machine learning & knowledge extraction

Review

Review on Learning and Extracting Graph Features for Link Prediction

Ece C. Mutlu [1,†], Toktam Oghaz [2,†], Amirarsalan Rajabi [2,†] and Ivan Garibay [1,2,*]

[1] Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816, USA; ece.mutlu@ucf.edu
[2] Department of Computer Science, University of Central Florida, Orlando, FL 32816, USA; Toktam.Oghaz@ucf.edu (T.O.); amirarsalan@knights.ucf.edu (A.R.)
* Correspondence: igaribay@ucf.edu
† These authors contributed equally to this work.

Received: 29 October 2020; Accepted: 11 December 2020; Published: 17 December 2020

Abstract: Link prediction in complex networks has attracted considerable attention from interdisciplinary research communities, due to its ubiquitous applications in biological networks, social networks, transportation networks, telecommunication networks, and, recently, knowledge graphs. Numerous studies utilized link prediction approaches in order sto find missing links or predict the likelihood of future links as well as employed for reconstruction networks, recommender systems, privacy control, etc. This work presents an extensive review of state-of-art methods and algorithms proposed on this subject and categorizes them into four main categories: similarity-based methods, probabilistic methods, relational models, and learning-based methods. Additionally, a collection of network data sets has been presented in this paper, which can be used in order to study link prediction. We conclude this study with a discussion of recent developments and future research directions.

Keywords: complex networks; graph analysis; proximity; supervised link prediction; unsupervised link prediction

1. Introduction

Online social networks [1], biological networks, such as protein–protein interactions and genetic interactions between organisms [2], ecological systems of species, knowledge graphs [3], citation networks [4], and social relationships of users in personalized recommender systems [5], are all instances of graphs of complex interactions, which are also referred to as complex networks. While these networks are almost always dynamic in nature, a vital query is how they change over time. More specifically, what are the future associations between entities in a graph under investigation. The problem of link prediction in graphs is one of the most interesting and long-standing challenges. Given a graph, which is an abstraction of relationships among entities of a network, link prediction is to anticipate future connections among entities in the graph, with respect to its current state. Link prediction models might

(i) exploit the similarity metrics as the input features,
(ii) embed the nodes into a low dimensional vector space while preserving the topological structure of the graph, or
(iii) combine the information that is derived from the two aforementioned points, with the node attributes available from the data set.

All of these models rely on the hypothesis that higher similarity between nodes results in a higher probability of connection [6].

Applications of link prediction include analyzing user–user and user–content recommendations in online social networks [5,7–9], reconstruction of the PPI (protein–protein interaction) network and reducing the present noise [10–12], hyper-link prediction [13], prediction of transportation networks [14], forecasting the behavior of terrorism campaigns and social bots [15,16], reasoning and sensemaking in knowledge graphs [17], and knowledge graph completion while using data augmentation with Bidirectional Encoder Representations from Transformers (BERT) [18,19]. Link prediction in these applications has been mostly investigated through unsupervised graph representation and feature learning methods that are based on the node (local) or path (global) similarity metrics that evaluate the neighboring nodes. Common neighbors, preferential attachment, Jaccard, Katz, and Adamic Adar are some of the most widely used similarity metrics that measure the likelihoods of edge associations in graphs. While these methods may seem to be dated, they are far from being obsolete. Despite the fact that these methods do not discover the graph attributes, they have remained popular for years, due to their simplicity, interpretability, and scalability. Probabilistic models, on the other hand, aim to predict the likelihood of future connections between entities in an evolving dynamical graph with respect to the current state of the graph. Another context under which the problem of link prediction is raised is relational data [20–23]. In this context, when considering the relational data set in which objects are related to each other, the task of link prediction is to predict the existence and type of links between pairs of objects [24]. However, the availability of labeled data allows for the supervised machine learning algorithms to provide new solutions for the link prediction task, including neural network-based methods for link prediction [25], which allow for learning a suitable heuristic than assuming strong relationships among vertices.

Similar surveys on the topic of link prediction exist, and this survey has benefited from them. The work of [26] provides a comprehensive review of the problem of link prediction within different types of graphs and the applications of different algorithms. Other related review papers on this topic include the works of [27,28]. The work of [28] reviews the progress of link prediction algorithms from a physical perspective, applications, and challenges for this line of research. While some of these reviews only focus on a specific set of methodologies that are proposed for link prediction, such as the work of [27], which presents an extensive review on relational machine learning algorithms, specifically designed for knowledge graphs, some important related methodologies are overlooked in the aforementioned studies. For instance, [28] does not discuss some important graph feature learning and neural network-based techniques that have been recently developed. Our effort has been to provide a review that includes the most recent approaches for the problem of link prediction that demonstrate promising results, but are not fully covered by exceptional similar surveys, such as the works of [26–28]. Thus, we believe that our study provides comprehensive information on the topic of link prediction for large networks, and it can help to discover the most related link prediction algorithms that are deliberately categorized into the proposed taxonomy. This study reviews similarity-based methods, including local, global, and quasi-local approaches, probabilistic and relational methods as unsupervised solutions to the link prediction problem, and, finally, learning-based methods, including matrix factorization, path and walk based link prediction models, and using neural networks for link prediction.

2. Background

A graph (complex network), denoted as $G = \langle V, E \rangle$, can be defined as the set of vertices (nodes) V, and the interactions among pairs of nodes, called links (edges) E, at a particular time t. It should be noted that in this problem setting, self-connections, and multiple links between nodes are not allowed and, accordingly, are not taken into account in the majority of link prediction problem settings [28]. The main idea behind applying feature extraction or feature learning-based methods for the link prediction problem is to use the present information regarding the existing edges in order to predict the future or missing link that will emerge at time $t^{'} > t$. The types of graphs can be classified into two main categories according to the direction of the information flow between interacted nodes;

directed and undirected graphs. Although many of the discussed methods in the next sections of this paper can provide solutions to the link prediction problem in directed graphs, the majority of the reviewed methods in this survey address the problem of link prediction for undirected graphs. The difference between the link prediction problem for these two graph categories arises from the additional information that is required for the directed graphs. This information refers to the origin of the associated link in directed graphs, in which $\langle v_x, v_y \rangle$ conveys the existence of a directed edge from node v_x to v_y and $\langle v_x, v_y \rangle \neq \langle v_y, v_x \rangle$ [29]. However, edges in undirected graphs have no orientation, and the relations among the node pairs are reciprocal. The set of nodes that are connected to node $v_x \in V$ are known as the "neighbors" of v_x, denoted as $\Gamma(v_x) \subseteq V$, and the number of edges that are connected to the node v_x is referred to as $|\Gamma(v_x)|$. Link prediction algorithms necessitate training and test sets to be compared in the case of model performance, similar to other machine learning methods. However, one cannot know the future links of a graph at time $t^{'}$, given the current graph structure. Therefore, a fraction of links from the current graph structure is deleted (Figure 1), and taken as the test set; whereas, the remaining fraction of edges in the graph is used for the training purpose. A reliable link prediction approach should provide higher probabilities for the edges that belong to the set of true positives than the set of nonexistent edges [30]. Apparently, by treating the link prediction task as a binary classification problem, conventional evaluation metrics of binary classification in machine learning can be applied in order to evaluate the performance of link prediction. Within the context of the confusion matrix, TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative) metrics can be used in order to assess performance. In this context, sensitivity, specificity, precision, and accuracy are computed, as follows ([31]):

$$\begin{aligned} \text{Sensitivity (Recall)} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN}+\text{FP}} \\ \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \end{aligned} \tag{1}$$

The most common standard metric that is used to quantify the performance of the link prediction algorithms is "the area under the receiver operating characteristic curve (AUC)" [32]. The AUC value represents the probability that a randomly selected missing link between two nodes is given a higher similarity score than the randomly selected pair of unconnected links. The algorithmic calculation of AUC is given by:

$$\text{AUC} = \frac{n' + 0.5n''}{n} \tag{2}$$

where n is the number of total independent comparisons and n' is the number of comparisons in which the missing link has a higher score than the unconnected link, while n'' is the number of comparisons when they show equal scores.

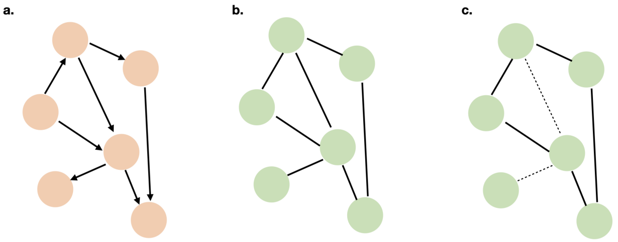

Figure 1. Imaginary representation of (**a**). Directed whole graph (**b**). Undirected whole graph (**c**). Undirected training graph.

