International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
***Article Received:** 07 June 2023 **Revised:** 25 July 2023 **Accepted:** 16 August 2023*

Forecasting the Missing Links in Heterogeneous Social Networks

Atika Gupta[1], Priya Matta[*2], Bhasker Pant[3]
[1]Graphic Era Deemed to be University
Graphic Era Hill University
Dehradun, India
atika04591@gmail.com
[2]Tula's Institute
Dehradun, India
mattapriya21@gmail.com
*Corresponding Author
[3]Graphic Era Deemed to be University
Dehradun, India
pantbhaskar2@gmail.com

Abstract—Social network analysis has gained attention from several researchers in the past time because of its wide application in capturing social interactions. One of the aims of social network analysis is to recover missing links between the users which may exist in the future but have not yet appeared due to incomplete data. The prediction of hidden or missing links in criminal networks is also a significant problem. The collection of criminal data from these networks appears to be incomplete and inconsistent which is reflected in the structure in the form of missing nodes and links. Many machine learning algorithms are applied for this detection using supervised techniques. But, supervised machine learning algorithms require large datasets for training the link prediction model for achieving optimum results. In this research, we have used a Facebook dataset to solve the problem of link prediction in a network. The two machine learning classifiers applied are LogisticRegression and K-Nearest Neighbour where KNN has higher accuracy than LR. In this article, we have proposed an algorithm **G**raph **S**ample **A**ggregator with **L**ow **R**eciprocity, ***(GraphSALR),*** for the generation of node embeddings in larger graphs which use node feature information.

Keywords- social network analysis, link prediction, criminal network, heterogeneous network.

I. INTRODUCTION

Online Social Network(OSN) is a structure of several actors who interact with each other over the internet. The nodes or the vertices represent the people, and the edges represent the interactions, influence or collaboration between these nodes. The exchange or the relationships shown are established due to the mutual interest of the nodes or when they are friends or colleagues. As these relations constantly change, new edges can be added or deleted at any time. So, a particular network snapshot at time *t1* may differ from that at time *t2*. If two accounts of the same person are on various networks, an interlayer link will exist between the nodes[1]. Various OSNs provide users different functionality, such as using Facebook or Twitter for personal interaction and LinkedIn for professional networks. Also, these social networks are highly complex and dynamic. One of the critical issues with the social network is the link prediction problem which has gained much attention over time because it is essential in data mining and analysis of the evolution of social networks[2].

Criminal networks are described as a group operational beyond the limitations of the law and making a profit using an unlawful manner which is generally gained by detriment to other individuals or organizations[3]. Due to this reason, researchers and scholars are significantly adapting to social network analysis to investigate the criminal trend, as it is a potent tool to analyze criminal networks and get richer insight into their behaviour.

Graph theory is highly used for analyzing these networks and has proved to be a powerful tool. Graph theory provides the theoretical structure, procedures and techniques for graph analysis[4]. Criminal networks, in particular, show a relatively high tendency to have hidden or missing links because of illicit activities' secret and covert nature. The characteristic of criminal networks, such as inadequate, unreliable and incorrectly captured data, is either caused by deliberate deception or illegal or unintentional human error. Therefore, the practice and technique of predicting hidden and missing links between nodes in most OSNs have significantly become relevant and essential in criminal networks.

Criminal network analysis is also helpful in investigating by producing link charts to identify the target and the key actors[5]. The analysis of relationships amongst individuals based on the activities and events obtained from various investigation

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
***Article Received:** 07 June 2023 **Revised:** 25 July 2023 **Accepted:** 16 August 2023*

activities. In this study, a computational issue involved with the social network analysis is evaluated, *the link-prediction problem*. There are many applications of this link-prediction problem such as Online Social Network(Recommend friends to connecta and suggest friends), E-commerce(product recommendation), Bioinformatics(predict protein-protein interaction), Citation Network(predict missing citations, predict future collaboration), Police and Military(Identify hidden group of terrorist, spotting the criminals).

We are trying to focus on spotting out the criminal from there network which can generate future links. Here, given a snapshot of the social network at time *t1*, we have to accurately predict the edges that will be added to this particular network with time and show the image of the same network at time *t2*. Most link prediction models utilize nodes, edges and topological features to predict the advantages that can be formed over time. There is not much work in the literature which describes the heterogeneous link prediction. The solutions available treat all the links as the same and destroy the heterogeneous data available for the network. Missing data is one of the significant challenges when analyzing social networks for criminal identification. When the investigator relies on the data gathered, which, when incomplete, can lead to false positives, missing data in network analysis can be referred to as missing nodes or missing edges.

In a criminal network, the investigator may get the benefit of identifying the connection between two individuals, which is still unknown, using the link prediction. However, this task is pretty challenging as the data is noisy and incomplete because the criminal does not want to leave traces behind. This may raise a concern about precision and accuracy.

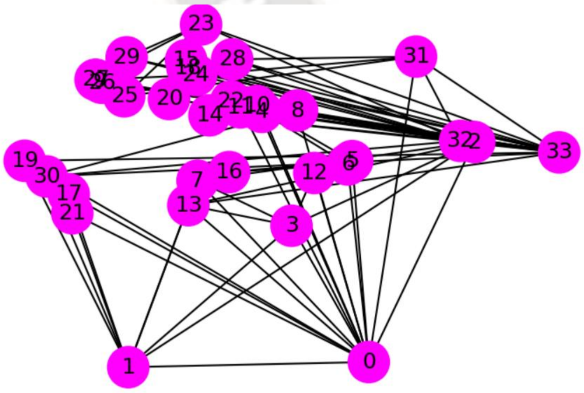


Figure 1. Example of a Social Network

"Figure 1.", shows an example of a social network where the nodes are connected.

II. LITERATURE REVIEW

Most of the existing work done in the literature divides the problem of link prediction into two categories: the learning-based approach and the similarity-based approach[6]. The similarity-based process computes the similarity between two nodes using various graph-based methods and then uses the ranking algorithm to find the link between two nodes. On the other hand, learning-based approaches utilize the computation models such as machine learning but suffer from model capacity. Also, the models are designed to evaluate the static view of the network, but not to forget social networks are highly dynamic and change with time, so we cannot predict a link based on a static snapshot.

