networks, designed for tasks whose inputs are graphs. In more general subject of geometric deep learning certain existing neural networks architectures can be interpreted as GNN's operating on suitably defined graphs.(Bessadok et al., 2022; Wu et al., 2020)

Architecture

The architecture of generic GNN's implements the following fundamental layers which are as follows: permutation layers, local layers, global layers.

Message passing layers

Message passing layers are permutation equivalent layers mapping a graph into an updated representation of the same graph. Formally they can be expressed as message passing neural networks.

Graph convolutional Networks

GCN's was first introduced by Thomas Kipf and Max Welling in 2017 (Reiser et al., 2021; Veličković, 2023), A GCN layer defines a first order approximation of a localized spectral filter on graphs. GCN's are widely used as a starting point for many graph learning tasks due to their simplicity and effectiveness. Some of them strengths including its **efficiency:** the formulation relies on sparse matrix multiplications, making it computationally efficient and scalable to large graphs. **Simplicity:** the model is straightforward to implement and has relatively few parameters compared to more complex architectures. [3], [4]

Strong baseline: GCN's perform remarkably well on mane node classification benchmarks often serving as a baseline to compare against. The primary operation of a GCN layer can be expressed with a single, elegant propagation rule that updates the features for all nodes in the graph simultaneously.

The GCN propagation rule

For a GCN layer, the process of generating the output features for the next layer, H(l+1), from the input features of the current layer, H(l), is defined by the following 1st formula provided in the equations section. This equation might see dense at first, but each component serves a distinct and understandable purpose

H(l) is the matrix of node embeddings at layer l, W(l) is the layer's trainable weight matrix, σ is a non-linear activation function, such as ReLu, applied element wise.

The most distinctive part of the GCN formula is the term $\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$. This component handles the core graph convolution and is constructed from the graph's structure; this full term performs a symmetric normalization of the adjacency matrix as this step is significant for stable training.

III. METHODOLOGY

The John Hopkins Time Series Dataset(Cai et al., 2024; Jin et al., 2024) was the primary source of data used, which included global confirmed cases, recovery cases and deaths cases. All these sources were integrated together to generate results using several machine learning models (RF, SVM, NB), graph model alone, hybrids of Analytics + graph, graph neural networks (GAT/GCN). [6], [7]

The following table presents the experimental results used to evaluate model accuracy and compare the performance of the different models.

The real data and results can be found below where models are trained and performed using John Hopkins Covid-19 Dataset**.**

A. Equations

The graph neural networks equation involves the following parameters, H (l) is the matrix of node embedding at layer l,

W(l) is the layer's trainable weight matrix, σ is a non-linear activation function, such as ReLu applied element wise,

1. Graph Convolutional Network (GCN)

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

2. Accuracy Formula

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy is the proportion of all the classifications that were correct, whether positive or negative. Mathematically it is defined by the above equation.

3. Recall (TPR) Equation

$$\text{Recall (or TPR)} = \frac{TP}{TP + FN}$$

The true positive rate, or the proportion of all actual positives that were classified correctly as positives, is also known as recall. False negatives are actual positives that were misclassified as negatives, which is why they appear in the denominator. For Example: in Gmail, The Recall measured the fraction of spam emails that were classified as spam, hence recall another name is given as 'Probability of detection'.

A hypothetical prefect model would have zero false negatives and therefore a recall (TPR) of 1.0, which is to say, a 100% detection rate.

4. Precision

$$Precision = \frac{TP}{TP} + FP$$

Precision is the proportion of all model's positive classifications that are actually positive. It is mathematically defined according to the above equation, taking the Gmail spam example: precision measured the fraction of spam emails classified as spam that were actually spam,

A perfect zero false model would have value of precision as 1.0. Precision improves as false positive decrease, while recall improves when false negative decrease. Increasing the classification threshold tends to decrease the number of false positives and increase the number of false negatives, while decreasing the threshold has the opposite effects. As a result, precision and recall often shows an inverse relationship, where improving one of them worsens the other.

5. Common Aggregation functions

The Aggregate functions is a primary operation within the message passing mechanism used in graph neural networks. Its purpose involves taking a set of feature vectors form a node's neighbors, which can vary in size, and condensing them into a single representative vector. The aggregator must be a permutation-invariant function, meaning the order in which neighbor features are processed does not affect the output. This property is essential for graphs, as there in no natural ordering to a node's neighbors.