One of the applications of link prediction is in recommender systems [5,33] that exploit information on users' social interactions in order to find their desired information according to their interests and preferences. Therefore, within this context, the following evaluation metrics are also used [34–36]:

$$\begin{aligned} \text{Precision at } n &= \frac{r}{n} \\ \text{Recall at } n &= \frac{r}{R} \\ \text{Mean Reciprocal Rank} &= \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \\ \text{Average Precision} &= \sum_{k=1}^{n} P(k)\Delta r(k) \\ \text{Mean Average Precision} &= \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \end{aligned} \tag{3}$$

In the above equations, Precision at n shows the number of relevant results (r) among the top n results or recommendations. In Recall at n, R presents the total number of relevant results. In order to calculate Mean Reciprocal Rank, first, the inverse of the ranking of the first correct recommendation is calculated ($\frac{1}{rank_i}$) and, then, an average over the total queries (Q) is taken. In order to calculate Average Precision, Precision at a threshold of k in the list is multiplied by the change in recall from items $k-1$ to k, and this process is summed over all of the positions in the ranked sequence of documents. The Mean Average Precision is then the average of all Average Precisions over total queries (Q).

In order to provide a few visualization examples for complex networks, Figure 2 demonstrates the network structure of the two different hashtag co-occurrence graphs (#askmeanything and #lovemylife) of the Instagram posts from 04/01/2020 to 04/08/2020. These two different figures clearly demonstrate the variability of the network structure, even in the same fields, i.e., Figure 2a. shows different sub-communities with its more sparse structure, while Figure 2b. represents a densely connected network example.

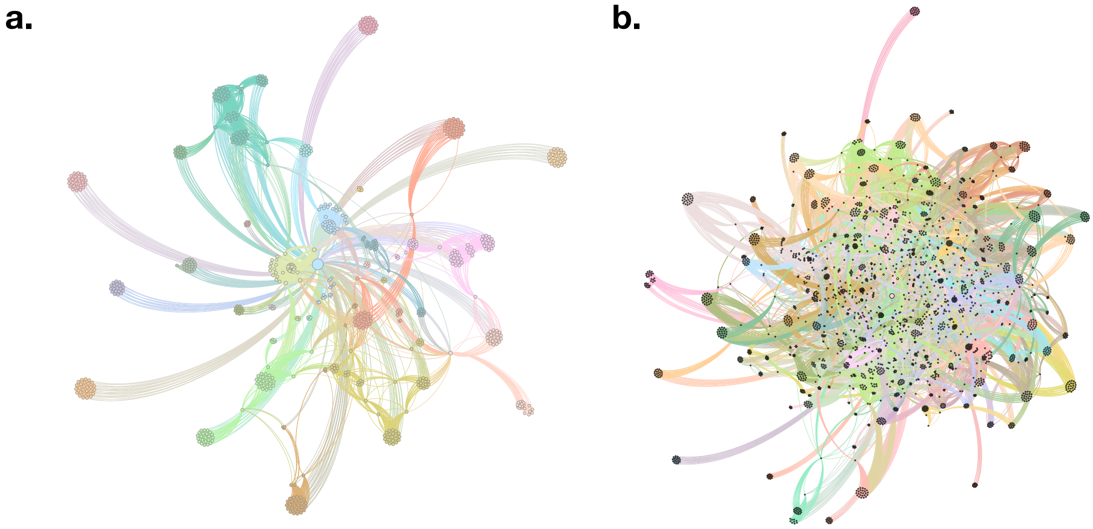

Figure 2. Hashtag co-occurrence graph (via Gephi) of (**a**). #askmeanything (757 nodes and 16046 edges), (**b**). #lovemylife (2748 nodes and 63413 edges). Each network is colored by the modularity ranking.

3. Similarity Based Methods

Similarity-based methods, which mainly focus on the topological structure of the graph, are the most straightforward and oldest link prediction metrics. These methods try to figure the missing links out by assigning similarity score, $s_{(v_x,v_y)}$, between node pairs (v_x and v_y) using the structural property of the graphs. These methods can be investigated under three main categories: local, quasi-local, and global approaches.

3.1. Local Similarity-Based Approaches

Local similarity-based approaches are based on the assumption that, if node pairs have common neighbor structures, they will probably form a link in the future. Because they only use local topological information based on neighborhood-related structures rather than considering the whole network topology, they are faster than the global similarity-based approaches. Many studies also showed their superior performance, especially on the dynamic networks [37]. However, they are restricted to compute the similarity of all possible combinations of the node pairs, since they only rank similarity between close nodes with a distance of less than two.

3.1.1. Common Neighbors (CN)

CN is one of the most extensive information retrieval metrics for link prediction tasks, due to its high efficiency, despite its simplicity. The idea behind CN is very intuitive; the probability of being linked for two nodes in the future is affected by the number of their common neighboring nodes, i.e., two nodes will highly probably establish a link if they have more shared nodes. The score of this metric can be defined, as follows:

$$s^{CN}_{(v_x,v_y)} = |\Gamma(v_x) \cap \Gamma(v_y)| \tag{4}$$

where $\Gamma(.)$ represents the set of adjacent nodes.

It should be noted that the resulting score using CN is not normalized, and only shows the relative similarity of different node-pairs by considering shared nodes between them. Newman used CN in order to show that the probability of collaboration between two scientists in the future can be estimated by their previous common collaborators [38].

3.1.2. Jaccard Index (JC)

The metric not only takes the number of common nodes into account as in CN, but it also normalizes it by considering the total set of numbers of shared and non-shared neighbors. The equation of this score that is proposed by Jaccard [39] is:

$$s^{JC}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x) \cup \Gamma(v_y)|} \tag{5}$$

3.1.3. Salton Index (SL)

SL is the metric that is also known as cosine similarity. It calculates the cosine angle between the two columns of the adjacency matrix and it is identified as the ratio of the number of shared neighbors of v_x and v_y to the square root of inner-product of their degrees [40], as follows:

$$s^{SL}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{\sqrt{|\Gamma(v_x)|.|\Gamma(v_y)}} \tag{6}$$

Wagner & Leydesdorff [41] showed that SI is an efficient metric, especially when the aim is to visualize the constructional pattern of relations in a graph.

3.1.4. Sørensen Index (SI)

The index, which is very similar to JC, is generated to make a comparison between different ecological samples [42], such that:

$$s^{SI}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x)| + |\Gamma(v_y)|} \tag{7}$$

The difference in using the summation of the degrees instead of the size of the union of their neighbors makes SI less outlier sensitive when compared to JC [43].

3.1.5. Preferential Attachment Index (PA)

Motivated by the study by Barabasi & Albert [44], new nodes joining the network are more likely to connect with the nodes with higher connections (hub) than the nodes with lower degrees, PA can be formulated as:

$$s^{PA}_{(v_x,v_y)} = |\Gamma(v_x)|.|\Gamma(v_y)| \tag{8}$$

3.1.6. Adamic-Adar Index (AA)

The metric is employed for the necessity of the comparison of two web-pages by Lada Adamic and Eytan Adar [45]. It simply uses the idea of giving more weight to the relatively fewer common neighbors, such that:

$$s^{AA}_{(v_x,v_y)} = \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} \frac{1}{\log|\Gamma(v_z)|} \tag{9}$$

where v_z refers to a common neighbor for nodes v_x and v_y (connected/linked to both).

Although this metric has similarities to CN, the vital difference is that the logarithm term penalizes the shared neighbors of the two corresponding nodes. It should be noted that while the other metrics include only two nodes (v_x and v_y) and/or their degrees in their equations so far, AA also relates familiar neighbors (v_z) to these two nodes (v_x and v_y).

3.1.7. Resource Allocation Index (RA)

Motivated by the physical process of resource allocation, a very similar metric to AA was developed by Zhou et al. [46] which can be formulated as:

$$s^{RA}_{(v_x,v_y)} = \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} \frac{1}{|\Gamma(v_z)|} \tag{10}$$

The difference in the denominator ($|\Gamma(v_z)|$) of RA rather than its logarithm ($log|\Gamma(v_z)|$) as in AA penalizes the contribution of common neighbors more. Many studies show that this discrepancy is insignificant, and the resulting performances of these two metrics are very similar when the average degree of the network is low; however, RA is superior when the average degree is high [47].