[7] the author's research is based on applying SNA models and machine learning metrics. They utilized the gradient boosting machine(GBM) technique to enhance link prediction accuracy on large datasets. To evaluate the performance of the algorithm constructed using GBM, an experiment was conducted using statistical and link prediction models on the smaller criminal dataset. The dataset used for the research was significant compared to other studies and showed the natural characteristics of the criminal network.

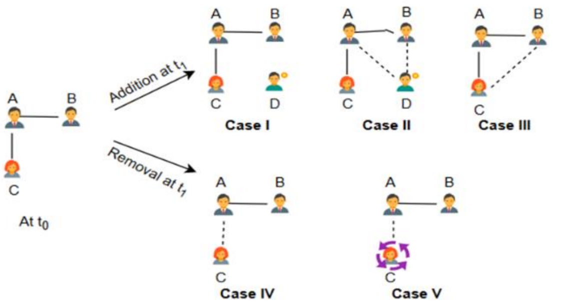


Figure 2. Transition from time t0 to t1.

[8] In the past decade, the problem of interlayer link prediction in a heterogeneous network made up of different OSNs has gained attention. For this purpose, features have been extracted from the profile and the content. These approaches have involved machine learning techniques for predicting links in multiplex networks.

[9] the author here has introduced a unified approach for the user identity linkage problem, which consists of two stages: feature extraction and model construction. The feature extraction phase gathers features like profile, content and network, whereas the model construction can use supervised, semi-supervised or unsupervised approaches. They explain that meta-path-based techniques can operate link prediction at the same time on different networks.

[10] the author of this article investigates if a user can be identified using tagging behaviour and profile attributes. The author found that different people use different keywords to

represent tags on social media networks. A method was proposed for user identification leveraging the user's tagging behaviour and profile attributes. The experiment results give better results than the other existing methods. However, only two SNS were involved in the experiment.

[11] The existing natural systems comprise many interacting real-world entities, their structure is multi-layered, and most of the research work considers it a homogenous network without considering the different types of networks and the link between them. In recent times, only some researchers have started feeling the heterogeneity of the networks. Compared to the homogenous networks, the heterogeneous network consists of richer semantic information that provides lots of challenges and opportunities for data mining.

[12] the author has proposed a method based on embedding and match based model for anchor link prediction, named PALE(Predicting Anchor Links via Embedding). This method comprises two stages: embedding and matching. The idea is to first predict the embedding by considering the regularities in the significant structure. Despite using human-defined structural features, the network embedding converts the features into low-dimensional space, by which frequencies can be observed, and insignificant details are discarded.

[13] The article addressed the problem of user identification linkage across SMNs and offered an innovative solution. A uniform network structure-based user identification solution is proposed along with a friend relationship-method-based algorithm FRUI. The experiment was conducted on Twitter, Facebook and the Foursquare dataset. The results reveal that the network structure can achieve better user identification. The challenge with the persisting study is that only a portion of identical users with different nicknames can be identified.

[14] the author of this article has studied the problem of identity matching across two or more social networks. This task is challenging due to the cost involved, heterogeneous attributes for each network, and incomplete and missing data. An unsupervised method is introduced to address this challenge and illustrated using three real datasets. The experiment results show that the algorithm proposed outperforms the datasets used.

III. PROBLEM STATEMENT

There are issues with these social such as a network that is not static. It is subjected to temporal changes. Static network means a single snapshot of a network is considered, where the nodes are at their fixed place for ages, no new node is coming to get added to the web, and no node is deleted from that network. But what would happen if the network changed from time to time? We know that in a real-world scenario, the network changes from time to time, and sites like Facebook, Twitter etc., are having dynamic nature. This concludes that we have some network topology at time t_0 and some other topology at time t_1, where (t_1>t_0).

$$\mathbf{G_{t0}(V,E): topology\ at\ time\ t=t_0}$$
$$\mathbf{G_{t1}(V',E'): topology\ at\ time\ t=t_1,}$$
$$\mathbf{Where,\ (t_1>t_0)}$$

If we consider these changes, there are five possible cases from time t_0 to time t_1. "Figure 2." explains these cases below:

- Case I: new nodes are added, forming no new link with the existing nodes.
- Case II: new nodes are added, creating new links with existing nodes.
- Case III: new nodes are not counted; existing nodes make new connections.
- Case IV: Some current links are deleted, but the nodes remain.
- Case V: some edges are removed along with the nodes.

When we consider the link prediction problem, we have to understand that there are two kinds of link prediction problems: one is known as missing link prediction, whereas the other is called future link prediction. Missing link prediction is when you are given a static network, and your task is to predict if you have missed some of the edges in the network. For example, you have a network, but the data was so noisy that you missed some of the edges, and now to complete your Graph, you are taking the help of link prediction. The second problem is future link prediction; it says that if you have been given a graph at time t_0, your task is to predict the edges which will be added at time t_1. Future link prediction can be further classified into two categories: periodic, where the link is expected on time, and the other is non-periodic, where the link is predicted based on a particular snapshot of the network[15]. The link prediction problem is very complex in itself. We have mapped this problem to some approaches like the heuristic model, which covers local similarity metrics and global similarity metrics, the Probabilistic model and theoretical models.

IV. METHODOLOGY

Link prediction is a popular task in social network analysis which involves predicting the likelihood of a connection forming between two nodes in a network.

In this section, we discuss the techniques used for link prediction methods. Irrespective of the style used, the main objective of link prediction is to:

- To connect the nodes successfully, with some similarity, but are not yet linked.
- If the nodes are similar and closer, they are more likely to connect and interact with each other.

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
Article Received: *07 June 2023* ***Revised:*** *25 July 2023* ***Accepted:*** *16 August 2023*

The similarity of the two nodes can be derived using the following:

- Level of edges
- Level of nodes
- Level of information about the nodes.

1. Node-Based Similarity Measures

A. Common Neighbourhood

The simplest solution to solve this problem is to look at the neighbour node. It says if the nodes share more common neighbours, they are more likely to establish a link in the future.

$$S_{CN}(x,y)=|N(x) \cap N(y)|$$

S_{CN} (x,y) is the forecasted score of the hidden link between the x and y nodes.

