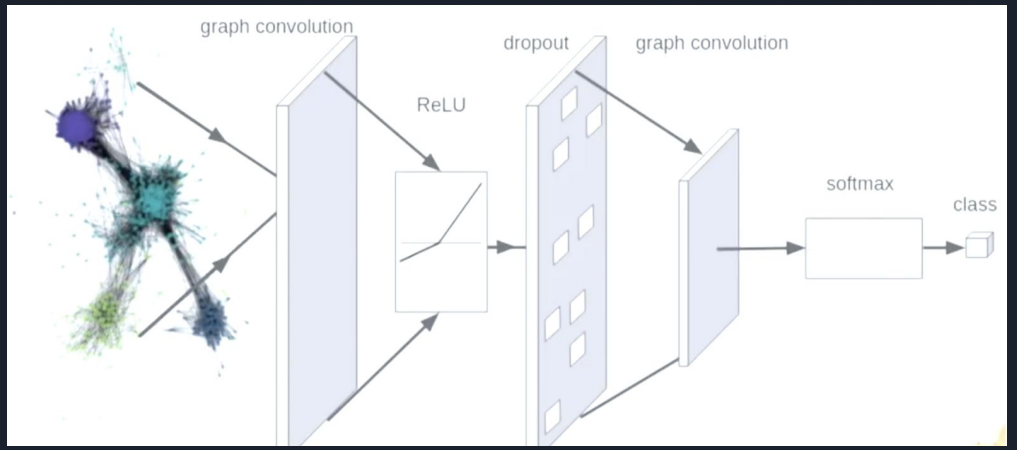
Graph Convolutional Network (GCN) Model Analyser

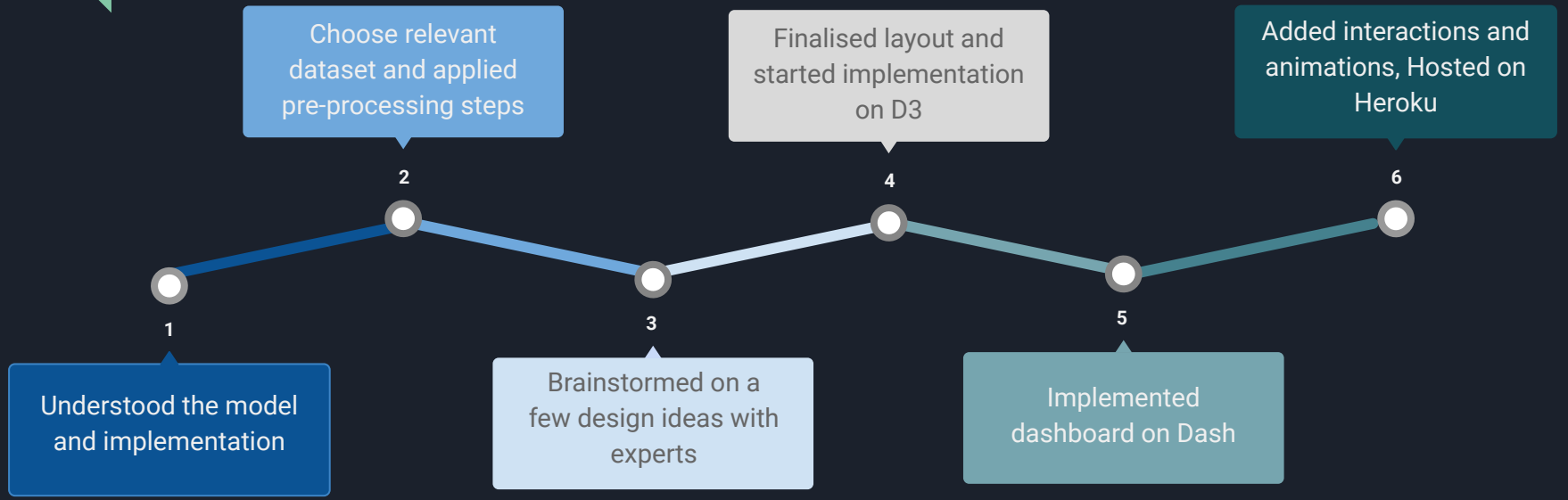
Mridula Gupta, Sarmishta Burujpalli

Objective

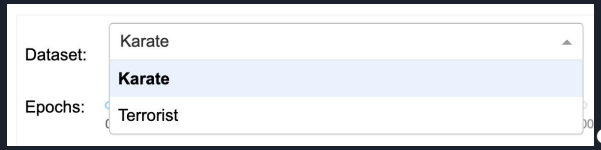
- Implement a system which can visualize the GCN Model
- Compare our results with the paper visualizations
- Add features to interact with the dashboard
- Should be extendable for future work



