Utrecht University

The Impact of Role Mining on Link Prediction in Financial Graph Networks

by

Pablo Anthony van Beek

Master thesis, Applied Data Science

Utrecht, Netherlands

June, 2024

Supervisors - Dr. I.R. Ioana Karnstedt-Hulpus & Prof. Dr. Yannis Velegrakis

Keywords: link prediction, role mining, financial graph network, cryptocurrencies

github.com/PablovanBeek/Master-Thesis/

Abstract

This thesis investigates the effects of role mining on learning-based link prediction models in financial graph networks. Link prediction is utilised in the financial domain for predicting transactions and finding anomalies. This benefits the effective recognition of fraud and the detection of potential financial opportunities. Role mining has yielded positive results in other domains for improving link prediction models. However, this research finds that role mining does not significantly improve link prediction in financial graph networks. Role mining likewise does not seem complementary to other graph-based features, such as node centrality features and node similarity features. The features were provided in various combinations to a random forest algorithm. Whereupon the performance was evaluated based on the recall score over various thresholds. The findings in this paper contribute to a better understanding of role mining in financial networks and its effects on link prediction. It opens the way for further research into the effective use of appropriate feature combinations for link prediction in financial networks, in which role mining could potentially be instrumental.

Table of Contents

1. Introduction....4
    1.2 Thesis Outline....5
2. Background....6
    2.1 Cryptocurrency Transaction Networks....6
    2.2 Link Prediction....7
        2.2.1 Applications in Social Networks....7
        2.2.3 Applications in Financial Transaction Networks....8
        2.2.4 Techniques....9
        2.2.5 Node Similarity Measures....10
        2.2.6 Node centrality measures....10
    2.3 Graph-Based Roles....12
        2.3.1 Community-Based Detection....12
        2.3.2 Feature-Based Detection....12
    2.5 Sampling....13
        2.5.1 Negative Sampling....13
3. Methods....15
    3.1 Data....15
    3.2 Sliding Window....15
    3.3 Features and Roles....17
        3.3.1 Node Centrality Features....17
        3.3.2 Node Pair Similarity Features....17
        3.3.3 Role Mining....17
    3.4 One-Class Classification....18
        3.4.1 Classification Model....18
        3.4.2 Sampling Method....18
        3.4.3 Performance Measures....18
4. Results....20
5. Discussion & Future Work....23
6. Conclusion....24
7. References....25

1. Introduction

Many biological, social, financial, and other information systems are effectively represented in graph networks. Representing these systems as graph networks allows for the application of statistical methods to interpret the interaction and relationships between entities, thereby facilitating the use of mathematical methods for knowledge mining. Graph networks consist of nodes and edges, where nodes represent real-world entities, and edges reflect the interaction that takes place between these entities. Link prediction is a problem that capitalises on this interaction property. This method attempts to calculate the likelihood of the existence of a link (interaction) between two nodes (Lin et al., 2022). This technique has various practical applications that differ based on the domain.

In social networks, it can reflect how likely people are to 'connect' with each other - becoming friends on Facebook or connecting on LinkedIn. Effectively estimating this probability can e.g. improve the recommendations in network suggestions (Liben-Nowell & Kleinberg, 2003). In the financial domain, it does not indicate the probability of a relationship between individuals but rather the probability that bank accounts have capital flows between them. This capability extends beyond merely predicting transaction likelihoods; it can highlight potential credit risks and detect fraudulent transactions. In order to detect fraudulent transactions, link prediction can be used to find anomalies in a transactional network. These anomalies refer to links that should not exist in a transaction network possibly reflecting, e.g. fraudulent activity. Link prediction can likewise be deployed to find logical - high likelihood - transactions that have yet to occur in the real world. This application is beneficial for, e.g., finding opportunities, loans and partnerships (Zambre & Shah, 2013).

Link prediction can be categorised into multiple underlying techniques, one of which includes learning-based frameworks (Kumar et al., 2020). Their effectiveness depends heavily on the feature set, which often consists of node- and similarity features (Liben-Nowell & Kleinberg, 2003). Consequently, this paper aims to answer the question: What is the impact of role mining on the performance of learning-based link prediction frameworks in financial graph networks?

Literature has shown that role mining can positively impact the results of multiple types of analyses (Rossi & Ahmed, 2015). This opens the door to exploring the impact of role mining in large financial graph networks.

This research determines that role mining has no significant positive impact on the performance of learning-based link prediction models in the proposed configuration. More research is needed with diverse configurations to ensure that role mining is not effective in financial networks.

1.2 Thesis Outline

Following the introduction, section 2 discusses background information regarding link prediction, financial graph networks, role mining, and sampling methods. Section 3 then explains the methodology of the thesis. Section 4 will provide the results. This will be followed by section 5, which discusses the results and future work. Section 6 concludes the main findings of the thesis.

2. Background

This section presents background information related to cryptocurrency transaction datasets. Then, the paper discusses the concepts of link prediction in social- and financial networks. Finally, role mining, and sampling methods for link prediction are discussed.

2.1 Cryptocurrency Transaction Networks

In the past decade, cryptocurrencies have become increasingly popular. Cryptocurrencies fulfil the ability to execute a transaction without an intervening controlling authority. Some cryptocurrencies, including Bitcoin, use a so-called blockchain to make this possible.

The blockchain is the domain where all transactions are registered. It is the blockchain that allows Bitcoin (and other cryptocurrencies) to function as the antagonist of the regular banking system by functioning in a decentralised manner with no overseeing authority. To ensure that Bitcoin functions in a decentralised manner while still being secure, cryptocurrency miners validate these transactions to ensure authenticity. They receive a small fee for the validation service (Wu et al., 2021).

The transparency of cryptocurrencies and their blockchains provides opportunities for transaction research. Previously, it could be problematic to obtain transaction data, but some cryptocurrencies overcome this problem as the transaction information is public. Cryptocurrency addresses can be perceived as bank accounts, and cryptocurrency trades between accounts can be perceived as transactions. Meta information, such as timestamps and volumes, is likewise often available (Gupta & Sadoghi, 2019).