3.1.8. Hub Promoted Index (HP)

The index is proposed for assessing the similarity of the substrates in the metabolic networks [48], and it can be defined, as follows:

$$s^{HP}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{\min(|\Gamma(v_x)|, |\Gamma(v_y)|)} \tag{11}$$

HP is determined by the ratio of the number of common neighbors of both v_x and v_y to the minimum of degrees of v_x and v_y. Here, link formation between lower degree nodes and the hubs is more promoted, while the formation of the connection between hub nodes are demoted [6].

3.1.9. Hub Depressed Index (HD)

The totally opposite analogy of HP is also considered by Lü and Zhou [28], and it is determined by the ratio of the number of common neighbors of both v_x and v_y to the maximum of degrees of v_x and v_y. Here, the link formation between lower degree nodes and link formation between hubs is promoted. However, the connection between hub nodes and lower degree nodes are demoted, such that:

$$s^{HD}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{\max(|\Gamma(v_x)|, |\Gamma(v_y)|)} \tag{12}$$

3.1.10. Leicht-Holme-Newman Index (LHN)

The index, which is very similar to SI, is defined as the ratio of the number of shared neighbors of v_x and v_y to the product of their degrees (the expected value of the number of paths of length between them) [49]. It can be represented by:

$$s^{LHN}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x)|.|\Gamma(v_y)|} \tag{13}$$

The only difference in the denominator as compared to SI shows that SI always assigns a higher score than LHN, i.e., $|\Gamma(v_x)|.|\Gamma(v_y)| \leq |\Gamma(v_x) \cap \Gamma(v_y)|$.

3.1.11. Parameter Dependent Index (PD)

Zhou et al. [50] proposed a new metric in order to improve the prediction accuracy for popular links and unpopular links. PD can be defined as:

$$s^{PD}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x)|.|\Gamma(v_y)|^{\beta}}, \tag{14}$$

where β is a free parameter and it can be tuned to the topology of the graph. One can easily recognizes that PD is degraded to CN, SL, and LHN when $\beta = 0$, $\beta = 0.5$, and $\beta = 1$, respectively.

3.1.12. Local Affinity Structure Index (LAS)

LAS shows the affinity relationship between a pair of nodes and their common neighbors. The hypothesis is that a higher affinity of two nodes and their common neighbors increases the probability of getting connected [51], such as:

$$s^{LAS}_{(v_x,v_y)} = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x)|} + \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_y)|} \tag{15}$$

3.1.13. CAR-Based Index (CAR)

When a node interacts with another neighbor node, it is called a first-level neighborhood; whereas, the interaction between the first-level neighbor node and its neighbor node is called the second-level neighborhood for the seed node. According to the local community paradigm (LCP) of Cannistraci [52], the researchers mostly consider the first-level neighborhood, because the second-level neighborhood is noisy; however, the second-level neighborhood carries essential information regarding the topology of the network. Therefore, CAR filters these noises and considers nodes that are interlinked with neighbors mostly. The similarity metric can be calculated, as follows:

$$s^{CAR}_{(v_x,v_y)} = |\Gamma(v_x) \cap \Gamma(v_y)| \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} \frac{|\Gamma(v_z)|}{2}. \tag{16}$$

3.1.14. The Individual Attraction Index (IA)

Dong et al. [53] proposed an index that relates not only to the common neighbors of the nodes individually, but also the effect of the sub-network created by those. The IA score can be formulated as:

$$s^{IA}_{(v_x,v_y)} = \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} \frac{|\Gamma(v_x) \cap \Gamma(v_y) \cap \Gamma(v_z)| + 2}{|\Gamma(v_z)|}. \tag{17}$$

Because IA considers the existence of links between all common neighbors, the algorithm is very time-consuming. Therefore, a simpler alternative is also proposed as:

$$s^{IA*}_{(v_x,v_y)} = \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} \frac{|\Gamma(v_x) \cap \Gamma(v_y)| + 2}{|\Gamma(v_x) \cap \Gamma(v_y)|.|\Gamma(v_z)|} \tag{18}$$

3.1.15. The Mutual Information Index (MI)

This method examines the link prediction problem while using information theory, and it measures the likelihood of conditional self-information when their common neighbors are known [54], and formulated as:

$$s^{MI}_{(v_x,v_y)} = -I(e_{v_x,v_y}|v_z), \tag{19}$$

where $v_z \in \Gamma(v_x) \cap \Gamma(v_y)$ and $I(.)$ is the self-information function for a node and it can be calculated by (20). Here, $I(e_{v_x,v_y}|v_z)$ means conditional mutual self-information of the existence of a link between v_x and v_y and their shared set of neighbors. The smaller value of $s^{MI}_{(v_x,v_y)}$ means the higher likelihood to be linked. If all of the link between common neighbors be independent of each other, the self-information of that node pair can be calculated as [6]:

$$I(e_{v_x,v_y}|v_z) = \log 2 \frac{|\{e_{v_x,v_y} : v_x, v_y \in \Gamma(v_z), e_{v_x,v_y} \in E\}|}{\frac{1}{2}|\Gamma(v_z)|(|\Gamma(v_z)| - 1)}. \tag{20}$$

3.1.16. Functional Similarity Weight (FSW)

This index is first used by Chou et al. in order to understand the similarity of physical or biochemical characteristics of proteins [55]. Their motivation is based on the Czekanowski–Dice distance that is used in [56] in order to estimate the functional similarity of proteins. This score can be defined as:

$$s^{FSW}_{(v_x,v_y)} = \left(\frac{2|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x) - \Gamma(v_y)| + 2|\Gamma(v_x) \cap \Gamma(v_y)| + \beta} \right)^2. \tag{21}$$

Here, β is used to penalize the nodes with very few common neighbors, and it is defined as:

$$\beta = \max(0, |\Gamma_{avg}| - (|\Gamma(v_x) - \Gamma(v_y)|) + (|\Gamma(v_x) \cap \Gamma(v_y)|)), \tag{22}$$

where $|\Gamma_{avg}|$ is the average number of neighbours in the network.

3.1.17. Local Neighbors Link Index (LNL)

Motivated by the cohesion between common neighbors and predicted nodes, both attributes, and topological features are examined in [57], as:

$$s^{LNL}_{(v_x,v_y)} = \sum_{v_z \in (\Gamma(v_x) \cap \Gamma(v_y))} w(v_z), \tag{23}$$

where $w(v_z)$ is the weight function that can be measured by:

$$w(v_z) = \frac{\sum_{v_u \in \Gamma(v_x) \cup v_x} \delta(v_z, v_u) + \sum_{v_v \in \Gamma(v_y) \cup v_y} \delta(v_z, v_y)}{|\Gamma(v_z)|}. \quad (24)$$

Here, $\delta(a, b)$ is a boolean variable that is equal to 1 if there exists a link between a and b; otherwise, it equals to 0.

3.2. Global Similarity-Based Approaches

Global similarity-based approaches, contrary to local ones, use the whole topology of the network to rank the similarity between node pairs; therefore, they are not limited to measure the similarity between nodes that are locating far away from each other. Although considering the whole topology of the network gives more flexibility in link prediction analysis, it also increases the algorithm's time complexity. Because an ensemble of all paths between node pairs is used, they can also be called path-based methods.

3.2.1. Katz Index (KI)

The metric, which is defined by Katz [58], sums over the sets of paths and is exponentially damped by length to be counted more intensively with shorter paths. This index can be formulated with a vector space:

$$s^{KI}_{(v_x,v_y)} = \sum_{i=1}^{\infty} \beta^i.|A^{\langle i \rangle}_{v_x v_y})|. \quad (25)$$

Here, A is the adjacency matrix and β is a free parameter ($\beta > 0$) that is also called a "damping factor". One can realize that KI yields to a very similar score when β is low enough as the paths that have higher lengths contribute less, and the similarity index is simply determined by the shorter paths [28].

In the case of $\beta < \frac{1}{\lambda^A_1}$, where λ^A_1 is the largest eigenvalue of the adjacency matrix, the similarity matrix can be written, as follows:

$$S^{KI} = (I - \beta A)^{-1} - I, \quad (26)$$

where I is the identity matrix.

3.2.2. Global Leicht-Holme-Newman Index (GLHN)

The idea behind GLHN is very similar to that of KI, since it also considers a high similarity for the nodes if the number of paths between these corresponding nodes is high [49], such that:

$$S^{GLHN} = \beta_1 (I - \beta_2 A)^{-1}, \quad (27)$$

where β_1 and β_2 are free parameters, and a smaller value of β_2 considers higher importance for the shorter paths, and vice versa.

