

### **Idea Of Code :**

The code is a simple example of a Graph Neural Network (GNN) using the *torch\_geometric* library.

We need to classify nodes to :

1. Benign users
2. Malicious users

### **What the code do ?**

Builds a small graph:

- There are 6 nodes (representing users).
- Each node has a small feature vector that indicates whether it is benign or malicious.
- Edges define which users are connected to each other  
(benign users are connected together, malicious users are connected together)

### **Puts the data in a suitable format:**

It uses the Data object from *torch\_geometric* to store:

- Node features (x)
- Graph edges (edge\_index)
- Node labels (y)

where 0 = benign, 1 = malicious

### **Defines a GraphSAGE model:**

The model consists of two **GraphSAGE** layers.

Each node representation is updated not only based on its own features, but also by **aggregating** information from its neighbors in the graph.

The final output of the model is a score log-probability for each class benign **vs** malicious for every node.

### **Trains the model:**

Uses the Adam optimizer with a suitable learning rate.

Uses a classification loss function (negative log-likelihood loss).

Runs several training epochs to minimize the loss and help the model learn how to distinguish between benign and malicious nodes.

### **Makes predictions:**

After training, the model is switched to evaluation mode.

It predicts a label (0 or 1) for each node in the graph.

The predicted labels are printed to check how well the model has learned.