The problem with this approach is that if one of the nodes is a hub node, then the intersection will increase. Another problem is that the familiar neighbourhood does not have any boundaries.

B. Jaccard coefficient

It overcomes common-neighbourhood problems and gives the scores generated by the previous algorithm in a normalized manner. It provides the probability score between the two nodes for the hidden links. The expression can be given as follows:

$$LP_{JC}(x, y) = \frac{|N(x) \cap N(y)}{N(x) \cup N(y)|}$$

Where LP_{JC}(x, y)is the forecasted score of the hidden link between node x and y, here denominator indicates the maximum possible familiar neighbours between node x and y.

C. Preferential attachment

It is generally seen that the chances of a link being established between two nodes are high if they have a higher degree than the node pair with a minor degree. If the nodes are highly connected, they will likely create new relationships with other nodes. The forecast score for hidden links can be calculated by multiplying the nodes' degrees. The expression is given below:

$$LP_{PA}(x, y) = N(x) * N(y)$$

LP_{PA}(x, y)is the forecasted score of the hidden link between node x and y using a Preferential Attachment.

D. Adamic Adar

According to this algorithm, a node pair is assigned a high score if the familiar neighbours are not shared with other nodes in the network. This idea is based on the fact that if the nodes are connected to a common neighbour, they are likely to establish a future link between them. The more the value between such pairs, the higher the chances of link establishment. The expression can be given as follows:

$$LP_{AA}(x, y) = \sum_{z \in \{N(x) \cap N(y)\}} \frac{1}{\log(N(z))},$$

LP_{AA}(x,y) is the forecasted score of the hidden link between node x and y using Adamic Adar.

E. Kartz measure

This algorithm falls under the global approach, which says that two nodes are likely to create a connection in future if they have a path between them. The chances of contact are better if the number of approaches is more and they are shorter as well. Kartz is the most exciting measure to quantify hidden links based on the number of paths available between these nodes. All the ways of less than the length of diameter are considered here. It can be calculated as the total sum of the existing paths between the nodes. The expression is given as follows:

$$LP_{KZ}(x, y) =\sum_{i=1}^{\infty} \beta i \,|path^{i}{}_{x,y}|$$

LP_{KZ}(x,y) is the forecasted score of the hidden link between node x and y using a Kartz measure.

F. SimRank

SimRank is also a global approach and considers the similarity score between the neighbourhood nodes. According to this measure, there is a high probability of establishing a new link between a pair of nodes if they share many similar neighbours. The expression is given as follows:

$$LP_{SR}(x, y) = \gamma \sum_{a \in N(x)} \sum_{b \in N(y) LPSR(a,b)} / |N(x)||N(y)| ,$$

LP_{SR}(x, y) is the forecasted score of the hidden link between node x and y using SimRank.

2. Machine Learning algorithms

Several machine-learning approaches can also solve the social network's link prediction problem. Some of the machine-learning approaches are discussed below:

A. Support Vector Machine(SVM)

It is a non-probabilistic machine learning model known as a support vector network. It falls under the category of supervised algorithms and can be used for both classification and regression. Given training set data, the algorithm classifies the new sample into two categories based on the feature set selected. One approach to using SVMs for link prediction is to represent nodes in the network as feature vectors and then train a binary classifier using the SVM algorithm. Each feature vector represents a node and contains information about the node's properties or characteristics, such as its degree, clustering coefficient, or other relevant measures.

B. Decision Tree(DT)

A Decision Tree is also a supervised machine learning algorithm which follows a tree-like structure which goes on by dividing the problem for reflection about a sample starting from the root to the conclusion, which is the leaves to the tree.

The decision tree can be constructed using various algorithms, such as ID3, C4.5, CART, and Random Forest. Once the decision tree is constructed, it can be used to predict whether two nodes are linked by traversing the tree from the root node down to a leaf node that corresponds to a class label.

C. Artificial Neural Network(ANN)

The idea is based on the biological neural network, where the nerve cells are used to process the information. The ANN model proceeds by dividing itself into layers where each layer has nodes connected to the previous node's layer based on the requirement. The networks established have different weights upon them. The other variation of this network is the feedforward and feedback networks.

One approach to using ANNs for link prediction is to use a multi-layer perceptron (MLP), which is a type of feedforward neural network that consists of one or more hidden layers of neurons between the input and output layers. Each neuron in the MLP receives inputs from the previous layer and computes a weighted sum, which is then passed through an activation function to produce an output. The weights and biases of the neurons are learned through backpropagation, which adjusts the weights to minimize the prediction error on the training data.

Another approach is to use a graph neural network (GNN), which is a type of neural network that operates directly on the graph structure of the network. GNNs can learn the node and edge representations in the graph by recursively aggregating the features of neighbouring nodes and edges.

D. Random Forest(RF)

Random forest collects different decision trees to create a forest that can be applied to regression and classification. So, the number of trees in the woods is directly proportionate to the accuracy of the result. When the dataset is large, it has more variables, making it difficult to cluster the data, so RF gives better accuracy for the data belonging to a similar group. The tree with the best classification is used based on the training set.

Random Forests have several advantages for link prediction tasks. They can handle both categorical and continuous features, handle missing data, and are relatively robust to outliers and noisy data. They are also easy to train and interpret and can be used to estimate the importance of different features in the prediction task.

E. K-nearest neighbor(KNN)

It is also a supervised machine-learning algorithm which is also used for both regression and classification. The basic assumption of this algorithm is that it assumes similar things appear closer to each other. The distance between the nodes is calculated based on Euclidean or other measures. The initialization of *K* depends on the number of chosen neighbours. KNN has several advantages for link prediction tasks. It is easy to implement, computationally efficient and can handle both categorical and continuous features. It is also a non-parametric method, which means it can capture complex relationships between features and labels without making any assumptions about the underlying distribution of the data.

3. ***Other Methods***

Here are some of the latest research developments in link prediction:

A. *Graph Neural Networks (GNNs):* GNNs have shown great success in link prediction by incorporating node features and the graph structure into the prediction model. Recent studies have explored different GNN architectures, such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE, to improve link prediction accuracy.

Unlike traditional neural networks that operate on vector data, GNNs operate directly on the graph structure of the network and can learn the representations of nodes and edges by aggregating information from their neighbouring nodes and edges.