3.2.3. SimRank (SR)

This index computes the similarity starting from the hypothesis "two objects are similar if they are related to similar objects", and it is recursively defined [59]. SR is equal to 1 when node $v_x = v_y$, otherwise:

$$s^{SR}_{(v_x,v_y)} = \gamma. \frac{\sum_{v_{z_1} \in \Gamma(v_x)} \sum_{v_{z_2} \in \Gamma(v_y)} s^{SR}_{(v_{z_1},v_{z_2})}}{|\Gamma(v_x)|.|\Gamma(v_y)|}, \quad (28)$$

where $\gamma \in [0, 1]$ is called decay factor and it controls how fast the effect of neighbor node pairs (v_{z_1} and v_{z_2}) reduces as they move away from the original node pairs (v_x,v_y). SR can be explained in terms of a random walk process, which is, $s^{SR}_{(v_x,v_y)}$ measures how long the two random walkers are expected to meet on a particular node, starting with the v_x and v_y nodes. Its applicability is constrained on large networks due to its computational complexity [47,60].

3.2.4. Pseudo-Inverse of the Laplacian Matrix (PLM)

Using Laplacian matrix $L = D - A$ rather than Adjacency matrix A gives an alternative representation of a graph, where D is the diagonal matrix of vertex degrees [61] ($D_{i,j} = 0$ and $D_{i,i} = \sum_j A_{i,j}$). The Moore–Penrose pseudo-inverse of the Laplacian matrix, represented by L^+, can be used in the calculation of proximity measures [62]. Because PLM is calculated as inner product cosine similarity, it is also called "cosine similarity time" in the literature [47], and can be calculated as:

$$s^{PLM}_{(v_x,v_y)} = \frac{L^+_{(v_x,v_y)}}{\sqrt{L^+_{(v_x,v_x)} L^+_{(v_y,v_y)}}}. \tag{29}$$

3.2.5. Hitting Time (HT) and Average Commute Time (ACT)

Motivated by random walk, as introduced by mathematician Karl Pearson [63], HT is defined as the average number of steps to be taken by a random walker starting from v_x to reach node v_y. Because HT is not a symmetric metric, one may consider using ACT, which is defined as the average number of steps to be taken by the random walker starting from v_x to reach the node v_y, and that from v_y to reach node v_x. Therefore, HT can be computed by:

$$s^{HT}_{(v_x,v_y)} = 1 + \sum_{v_z \in \Gamma(v_x)} P_{v_x,v_z} s^{HT}_{(v_z,v_y)}. \tag{30}$$

Here, $P_{i,j} = D^{-1}A$, where A and D are the adjacency and the diagonal matrix of vertex degrees [47]. Accordingly, ACT can be formulated as:

$$s^{ACT}_{(v_x,v_y)} = s^{HT}_{(v_x,v_y)} + s^{HT}_{(v_y,v_x)}. \tag{31}$$

For the sake of computational simplicity, ACT can be computed in a closed form using the pseudo-inverse of the Laplacian matrix of the graph, as follows [62]:

$$s^{ACT}_{(v_x,v_y)} = m(L^+_{(v_x,v_x)} + L^+_{(v_y,v_y)} - 2L^+_{(v_x,v_y)}). \tag{32}$$

One challenge of HT and ACT is that it gives very small proximity measures when the terminal node has high stationary probability π_{v_y}, regardless of the identity of the starting node. This problem can be solved by normalizing the scores as $-s^{HT}_{(v_x,v_y)}.\pi_{v_y}$ and $-(s^{HT}_{(v_x,v_y)}.\pi_{v_y} + s^{HT}_{(v_y,v_x)}.\pi_{v_x})$, respectively [37].

3.2.6. Rooted PageRank (RPR)

PageRank (PR) is the metric that is used by Google Search in order to determine the relative importance of the webpages by treating links as a vote. Motivated by PR, RPR defines that the rank of a node is proportional to the likelihood that it can be reached through a random walk [47], such that:

$$s^{RPR}_{(v_x,v_y)} = (1 - \beta)(1 - \beta P_{v_x,v_y})^{-1}. \tag{33}$$

Here, $P_{i,j} = D^{-1}A$, where A is the adjacency matrix and D is the diagonal matrix of vertex degrees. It should be noted that one can calculate the PR by averaging the columns of RPR [7].

3.2.7. Escape Probability (EP)

The metric, which can be derived from RPR, measures the likelihood that the random walk starting from node v_x visits node v_y before coming back to the node v_x again [64]. Let $Q(v_x, v_y)$ be equal to $(1 - \beta D^{-1} A)^{-1} = s^{RPR}_{(v_x,v_y)} / (1 - \beta)$; the equation of EP can be written, as follows [7]:

$$s^{EP}_{(v_x,v_y)} = \frac{Q(v_x, v_y)}{Q(v_x, v_x).Q(v_y, v_y) - Q(v_x, v_y).Q(v_y, v_x)}. \tag{34}$$

3.2.8. Random Walk with Restart (RWR)

In a random walk (RW) algorithm, the probability vector of reaching a node starting from the node v_x can be defined as:

$$\vec{p_{v_x}}(t) = M^T \vec{p_{v_x}}(t-1), \tag{35}$$

where M is called the transition probability matrix, and it can be calculated by $A_{i,j} / \sum_k A_{i,k}$, where A is the adjacency matrix [65]. Because RW does not yield a symmetric matrix, the metric of RWR, very similar to RPR, looks for the probability that a random walker starting from node v_x visits node v_y and comes back to the initial state node v_x at the steady-state, such that:

$$s^{RW}_{(v_x,v_y)} = \vec{p_{v_x}}^{v_y} + \vec{p_{v_y}}^{v_x}. \tag{36}$$

3.2.9. Maximal Entropy Random Walk (MERW)

The basic MERW algorithm, which is based on the maximum uncertainty principle, was proposed as a result of the necessity in order to define uniform path distribution in Monte Carlo simulations [66]. However, its applications on the stochastic models are very recent [67]. Li et al. [68] proposed MERW, which maximizes the entropy of a random walk, as follows:

$$\lim_{l \to \infty} \frac{-\sum_{A^t_{v_x v_y} \in A^t} p(A^t_{v_x v_y}) \ln p(A^t_{v_x v_y})}{t}. \tag{37}$$

Here, $p(A^t_{v_x v_y})$ is the multiplication of the iterative transition matrices $(M_{v_x v_z}.M_{v_z v_q}...M_{v_q v_y})$, where M_{ij} can be calculated, as follows:

$$M_{v_i v_j} = \frac{A_{v_i v_j}}{\lambda} \frac{\psi_{v_j}}{\psi_{v_i}}, \tag{38}$$

where A is the adjacency matrix and ψ is the normalized eigenvector with normalization constant λ [6].

3.2.10. The Blondel Index (BI)

The index is proposed by Blondel et al. [69] in order to measure the similarity for the automatic extraction of synonyms in a monolingual dictionary. Although BI is used to quantify the similarity between two different graphs, Martinez et al. show that investigating the similarity of two vertices in a single graph can also be evaluated in an iterative manner, as:

$$S(t) = \frac{AS(t-1)A^T + A^T S(t-1)A}{||AS(t-1)A^T + A^T S(t-1)A||_F}, \tag{39}$$

where $S(t)$ refers to the similarity matrix in iteration t and $S(0) = I$. $||M||_F$ is the Frobenius matrix norm and it can be calculated, as follows:

$$||M_{m \times n}||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} M^2_{i,j}}. \tag{40}$$

The similarity metric is obtained when $S(t)$ is converged, such that $s^{BI}_{(v_x,v_y)} = S_{v_x,v_y}(t = c)$, where $t = c$ denotes the steady state level.

3.3. Quasi-Local Similarity-Based Approaches

The trade-off between the efficiency of the information regarding the whole network topological structure for the global approaches and the less time complex algorithms for the local-based methods have resulted in the emergence of quasi-local similarity-based methods for link prediction. Similarly, these approaches are limited in the calculation of the similarity between arbitrary node pairs. However, quasi-local similarity methods provide an opportunity for computing the similarity between a node and the neighbors of its neighbors. Although some of the quasi-local similarity-based methods consider the whole topology of the network, their time complexity is less than that of global similarity-based approaches.