Roadmap

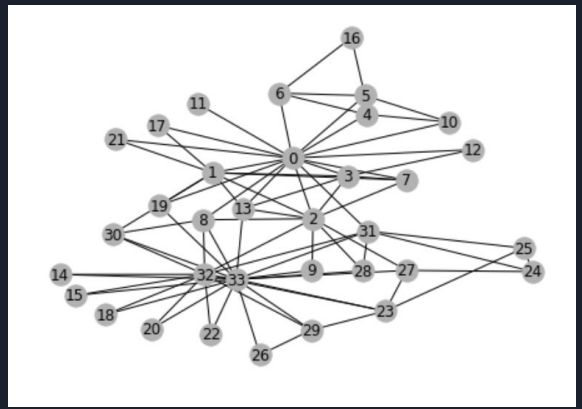

1
Understood the model and implementation
2
Choose relevant dataset and applied pre-processing steps
3
Brainstormed on a few design ideas with experts
4
Finalised layout and started implementation on D3
5
Implemented dashboard on Dash
6
Added interactions and animations, Hosted on Heroku

Datasets



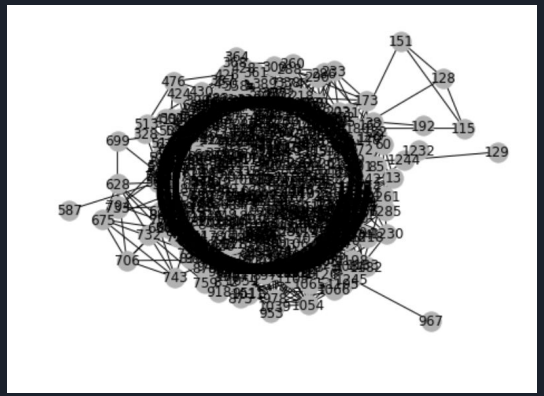
Karate Club Dataset(Smaller Dataset)

- Nodes - 34
- Classes - 4 categories of communities
- Features - Identity matrix
- Graph -
  - Nodes and relationship between nodes
  - Nodes and respective classes
- Usage- To compare the results with the paper



Terrorist Attack (Large Dataset)

- Nodes - 1293 attacks
- Classes - 6 categories of attacks
- Features - 106 distinct features
- Graph -
  - Nodes represent terrorist attack, relationship based on location
  - Around 3k edges connecting co-located terrorist attacks
- Usage- To make model generic
- Pre-processing Steps - Made changes to fit the GCN model's input while training



Dashboard Overview

Dashboard in Dash with 3 views:

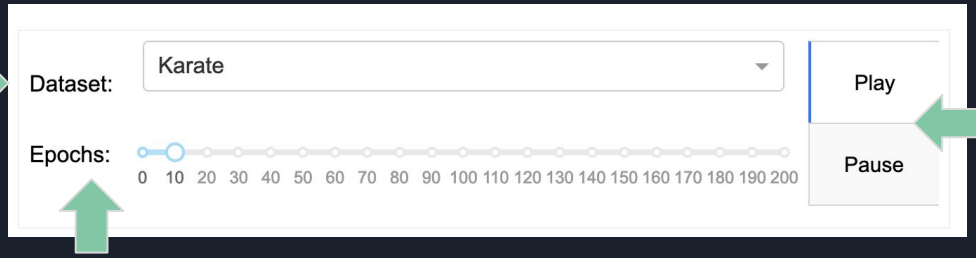
1. Model Training Overview - shows evolution of node embeddings
2. Hidden layer activations - shows node grouping after first Conv. layer
3. Feature Heatmaps - shows relevant/non-relevant features