- **Mean Aggregation**

The mean aggregator computes the element-wise average neighbor feature vectors. It provides a summary of neighborhood features, normalized by the number of neighbors. The primary characteristics of mean aggregation is its smoothing effect. By averaging, it ensures that the magnitude of the aggregated vector does not gros with the number of neighbors. This makes it a stable and often effective default choice. However, this same property can be drawback. If one neighbor has a particularly informative feature, its signal may be diluted by the average, especially in dense neighborhood.

- **Sum Aggregation**

The sum aggregator simply adds the feature vectors of all neighbors together. This is perhaps the most direct way to combine information from the neighbors, Unlike the mean, the sum is sensitive to the size of the neighborhood. The resulting aggregated vector's magnitude will scale with the node's degree.

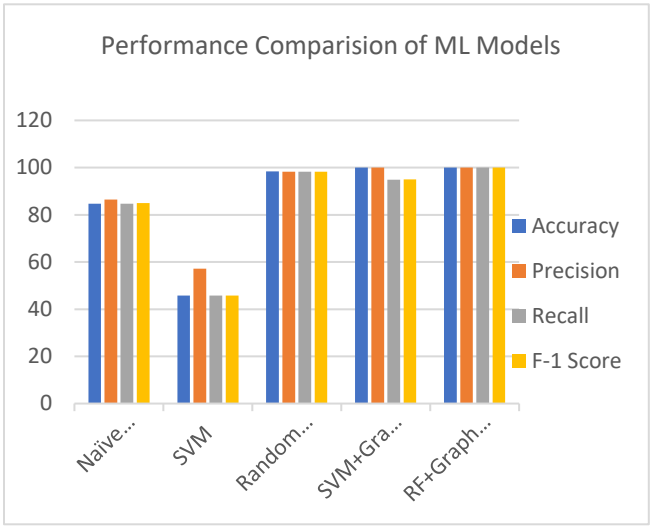
- **Max Aggregation**

Max aggregation, also known as max pooling, takes the element-wise maximum across all neighbor feature vectors. Instead of summarizing the entire neighborhood, it identifies the most prominent feature values among the neighbors. Max pooling is effective at capturing representative or salient features.

Model Training:

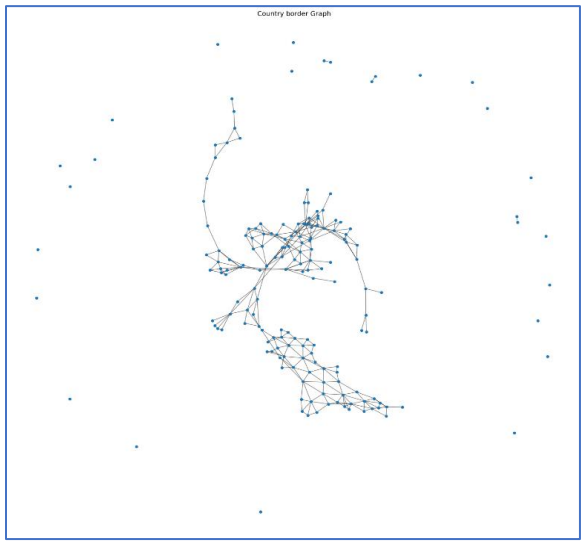
Model training follows a set of steps required to train the GAT/GCN model that help forecasting historical COVID-19 cases and comparing its performance with the previous machine learning models and hybrid models of graphs(Song et al., 2023; Wang et al., 2023), training a GCN/GAT model involves several steps: loading the data, constructing the graph, engineering features, creating the edge index, creating a PyTorch Geometric data object, building the GCN model, training the GCN, and plotting the training loss. The process is then repeated for the GAT model, followed by a comparison of the GCN and GAT results and the finalization of the model for forecasting future cases.

A representation of performance of previous machine learning models (Naïve Bayes, SVM, Random Forest, SVM+ Graph, RF+ Graph) can be found below:



A representation of a graph constructed using john Hopkins historical COVID-19 dataset shows a meaningful insight as it shows the connected countries sharing land borders (here nodes represents countries). [7]