The problem with which GNN suffers is the problem of over-smoothing. After some iterations, the interpretation collected from all the nodes in the graph becomes similar to each other.

GraphSAGE[16] overcomes the challenges which were imposed by GCN which were difficulty when learning from larger networks and difficulty to generalize the unseen nodes. It has two steps: Sample and Aggregate.

$$h_u^{(k)} = \sigma(W \cdot MEAN(\{h_u^{(k-1)}\} \cup \{h_v^{(k-1)}, \forall \nu \in N(u)\})$$

Where h_u is the hidden layer of node *u* at time *t+1*, σ is the ReLU and *W* is self-loop.

B. *Multi-Relational Link Prediction:* In real-world social networks, there are often multiple types of relationships between nodes. Multi-relational link prediction aims to predict links between nodes of different types. Recent research has explored novel approaches such as knowledge graphs and embeddings to improve multi-relational link prediction accuracy.

C. *Temporal Link Prediction:* Social networks are dynamic and links between nodes can change over time. Temporal link prediction aims to predict future links based on the past behaviour of nodes in the network. Recent research has explored deep learning methods, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory

(LSTM), to capture temporal patterns and improve link prediction accuracy.

D. *Negative Sampling:* Negative sampling is a widely used technique in link prediction to balance the positive and negative examples in the training data. Recent research has explored different sampling strategies to improve the performance of link prediction models. For example, adaptive negative sampling, where the sampling distribution is dynamically adjusted during training, has been shown to improve model performance.

E. *Explainable Link Prediction*: Explainable link prediction aims to provide interpretable explanations for link prediction results. Recent research has explored different methods such as attention mechanisms, decision trees, and rule-based models to provide explanations for link prediction results.

Overall, the field of link prediction on social networks is rapidly advancing, and many exciting developments are expected to improve the accuracy and interpretability of link prediction models.

V. EXPERIMENT AND RESULTS

If a graph can be represented as a structured dataset with features, then machine learning algorithms can be applied to discover the unconnected nodes in that Graph.
Consider the Graph given below:

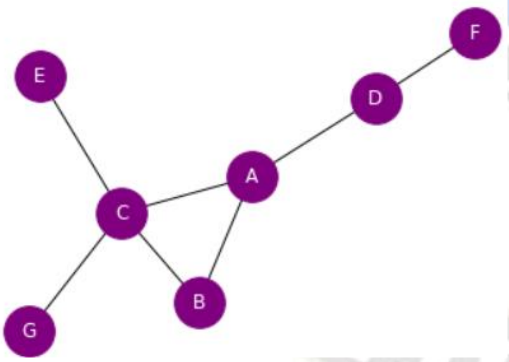


Figure 3. Graph at time t

The above Graph has some of the connected pairs like AB, AC, BC, CG, CE, AD, DF. Here there is a possibility of developing new connections between the unconnected nodes, as shown in the below Graph:

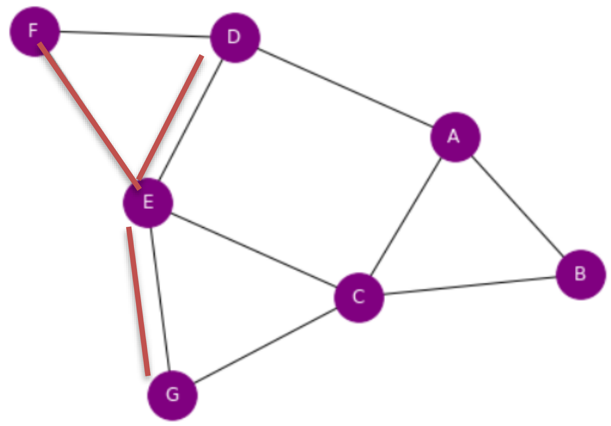


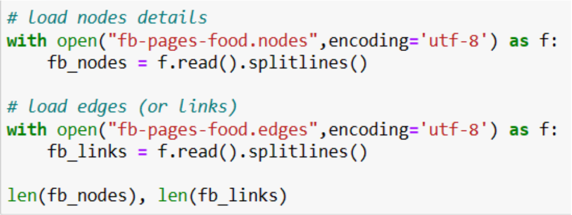
Figure 4. Graph with new possibilities

The red link is the edge that can perform new connections with time. The objective is to predict a new link between any two unconnected nodes. The loose pairs can be obtained from Fig. 2., which are: ED, EF, EG, and BG. For further graph processing, we need to extract features from it, but we still know the target variable. To find out the target variable, assign value '1' to the newly formed link, but before that, assign '0' to the nodes that still do not have any connection. So the data will look like below:

Feature	Link
E-D	1
E-F	1
E-G	1
B-G	0

TABLE I. ASSIGNING FEATURES

We can predict the link by comparing a graph at two different times. But in a real-world scenario, we will only have data from the present time. For our experiment, we have considered Facebook's dataset. The dataset consists of food joints and chefs across the Globe. The task is to predict the future links that can be common likes between unconnected nodes: Facebook pages. We have used python for this experiment, where the first task is to load the dataset.



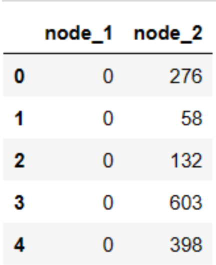
```
# load nodes details
with open("fb-pages-food.nodes",encoding='utf-8') as f:
    fb_nodes = f.read().splitlines()

# load edges (or links)
with open("fb-pages-food.edges",encoding='utf-8') as f:
    fb_links = f.read().splitlines()

len(fb_nodes), len(fb_links)
```

The output of the above code is

(621, 2102), where 621 is the nodes, and 2102 is the links present.