Hosted on Heroku: https://my-dash-app-gcn3.herokuapp.com

Live Demo

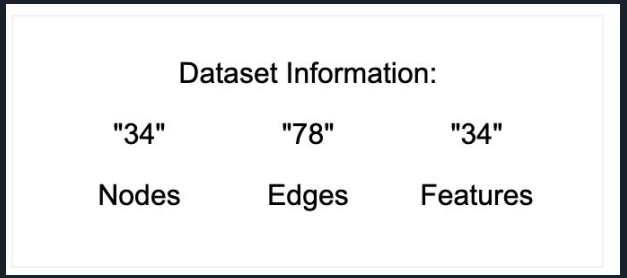
Dashboard Overview - Control Panel

Dataset Filter - Choose between Karate and Terrorist Datasets

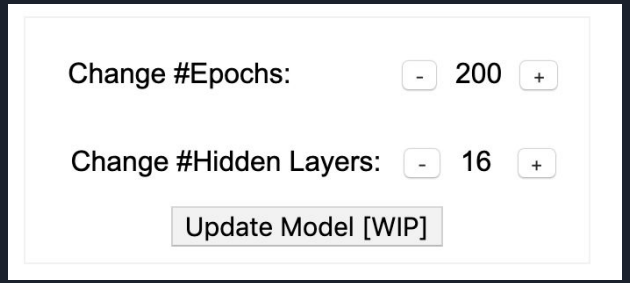


Play Pause functionality

Epoch Slider - Displays the current epoch, can be used to move



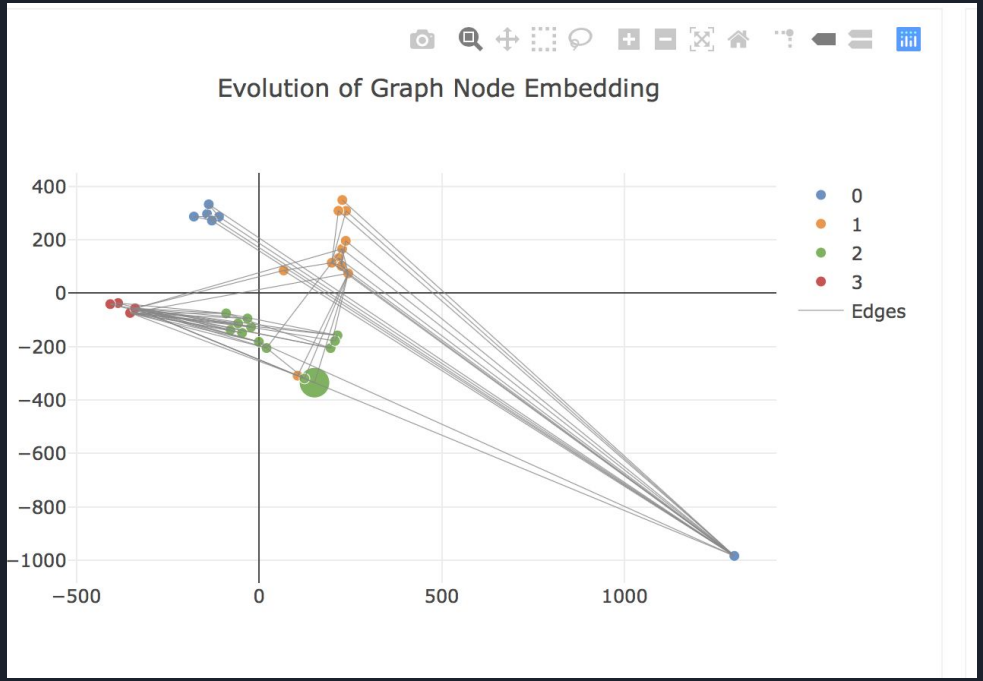
Dataset details can be seen on the screen