Most cryptocurrencies' blockchains, being public information, offer the possibility of analysing transactions using a complex graph network (Wu et al., 2021). This has created opportunities for multiple types of research, e.g., network evolution, price prediction and fraud detection (Greaves & Au, 2015; Liang et al., 2018; Zambre & Shah, 2013).

To analyse these transactions, they are often placed in a graph network. In these networks, the cryptocurrency accounts are considered as the nodes, and the transactions are considered as the edges. A graph can be defined as G = (V, E), where V represents all nodes and E represents all edges in the graph (Wu et al., 2020). Utilising this type of network makes it possible to examine the characteristics of a node, which can be used to calculate node features. The node features matrix can be defined as $Xv \in Rd$, where Xv is the feature vector of node v and Rd is the number of dimensions. Subsequently, the neighbourhood of the nodes can be represented as $N(v) = \{u \in V \mid (v, u) \in E\}$, where $N(v)$ represents all nodes that are connected by edge(v, u). Transaction networks can often be modelled as directed graph networks. This implies that for

$\forall e \in E$, there is an orientation within the transactions in G; thus, in a directed network the edge (V_1, V_2) does not imply the existence of the edge (V_2, V_1) (Wu et al., 2020).

2.2 Link Prediction

2.2.1 Applications in Social Networks

In social graph networks, link prediction is indispensable in uncovering missing connections and/or future connections. The relevance of link prediction in social networks gained momentum in recent years with the rise of the internet and social media, which extended individuals' social lives into an online world. Link prediction has proven to be valuable in this online social context as it can suggest new contacts, resulting in expanding an individual's social network (Wang et al., 2014).

In these networks, individuals are represented as nodes, and edges represent the relationships between them. The most challenging aspects are the dynamic nature of social connections and the incomplete information on social media platforms (Wang et al., 2014).

A considerable amount of research has been dedicated to the link prediction problem in social graph networks. Consequently, numerous methodologies have been developed to address the challenge of improving the accuracy of predicting future or missing links in these networks. Each methodology has an underlying rationale based on graph-derived characteristics, which often include a form of community detection, graph structure analysis, or a hybrid application of these (Liben-Nowell & Kleinberg, 2003; Abnar et al., 2015).

Community detection

As community detection is often part of the rationale behind link prediction methodologies. Identifying underlying communities, roles, or clusters within a network can uncover relationships that positively impact the performance of link prediction algorithms (Pulipati et al., 2021). As nodes in the same community often share the same characteristics and have a relatively high interconnectedness, their probability of having a link increases (Liben-Nowell & Kleinberg, 2003).

Community detection methods are used to improve link prediction models by integrating community-derived characteristics. (Biswas & Biswas, 2017; Liben-Nowell & Kleinberg, 2003). However, each method makes strong assumptions about when nodes should have a link. If these assumptions fail, the methods are limited in their effectiveness. Consequently, studies are conducted on heuristics derived from the network rather than predetermined heuristics. For

instance, Zhang and Chen (2018) use local subgraphs around each target link, allowing the training of a function mapping that recognises patterns that are suited to the network. They utilise Graph Neural Networks (GNNs) to study the heuristics from the local subgraphs.

Graph structure

The structure of a social graph network also plays a significant role in the link prediction problem. Traditional solutions mainly rely on methods based on neighbouring nodes' characteristics. However, topological information, which refers to the structural properties of the network, likewise retains valuable information for link prediction (Muniz et al., 2018). Methodologies similar to Structure Enhanced Graph neural network (SEG) reflect that the structure of a graph contains valuable information with respect to link prediction. SEG uses a path labelling method to capture the surrounding topological information of target nodes. After this, this information is eventually utilised in a GNN model, yielding state-of-the-art results (Ai et al., 2022).

Both community detection and structural features from a social graph network graph hold valuable information for link prediction in social graph networks. Both methodologies capture underlying information that benefits the performance of link prediction algorithms.

2.2.3 Applications in Financial Transaction Networks

Transaction graph networks have analogous semantics to those of social networks, but there are topological distinctions within these networks. Financial graph networks do not directly mirror social graph networks, as funds often flow beyond the extent of social circles to e.g. larger organisations. These organisations are often not part of an individual's social network. Additionally, transaction graphs often have a temporal element, representing the time and sequence of a transaction (Xiang et al., 2022). This is frequently absent in social networks, as the emphasis is more on the dynamic nature of connections (Wang et al., 2014), rather than on the temporal element; once a transaction has taken place, it cannot be removed.

Link prediction is a fundamental task within financial graph analysis. It estimates the probability of a transaction occurring between two nodes (Lin et al., 2022), denoted as $P((u, v) \in E$, where P is the probability of a transaction occurring between two nodes.

In the context of cryptocurrency networks, link prediction has several applications; e.g., it is used for predicting future transactions and recognising fraud (Zambre & Shah, 2013). By leveraging link prediction techniques, researchers can gain insights into the dynamics of a transaction network.

According to Yang et al. (2014), link prediction can solve two different problems in transaction graphs:

1. **Predict future transactions;** link prediction can forecast future transactions. The subgraph $G[t, t']$ is used to divide the transactions in the training and testing split, where G consists of all of the edges (transactions) that occurred between t and t'.
   For the training, the model is provided $G[t_0, t'_0]$, which expresses all of the data within the training interval. Following this, the model attempts to predict which edges E are present in the test split $G[t_1, t'_1]$ but not in the training split $G[t_0, t'_0]$ (Liben-Nowell & Kleinberg, 2003).
2. **Discover missing links in a network;** Link prediction is able to uncover missing links within a given graph $G[t, t']$. G consists of all edges (transactions) that are documented between t and t'. In this circumstance, the model is used to uncover the missing links (undocumented edges) in the same graph. This implies that $G[t, t']$ is both the train and test graph (Liben-Nowell & Kleinberg, 2003).

