Alexandria National University Faculty of Computers and Data Science Cyber Security Program

Social Networks

Name: Kareem Ahmed Helmy Al-Abd

Student ID: 2205069

This project involves implementing a simple **GraphSAGE** model for classifying nodes in the graph under either of the two categories: Benign users or Malicious users.

1. Graph Structure

Total nodes in this graph are 6.

Each node contains two feature values:

- Feature values of Benign nodes are [1.0, 0.0] (representing the nodes are Benign while the other value is 0).
- Feature values of Malicious nodes are [0.0, 1.0] (representing the nodes are Malicious while the other value is 0).

There are 3 Benign nodes (0, 1, 2) and 3 Malicious nodes (3, 4, 5). Connection between nodes is defined by edges:

- The 3 Benign nodes (0, 1, 2) are Fully Connected.
- The 3 Malicious nodes are Fully Connected.
- Also, there is another connection defined from Node 2 (which belongs to Benign nodes) and Node 3 (which belongs to Malicious nodes) represented with only one edge because this may be the first interaction between the Benign and Malicious users.

2. Node Features (x)

These are feature values of nodes in the graph.

These values describe the type of nodes in the graph.

- Values of Benign nodes are: [1.0, 0.0].
- Values of Malicious nodes are: [0.0, 1.0].

These values are stored in the Node x.

3. Edge List (edge_index)

The edges in this graph are stored in the Node index.

This index contains values in pairs.

Values in each pair are edges of the graph representing undirectional edges.

The representation of connection between nodes 0-1 in this index are [0, 1] and [1, 0].

4. Labels (y)

These define the labels of nodes in this graph:

- Nodes 0-2 are Benign nodes and are represented with 0.
- Nodes 3-5 are Malicious nodes and are represented with 1.

These labels are utilized while training in computing the Loss.

5. GraphSAGE Model

The model used here is **GraphSAGE**, which works with graph-structured data. It consists of two layers of **SAGEConv** convolution operations, where information from neighboring nodes is aggregated to update each node's features.

- **First Layer (conv1)**: Aggregates information from neighboring nodes and applies a **ReLU activation function** to introduce non-linearity.
- **Second Layer (conv2)**: Aggregates features again and generates **class scores** (whether the node is benign or malicious). The output is processed through **Softmax** to convert it into probabilities for each class.

6. Training the Model

The model is trained using the **Adam optimizer** with a learning rate of 0.01 to adjust the model's parameters.

- **Loss function**: The **Negative Log-Likelihood Loss** (nll_loss) is used for classification tasks, where the model outputs probabilities for the classes.

The model is trained for 50 epochs, with the optimizer adjusting parameters based on the computed gradients from the loss.

8. Predicted Labels

After training, the model predicts the labels for the 6 nodes, and in this case, it classifies all nodes as malicious (label 1). This indicates that the model struggled to differentiate between benign and malicious nodes, possibly due to limited feature information and reliance on graph structure alone.

Predicted labels: [1, 1, 1, 1, 1, 1]