Model Parameters Update - Future Work

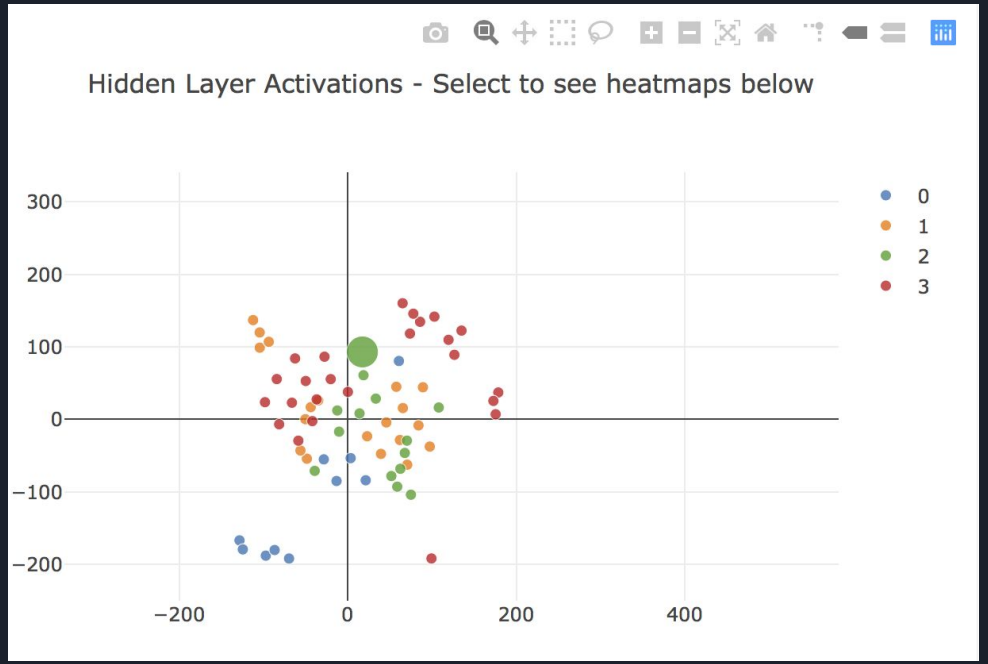
Dashboard - Evolution of Graph Node Embedding

a. Used DGL library to train the model in graph format
b. Follows technique of message passing between nodes
c. Networkx library- used to visualize in graph structure



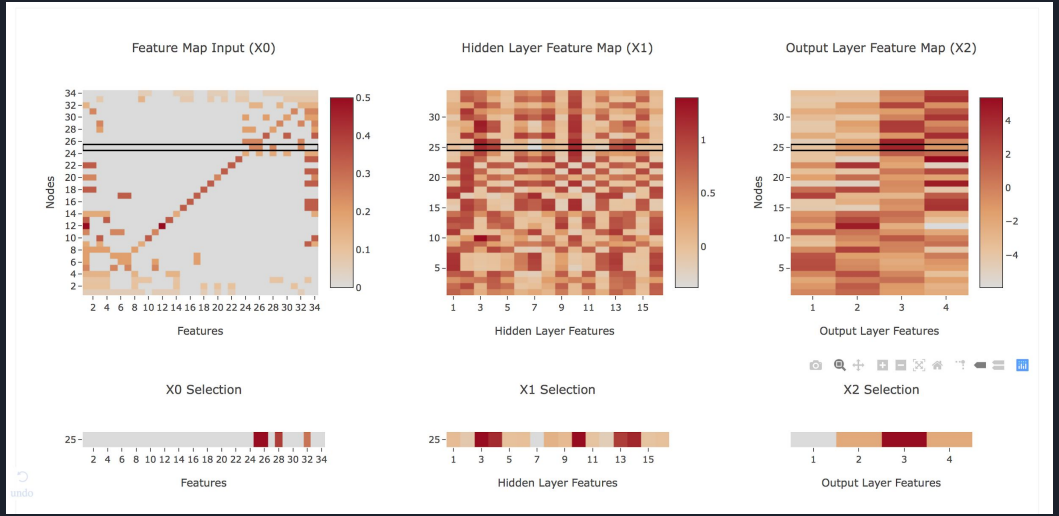
Dashboard - Hidden Layer Activations

a. Trained model & extracted activations of first layer
b. Applied T-SNE to reduce dimensionality
c. Scatter plot to visualize node groupings



Dashboard - Feature Maps across layers

a. Trained model and extracted features at end of each layer
b. Heatmap to visualize the relevance of features
c. Can be used for comparison with other nodes or see individual values