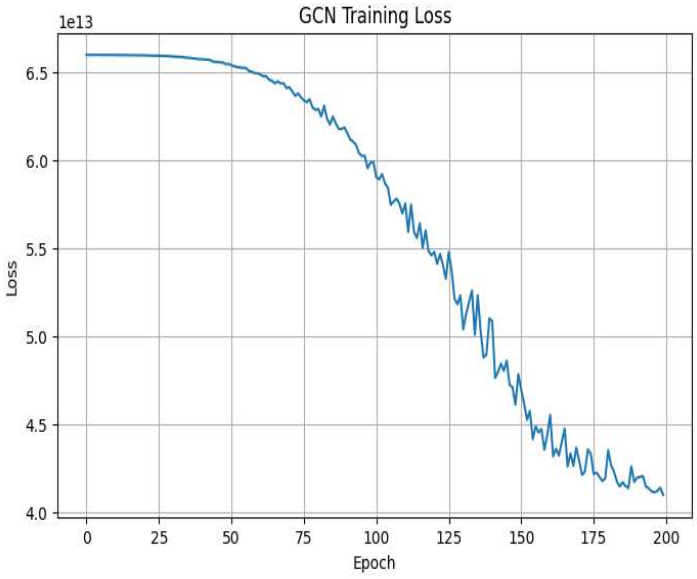
Country border graph



GCN (Graph convolutional networks)

GCN model construction involves a plenty of steps for example, defining a GCN Model, training the model, and finally plotting a training loss graph that illustrates how well the model is learning in order to understand the relationship between neighbor countries and historical COVID-19 case trends,

The 1st step is training the model and plotting a training loss graph, followed by setting graph dataset (data [201, 1142], edge index= [2, 510], y [201, 1]), total of training nodes 160, testing nodes 41. The following constructed graph below shows how well the model is learning.

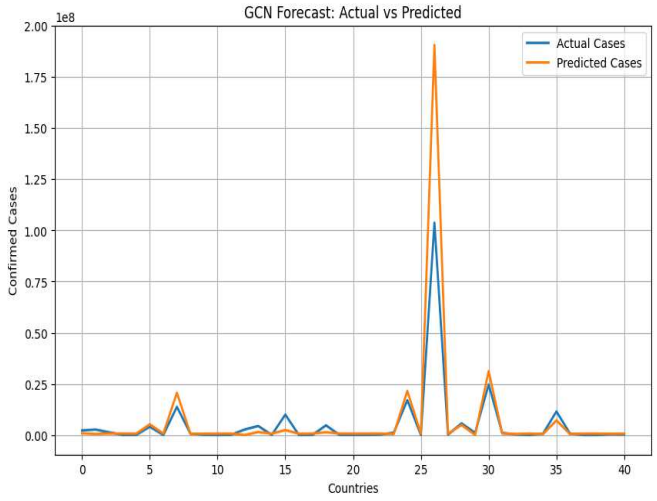


GCN – training loss graph gradually decreases steadily from about 6.6 x 1013 to 4.1 x 1013.

The graph depicts the training loss graph of GCN over 200 epochs. The loss tends to decrease generally over time indicating the model is learning the underlying relationship of COVID-19 cases and neighboring countries trends, on examination it is found that the GCN model training loss shows fluctuations after 120 epochs which can be very normal in such cases, overall GCN model learn well as the loss decreases and there is no training instability.

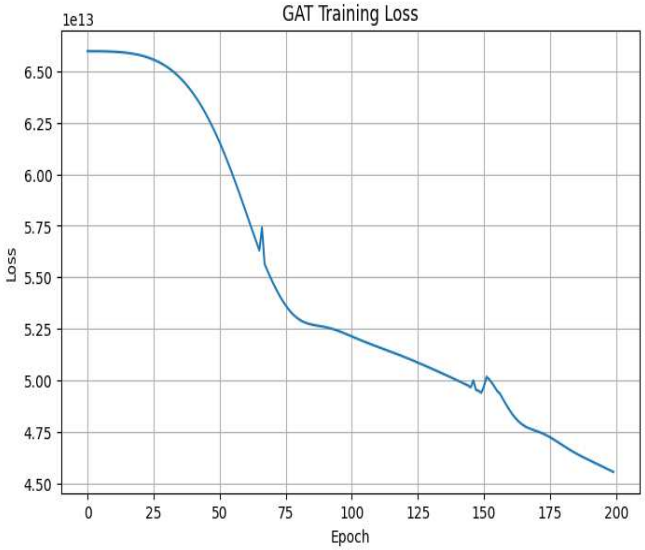
GCN forecasting graph construction of actual vs predicted COVID-19 cases. The graph clearly identifies that there are few countries with very high cases (shows higher peak) and few with very less cases (shows lower peak), orange line defines predicted cases and blue line with actual cases. Smaller countries with low case counts are compressed to near zero.