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
***Article Received:** 07 June 2023 **Revised:** 25 July 2023 **Accepted:** 16 August 2023*

After creating a data frame to see which node connects with which other nodes, we can easily create a graph, as shown.



	node_1	node_2
0	0	276
1	0	58
2	0	132
3	0	603
4	0	398

The data frame shows node 0 connects with nodes '276',' 58',' 132',’603' and '398'.
The Graph created for the dataset is:

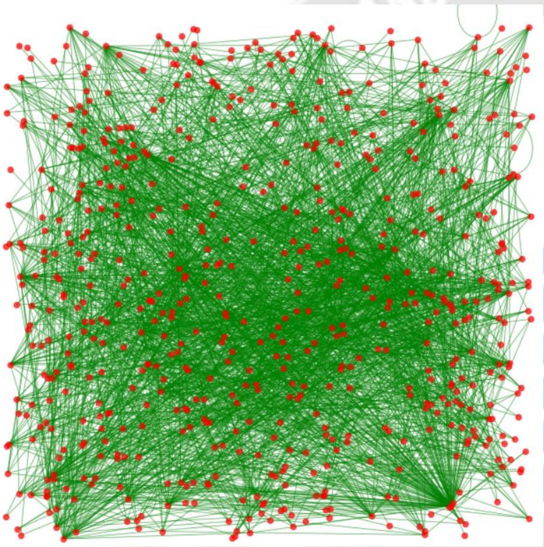


Figure 5. Arrangement of FB pages as Graph

“Figure 5.” shows the arrangement of Facebook pages as a graph where red dots show the nodes and green lines show the edges.

A dataset is prepared from the undirected graph, with a node pair feature and the binary target variable.

1. Retrieve unconnected pair for negative sample
2. Remove Links from connected pairs for positive samples.

The output obtained after this:

```
0    19018
1     1483
```

We have above 1400 edges which we can drop to create a positive training example.

We have applied three machine-learning algorithms:

1. logistic regression
2. K-Nearest Neighbour

The results obtained are listed below in terms of accuracy.

Classifier	Accuracy
Logistic Regression	0.809
KNN	0.904

TABLE II. RESULTS

VI. FEATURES OF A CYBER-CRIMINAL NODE

As we have seen in the previous part, where we discussed the graph-based methods of link prediction, the nodes which are passing information are either the complete network nodes or 2-hop nodes. Here are some of the features which can be seen in the nodes identified as cyber-criminal nodes. The features are listed below:

1. *High degree centrality*: A cybercriminal node may have a high number of connections to other nodes in the social network, indicating that they can communicate and interact with a large number of other users.

$$C_D(v)=deg(v)$$

Where v is the number of vertices.

2. *Low clustering coefficient*: A cybercriminal node may have a low clustering coefficient, indicating that they are less likely to be part of a tightly connected group of users. This may be because they are operating alone, or because they can manipulate others into participating in their activities without forming strong social bonds.

$$C(i) = 2T(i)/(k(i)(k(i)-1))$$

where T(i) is the number of triangles (i.e., sets of three nodes that are fully connected) that include node i, and k(i) is the degree of node i.

3. *High betweenness centrality*: A cybercriminal node may have a high betweenness centrality, indicating that they are a key node in the network for facilitating communication and interactions between other nodes. This may be because they can manipulate or influence others, or because they have access to valuable resources or information.

$$C_B(n_i)=\sum_{j<k} g_{jk}(n_i)/g_{jk}$$

Where g_{jk} is the number of the shortest path connecting j and k and $g_{jk}(n_i)$ is the number on which i is.

4. *Low reciprocity*: A cybercriminal node may have a low level of reciprocity, indicating that they are less likely to have mutual connections with other nodes in the network. This may be because they are operating under

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
***Article Received:** 07 June 2023 **Revised:** 25 July 2023 **Accepted:** 16 August 2023*

a false identity or because they are actively trying to avoid detection.

low reciprocity = 1 - (R/T)

where R is the number of reciprocated edges and T is the total number of edges in the graph.

5. *High activity level*: A cybercriminal node may be highly active on the social network, posting frequently and engaging with a large number of other users. This may be because they are attempting to spread malicious content or to manipulate others into participating in their activities.

The proposed algorithm is inspired by an existing study in [16] which had proposed an algorithm GraphSAGE, which works by iterative sampling and aggregating node neighbourhoods to generate node embeddings. T he algorithm consists of the following steps:

1. Initialization: For each node in the graph, assign it an initial embedding vector.
2. Sampling: For each node in the graph, sample a fixed-size set of its neighbouring nodes.
3. Aggregation: Aggregate the embeddings of the sampled neighbours using a neural network. This generates a new embedding for the original node.
4. Repeat steps 2 and 3 for a fixed number of iterations, updating the node embeddings each time.