2.2.4 Techniques

There are various techniques to calculate the probability of links occurring. The techniques use different methods of calculating the closeness or similarity between a set of nodes (Liben-Nowell & Kleinberg, 2003). Kumar et al. (2020) Primarily distinguish between four underlying link prediction techniques:

1. **Similarity-based methods** are considered the simplest form of link prediction. Similarity-based methods use a similarity score $S(x, y)$, that is calculated for each set of nodes in G. This score is estimated based on the characteristics of the nodes. The nodes are assigned scores for their similarity.
2. **Probabilistic and maximum likelihood models** use an objective function with multiple parameters to calculate the probability of a non-existing link (x, y). This is evaluated with the conditional probability $P(Axy = 1|\theta)$. These models are often computationally expensive and complex, making them unsuited to apply to large complex networks.
3. **Link prediction using dimensionality reduction** uses the fundamental properties of dimensionality reduction to group similar nodes. Dimensionality reduction allows nodes to be represented in the embedding space rather than the original network. In this embedding space, nodes that are located in each other's proximity are assigned a high probability of having a link.

4. **Learning-based frameworks for link prediction**: Unlike previous methods, which use similarity or probabilistic methods to calculate the probability of a link, these methods use graph features. These models are typically trained on features extracted from the graph, nodes, and/or edges. These models are trained as classification or anomaly detection tasks. In their paper, Liben-Nowell and Kleinberg (2003) demonstrate that node similarity features can positively influence link classification. One of the biggest challenges is selecting the correct feature set. This leaves room for improvement as the feature selection significantly impacts the performance of the learning-based frameworks.

2.2.5 Node Similarity Measures

These features examine the similarities between two different nodes, which are then expressed as a value. A higher similarity between two nodes indicates a higher link probability between them and can effectively be used in link prediction (Liben-Nowell & Kleinberg, 2003).

- **The Jaccard Coefficient** compares the intersection of the features of (x, y) to the union of the features of (x, y). In the context of a network, (A) and (B) represent the neighbours of the nodes (x, y); this can be formally denoted as: $Jaccard(A, B) = \frac{|A \cup B|}{|A \cap B|}$. This score represents the ratio of shared neighbours compared to the total unique neighbours of both nodes.
- **Dice-Sørensen coefficient** is similar to Jacob's similarity, but it is more robust against outliers. It measures the ratio of shared neighbours to the average total number of neighbours. The shared neighbours are denoted as: $|\Gamma(x) \cap \Gamma(y)|$, whereas the number of neighbours (degrees) for the nodes (x, y) is denoted as: $k_x + k_y$. This results in the formula $S(x, y) = \frac{2|\Gamma(x) \cap \Gamma(y)|}{kx + ky}$ Kumar et al. (2020).

2.2.6 Node centrality measures

Node centrality measures are essential in graph analysis. They express the centrality of nodes in numerical terms. As Kumar et al. (2022) demonstrate, these measures can also be used for link prediction. Centrality measures can be used to quantify how similar nodes are and thus can be of value in the link prediction problem.

- **In-degree & Out-degree** are the most straightforward centrality measures. They indicate how many in- and outgoing connections a node has. In an undirected network, it is impossible to distinguish between outgoing and incoming edges. In directed networks, it

is possible to distinguish between incoming edges (in-degree) and outgoing (out-degree). The indegree and outdegree can be defined as:

$Degree_{in}(x) = |\{ y \in V : (y, x) \in E\}|$ - This measure indicates how many edges are pointing towards the node x.

$Degree_{out}(x) = |\{ y \in V : (x, y) \in E\}|$ - This measure indicates how many edges point out of the node x.

- **Eigenvector centrality** builds upon the degree centrality; it also considers the importance of adjacent nodes. The Eigenvector centrality can be defined as:

  $Eigenvector\ Centrality_x = \frac{1}{\lambda} \sum_{y \in V} A_{x,y} C_y$. In this formula, λ is a constant value, $A_{x,y}$ indicates if a link between nodes (x, y) is present and C_y is the eigenvector value of the node y. The underlying concept of this formula is that higher-scoring nodes are more central in the network and, thus have a higher eigenvector value (Bonacich, P., 1987).
- **PageRank centrality** is a measure developed initially by Google (Page, Brin, Motwani, & Winograd, 1998). The algorithm originally aimed to rank webpages based on their incoming and outgoing hyperlinks. This concept can also be applied to graph networks, where edges replace hyperlinks, and the websites are considered nodes. The PageRank formula can be defined as:

  $PageRank(x) = (1 - d) + d(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \ldots + \frac{PR(T_N)}{C(T_N)})$. In this formula, $PageRank(x)$ is the PageRank value of node x. $C(T_n)$ refers to the amount of outbound edges from the node x. PR(T_n) refers to the PageRank score for the nodes with outgoing nodes to the node x. d Is a damping factor that is used as an indicator for the algorithm, how quickly it should move to a random node rather than continuing following edges.
- **Average Shortest Path Length (ASPL)** is a measure that specifies the average number of steps (edges) needed to reach all other nodes from a specific node. ASPL can be defined as:

  ASPL $(v_i) = \frac{1}{n*(n-1)} * \sum_{i \neq j} d(v_i, v_j)$**.** In this formula, v_i is a node, where $d(v_i v_j)$ represents the shortest distance between v_i and v_j**.**

2.3 Graph-Based Roles

Role mining is a technique for analysing the roles or functions of nodes within a graph. Node roles can help indicate if nodes are associated with each other due to their similar roles or functions within a graph.

2.3.1 Community-Based Detection

Roles can be uncovered by analysing their community. Graph-based role-mining techniques identify these communities by examining the relationships and patterns that can be inferred from their links. In essence, these algorithms locate nodes that are very well connected and thus form a community. (Abnar et al., 2015; Rossi & Ahmed, 2015).

One example of an algorithm based on this community detection technique is SocioRank (Rafique et al., 2019). SocioRank combines community detection with role mining. It does this by using the degree, the betweenness centrality, and the closeness centrality. It iteratively removes links with the highest centrality values. The idea of removing these high-centrality value links is that once the links that interconnect these communities are removed, it will become easier to distinguish the different communities.