3.3.1. The Local Path Index (LPI)

The index, which is very similar to the well-known approaches KI and CN, considers the local path with a wider perspective by not only employing the information of the nearest neighbors, but also the next two and three nearest neighbors [46,70], such that:

$$S^{LP} = A^2 + \beta A^3, \tag{41}$$

where A is the adjacency matrix, β is a free parameter to adjust the relative importance of the neighbors within the length $l = 2$ distances and length $l = 3$ distances. The metric can also be extended for the higher orders as:

$$S^{LP^{(L)}} = \sum_{l=2}^{L} \beta^{l-2} A^l. \tag{42}$$

The neighbors within the length of three distances are preferable due to increasing complexity in the higher orders of LP. One can easily realize that this similarity matrix simplifies to CN when $l = 2$ and may produce a very similar result to KI given low β values without the inverse transform process. The similarity between two nodes can be evaluated via $s^{LP}_{(v_x,v_y)} = S^{LP}_{v_x,v_y}$.

3.3.2. Local (LRW) and Superposed Random Walks (SRW)

Although the random walk-based algorithms perform well, the sparsity and computational complexity regarding massive networks are challenging for these algorithms. Thus, Liu and Lü proposed the LRW metric [71], in which the initial resources for the random walker are assigned based on their importance in the graph. LRW considers the node degree as an important feature and it does not concentrate on the stationary state. Instead, the number of iterations is fixed in order to perform a few-step random walk. LRW can be formulated, as:

$$s^{LRW}_{(v_x,v_y)}(t_c) = \frac{|\Gamma(v_x)|}{2|E|} p^{\vec{v}_x}_{v_y}(t_c) + \frac{|\Gamma(v_y)|}{2|E|} p^{\vec{v}_y}_{v_x}(t_c). \tag{43}$$

Because superposing all of the random walkers starting from the same nodes may help to prevent the sensitive dependency of LRW to the farther neighboring nodes, SRW is proposed as:

$$s^{SRW}_{(v_x,v_y)}(t_c) = \sum_{t=1}^{t_c} s^{LRW}_{(v_x,v_y)}(t). \tag{44}$$

3.3.3. Third-Order Resource Allocation Based on Common Neighbor Interactions (RACN)

Motivated by the RA index, Zhang et al. [72] proposed RACN, in which the resources of nodes are allocated to the neighbors as:

$$s^{RACN}_{(v_x,v_y)} = \sum_{v_z \in \Gamma(v_x) \cap \Gamma(v_y)} \frac{1}{|\Gamma(v_z)|} + \sum_{e_{v_i,v_j} \in E, |\Gamma(v_j)| < |\Gamma(v_i)|} \left(\frac{1}{|\Gamma(v_i)|} - \frac{1}{|\Gamma(v_j)|}\right), \tag{45}$$

where $v_i \in \Gamma(v_x)$ and $v_j \in \Gamma(v_j)$. The superiority of the RACN over the original RA has been shown in [25] while using varying datasets.

3.3.4. FriendLink Index (FL)

The similarity of two nodes is determined according to the normalized counts of the existing paths among the corresponding nodes with varying length L. The formulation for the FL index is as follows:

$$s^{FL}_{(v_x,v_y)} = \sum_{l=1}^{L} \frac{1}{l-1} \frac{|A^{l}_{v_x,v_y}|}{\prod_{j=2}^{l} (|V| - j)}, \tag{46}$$

where $|V|$ is the number of vertices in the graph. The metric is favorable, due to its high performance and speed [73].

3.3.5. PropFlow Predictor Index (PFP)

PFP is a metric that is inspired by Rooted PageRank, and it simply equals the probability that the success of random walk starts from node v_x and terminates at node v_y in not more than l steps [74]. This restricted random walk selects the links based on weights, denoted as ω [47], such that:

$$s^{PFP}_{(v_x,v_y)} = s^{PFP}_{(v_a,v_x)} \frac{\omega_{v_x v_y}}{\sum_{v_z \in \Gamma(v_x)} \omega_{v_x v_y}}. \tag{47}$$

The most important superiority of PFP is its widespread use in directed, undirected, weighted, unweighted, sparse, or dense networks.

4. Probabilistic Methods

Probabilistic models are supervised models that use Bayes rules. The most important drawback of some of these models is their being slow and costly for large networks [24]. In the following, we introduce the five most important probabilistic methods of link prediction.

4.1. Hierarchical Structure Model

This model was developed based on the observation that many real networks present a hierarchical topology [75]. This maximum likelihood-based method searches for a set of hierarchical representations of the network and then sorts the probable node pairs by averaging over all of the hierarchical representations explored. The model was first proposed in the work of [76], in which it develops a hierarchical network model that can be represented by a dendrogram, with $|N|$ leaves and $|N-1|$ internal nodes. Each leaf is a node from the original network and each internal node represents the relationship of the descendent nodes in the dendrogram. A value of p_r is also attributed to each internal node r, which represents the probability with which a link exists between the branches descending from it. If D is a dendrogram that represents the network, the likelihood of dendrogram with a set of internal node probabilities (p_r) is:

$$\mathcal{L}(D, \{p_r\}) = \prod_{r \in D} p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}. \tag{48}$$

In the above equation, E_r is the number of links that connect nodes that have a node r as their lowest common ancestor in D. L_r and R_r represent the number of leaves in the left and right subtrees that are rotted in r, respectively. Setting $p_r^* = \frac{E_r}{L_r R_r}$ maximizes the likelihood function (48). Replacing p_r with p_r^* in the likelihood function, the likelihood of a dendrogram at its maximum can be calculated by:

$$\mathcal{L}(D) = \prod_{r \in D} \left[(1 - p_r^*)^{1-p_r^*} p_r^{*p_r^*} \right]^{L_r R_r} . \tag{49}$$

These equations are then utilized to perform link prediction. After a Markov Chain Monte Carlo method is used to sample a large number of dendrograms with probabilities proportional to their likelihood, the connection probability between two nodes v_i and v_j is estimated by averaging over all the sampled dendrograms. This task is performed for all sampled dendrograms and, subsequently, the node pairs are sorted based on the corresponding average probabilities. The higher the ranking, the more likely that the link between the node pair exists. A major drawback of the hierarchical structural model is its computational cost and being very slow for a network consisting of a large set of nodes.

4.2. Stochastic Blockmodel

Stochastic block models are based on the idea that nodes that are heavily interconnected should form a block or community [77]. In a stochastic block model, nodes are separated into groups and the probability that two nodes are connected to each other is merely dependent on the group to which they belong [78]. Stochastic block models have been successfully applied to model the structure of complex networks [79,80]. They have also been utilized to predict the behavior in drug interactions [81]. The work of [82] uses a block model in order to predict conflict between team members. Ref. [83] also utilizes a stochastic block model in order to develop a probabilistic recommender system.

As noted above, the probability that two nodes i and j are connected depends on the blocks that they belong to. A block model $M = (P, Q)$ is completely determined by the partition P of nodes into groups and the matrix Q of probabilities of linkage between groups. While numerous partitions (models) can be considered for a network, the likelihood of a model A^O can be calculated by the following [78,84]:

$$\mathcal{L}(A^O|P, Q) = \prod_{\alpha \leq \beta} Q_{\alpha\beta}^{l_{\alpha\beta}^O} (1 - Q_{\alpha\beta})^{r_{\alpha\beta} - l_{\alpha\beta}^O}. \tag{50}$$

In Equation (50), $l_{\alpha\beta}^O$ is the number of links in A^O between nodes in groups α and β of P, and $r_{\alpha\beta}$ is the maximum number of links possible, which is $|\alpha||\beta|$ when $\alpha \neq \beta$ and $\binom{|\alpha|}{2}$ when $\alpha = \beta$. Setting $Q_{\alpha\beta}^* = \frac{l_{\alpha\beta}^O}{r_{\alpha\beta}}$ maximizes the likelihood function (50). By applying Bayes theorem, the probability (reliability) of a link with maximum likelihood can be computed.

Similar to the hierarchical structure model that is discussed in Section 4.1, a significant shortcoming of this method is that it is very time-consuming. While the Metropolis algorithm [85] can be utilized to sample partitions, this approach is still impractical for a large network. An example of blockmodel likelihood calculation is illustrate in Figure 3.

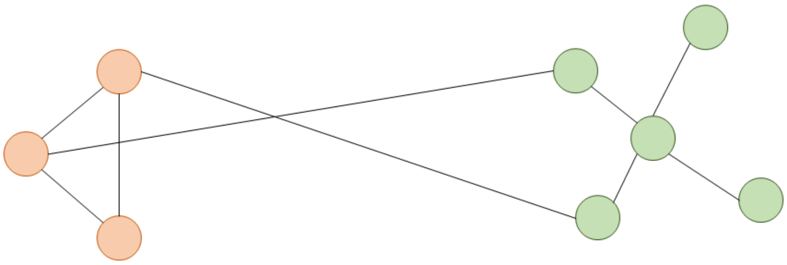

Figure 3. An example of Blockmodel likelihood calculation Here, a probable partitioning is presented. The block on left is α and the block on right is β. $Q^*_{\alpha\alpha} = 1$, $Q^*_{\alpha\beta} = \frac{2}{12}$, and $Q^*_{\beta\beta} = \frac{5}{6}$. Hence, likelihood is calculated, as follows: $1^3 \times 1 \times \frac{2}{15}^2 \times \frac{13}{15}^{13} \times \frac{1}{2}^5 \times \frac{1}{2}^5 \approx 2.701 \times 10^{-6}$.