The largest peak around 26-27 is overestimated by the model, the graph represents the baseline of GCN forecasting results before refinement.



GAT (Graph attention network):

GAT model construction follows the same steps as for GCN model, we first create a training loss graph using the same graph dataset for GAT plotting a graph to check how well GAT model performs and learns the relationship between neighbor countries and COVID-19 cases trends. The following training loss graph of GAT model below illustrates weather GAT trained well or there is instability in its training.



as per the graph, GAT model at the beginning of the model training the epochs is relatively high meaning the model does not learn the relationship between the countries and COVID-19 cases trends, as the training of GAT model progresses, the loss decreases over time indicating that the model learns the meaningful representations from both the graph and historical infection spread from the john Hopkins timed series COVID-19 dataset.

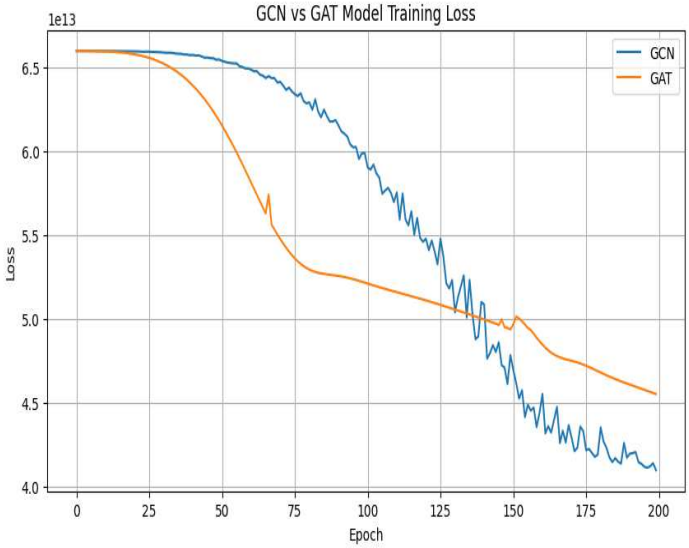
On deep analysis the GAT model during its training period, shows slight fluctuations on 60 epochs and on around 145- 151 as the loss decreases.

GCN and GAT models comparison:

During the initial model training period of both GCN (Graph convolutional networks) and GAT (graph attention networks) tends to learn meaningful insights and understand the relationship, a well-constructed comparison graph between training loss of GCN and GAT were created to compare the results obtained. (Khemani et al., 2024)

The following figure represents the comparison of both the GCN/GAT model training loss performance.

On comparison of both the models GAT model (represented by orange line) relatively has less loss decreases rate, GCN on the other hand tends to have more loss decrease rate over time. This illustrates that during training of models GCN tends to learn and understand the relationship much more as compared to the GAT model which has slightly less loss decrease during training progression, GCN shows much fluctuations immerging from epoch 120 all the way down to epoch 200, GAT shows slight difference at epoch 65 and epoch 151-154.



Parameters	Values
Dataset	John Hopkins Covid-19 timed series dataset
No. of countries	201
No. of countries used	154
No. of time steps	1143
Node features	1142
Graph edges	154
No. of epochs	200

The table the sum of all value/datasets used during model training.

