EXPLORATORY PROJECT REPORT

Link prediction techniques and performance

By -

Devasani Ramu (22075027) & **Vishnu Lal Maurya (22075094)**

Under the guidance of -

Dr. Bhaskar Biswas

INDEX

1. INTRODUCTION

In the world of studying networks, like social media or biological systems, predicting future connections between things is important. Link prediction is all about guessing these connections based on what we already know. This project is all about looking at different ways to do this and seeing how well they work in different situations. We'll be testing out these methods to see which ones do the best job.

2. OBJECTIVE

- To understand the concept of link prediction and its significance in network analysis.
- To explore various link prediction techniques including traditional methods and machine learning-based approaches.
- To implement selected link prediction techniques on different types of networks.
- To evaluate the performance of these techniques using appropriate metrics and datasets.

3. METHODOLOGY

- Traditional link prediction algorithms: Jaccard Coefficient, Adamic-Adar, Preferential Attachment, Katz , Random Walk with Restart.
- Machine learning-based approaches: Node embeddings, Message Passing using Graph Sage
- Evaluation metrics: AUROC, AUPR, Recall@10, Hits@k
- Dataset selection: Football, Karate Club Graph, Dolphin Social Network
- Implementation: Implement selected link prediction techniques using libraries and frameworks given as NetworkX and Pytorch.

4. LINK PREDICTION TECHNIQUES

1. Jaccard Coefficient

$$J(A, B) = |A \cap B| / |A \cup B|$$

$|A \cap B|$ denotes the count of common neighbors of A and B.

$|A \cup B|$ denotes the count of nodes that are neighbors to either A or B

2. Adamic Adar Index

$$AA(u, v) = \sum_{w \in N(u) \cap N(v)} \frac{1}{\log(|N(w)|)}$$

N(u) represents the set of neighbors of node u.

N(v) represents the set of neighbors of node v.

$N(u) \cap N(v)$ represents the intersection of the neighbor sets of nodes u and v.

$|N(w)|$ is the degree (number of neighbors) of node w, which is a common neighbor of nodes u and v.

3. Preferential Attachment

$$PA(u, v) = k_u \times k_v$$

K_u is the degree (number of connections) of node u.

K_v is the degree of node v.

4. Katz Index Formula

$$Katz(u,v) = \sum_{l=1}^{\infty} \beta^{l} \times N_{uv}^{(l)}$$

β is a parameter (0 < β < 1) that controls the contribution of indirect connections. A smaller value of β gives more weight to shorter paths, while a larger value gives more weight to longer paths.

$N^{(l)}_{uv}$ is the number of paths of length l between nodes u and v.

5. Random Walk with Restart

Transition Matrix:
The transition matrix **M** is defined such that a random walker iteratively moves to an arbitrary neighbor and returns to the same starting vertex with probability **α**. The transition matrix can be expressed as:

$$M = (1-\alpha)\cdot I + \alpha \cdot A$$

Where:
I is the identity matrix.
A is the adjacency matrix of the graph.
RWR(v) is the **RWR** score of vertex v.
α is the restart probability.
P is the seed vector (a vector with zeros for all components except the element corresponding to the starting vertex).

$$\text{RWR}(v) = (1-\alpha)\cdot M \cdot \text{RWR}(v) + \alpha \cdot \text{P}$$
$$S(u,v) = \text{RWR}(u) + \text{RWR}(v)$$

6. Link Prediction using Graph Sage neural network

GNN **Graph Neural Networks** layers have two primary components:

- **Message passing** between nodes and their neighbors
- **Aggregation** of received messages for each node to update the node's embedding