4.3. Network Evolution Model

Ref. [86] proposed a network topology based model for link prediction. In this model, probabilistic flips of the existence of edges are modeled by a "copy-and-paste" process between the edges [86]. The problem of link prediction is defined, as follows: the data domain is represented as a graph $G = (V, s)$, where V is the set of nodes of the network and $s : V \times V \rightarrow [0, 1]$ is an edge label function. $s(v_i, v_j)$ indicates the probability that an edge exists between i and j. $s^{(t)}$ shows the edge label function at time t, and its of Markovian nature, i.e., $s^{(t+1)}$ only depends on $s^{(t)}$. The fundamental idea behind the proposed network edge label copy-and-paste mechanism is that, if a node has a strong influence on another node, the second nodes association will be highly affected by the second node. The probability of an edge existing between nodes i and j at time $t + 1$ is as follows:

$$\begin{aligned} s^{t+1}(v_i, v_j) = & \frac{1}{|V| - 1}\Big(\sum_{k \neq i,j} w_{v_k v_j} s^{(t)}(v_k, v_i) + w_{v_k v_i} s^{(t)}(v_k, v_j)\Big) + \\ & \Big(1 - \frac{1}{|V| - 1} \sum_{k \neq i,j} w_{v_k v_j} + w_{v_k v_i}\Big) s^{(t)}(v_i, v_j), \end{aligned} \tag{51}$$

where $w_{v_k v_j}$ is the probability that an edge label is copied from node v_k to node v_j. In Equation (51), the first term represents the probability that the edge label for (v_i, v_j) is changed by copy and pasting. The second term represents when the same edge label is unchanged. The linkages are obtained by iteratively updating Equation (51) until convergence. The objective function according to which the parameters are set is solved by an expectation maximization type transductive learning.

4.4. Local Probabilistic Model

The work of [87] proposed a local probabilistic model for link prediction, in which the focus of the original paper is particularly in the context of evolving co-authorship networks. Given the candidate link, e.g., nodes v_i and v_j, first, the central neighborhood set of v_i and v_j are determined, which is the set of nodes that are the most relevant to estimating the co-occurrence probability. The central neighborhood sets are chosen from the nodes that lie along paths of shorter length between v_i and v_j. Ref. [87] proposes an algorithm in order to determine central neighborhood set, which is, as follows: first, collecting all of the nodes that lie on length-2 simple paths, then those on length-3 simple paths, and so on. The paths are then ordered based on the frequency scores and the ones with the highest scores are chosen [87]. A path length threshold is also considered for the sake of decreasing computational cost ([87] proposes a threshold of 4 for their specific problem). Next, they form a transaction dataset that is formed by a chronological set of events (co-authoring articles). A non-derivable itemset mining is performed on this dataset, which results in all non-redundant itemsets along with their frequencies. In the end, a Markov Random Field (MRF) graph model is trained while using the derived dataset. The resulting final model gives the probability of the existence of each link v_i and v_j.

4.5. Probabilistic Model of Generalized Clustering Coefficient

This method that was proposed by [88] focuses on analyzing the predictive power of clustering coefficient [88]. The generalized clustering coefficient $C(k)$ of degree k is defined as [88]:

$$C(k) = \frac{\text{number of cycles of length } k \text{ in the graph}}{\text{number of paths of length } k} \tag{52}$$

As explained in [88], generalized clustering coefficients describe the correlation between cycles and paths in a network. Therefore, the probability of formation of a particular link is determined by the number of cycles (of different lengths) that will be constructed by adding that link [88]. The concept of *cycle formation model* is explained, as follows: a cycle formation model of degree k ($k \geq 1$) is governed by k link generation mechanisms, $g(1), g(2),..., g(k)$, which are each described by $c_1, c_2,..., c_k$. If $P_{v_iv_jk}$ shows a path from v_i to v_j with length k, then $c_k = P((v_i, v_j) \in E||P_{v_iv_jk}| = 1)$ (the probability that there is a link between i and j, given that there is one path of length k between them). We know that, if there is more than one path with length k from v_i to v_j, then the probability that there is a link between them increases (see Figure 4 for instance). Therefore:

$$\begin{aligned} &\text{if } \mathrm{P}((v_i, v_j) \in E||P_{v_i,v_j,k}| = 1) = c_k \ \& \ |P_{v_i,v_j,k}| = m \ \rightarrow \\ &\mathrm{P}((v_i, v_j) \in E||P_{v_i,v_j,k}| = m) = \frac{c_k^m}{c_k^m + (1 - c_k)^m} \end{aligned} \tag{53}$$

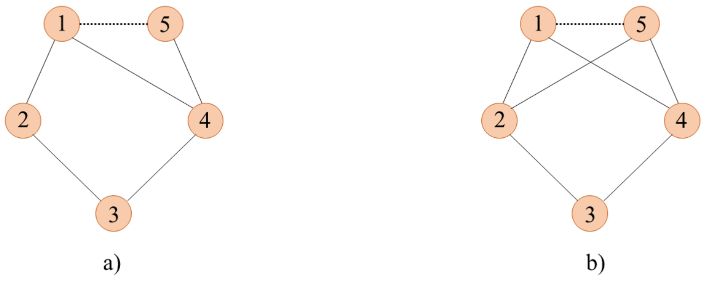

Figure 4. (**a**) There is only one path of length two from node 1 to node 5 in the network and $\mathrm{Pr}_2((1, 5) \in E) = c_2$. (**b**) There are two paths of length 2 from the node 1 to node 5, therefore $\mathrm{Pr}_2((i, j) \in E) = \frac{c_2^2}{c_2^2+(1-c_2)^2}$.

Because of the fact that the total link occurrence probability between v_i and v_j is a result of the effect of multiple mechanisms of cycle formation model of degree k ($CF(k)$) is calculated by:

$$\begin{aligned} &P_{m_2,...,m_k} = \mathrm{P}((v_i, v_j) \in E||P_{v_iv_j2}| = m_2, ..., |P_{v_iv_jk}| = m_k) = \\ &\frac{c_1 c_2^{m_2} ... c_k^{m_k}}{(c_1 c_2^{m_2} ... c_k^{m_k}) + (1 - c_1)(1 - c_2)^{m_2} ... (1 - c_k)^{m_k}} \end{aligned} \tag{54}$$

5. Relational Models

One drawback of the previously mentioned methods is that they do not incorporate vertex and edge attributes to model the joint probability distribution of entities and links that associate them [24]. Probabilistic Relational Models (PRM) [21] is an attempt to use the rich logical structure of the underlying data that is crucial for complicated problems. One major limitation of Bayesian networks is the lack of the concept of an "object" [22]. Bayesian PRMs [20,21] include the concept of an object in the context of Bayesian networks, in which each object can have their attributes and relations exist between objects and their attributes. Figure 5 is an example of a schema for a simple domain. A relational model consists of a set of classes, $\mathrm{Y} = \{Y_1, Y_2, ..., Y_n\}$. In Figure 5, $\mathrm{Y} = \{\text{Journalist}, \text{Newspaper}, \text{Reader}\}$.

Each class also contains some descriptive attributes, the set of which is shown with $A(Y)$. For example, Journalist has attributes Popularity, Experience, and Writing skills. In order for objects to be able to refer to other objects, each class is also associated with a set of reference slots, which is shown by $Y.\rho$. Slot chains also exist, which are references between multiple objects (similar to $f(g(x))$). $Pa(Y.A)$ shows the set of parents of $Y.A$. For instance, in Figure 5, a journalist's Popularity depends on her Experience and Writing skills. Dependency can also be a result of a slot chain, meaning that some attributes of a class depend on some attributes of another class. The joint probability distribution in a PRM can be calculated, as follows [22]:

$$P(I|\sigma_r, S, \theta_S) = \prod_{Y_i} \prod_{A \in A(Y_i)} \prod_{y \in \sigma_r(Y_i)} P(I_{y.A}|I_{Pa(y.A)}) \tag{55}$$

In Equation (55), I shows an instance of a schema S, which specifies for each class Y, the set of objects in the class, a value for each attribute $y.A$, and a value for each reference slot $y.\rho$. Additionally, σ_r is a relational skeleton, which denotes a partial specification of an instance of a schema, and it specifies the set of objects for each class and the relations that hold between the objects [22].