2.3.2 Feature-Based Detection

Another role-mining technique is to examine the network's structure. The distance between two nodes may be relatively large, indicating that they are not (in)directly linked. However, if the structural similarity between the two nodes is high, a role-mining algorithm that examines the network structure would likely assign the same role to the two nodes. A graph's structure can be represented by structural features. (Henderson et al., 2012).

Henderson et al. (2012) introduced RolX, an unsupervised algorithm for discovering roles through structural analysis. RolX needs no prior knowledge of the network; it can determine the appropriate number of roles itself.

The algorithm assigns each node a probability for each role that indicates how well it fits into that role. The algorithm calculates these probabilities by describing each node in the network as a feature vector. RoLX can determine the importance of certain features and adjust its feature set accordingly. These features can include, for example, the number of triangles a node participates in and the number of neighbours. RoLX is scalable for large networks as it has a linear time complexity relative to the number of nodes.

RoleSim is a different algorithm that calculates how structurally similar nodes are. It works by iteratively comparing nodes' neighbours; if nodes' neighbours are structurally similar, RoleSim

assumes that the nodes themselves are similar. This allows RoleSim to identify nodes that have a similar structural role within a network (Jin et al., 2011).
RoLX and RoleSim are both valid methods for calculating nodes' roles within a complex network. By leveraging these methods, researchers can gain insight into a role's function based on its structural similarities within a graph network.

2.5 Sampling

Sampling is an essential technique for analysing a graph network. It encompasses the mechanism for selecting a subset of edges and nodes from graph G while aiming to maintain the structural properties of G. In this context, the graph is defined as $G = (V, E)$ and the subgraph is as defined as $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$.
This is valuable because graph networks such as biological, social, and financial systems can be vast. The size of graphs has a negative impact on the computational time of graph analyses; in certain scenarios, the size of a graph makes it computationally infeasible to perform certain analyses. Finding sub-graphs that replicate the characteristics of the original graph ensures that the insights found in the sub-graph can effectively be translated into insights for the original graph (Leskovec & Faloutsos, 2006).

2.5.1 Negative Sampling

Graphs represent data through nodes and the relationships (E) between them. Therefore, in the context of link prediction, a graph only represents positive relationships. Relations that do not exist are not represented in a graph. The relations that do not exist in G can be expressed as $E' = \{(u, v) \in V \times V \mid (u, v) \notin E\}$, where (u, v) is a pair of nodes and $V \times V$ represent all possible combinations of nodes.
It is possible to train one-class classification models with only a positive class. However, classifiers where negative relations have been inserted in the training phase offer increased performance (Yousef et al., 2010). There are numerous techniques for creating these negative relationships. Below is a concise overview of some of these techniques.

- **Random Node Selection** is a less complex graph sampling technique. This technique randomly selects nodes (X, Y), which are then linked in the graph. This technique does not take the characteristics of the graph into account. This results in the technique not

retaining the power-law degree distribution of the original graph, possibly affecting the structure and, thus, characteristics of the graph (Leskovec & Faloutsos, 2006).

- **Random Node Neighbor** selects a node at random. After this, all outgoing neighbours of this node are stored. Then, new edges are sampled between nodes that have no direct connection. This provides an effective representation of the out-degree of the original graph. However, this method does not consider the in-degree of nodes, nor does it consider the community structure of the original graph (Leskovec & Faloutsos, 2006).
- **Random Walk** selects a node at random. After this, the algorithm performs a random walk over the graph. With each step, the algorithm increases the possibility of returning to the original node. The algorithm might get stuck in a node without outgoing edges or get stuck in an isolated component. The algorithm overcomes this issue by selecting a different starting node when it has not reached the desired sample size within a certain amount of steps. During the random walk, the nodes visited are stored; when the random walk is finished, random pairs of nodes are generated from the stored nodes. This algorithm considers the in-degree and out-degree of nodes, as well as the network structure. However, it might be biased towards high-degree nodes during the random walks, and it can be computationally expensive (Leskovec & Faloutsos, 2006).

3. Methods

This section explains the methodology behind this study, which examines whether structural role mining can positively affect performance when added to the feature set of a learning-based framework link prediction model. The research relies on a combination of node centrality features, node pair features, and role probabilities. Multiple random forest models with different combinations of feature sets are trained to investigate the effectiveness of role mining.

3.1 Data

To test the methodology, the Temporal Graph Benchmark (TGB) coin dataset is used (Huang et al., 2023). TGB is a collection of diverse benchmark datasets and evaluation tools. Their aim is to improve the availability of temporal datasets and to streamline the evaluation of machine learning tasks on these temporal datasets. By doing so, they hope to facilitate the development of new methods and improve the understanding of temporal networks.
The Temporal Graph Benchmark (TGB) coin dataset is a cryptocurrency transaction dataset based on the Stablecoin ER20 transaction dataset. It is a medium-size dataset (Huang et al., 2023) used for link prediction.
The dataset, which was collected from 1 April 2022 to 1 November 2022, has 638,486 nodes and 22,809,486 edges (Shamsi et al., 2022).
The following transaction (meta)data is used during the study:

- **Sender Adress:** The unique wallet address of the sender.
- **Receiver Adress:** The unique wallet address of the receiver.
- **Timestamp:** The time at which a transaction occurred.

It is important to note that the temporal aspect was not included in the training for the models. It was only used to gain insights while analysing the performance of the model.

3.2 Sliding Window

A sliding window approach is used to accurately capture the financial data's temporal aspect. This is a systematic sampling approach. Using this approach, it is possible to capture the developments of patterns and temporal aspects over time. Figure 1 gives a visual representation of this sampling method. The shift size refers to the time length between the start points of two subsequent windows. These windows can overlap, but they cannot have the same starting point.