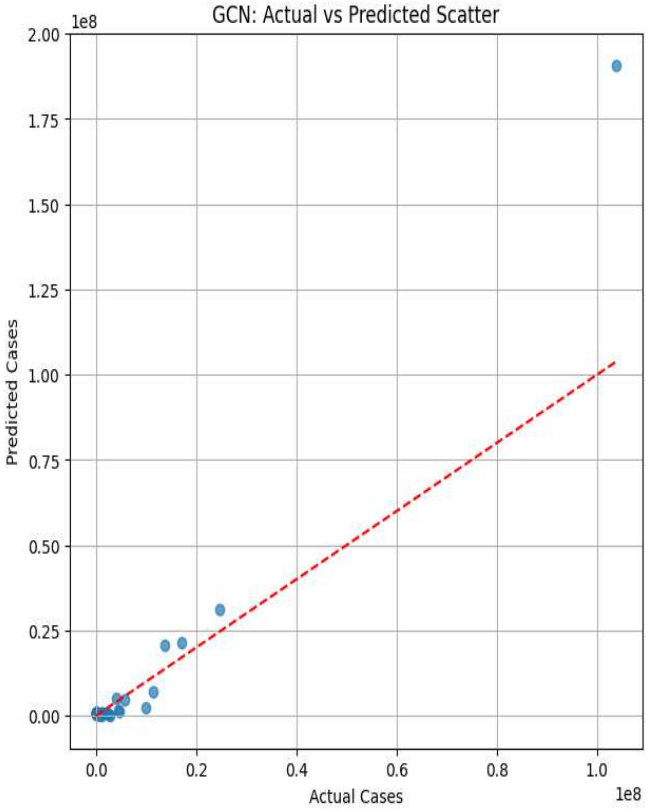
IV. RESULTS

Based on the computed results from the GCN and GAT models, a scatter plot was generated for the GCN model to compare the actual COVID-19 case values with the predicted values. In addition, a residual-error distribution histogram was created to illustrate the differences between the actual and predicted values. The results were further analyzed through a comparative performance table assessing the GCN and GAT models using the mean absolute error, mean absolute percentage error, root mean squared error, and coefficient of determination ().

GCN actual vs predicted values scatter plot:

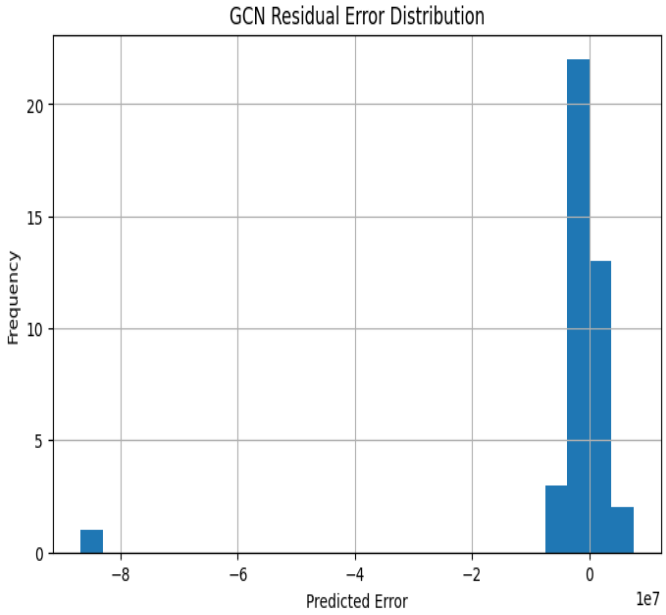
The following scatter plot mainly depicts the comparison of actual COVID-19 values and prediction values produced by the GCN model during forecasting infection spread.

The scatter plot of GCN model, horizontal axis illustrates values confirmed cases, vertical axis showing predicted values, followed by a dashed diagonal red line which suggests a prefect prediction made by the GCN model hence actual value = predicted value, each dot/point on the scatter plot equals to one observation/prediction. The dots farther from the line shows prediction errors, whereas dots closer to line shows accurate predictions. The overall distribution of the points on the scatter plot gives a visual indication of how well GCN model understands the COVID-19 data.



GCN Residual error distribution: this distribution was formed for demonstrating the difference between the actual COVID-19 values minus the prediction values made by GCN model. Let's take a look on the graph and examine the difference between these values, on analysis it was found that the larger the residual value gives an indication of greater errors in prediction made by the model, smaller the residual value corresponds to highly accurate predictions, However an important point to be noted that presence of some larger residual might indicate cases for model finding it difficult to make predictions mainly for countries with very unusually extreme data or rapidly changing infection.

The following histogram shows difference between actual COVID-19 case values with that of predictions made by GCN model.



For the final one we have GCN vs GAT performance metrics comparison table which compares both the models mean absolute error, mean absolute percentage error, root mean squared error, and the coefficient of determination R2. According to the research it is found that lower the value of metrics means better prediction by the model while higher the value means greater inaccuracy in prediction made by a specific model, in terms of the coefficient of determination R2 equals to better result and vice versa. These data give us a clear view for deciding which graph neural network (GNN) architecture performs better on the COVID-19 forecasting tasks.

A table is presented below comparing all these data of our GCN and GAT models:

GCN vs GAT Performance Metrics

Model	MAE	RMSE	MAPE	R2
GCN	3582992.00	1892.879	6745.285	0.2984
GAT	1625992.25	1275.144	6745.285	0.9668

V. LIMITATIONS OF STUDY

GCN, GAT models both illustrates strategic performance in modeling/learning COVID-19 transmission patterns; however, several limitations need to be considered, graph constructed is based only on geographical border connectivity, and does not include factors like air travel, mobility features, vaccination rates, etc. Secondly, models trained relies heavily on global historical confirmed cases and hence cannot account for sudden changes caused by new variant, Finally the current implemented models focus on single step prediction rather than a recursive multi step forecasting.