Let $\mathbf{h}_v^{(l)}$ be the **node embedding** vector of node v at layer ll of the GNN. The embedding at the next layer $\mathbf{h}_v^{(l+1)}$ is determined by the messages received from neighbors, the aggregation function applied to these messages, and how this result is combined with $\mathbf{h}_v^{(l)}$. We can formally express this using the following equation:

$$h_v^{(l+1)} = \text{UPDATE}(h_v^{(l)}, \text{AGG}(\{m_u^{(l+1)}, \forall u \in N(v)\}))$$

where **N(v)** is the set of neighbors of **v**, $\mathbf{m}_u^{(l+1)}$ is the message passed by neighbor **u** in layer **l+1**, AGG is the aggregation function applied to messages, and UPDATE is the update function to compute the updated embedding. Different GNN layer architectures use different message, aggregation, and update functions. We will use the **GraphSAGE** framework here.

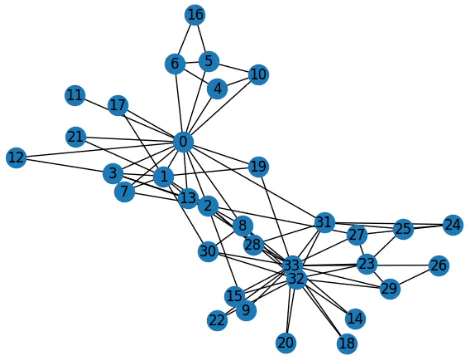
$$h_v^{(l+1)} = W_1 \cdot h_v^{(l)} + W_2 \cdot \text{mean}(\{h_u^{(l)}, \forall u \in N(v)\})$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices

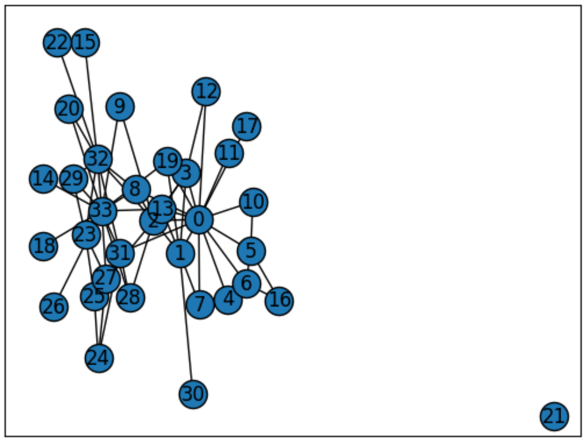
5. DATASETS USED

1. Karate Dataset

Number of nodes : 34
Number of edges : 78
Number of connected components : 1

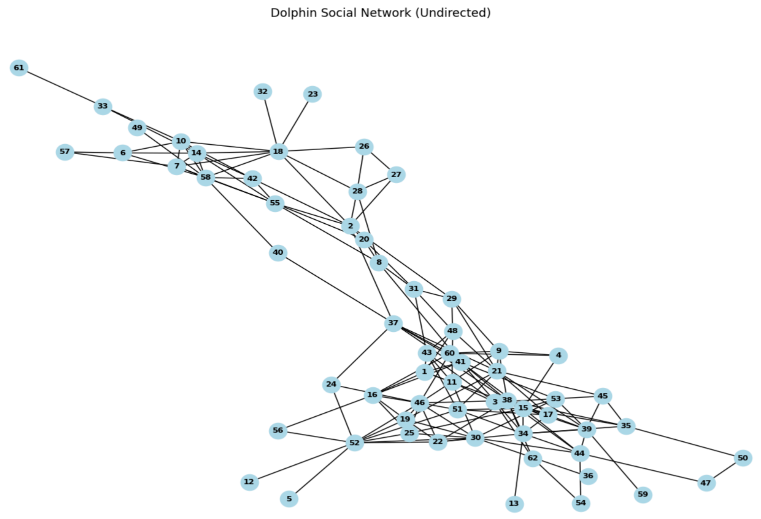


Number of edges deleted : 19
Number of edges remaining : 59

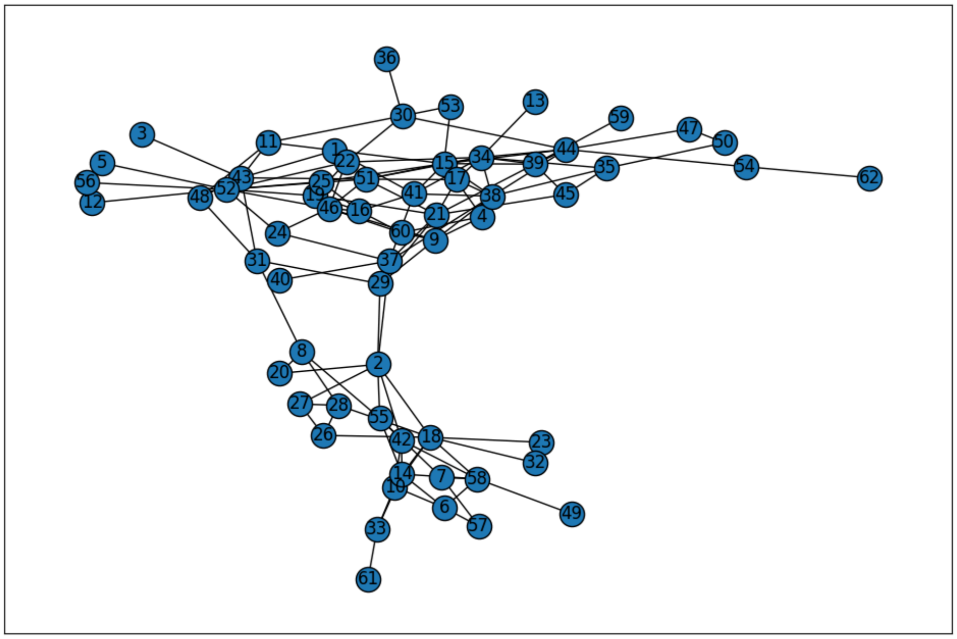


2. Dolphin Dataset

Number of nodes : 62
Number of edges : 159
Number of connected components : 1

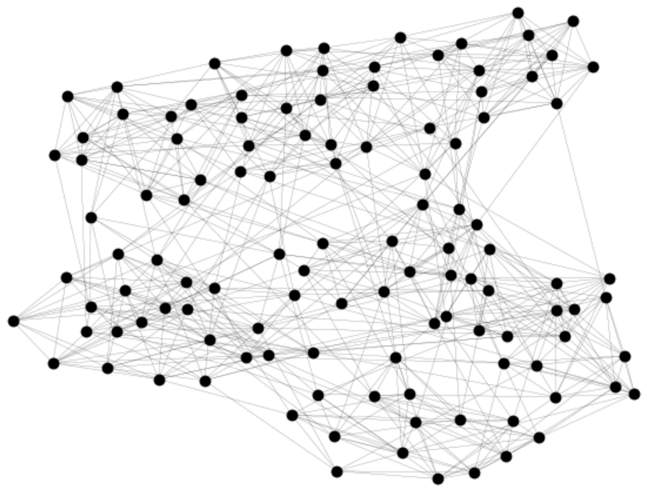


Number of edges deleted : 31
Number of edges remaining : 128

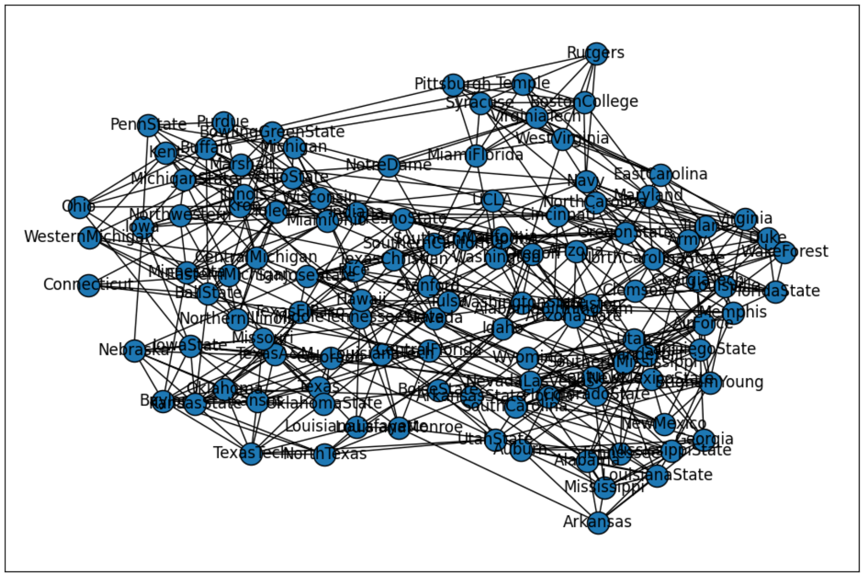


3. Football

Number of nodes : 115
Number of edges : 613
Number of connected components : 1



Number of nodes deleted : 122
Number of edges remaining : 491



4..OGBL DDI graph object:

Number of nodes |V|: 4267
Number of (training) edges |E|: 2135822
Is undirected: True
Average node degree: 500.54
Number of node features: 0
Has isolated nodes: False
Has self-loops: False

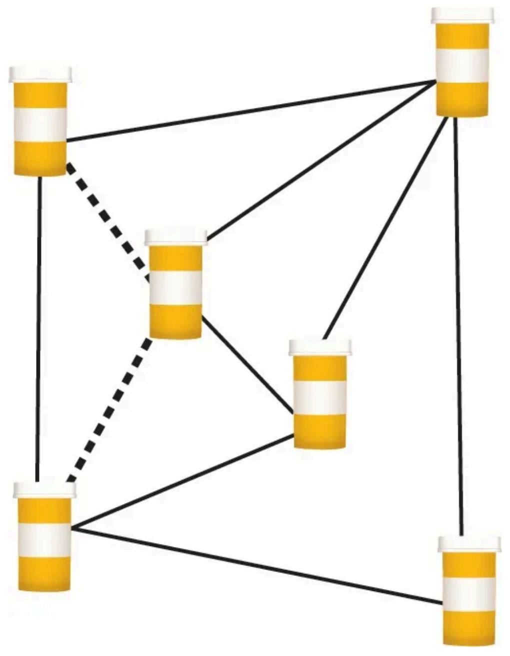


Illustration of a sample DDI graph. The DDI graph is undirected, unweighted, and homogenous (i.e. there is only a single “type” of node and a single “type” of edge). In this visualization, solid lines are known interactions, and dashed lines represent “missing edges”, i.e., edges that represent interactions that we would like to predict.

After data splitting
Number of training positive edges: 1067911
Number of validation positive edges: 133489
Number of validation negative edges: 101882
Number of test positive edges: 133489
Number of test negative edges: 101882

6. EXPERIMENTAL ANALYSIS AND RESULTS

1. AUROC results

	Jaccard Coefficient	Adaamic Adar	Preferential Attachment	Katz Index	Random Walk with Restart
Karate	0.5478914677999347	0.6139806036831208	0.6139806036831208	0.2706422018348624	0.8545893719806763
Dolphin	0.6664426523297491	0.6742191500256016	0.6058307731694829	0.8168754850870386	0.6080421664307532
Football	0.8463549784480441	0.8306491922047489	0.3370341606561254	0.816070374273717	0.3199328481330457