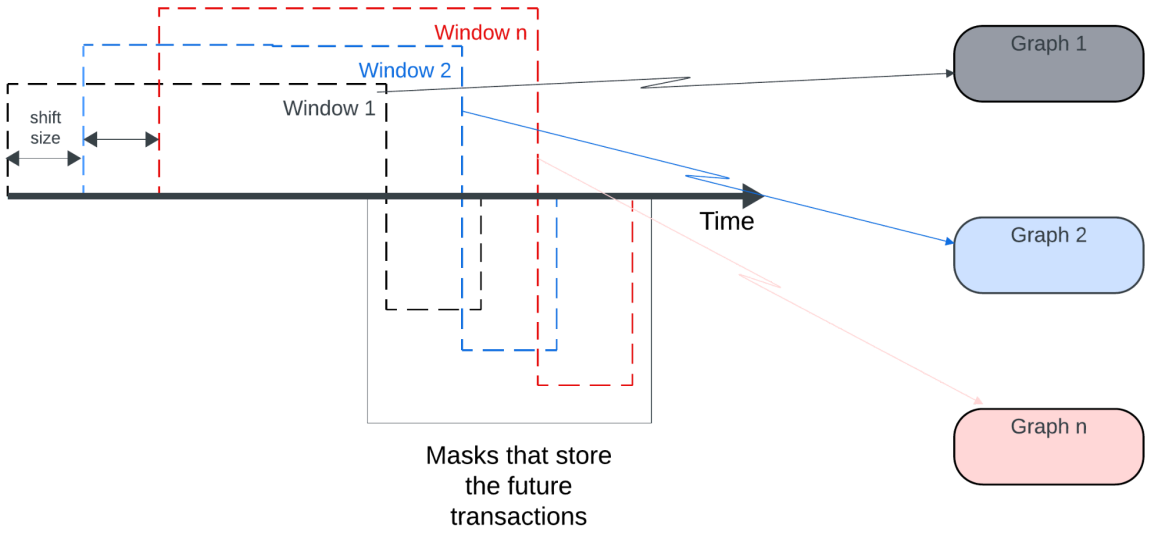

Figure 1: Sliding window visualisation

In this thesis, each new training window starts exactly one day after the previous training window. The test window starts after the last train mask to ensure that there is no data leak during the sampling. Three training windows and one test window are used, each three days long. Table 1 shows the total number of edges and nodes per window.

A mask is stored for each window; these masks store the transactions that occur within a six-day time period following the respective window. The masks are then compared with their respective window. If a node from the mask does not appear in the window, it is removed from the mask. Otherwise, this would cause issues with the role and feature calculations.

	Nodes	Edges
Train window 1	60,371	218,462
Train window 2	60,405	214,670
Train window 3	65,261	229,120
Test window	71,429	255,416

Table 1: Amount of nodes and edges per window

3.3 Features and Roles

Features and roles represent the characteristics of the nodes quantitatively. In this study, three different types of node features are distinguished: node centrality features, node pairs similarity features, and roles.

3.3.1 Node Centrality Features

These features represent the importance and centrality of the nodes in the network, each emphasising different aspects of a node. The centrality features used in this study are:

- **In-degree:** Number of incoming connections
- **Out-degree:** Number of outgoing connections
- **Eigenvector centrality:** Considers the node's importance as well as a node's degree.
- **PageRank centrality:** Developed by Google to determine the importance of nodes.
- **Average Shortest Path Length (ASPL):** Average number of steps needed to reach all other nodes.

3.3.2 Node Pair Similarity Features

Node pair similarity features quantify the similarity between two nodes. In the context of link prediction, the similarity between two nodes correlates directly with the probability of these nodes having a link (Liben-Nowell & Kleinberg, 2003).
This study includes two node similarity features.

- **The Jaccard Coefficient:** Compares the intersection with the union of the neighbours of two nodes.
- **The Dice-Sørensen Coefficient:** Is similar to the Jaccard Coefficient but is more robust against outliers.

The limitations of the igraph Python library in terms of the availability of the number of node similarity features and time constraints limited the scope of the study in this section.

3.3.3 Role Mining

In this study, the approach chosen was to opt for a role mining algorithm that examines the structure of a network: RolX (Henderson et al. 2012). Since the previously mentioned features are already based on communities (Abnar et al., 2015; Rossi & Ahmed, 2015), the rationale is

that the combination of community features and network structure will improve the overall results. Instead of assigning each node to the role with the highest probability, RolX calculates each node's percent membership to each role. The algorithm is not provided with any parameters; it detects the amount of roles itself.

3.4 One-Class Classification

The classification model is trained on valid and invalid edges. The model's performance is estimated on a test set that only holds true positive values, as the goal is to estimate the model's real-world performance, contrary to estimating the algorithm's performance on detecting artificially generated samples. Using a test set that only holds positive values implies that this model is used for a one-class classification task.

3.4.1 Classification Model

The classification model utilised is the sklearn Random Forest classification model. The model is configured with 100 trees (n_estimators=100), and the gini impurity (criterion='gini') is used for the split of the criteria. This Random Forest implementation aims to act as a straightforward to-replicate baseline rather than a high-performance model, as the main goal is to demonstrate the difference between the different combinations of features rather than creating a high-performance model.

3.4.2 Sampling Method

During the training, the model is trained on an equal split between valid and invalid edges. These invalid edges are artificially created using a random node sampling approach. During the sampling, sources and destination nodes can be randomly selected for a new artificial link. This approach implies that potential graph characteristics can be lost during the sampling process.

This is a possible limitation in the study as this might affect the results, but due to the limited scope of this study this approach has been chosen.

3.4.3 Performance Measures

By using only positive values during the test phase, false positive (FP) values and true negative (TN) values are missing. This restricts the metrics for performance analysis to metrics that exclusively use true negative (TN) values and true positive values (TP). The option of admitting artificially generated samples to overcome this issue is not preferred, since the goal is to test the

models' real-world performance rather than recognise the artificially generated samples. This implies that the model is performing a one-class classification task; thus, recall is used to evaluate the performance of the models.

- **Recall** expresses the number of true positives (TP) over the number of true positives (TP) plus the number of false negatives (FN). This can be defined as: $Recall = \frac{TP}{TP + FN}$

In the optimal scenario, a model would have a score of 1.0, implying that it retrieved all of the links, while a lower score implies that it was unable to retrieve all of the links. To effectively compare the models' performance, the recall scores are compared on various recall thresholds.