Contribution

Mridula

Backend -

1. Generated the dataset for weight and feature matrices
2. Created the datafiles for Viz 2 and 3

Frontend -

1. Started with Network and Heatmaps charts in D3
2. Implemented Viz 2 and 3 in Dash
3. Added Epocs slider and animations to all the charts
4. Added interactions within the charts
5. Deployed the app on Heroku

Sarmishta

Backend -

1. Implemented tSNE
2. Implemented Viz 1 in offline notebook
3. Created the datafiles for Viz 1

Frontend -

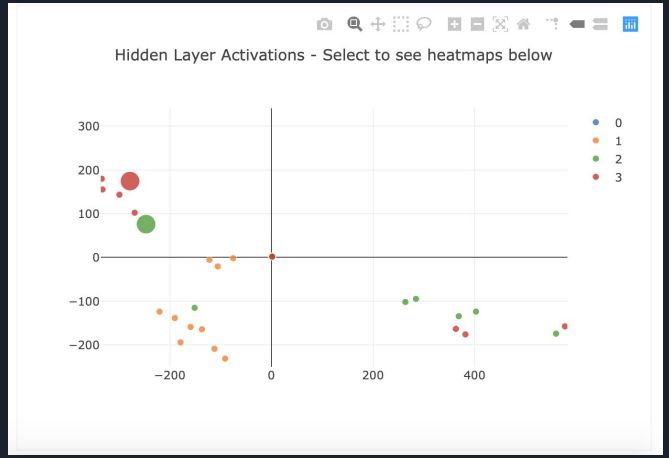
1. Started with tSNE visualization in D3
2. Implemented Viz 1 in Dash
3. Made the colors and legend consistent

Documentation -

1. Worked on the report

Unexpected Technical Issues

- Processing Issues - App worked perfectly on Server, failed on local machine after adding heavy interactions and full datasets
- Libraries Installation - Some libraries did not work on local machine after multiple attempts, so had to work on Kaggle
- Dash -
  - Not a lot of support available
  - Limited and buggy
- Kaggle -
  - Exporting data to csv takes longer
  - Sometimes exports wrong values

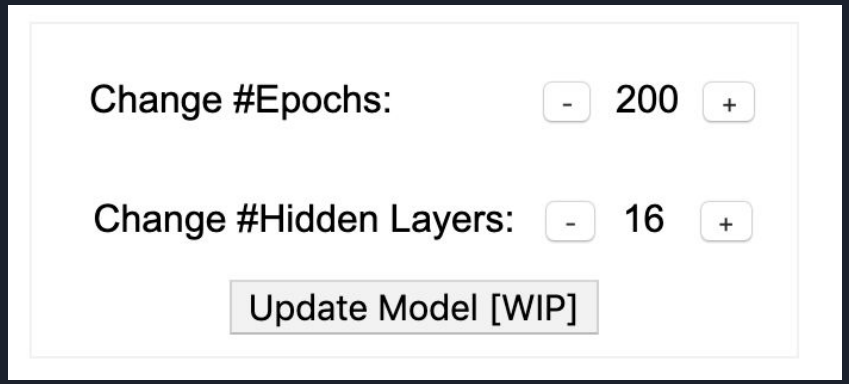


Case studies/Analysis results

- Used the GCN model to train on both datasets, making adjustments to increase the accuracy
- Extracted the intermediary parameters successfully
- Effectively visualized the GCN model for better understanding
- Dashboard supports selection of datasets
- Possibility to extend the dashboard for advanced features
- Dashboard can be extended for larger graph datasets/relevant use cases

Limitations and Future Work

- Make graph visualisation(large dataset) and interactions perfect
- Provide reset option for selections
- Support for updating the model
- Support to update training parameters like epochs, hidden layers, etc.
- Extend support to add more meaningful datasets



References

[1] https://dl.acm.org/citation.cfm?id=1248632

[2] https://arxiv.org/abs/1609.02907

[3] https://www.youtube.com/watch?v=UAwrDY_Bcdc&t=2s

[4] https://linqs.soe.ucsc.edu/data

[5]https://dash.plot.ly/getting-started

[6]https://discuss.pytorch.org

Thank you!!!