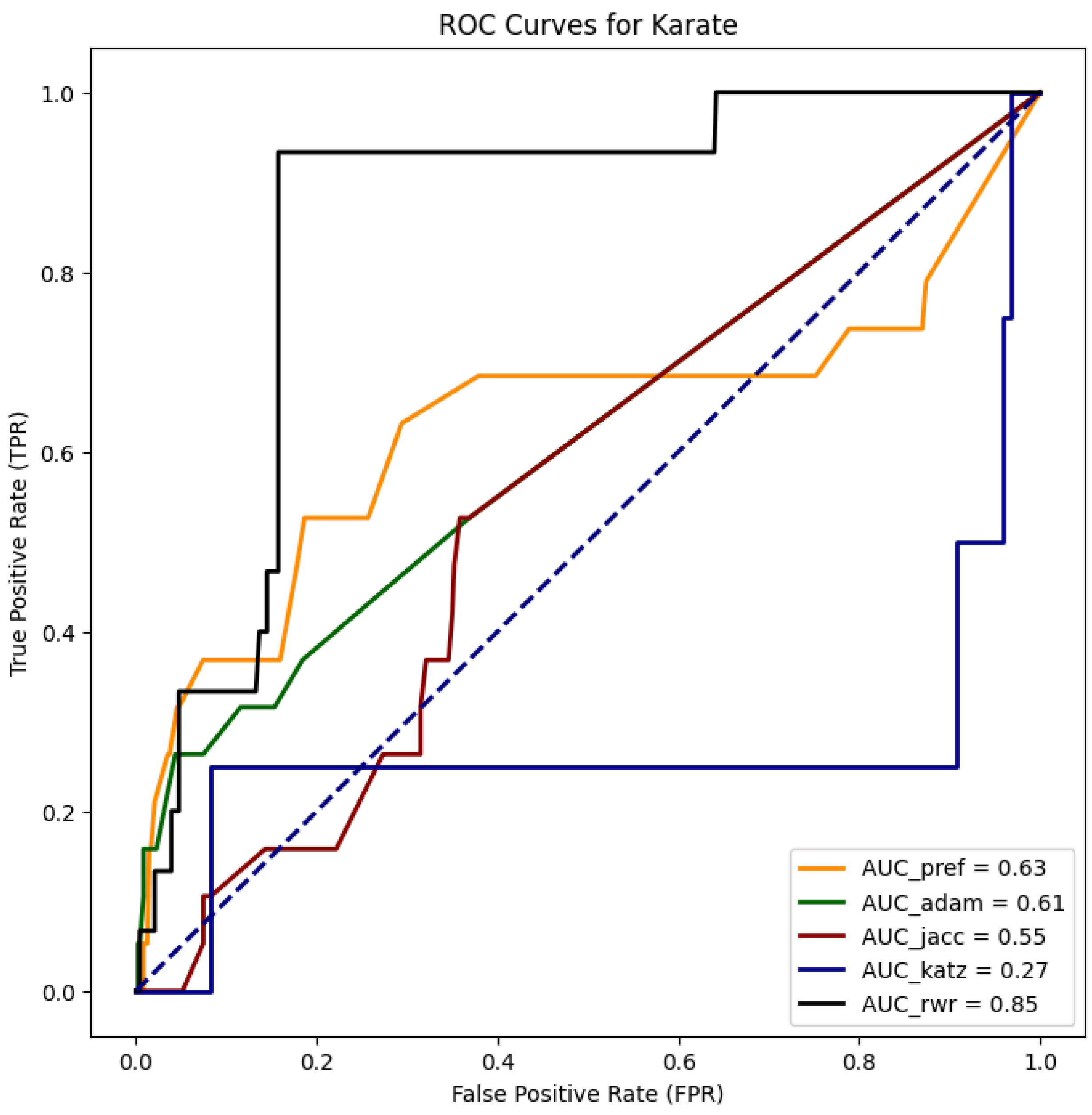
2. AUPR results

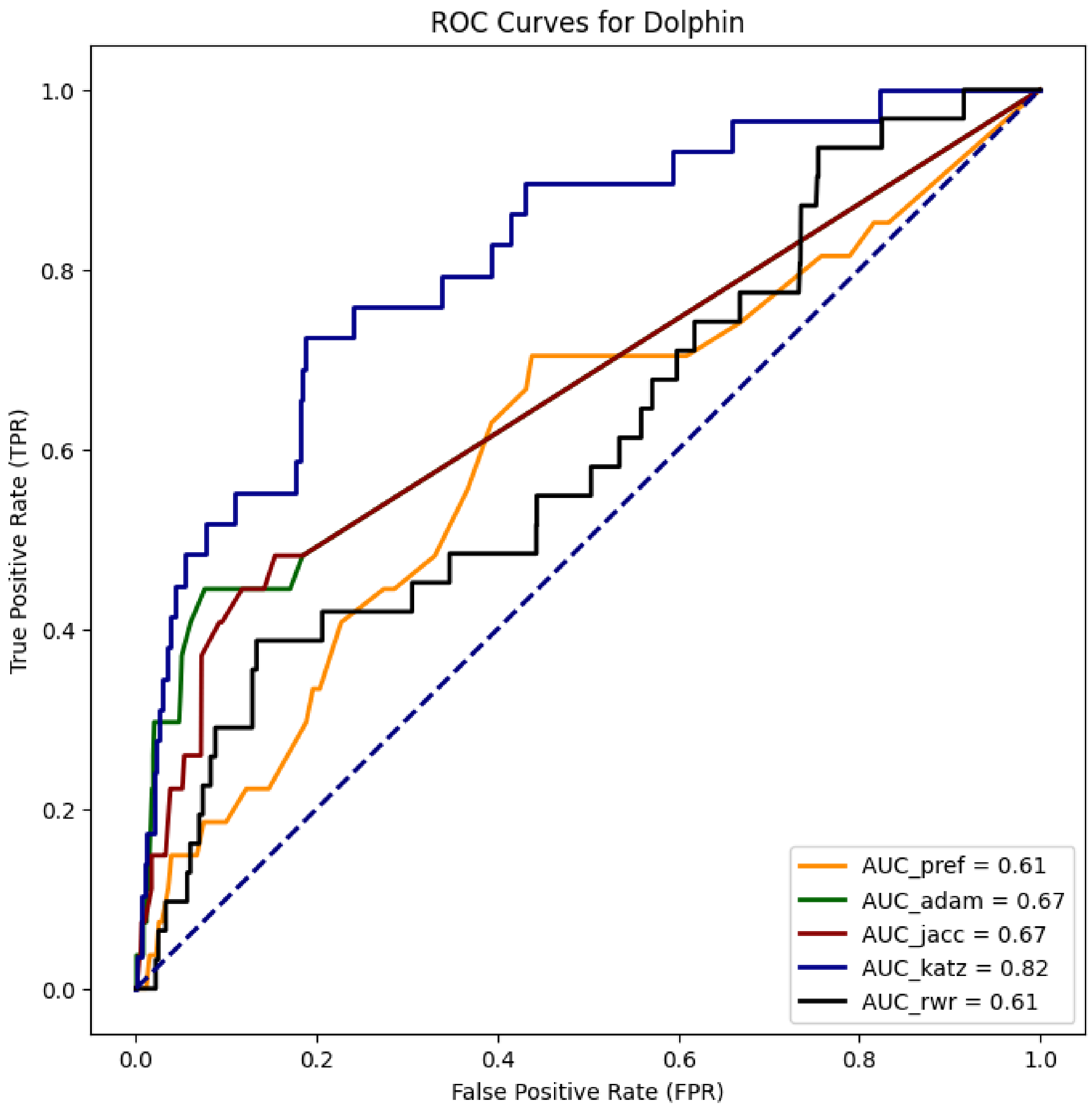
	Jaccard Coefficient	Adaamic Adar	Preferential Attachment	Katz Index	Random Walk with Restart
Karate	0.040329571690674124	0.12191699605365841	0.12191699605365841	0.09652545522110739	0.1443073475283213
Dolphin	0.04908800289835095	0.8306491922047489	0.02559764589728517	0.12773603737942216	0.031179647988329676
Football	0.1948727421360177	0.15528768506722235	0.007364120724981913	0.17925603817806487	0.013823827492745624