4. Results

The performance of various models is evaluated by examining the recall score across various thresholds, presented in Figure 2. The recall metric indicates a model's ability to predict all positive samples correctly, implying that a higher recall score results in a lower quantity of false negatives. Table 2 provides a high-to-low overview of the Area Under the Curve values.

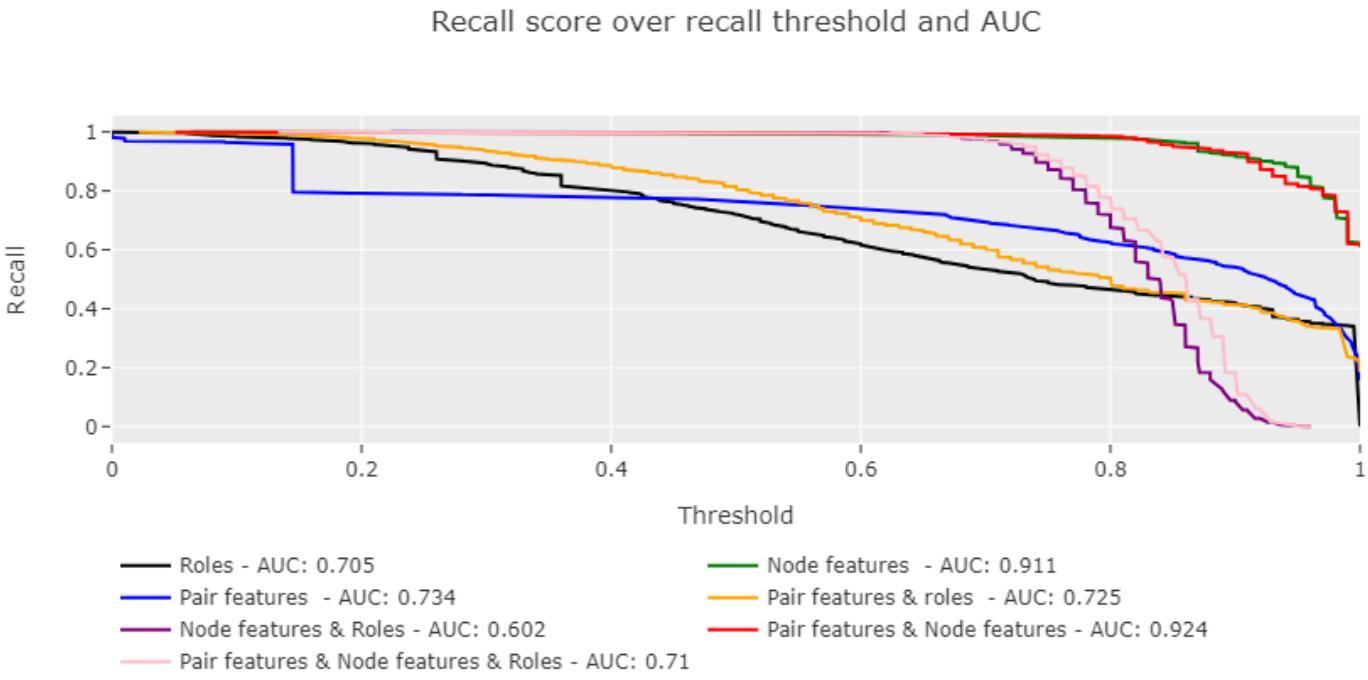

Figure 2: Impact of feature sets over varying recall thresholds.

Performance overview for the feature combinations:

- **Role Features:** The recall curve for the role probability feature (black line) starts decreasing over the lower threshold settings, indicating relatively low performance until the 0.8 threshold level. At the 0.8 level, the model performs better than other feature sets. However, the overall performance is still quite inadequate, this results in an AUC score of 0.705. Indicating a limitation for the role probabilities to be utilised in a link prediction task.
- **Node Features:** The node features (green line) are the best-performing non-feature combination in this test with a 0.911 AUC. Their performance stays close to perfect until

just after the 0.8 threshold mark, indicating high effectiveness compared to the other non-feature combination sets.

- **Pair Features:** The recall curve for the pair features (blue line) quickly jumps to a 0.8 recall score. The curve stays at this level until the 0.6 threshold, after which performance deteriorates, resulting in a 0.734 AUC. It performs worse than the role probabilities in the lower threshold range but better in the higher threshold range.
- **Pair Features & Roles:** The combination of pair features and roles achieves average results across all threshold levels acknowledged by the AUC score of 0.725 (orange line). The AUC has decreased compared to using the pair features independently, revealing that the role probabilities and the pair features are non-complementary features in this scenario.
- **Node Features & Roles:** Integrating the role probabilities with the node features (purple line) results in worse performance in the higher threshold range compared to only using the node features, resulting in an AUC score of 0.602. This indicates that the role probabilities are not complementary to the node features.
- **Pair Features & Node Features:** Combining the node features with the node similarity features (red line) improves performance in the high threshold range, resulting in the best-performing combination with an AUC of 0.924.
- **Pair Features & Node Features & Roles:** The combination of the three sets (pink line) performs worse than using the combination of pair features and node features. Indicating that role probabilities are non-complementary in this scenario. The AUC dropped significantly to 0.710 compared to the 0.924 that was achieved by the combination of pair features and node features.

Role probabilities	Node features	Pair features	AUC score
	x	**x**	0.924
	x		0.911
		x	0.734
x		**x**	0.725
x	**x**	**x**	0.710
x			0.705
x	**x**		0.602

Table 2: Overview of AUC scores for the different feature combinations.

The analysis shows that combining the different sets, overall reveals positive results. The combination of the three sets does not achieve the highest result. It occurs that the combination of the pair features and the node features is the most robust. This is in line with the performance of the node features, which by themselves perform the strongest of the three different sets. However, results diminish when combined with the role probabilities. Role probabilities score low across all feature combinations; they don't seem to be complementary to other features and are not impactful by themselves compared to the other features.

The limited performance of role probabilities can be justified by several assumptions. One of the arguments is that the performance of the role mining-based model rapidly deteriorates as time in the mask progresses. We can invalidate this assumption by examining Figure 3.