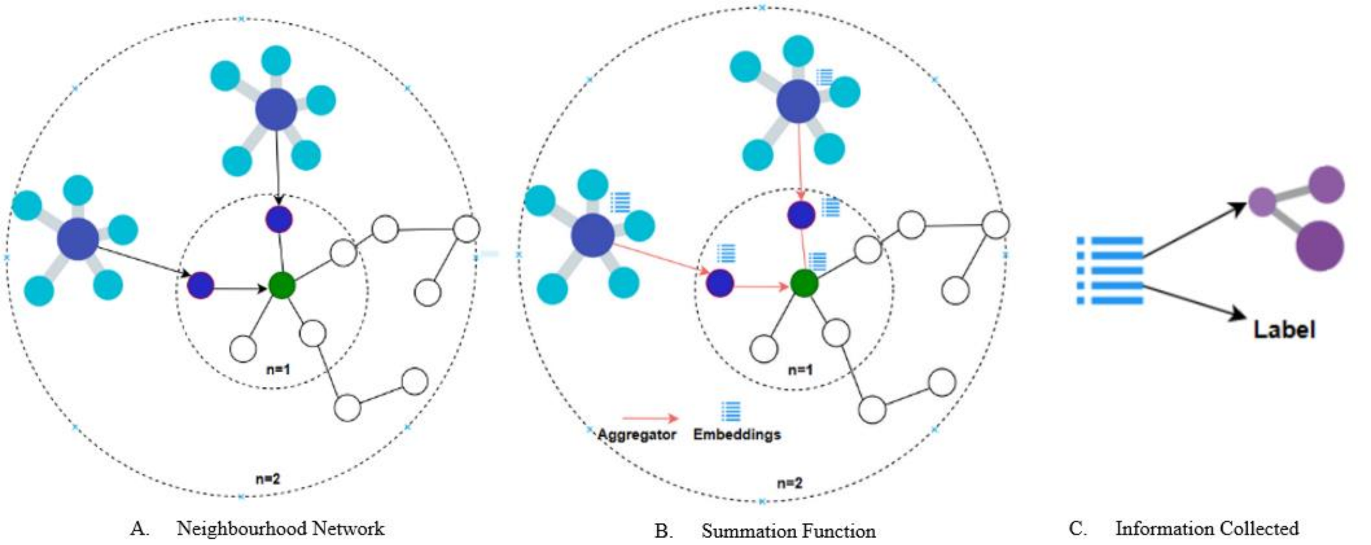


Figure 1. An Illustration of GraphSALR

5. Use the final node embeddings as features for downstream machine-learning tasks.

The aggregation step is performed by a neural network that takes as input the embeddings of the sampled neighbours and produces a new embedding for the original node. The neural network can be designed in various ways, such as a simple mean or max pooling function, or a more complex neural network architecture such as a multi-layer perceptron or a graph convolutional network.

GraphSAGE is designed to handle large graphs that do not fit into memory, by using a sampling strategy that selects a subset of the nodes and edges to be processed at each iteration. This makes it possible to generate node embeddings for very large graphs using limited computational resources.

GraphSAGE has been shown to achieve state-of-the-art performance on a variety of graph-based machine-learning tasks, including node classification, link prediction, and graph classification. It is widely used in industry and academia for

So, a criminal node can be defined by the given features. Considering these factors, we are proposing a new algorithm GraphSALR, which is Graph **S**ample **A**ggregator with **L**ow **R**eciprocity.

VII. PROPOSED METHOD: GRAPHSALR

The focus of our research is to find out a cybercriminal node in a social network. Cybercriminals may be involved in any the cybercriminal activities like cyber frauds, cyberbullying, identity theft and there are many more. With the advancement of technology, social media has become a powerful tool to commit crimes. Although social media can be used to commit crimes, it can also be used to detect and predict criminal activity.

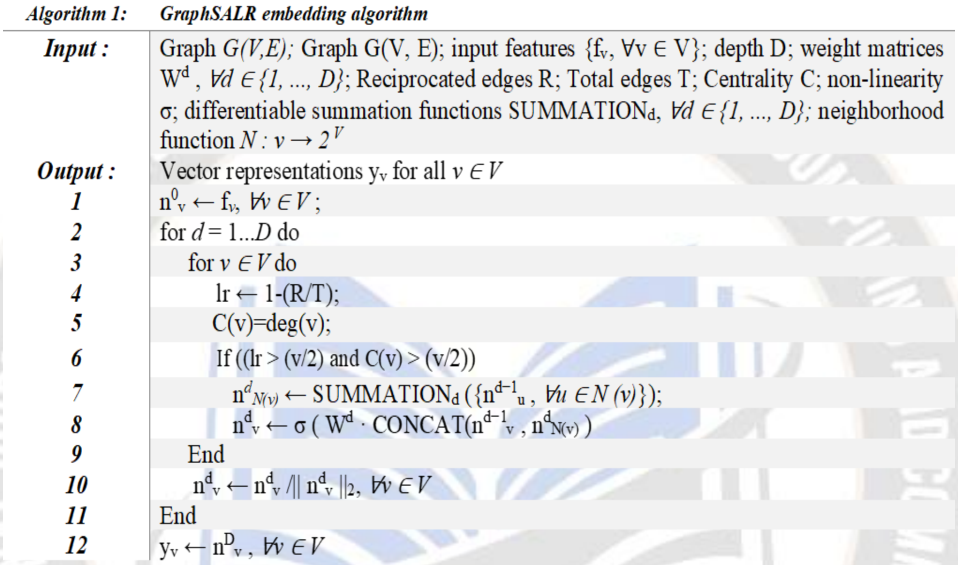
We are trying to uncover any links which are hidden or missing and that can identify a criminal. The problem becomes a problem of link prediction, which is being studied for a long. Many methods have been used for link prediction like GNN, GCN, and GAT including machine learning algorithms.

analyzing social networks, recommendation systems, and other types of large-scale graph data.

Most of the existing approaches to graph embedding use matrix-factorization-based methods which do not generalize the unseen data because they are predicting based on a node in a fixed graph.

There are several approaches to learning over graphs, including:

1. Graph Neural Networks (GNNs): GNNs are a type of deep learning model designed to operate on graphs. They can capture the structural information of the graph and can be used for tasks such as node classification, graph classification, and link prediction.
2. Message Passing: Message passing is a fundamental operation in GNNs. It involves passing information between neighbouring nodes in the graph. Each node updates its representation based on the representations of its neighbours and sends updated representations to its neighbours.
3. Graph Convolutional Networks (GCNs): GCNs are a type of GNN that use graph convolution operations to learn node representations. These operations involve aggregating information from neighbouring nodes and updating the node representation based on the aggregated information.



Algorithm 1: ***GraphSALR embedding algorithm***

```
Input :   Graph $G(V,E)$; Graph G(V, E); input features $\{f_v, \forall v \in V\}$; depth D; weight matrices
          $W^d$, $\forall d \in \{1, ..., D\}$; Reciprocated edges R; Total edges T; Centrality C; non-linearity
          $\sigma$; differentiable summation functions $SUMMATION_d$, $\forall d \in \{1, ..., D\}$; neighborhood
          function $N : v \rightarrow 2^V$
Output :  Vector representations $y_v$ for all $v \in V$
          $n^0_v \leftarrow f_v, \forall v \in V$;
          for $d = 1...D$ do
              for $v \in V$ do
                  $lr \leftarrow 1-(R/T)$;
                  $C(v)=deg(v)$;
                  If ((lr > (v/2) and C(v) > (v/2))
                      $n^d_{N(v)} \leftarrow SUMMATION_d(\{n^{d-1}_u, \forall u \in N(v)\})$;
                      $n^d_v \leftarrow \sigma(W^d \cdot CONCAT(n^{d-1}_v, n^d_{N(v)}))$
              End
              $n^d_v \leftarrow n^d_v / \| n^d_v \|_2, \forall v \in V$
          End
          $y_v \leftarrow n^D_v, \forall v \in V$
```

4. Graph Attention Networks (GATs): GATs are a type of GNN that use attention mechanisms to weigh the contributions of neighbouring nodes when updating a node representation. This allows the model to selectively focus on important nodes in the graph.
5. Graph Embeddings: Graph embeddings are low-dimensional vector representations of nodes or graphs that preserve some of the structural information of the original graph. These embeddings can be used as inputs to machine learning models for downstream tasks.