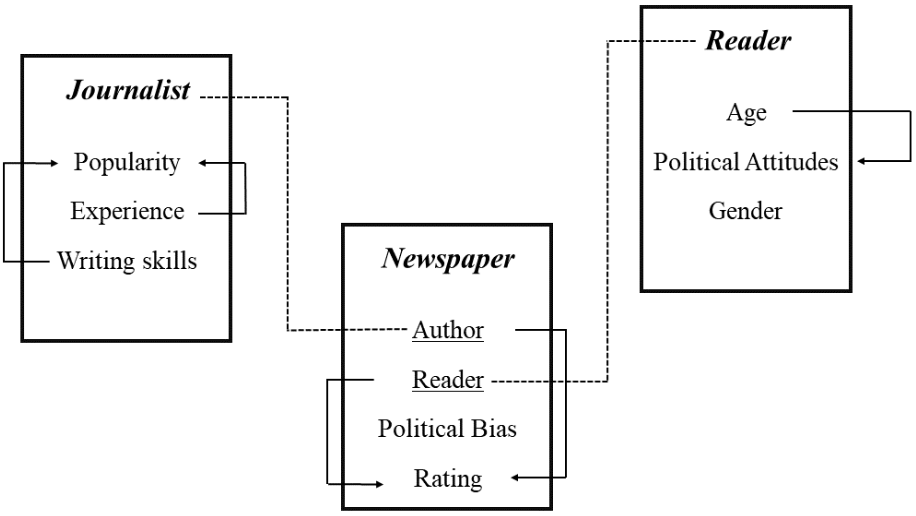

Figure 5. An example of a relational schema for a simple domain. The underlined attributes are reference slots of the class and the arrows show the types of objects to which they are referring.

The task of link prediction can then be performed by considering the probability of the existence of a link between two objects in the relational model [23]. The work of [89] shows that deriving the distribution of missing descriptive attributes will benefit from the estimation of link existence likelihood. Besides, a Relational Bayesian Network, in which the model graph is a directed acyclic graph, the Relational Markov Network is also proposed [90,91], in which the graph model is an undirected graph and it can be utilized for the task of link prediction. Relational Markov Networks address two shortcomings of directed models: They do not constrain the graph to be acyclic, which allows for various possible graph representations. Additionally, they are well suited for discriminative training [92].

There exist other relational models for the task of link prediction. The DAPER model is a directed acyclic type of probabilistic entity-relationship model [93]. The advantage of the DAPER model is being more expressive than the aforementioned models [94]. Other Bayesian relational models in the literature include stochastic relational model [95], which models the stochastic structure of entity relationships by a tensor of multiple Gaussian processes [28], relational dependency network [96,97], and parametric hierarchical Bayesian relational model [98].

6. Learning-Based Methods

The feature extraction-based methods that are discussed earlier in this paper provide a starting point for the systematic prediction of missing or future associations available through learning the effective attributes. Among these effective features for link prediction, employing the topological attributes that can be extracted from the graph structure is the foundation of all learning-based link prediction algorithms, from which the pair-wise shortest distance attribute is the most common topological feature. Besides the topological attributes, some machine learning models benefit from the node and domain specific attributes, referred to as the aggregated and proximity features, respectively [99].

The introduction of supervised learning algorithms to the problem of link prediction led to the state-of-the-art models that achieve high prediction performances [100]. These models view the problem of link prediction as a classification task. In order to approach the link prediction problem, supervised models are supposed to tackle a few challenges, including the unbalanced data classes that result from the sparsity property of real networks, and the extraction of the topological, proximity, and aggregated attributes as independent informative features [101]. There is extensive literature on the classification models for link prediction, including the application of traditional machine learning methods into this field of research. Support Vector Machines, K-nearest Neighbors, Logistic Regression, Ensemble Learning, and Random Forrest, Multilayer Perceptron, Radial Basis Function network, and Naive Bayes are just a few of the supervised learning methods that are extensively used in link prediction. A comparison between a few of these supervised methods has been presented in [99], where, surprisingly, SVM with RBF kernel is reported to be very successful in the accuracy and low squared error of the model.

Although the traditional machine learning models for link prediction rely on user-defined feature encoding, the evolution of these models has led to the generation of automatic feature encoders, which prevent hand-engineered attributes [101]. These models aim to learn graph encoding, node, and/or domain-related features into low-dimensional space, and are referred to as representation learning or graph embedding-based models for link prediction. These methods can be trained while using neural networks or dimensionality reduction algorithms [102]. The applications of graph analysis and representation learning has led to the development of advanced language models that focus on language understanding, relation discovery, and question answering. Knowledge graphs, which represent sequences of relations between named entities within a textual content, are being widely investigated for the task of link prediction, relation prediction, and knowledge graph completion [103]. Although many of the reviewed methods in this survey are applicable to different applications and graph types, knowledge graphs and their embedding methods are dependent to directed relationships. Examples of recent methods for knowledge graphs are Relational Graph Convolutional Neural Networks (R-GCN) [104], which are able to extract features from a given data and, accordingly, generate a directed multigraph, label node types, and their relationships in the generated graph, and, finally, generate a latent knowledge-based representation that can be used for node classification as well as link prediction. Other language models, such as Bidirectional Encoder Representations from Transformers (BERT) [18], which use pre-trained language models, and their variations, including Knowledge Graph BERT (KG-BERT) [105] and Knowledge-enabled BERT (K-BERT) [103], can extract node and relation attributes for knowledge graph completion and link prediction [16]. A comprehensive review on embedding methods that are designed for knowledge graphs is available in [3].

The tasks of vertex representation learning and vertex collocation profiling (VCP) for the purpose of topological link analysis and prediction were introduced in [106,107], respectively. Comprehensive information on the surrounding local structure of embedded pairs of vertices v_x and v_y in terms of their common membership in all possible subgraphs of n vertices over a set of r relations is available from their VCP, written as $VCP_{x,y}^{n,r}$, and the VCP elements are closely related to isomorphic subgraphs. Thus, this method helps in the understanding of link formation mechanism from the nodes and graph representation.

Mapping the graph to a vector space is also known as encoding. On the contrary, the reconstruction of the node neighborhood from the embedded graph is referred to as decoding. Graph representation can be learned via supervised or unsupervised methods while using an appropriate optimization algorithm in order to learn the embeddings [101]. This mapping can be defined for graph G = < V, E > as $f : v_x \rightarrow v_{x'} \in \mathbb{R}^d, \forall x \in [n]$, such that $d \ll \mid V \mid$, where n denotes the total number of vertices, v_x is a sample node that has been embedded to d-dimensional vector space, and the embedded node is represented by $v_{x'}$. Figure 6 illustrates the procedure of node and graph representation.

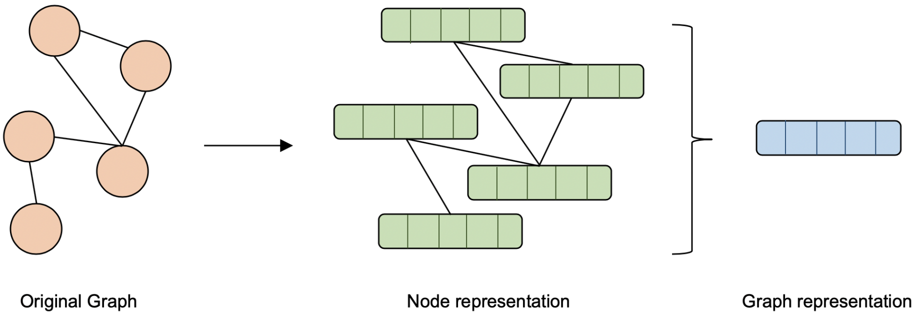

Figure 6. An example of node and graph representation. Here the node representation vectors are aggregated to generate a single graph representation.

Representation learning algorithms for the task of link prediction can be divided into categories based on their decoder function, a similarity measure for graphs, and the loss function in the models [101]. Therefore, we categorize these methods into

(i) Matrix Factorization-Based Models,
(ii) Path and Walk-Based Models, and
(iii) Deep Neural Network-Based Methods.

6.1. Matrix Factorization-Based Methods

These methods are able to extract latent features with additional graph features for link prediction. In these models, the vector representation of the topology-related features produces an N-dimensional space, where $N = |V|$ is the number of vertices in the network. The main purpose of matrix factorization-based methods is to reduce the dimensionality while also preserving the nonlinearity and locality of the graph via employing deterministic measures of node similarity in the graph. However, the global structure of the graph topology may be generally lost [108].