Figure 3 illustrates the cumulative recall score for the role probabilities subset from April 16 to April 21. The X-axis represents the time period, which is divided into six days. The y-axis represents the recall score on a scale from 0 to 1.

The recall curve shows a linear trend. This indicates that the model's recall is stable over the six-day period. In other words, the model's performance does not seem to be impacted by the length of the mask, indicating that the model's performance is stable over at least a six-day timespan.

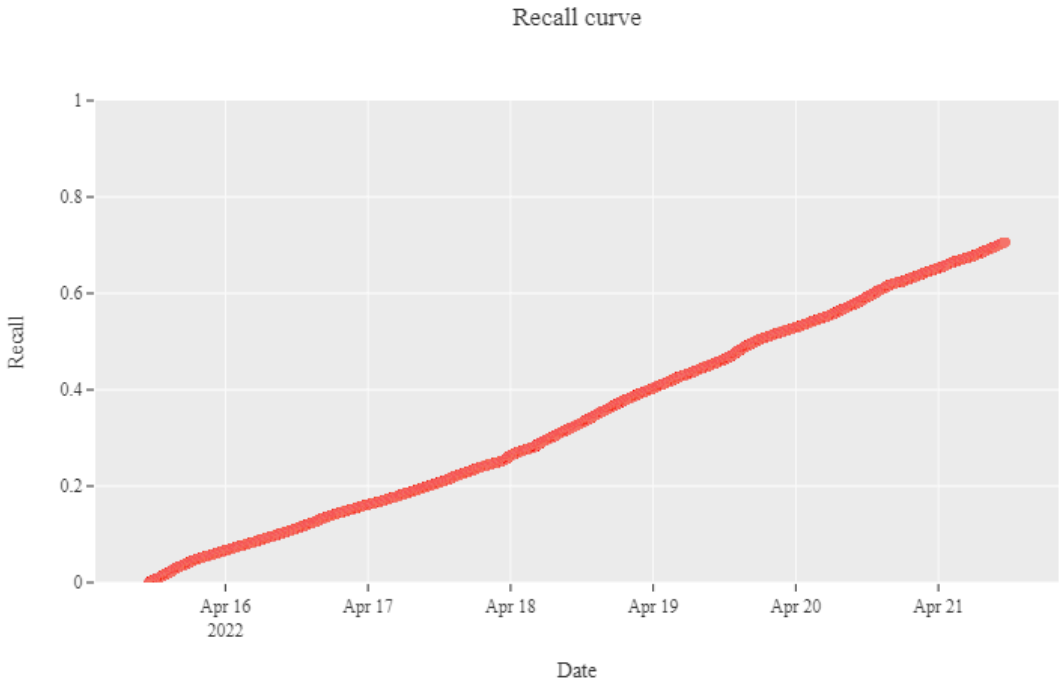

Figure 3: Cumulative recall score over a six-day time period for the role probability subset.

5. Discussion & Future Work

The results reveal that node features are more effective for link prediction than node similarity features. This contradicts studies from social graph networks, in which node pair similarity features are mostly praised for their simplicity in combination with their performance (Liben-Nowell & Kleinberg, 2003; Kumar et al., 2020). A possible explanation for this contradiction are the topological dissimilarities of social networks and financial transaction networks (Xiang et al., 2022; Wang et al., 2014). We expect that in financial networks nodes with a high degree are more prevalent, thus impacting the effectiveness of centrality measures. However, these are current assumptions that should be precluded by future work. However, it can be stated with certainty that there is a structural topological difference between financial networks and social networks; and that they, therefore, assumably respond differently to link prediction feature sets.

Furthermore, it is interesting to observe that role probabilities do not seem to complement other feature sets, particularly since the role probabilities were calculated based on the network's topology while the other features were community-based. It is important to investigate whether topological information from the graph can be beneficial for learning-based link prediction in financial graphs. In addition, the performance of link prediction models that are based on learning-based frameworks is heavily dependent on the utilised feature set (Liben-Nowell and Kleinberg, 2003). This suggests that the role probabilities, as implemented within the current methodology, may not effectively integrate with the other graph features. From this philosophy, more research is needed to investigate how role probabilities can successfully be integrated with other graph features.

6. Conclusion

This study examines the impact of role mining on learning-based link prediction models in financial networks. We use various features derived from the transaction graph, including role probabilities, node similarity features, and node centrality features, in various combinations to compare their effectiveness.

Role mining does not enhance the performance of learning-based link prediction models; its integration with other features is non-complementary. Specifically, the combination of role probabilities and node features is ineffective. Adding the role probabilities to the node features results in a performance loss of just over 0.3 AUC score. This highlights the subtle essence of feature selection for learning-based models.

Future research should examine these features in different configurations, generalising the results and finding the limitations of this study to provide a better understanding of the effects of role mining on link prediction in financial graph networks.

The study's results are evident; role mining does not have a positive effect on link prediction in financial networks in the presented configuration. Additional research is needed to examine the effectiveness of role mining in financial networks with different configurations to ensure that it is not effective.

7. References

1. Abnar, A., Takaffoli, M., Rabbany, R., & Zaïane, O. R. (2015). SSRM: structural social role mining for dynamic social networks. *Social Network Analysis And Mining*, *5*(1). https://doi.org/10.1007/s13278-015-0292-y
2. Ai, B., Qin, Z., Shen, W., & Li, Y. (2022, 14 januari). *Structure Enhanced Graph Neural Networks for Link Prediction*. arXiv.org. https://arxiv.org/abs/2201.05293
3. Biswas, A., & Biswas, B. (2017). Community-based link prediction. *Multimedia Tools And Applications*, *76*(18), 18619–18639. https://doi.org/10.1007/s11042-016-4270-9
4. Bonacich, P. (1987). Power and centrality: A family of measures. *American journal of sociology*, *92*(5), 1170-1182.
5. Greaves, A., & Au, B. (2015, 8 December). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. stanford.edu. Geraadpleegd op 27 mei 2024, van https://snap.stanford.edu/class/cs224w-2015/projects_2015/Using_the_Bitcoin_Transaction_Graph_to_Predict_the_Price_of_Bitcoin.pdf
6. Gupta, S., & Sadoghi, M. (2019). Blockchain Transaction Processing. In *Springer eBooks* (pp. 366–376). https://doi.org/10.1007/978-3-319-77525-8_333
7. Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., & Li, L. (2012). RolX. *KDD '12: Proceedings Of The 18th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. https://doi.org/10.1145/2339530.2339723
8. Huang, S., Poursafaei, F., Danovitch, J., Fey, M., Hu, W., Rossi, E., Leskovec, J., Bronstein, M., Rabusseau, G., & Rabbany, R. (2023, 3 juli). *Temporal Graph Benchmark for Machine Learning on Temporal Graphs*. arXiv.org. https://arxiv.org/abs/2307.01026