"Figure 6." shows an illustration of the model proposed called GraphSALR, which has three different graphs: A. Neighbourhood Network, B. Summation Function and C. Information Collected. The first image A describes a general structure of a network, where the green node is the central node which is collecting information from its neighbourhood. But, the information is collected from the neighbours who have high centrality degree and low reciprocity because such nodes can be identified as the criminal node. The second image B suggests that the information which is collected from such nodes is aggregated and passed to the central node. And lastly, image C gives an overview of what information is collected from this method.

As a substitute for training a distinctive embedding vector for every node, we train a set of summation functions which learn to combine feature material from a node's local neighbourhood. Every aggregator function combines data from a distinct number of hops or explores the depth, far from a given node. At examination or implication time, we utilize our trained system to create embeddings for completely hidden nodes by utilizing the learned summation functions.

A. Generating Node Embedding

Here, we are going to describe the node embedding generation. The input is Graph *G(V,E),* where *V* is the vertices and *E* is the edges. The features of all the nodes are defined by $f_v, \forall v \in V$, the point to be noted here is that the features of the nodes include the feature which is described for a node to be a criminal node. We are considering mainly the feature of low reciprocity which suggest that they are less likely to have mutual connections with other nodes in the network. The reciprocity will be calculated as:

$$\textbf{\textit{low reciprocity = 1 - (R/T)}}$$

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
Article Received: *07 June 2023* ***Revised:*** *25 July 2023* ***Accepted:*** *16 August 2023*

where R is the number of reciprocated edges and T is the total number of edges in the graph. An iteration of the algorithm is described. For every outer loop in Algorithm 1, the procedure is as follows: *d* signifies the existing step in the outer iteration (which is also the depth) and n^k is a node's interpretation at this step. In the first iteration, every node $v \in V$ sums the interpretation of the node which is in its neighbourhood, $(\{n^{d-1}_u, \forall u \in N(v)\})$, in a single vector $n^d_{N(v)}$. It is to be noted that this summation depends on the "if" condition and for loop. The "if" condition "If ((lr > (v/2) and C(v) > (v/2))" here specifies that if the reciprocity is low and centrality is greater then only take the aggregation of the nodes because it signifies that the node can be criminal in the network. After summation of the neighbouring feature vectors, GraphSALR then combines the node's present representation, n^{d-1}_v, with the combined neighbourhood vector, $n^{d-1}_{N(v)}$, and this concatenated vector representation is given a fully associated layer with nonlinear activation function σ, that transforms the representations which is to be used at the following stage of the algorithm (i.e., n^d_v, $\forall v \in V$).

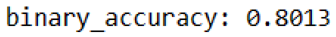
B. *Learning Parameters*

We have applied a graph-based loss function which generates useful and analytical representation in an unsupervised environment. These representations were generated from the node's local neighbourhood features, and the loss function calculated promotes similar representation for the nearer nodes keeping the distant nodes distinct.

VIII.EXPERIMENT

The experiment is conducted on a citation network which consists of a total of 5429 links. Each paper published in the dataset is described by a 0/1 value word vector which denotes the absence or presence of any word in the dictionary, which consists of 1433 unique words. The experiment is conducted using Python. The number of walks used for the experiment is three whereas the length taken is five. The experiment has a batch size of forty.

The experiment for link prediction outperforms and has an accuracy of 0.80 on the citation network.



```
binary_accuracy: 0.8013
```

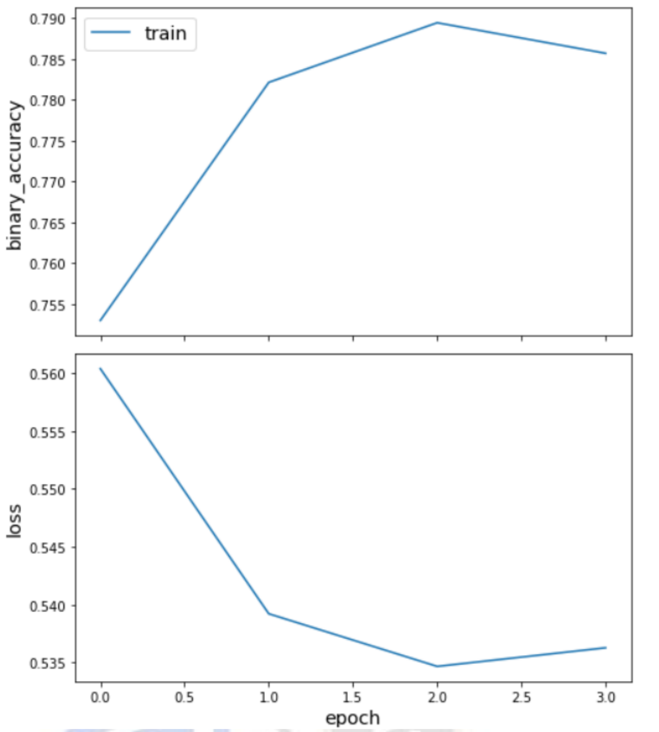


Figure 7. Visualized Graph

The node embedding for GraphSALR is also visualized in the graph in Figure 8. We also create a link generator for selection and running train and test link patterns to the model. The link generators fundamentally "plot" pairs of nodes (citing-paper, cited-paper) to the contribution of GraphSALR: they take mini-batches of node pairs, sample 3-hop subgraphs with (citing-paper, cited-paper) main nodes mined from those pairs, with the consistent binary labels representative whether those pairs signify true or false citation links.