VI. FUTURE WORK

Future research work can upgrade this demonstrated framework by integrating additional features like, air travel, mobility features, vaccination coverage, all into a graph representation. More advanced GNN's like Graph SAGE, graph transformers, TGN. (Banerjee et al., 2022; Schmidt et al., 2024)

Future studies can implement a recursive multi step forecasting to predict infection spread.

VII. CONCLUSIONS

This following research-based study presented a graph-based deep learning of COVID-19 cases prediction via graph neural networks like graph convolutional networks (GCN) and graph attention networks (GAT). The proposed methodology included countries as connected nodes within the graph, enabling neural networks to interpret spatial arrangement of neighboring countries and understand historical patterns of infection spread, the john Hopkins COVID-19 dataset was used for experimental evaluation as both GCN, GAT models demonstrated effective learning of meaningful insights of pandemic data, the comparison of training loss, scatter plots, residual error distribution shows a comprehensive assessment of model performance.

Overall, it shows that GCN, GAT are potential and an effective approach for analyzing and predicting infectious diseases.

VIII. ACKNOWLEDGMENT

The following research paper involves the use of artificial intelligence applications like (ChatGPT, and claude.AI). All the information provided in the paper had been gathered from the internet and details regarding the used sources are provided in the reference section.

IX. Competing Interest Declaration

The author declares that there are no competing interests.

X. Ethics Declaration

Ethics declaration: not applicable.

XI. Author Contribution Declaration

Tanmay Sharma: Conceptualization, methodology, data analysis, model development, visualization, writing, and editing.

XII. Data Availability Declaration

The data used in this study are publicly available datasets, including COVID-19 data from the Johns Hopkins University CSSE repository and publicly available geographic data. [7] Johns Hopkins Confirmed Cases CSV.

XIII. Consent to Participate declaration

Consent to Participate declaration: not applicable.

XIV. Consent to Publish declaration

Consent to Publish declaration: not applicable.

XV. Funding Declaration

No funding was received for this research work.

XVI. REFERENCES

[1] D. Woodward. E. B. Boltayevich, L. Vasudevan, M. Almusawi, H. Jani, and Y. Aziz, "Accurate COVID-19 Detection Using, Computing Models Using Artificial Intelligence and Big Data Analytics," International Conference on Sustainability, Innovation and Technology (ICSIT), 2025.

[2] Z. A. El Mouden, R. M. Taj, A. Jaikimi, and M. Hajar, "Towards Using Graph Analytics for Tracking Covid-19," Procedia Computer Science, vol. 177, pp. 204-211, 2020, doi: 10.16/j.procs.2020.10.029.

[3] Graph convolutional networks (GCN) explained -apxml.com

[4] [4]https://apxml.com/courses/introduction-to-graph-neural-networks/chapter-3-foundational-gnn-architectures/graph-convolutional-networks-gcn.

[5] https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

[6] . Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. The Lancet Infectious Diseases, 20(5), 533–534.

[7] Johns Hopkins Confirmed Cases CSV.

[8] https://www.naturalearthdata.com/?utm_source=chatgpt.com

[9] Huang, S., Yang, J., Fong, S., & Zhao, Q. "Artificial intelligence in the diagnosis of COVID-19: challenges and perspectives." *International Journal of Biological Sciences*, 17(6), 1581–1587, 2021. https://www.ijbs.com/v17p1581.htm?utm_source=chatgpt.com

[10] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. "A Comprehensive Survey on Graph Neural Networks." *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 4–24, 2021. https://ieeexplore.ieee.org/document/9046288

[11] Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions." *Journal of Big Data*, 11, 18, 2024. https://link.springer.com/article/10.1186/s40537-023-00876-4

[12] Song, K., Park, H., Lee, J., Kim, A., & Jung, J. "COVID-19 infection inference with graph neural networks."

Scientific Reports, 13, 11469, 2023. https://www.nature.com/articles/s41598-023-38314-3?utm_source=chatgpt.com

[13] Wang, L., Adiga, A., Chen, J., Sadilek, A., Venkatramanan, S., &Marathe, M. "CausalGNN: Causal-Based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 2022/2023 publication record. https://aaai.org/papers/12191-causalgnn-causal-based-graph-neural-networks-for-spatio-temporal-epidemic-forecasting/?utm_source=chatgpt.com.

[14]