9. Jin, R., Lee, V. E., & Hong, H. (2011, 18 februari). *Axiomatic Ranking of Network Role Similarity*. arXiv.org. https://arxiv.org/abs/1102.3937

10. Kumar, A., Sr., Singh, S. S., Singh, K., Biswas, B., & Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221–005, India. (2020). Link prediction techniques, applications, and performance: A survey. In *Physica A* (Vol. 553, p. 124289).

11. Kumar, S., Mallik, A., & Panda, B. S. (2022). Link prediction in complex networks using node centrality and light gradient boosting machine. *World Wide Web*, *25*(6), 2487–2513. https://doi.org/10.1007/s11280-021-01000-3

12. Leskovec, J., & Faloutsos, C. (2006). Sampling from large graphs. *KDD '06: Proceedings Of The 12th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. https://doi.org/10.1145/1150402.1150479

13. Liang, J., Li, L., & Zeng, D. (2018). Evolutionary dynamics of cryptocurrency transaction networks: An empirical study. *PloS One*, *13*(8), e0202202. https://doi.org/10.1371/journal.pone.0202202

14. Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks. *CIKM '03: Proceedings Of The Twelfth International Conference On Information And Knowledge Management*, 556–559. https://doi.org/10.1145/956863.956972

15. Lin, D., Wu, J., Xuan, Q., & Tse, C. K. (2022). Ethereum transaction tracking: Inferring evolution of transaction networks via link prediction. *Physica A*, *600*, 127504. https://www.sciencedirect.com/science/article/pii/S0378437122003600

16. Moya, M., Koch, M., & Hostetler, L. (1993). One-class classifier networks for target recognition applications. In *Proceedings of the World Congress on Neural Networks* (pp. 797-801). International Neural Network Society, INNS, Portland, OR.
17. Muniz, C. P., Goldschmidt, R., & Choren, R. (2018). Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowledge-based Systems*, *156*, 129–137. https://doi.org/10.1016/j.knosys.2018.05.027
18. Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). *The PageRank Citation Ranking: Bringing Order to the Web* (). Stanford Digital Library Technologies Project .
19. Pulipati, S., Somula, R., & Parvathala, B. R. (2021). Nature inspired link prediction and community detection algorithms for social networks: a survey. *International Journal Of System Assurance Engineering And Management*. https://doi.org/10.1007/s13198-021-01125-8
20. R. A. Rossi and N. K. Ahmed, "Role Discovery in Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 4, pp. 1112-1131, 1 April 2015, doi: 10.1109/TKDE.2014.2349913.
21. Rafique, W., Khan, M., Sarwar, N., & Dou, W. (2019). SocioRank*: A community and role detection method in social networks. *Computers and Electrical Engineering*, 76, 122–132. https://doi.org/10.1016/j.compeleceng.2019.03.010
22. Shamsi, K., Gel, Y. R., Kantarcioglu, M., & Akcora, C. G. (2022). Chartalist: Labeled graph datasets for UTXO and account-based blockchains. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, November 29-December 1, 2022, New Orleans, LA, USA* (pp. 1–14).

23. Wang, P., Xu, B., Wu, Y., & Zhou, X. (2014). Link Prediction in Social Networks: the State-of-the-Art. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1411.5118

24. Wu, J., Liu, J., Zhao, Y., & Zibin Zheng. (2021). Analysis of Cryptocurrency Transactions from a Network Perspective: An Overview. In *Journal of Network and Computer Applications* (pp. 1–24) [Journal-article]. https://arxiv.org/pdf/2011.09318

25. Wu, Z., Pan, S., Chen, F., Long, G., Zabg, C., & Yu, P. (2020, March 24). *A comprehensive survey on graph neural networks*. IEEE Journals & Magazine | IEEE Xplore. Retrieved May 28, 2024, from https://ieeexplore.ieee.org/abstract/document/9046288?casa_token=tYwDyWpOeIQAAAAA:n1gHAtc7JAUHkakg8Lx_VS-0UVUfHEVcOceO1A5WmX0a-_32_xSGUeTkNBRHvRzC7ckphaj_

26. Xiang, S., Cheng, D., Shang, C., Zhang, Y., & Liang, Y. (2022). Temporal and Heterogeneous Graph Neural Network for Financial Time Series Prediction. *Proceedings Of The 31st ACM International Conference On Information &Amp; Knowledge Management*. https://doi.org/10.1145/3511808.3557089

27. Yang, Y., Lichtenwalter, R. N., & Chawla, N. V. (2014). Evaluating link prediction methods. *Knowledge and Information Systems*, *45*(3), 751–782. https://doi.org/10.1007/s10115-014-0789-0

28. Yousef, M., Najami, N., & Khalifav, W. (2010). A comparison study between one-class and two-class machine learning for MicroRNA target detection. *Journal Of Biomedical Science And Engineering*, *03*(03), 247–252. https://doi.org/10.4236/jbise.2010.33033

29. Zambre, D., & Shah, A. (2013, 10 December). *Analysis of Bitcoin Network Dataset for Fraud*. stanford.edu. https://snap.stanford.edu/class/cs224w-2013/projects2013/cs224w-030-final.pdf

30. Zhang, M., & Chen, Y. (2018). *Link Prediction Based on Graph Neural Networks*. https://proceedings.neurips.cc/paper/2018/hash/53f0d7c537d99b3824f0f99d62ea2428-Abstract.html