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7697
***Article Received:** 07 June 2023 **Revised:** 25 July 2023 **Accepted:** 16 August 2023*

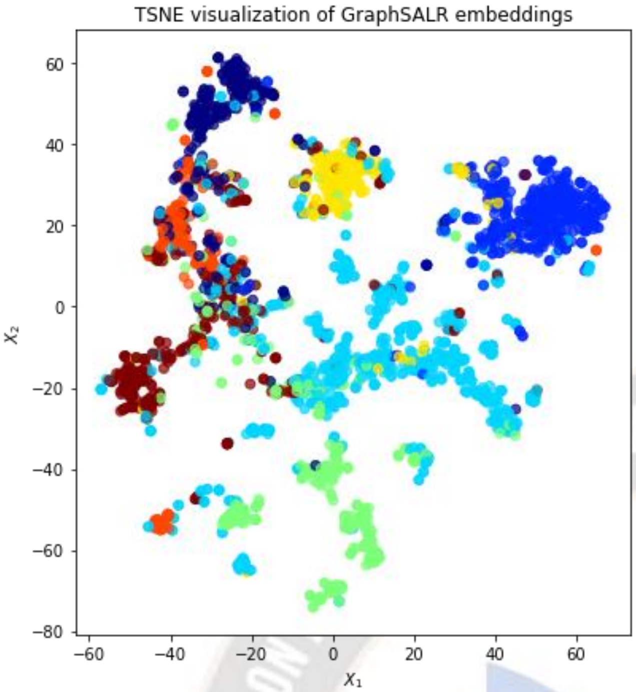


Figure 8. NE Visualization

The prediction is also plotted below:

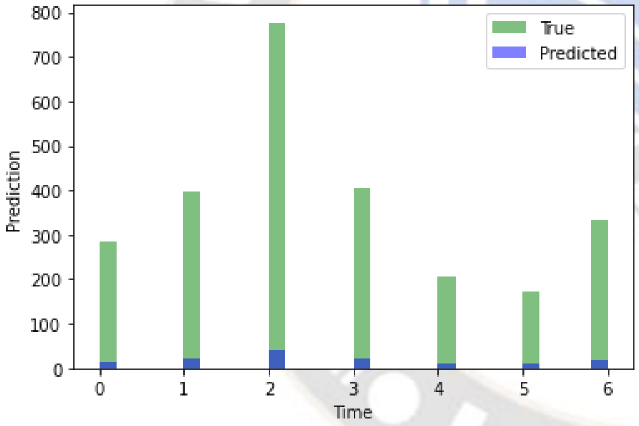


Figure 9. Prediction with time

IX. CONCLUSION AND FUTURE WORK

The identification of missing links and prediction of future links has gained the attention of scholars and researchers in the past few years. In literature, different methods and models have been proposed for it, amongst which many are based on common neighbour techniques. There are many other techniques which are used along with common neighbours such as Jaccard Coefficient, Preferential attachment, Adamic Adar, Kartz measure and many others. Link prediction problems now use machine learning methods. We have applied the link prediction method to a Facebook dataset. Two machine learning classifiers were then applied: LR and KNN where KNN shows better results than LR. We have also proposed an algorithm GraphSALR for link prediction. For future work, we can look for more elaborate experiments and other techniques to increase the accuracy of link prediction.

REFERENCES

[1] Tang, R., Jiang, S., Chen, X., Wang, H., Wang, W., & Wang, W. (2020). Interlayer link prediction in multiplex social networks: an iterative degree penalty algorithm. Knowledge-Based Systems, 194, 105598.

[2] Yao, L., Wang, L., Pan, L., & Yao, K. (2016). Link prediction based on common neighbours for dynamic social network. Procedia Computer Science, 83, 82-89.

[3] Ficara, A., Cavallaro, L., Curreri, F., Fiumara, G., De Meo, P., Bagdasar, O., ... & Liotta, A. (2021). Criminal networks analysis in missing data scenarios through graph distances. PLoS one, 16(8), e0255067.

[4] Lim, M., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Hidden link prediction in criminal networks using the deep reinforcement learning technique. Computers, 8(1), 8.

[5] Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., & Piccardi, C. (2016). Link prediction in criminal networks: A tool for criminal intelligence analysis. PloS one, 11(4), e0154244.

[6] Wang, P., Xu, B., Wu, Y., Zhou, X.. Link prediction in social networks: the state-of-the-art. Science China Information Sciences 2015; 58(1):1–38.

[7] Budur, E.; Lee, S.; Kong, V.S. Structural Analysis of Criminal Network and Predicting Hidden Links using. Mach. Learn. 2015.

[8] K. Shu, S. Wang, J. Tang, R. Zafarani, H. Liu, User identity linkage across online social networks: A review, ACM SIGKDD Explor. Newsl. 18 (2) (2017) 5–17.

[9] Mittal, P. ., & Navita. (2023). Early-Stage Detection of Covid-19 Patient using ML Model: A Case Study. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 84–89. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2480

[10] R. Feng, Y. Yang, W. Hu, F. Wu, Y. Zhang, Representation learning for scalefree networks, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[11] D. Zhao, N. Zheng, M. Xu, X. Yang, J. Xu, An improved user identification method across social networks via tagging behaviors, in: 30th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2018, pp. 616–622

[12] Shi, C., Li, Y., Zhang, J., Sun, Y., & Philip, S. Y. (2016). A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37.

[13] Man, T., Shen, H., Liu, S., Jin, X., & Cheng, X. (2016, July). Predict anchor links across social networks via an embedding approach. In Ijcai (Vol. 16, pp. 1823-1829).

[14] Zhou, X., Liang, X., Zhang, H., & Ma, Y. (2015). Cross-platform identification of anonymous identical users in multiple social media networks. IEEE transactions on knowledge and data engineering, 28(2), 411-424.

[15] Mr. Rahul Sharma. (2013). Modified Golomb-Rice Algorithm for Color Image Compression. International Journal of New Practices in Management and Engineering, 2(01), 17 - 21.

Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/13

[16] Kong, C., Gao, M., Xu, C., Qian, W., & Zhou, A. (2016, April). Entity matching across multiple heterogeneous data sources. In International conference on database systems for advanced applications (pp. 133-146). Springer, Cham.

[17] Kumari, A., Behera, R. K., Sahoo, K. S., Nayyar, A., Kumar Luhach, A., & Prakash Sahoo, S. (2020). Supervised link prediction using structured-based feature extraction in social network. Concurrency and Computation: Practice and Experience. doi:10.1002/cpe.5839

[18] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems, 30.