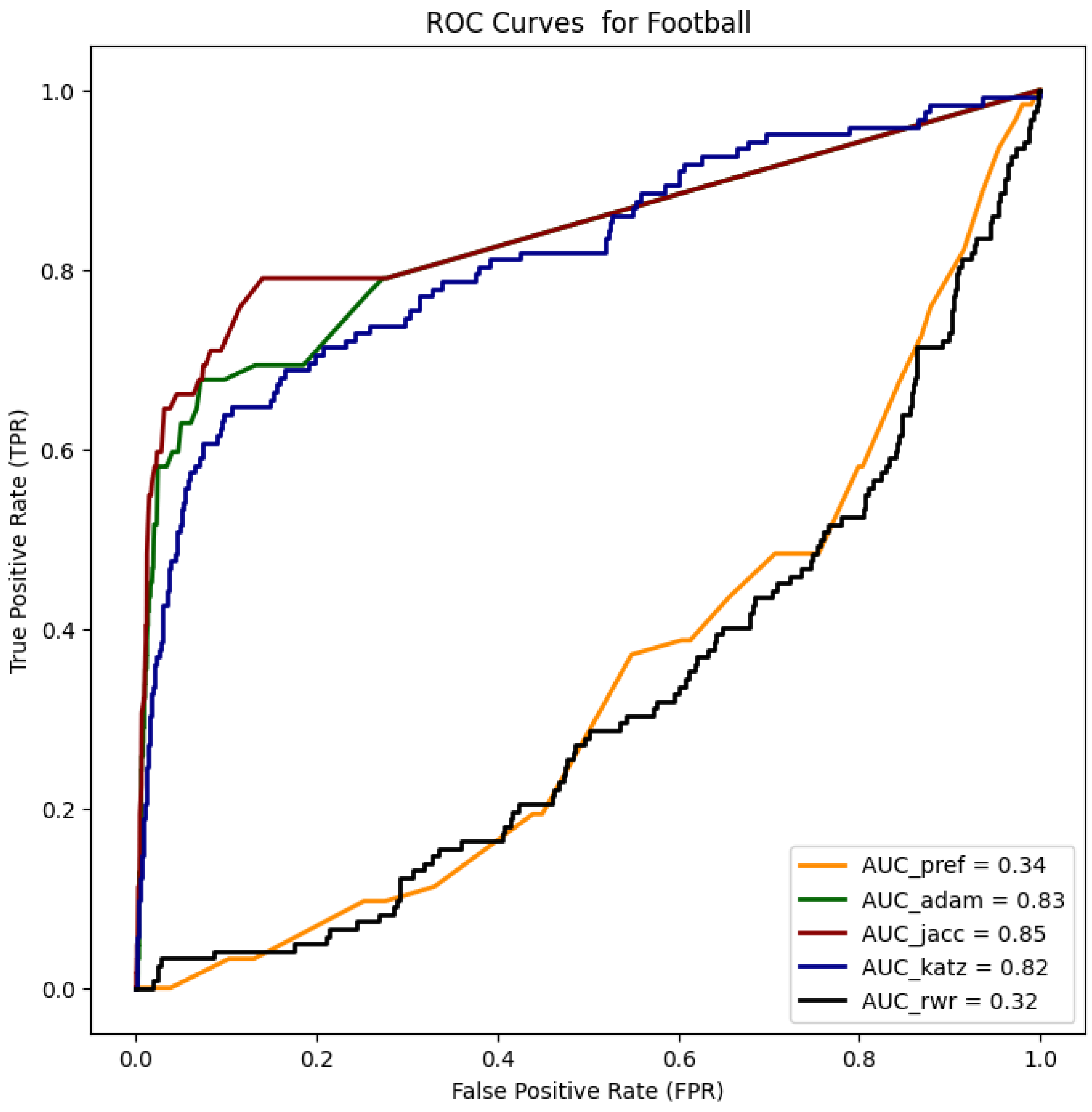
3. Recall@10 results

	Jaccard Coefficient	Adaamic Adar	Preferential Attachment	Katz Index	Random Walk with Restart
Karate	0.0	0.0	0.0	0.0	0.06666666666666667
Dolphin	0.037037037037037035	0.0	0.0	0.034482758620689655	0.0
Football	0.04838709677419355	0.0	0.0	0.00819672131147541	0.0

Plots:


ROC Curves for Karate
True Positive Rate (TPR)
False Positive Rate (FPR)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
AUC_pref = 0.63
AUC_adam = 0.61
AUC_jacc = 0.55
AUC_katz = 0.27
AUC_rwr = 0.85

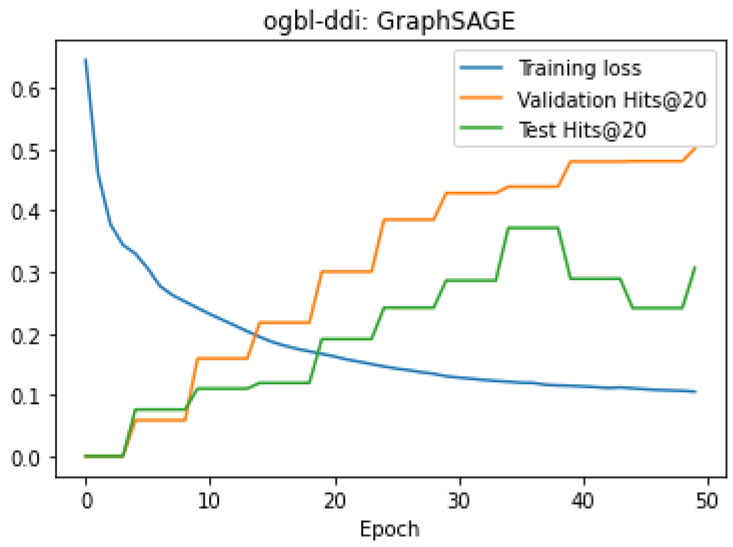

ROC Curves for Dolphin
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
AUC_pref = 0.61
AUC_adam = 0.67
AUC_jacc = 0.67
AUC_katz = 0.82
AUC_rwr = 0.61


ROC Curves for Football
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
AUC_pref = 0.34
AUC_adam = 0.83
AUC_jacc = 0.85
AUC_katz = 0.82
AUC_rwr = 0.32

Evaluation for Graph Sage GNN

For the evaluation of Graph Sage model on the ogbl-ddi dataset we have used **Hits@20 metric**

Hits@K is equal to the ratio of positive edges that have a prediction value higher than the K-th highest value negative edge.



7. CONCLUSION

In this survey, we have gone through various methods of link prediction. We implemented similarity-based methods such as the Jaccard Coefficient, Adamic Adar, and Preferential attachment, We also implemented some global methods such as Katz index and Random Walk with Restart.

After analyzing the plots of different datasets, we found out that the Katz index performs well when the number of nodes and edges are high as in the case of football and dolphin datasets.
In the football dataset, the similarity-based methods did better than the global methods.
In the karate dataset, the performance of all the techniques is not so good as the number of nodes and edges is very low.
Admic Adarm Jaccard Coefficient and Preferential Attachment techniques performed average in all the datasets.

In Graph Neural Network, we implemented Graph Sage in a simple way with two layers of message passing. Hits@20 metric reached 0.5 after the training and validation which shows that out of the top 20 predicted scores of edges, 10 of the edges exist . Adding more layers and attributes and running can further improve the results.