SVD is one of the commonly used methods as a result of its feasibility in low-rank approximations [109,110]. Here, the link function $L(.)$ is defined as $G \approx L(U\Lambda U^T)$, where $U \in \mathbb{R}^{|V|\times k}, \Lambda \in \mathbb{R}^{k\times k}$, and k denotes the number of latent variables in SVD. The similarity $s(v_x, v_y)$ between the node pairs v_x and v_y is defined by $L(u_{v_x}^T \Lambda u_{v_y})$.

In [111], a latent feature learning method for link prediction has been proposed by defining a latent vector $\overrightarrow{l_{v_x}}$ and a feature vector $\overrightarrow{a_{v_x}}$ for each node v_x, a weight vector W_v for node features, a weight vector $\overrightarrow{w_e}$ for edge features, and a vector of features $\overrightarrow{b_{v_x,v_y}}$ for each edge. This model computes the prediction of edge formation as:

$$s(v_x, v_y) = \frac{1}{1 + exp(-\overrightarrow{l_{v_x}}^T F \overrightarrow{l_{v_y}} - \overrightarrow{a_{v_x}}^T W_v \overrightarrow{a_{v_y}} - \overrightarrow{w_e}^T \overrightarrow{b_{v_x,v_y}})}, \tag{56}$$

where, F is the scaling factor for each edge.

The inner-product-based embedding models for link prediction embed the graph based on a pairwise inner-product decoder, such that the node relationship probability is proportional to the dot product of node embeddings:

$$DEC(v_{x'}, v_{y'}) = v_{x'}^T v_{y'}, \tag{57}$$

$$\mathcal{L} = \sum_{(v_x,v_y)\in D} ||DEC(v_{x'}, v_{y'}) - s_G(v_x, v_y)||_2^2. \tag{58}$$

Graph Factorization (GF) [112], GraRep [113], and HOPE [110] algorithms are examples of the inner-product-based methods for link prediction. Graph factorization model partitions the graph by minimizing the number of neighboring nodes, rather than applying edge cuts, as the storage and exchange of parameters for the latent variable models and their inference algorithms are related to nodes. HOPE [110] focuses on the representation and modeling of directed graphs, as directed associations can represent any type of graph. This model preserves the asymmetric transitivity for directed graph embeddings. The asymmetric transitivity property captures the structure of the graph by keeping the correlation between the directed edges, such that the probability of the existence of a directed edge from v_x to v_y is high if a directed edge exists for the opposite direction. HOPE supports classical similarity measures as proximity measurements in the algorithm, including the Katz Index (KI), Rooted PageRank (RPR), Common Neighbors (CN), and Adamic-Adar (AA).

6.2. Path and Walk-Based Methods

The developed models for link prediction that are designed based on random walk statistics prevent the need for any deterministic similarity measures. In these algorithms, similar embeddings are being produced for nodes that co-occur on graph short random walks. These algorithms investigate the node features, including node centrality and similarity via graphs exploration and sampling with random walks or search algorithms, such as Breadth First Search (BFS) and Depth First Search (DFS) [114]. The random walk-based models for graphs can be divided into many different categories, according to varying perspectives. One possible division for these models includes categorization that is based on their embedding output, for instance, local structure-preserving methods, global structure-preserving methods, and the combination of the two [115].

Representations with BFS provide information regarding the similarity of nodes in the case of their roles in the network, for instance, representing a hub in the graph [102]. On the contrary, random walks with DFS can provide information regarding the communities that nodes belong to. These algorithms have been recently applied along with generative models to introduce edges and nodes directly to the graph [116]. Community aware random walk for network embedding (CARE), as introduced in [117], is another approach for the task of link prediction and multi-label classification. This model builds customized paths that are based on local and global structures of network, and uses the Skip-gram model to learn representation vectors of nodes.

In comparison to walk-based methods, link prediction that is based on meta path similarity has been introduced in [118], which operates a similarity search among the same type of nodes. Thus, meta path-based methods extend link prediction to heterogeneous networks with different types of vertices. In this model, a meta path refers to a sequence of relations between object types and defines a new composite relation between its starting type and ending type. The similarity measure between two objects can be defined according to random walks used in P-PageRank, pairwise random walk used in SimRank, P-PageRank, or SimRank on the extracted sub-graph or, finally, using PathSim, which captures the subtle semantics of similarity among peer objects [118]. PathSim calculates the similarity of two peer objects as:

$$s(v_x, v_y) = \frac{2 \times |\{p_{v_x,v_y} : p_{v_x,v_y \in \mathcal{P}}\}|}{|\{p_{v_x,v_x} : p_{v_x,v_x \in \mathcal{P}}\}| + |\{p_{v_y,v_y} : p_{v_y,v_y \in \mathcal{P}}\}|}, \tag{59}$$

where $\mathcal{P}$ refers to the meta path defined on the graph of network schema, p_{v_x,v_y} is a path instance between v_x and v_y, and p_{v_x,v_x} and p_{v_y,v_y} are the same concept for vertices v_x and v_y. An application of using meta-path for link prediction is in [119], which predicts drug target interactions (DTI) on the observed topological features of a semantic network in the context of drug discovery.

6.3. Neural Network-Based Methods

In order to avoid strong assumptions for every heuristic related to node similarities and edge formation, link prediction algorithms that are based on neural networks have been proposed that automatically learn a suitable heuristic from a given network. In [25], a mapping function for the subgraph patterns to link existence is being learned by extracting a local subgraph around each target link. Thus, this model automatically learns a "heuristic" that suits the graph. The powerful capabilities and simplicity of using neural network-based methods have led to the generation of a family of complex encoder-decoder-based representation learning models, such as Graph Neural Networks (GNNs) [120,121] and Graph Convolutional Neural Networks (GCNs) [104,122–124].

Although the general concept of graph neural networks was first presented in [121], many neural network-based algorithms for representation learning and link prediction have been proposed, including SEAL [25], which uses GNNs to learn general graph structure features for link prediction from local enclosing subgraphs. Besides models that consider graph structure features, latent and explicit features are also investigated in the literature for link prediction. Furthermore, efficient strategies for capturing multi-modality for graphs, for instance, node heterogeneity, have been originated from neural network-based models. Another extension for graph embedding methods that have become achievable by neural networks, is the embedding of subgraphs $(S \subset V)$. The attribute aggregation procedure in different neural network architectures may vary according to their connection types, and the usage of filters or gates in the propagation step of the models [115].

In order to learn the information on the neighboring nodes, GNNs [121] aim to learn a state embedding $h_{v_x} \in \mathbb{R}^s$ iteratively, where s is the dimension for the vector representation of node v_x. By stacking the states for all of the nodes, the constructed vectors H, and the output labels O can be represented as:

$$H = F_g(H, X), \tag{60}$$

$$O = O_g(H, X_N), \tag{61}$$

where F_g is the global transition function, O_g is the global output function, X refers to the feature vector, and X_N stands for the feature vector for all nodes. The updates per iteration can be defined as:

$$H^{t+1} = F(H^t, X), \tag{62}$$

where t denotes the t_th iteration. In this algorithm, the learning of the representations can be achieved by a supervised optimization method, such as the gradient-descent method.

The SEAL [115] algorithm that has been designed for the task of link prediction considers enclosing subgraph extraction for a set of sampled positive (observed) and negative links in order to prepare the training data for GNN and uses that information to predict edge formations. The GNN model receives the adjacency matrix (A) and node information matrix (X) as input, where each row of X corresponds to a feature vector of a vertex. The process of X preparation for each enclosing subgraph includes three components of structural node labels based on Double-Radius Node Labeling (DRNL), node embeddings, and node attributes. Another neural network-based model for the task of link prediction is HetGNN [120], which considers heterogeneous networks. This model starts with a random walk with restart strategy and samples a fixed size of correlated heterogeneous neighbors to group them based upon node types. Subsequently, neural network architecture with two modules is used in order to aggregate feature information of sampled neighboring vertices. The deep feature interactions of heterogeneous contents are captured by the first module, which generates content embedding for each vertex. The aggregation of content embeddings of different neighboring types is being done by the second module. HetGNN combines these outputs in order to obtain the final node embedding.

Multi-layer Perceptrons (MLPs) are neural network-based representation learning algorithms that approach graph embedding via message passing, in which information flows from the neighboring nodes with arbitrary depth. Message Passing Neural Networks (MPNNs) [125] further extend GNNs and GCNs by proposing a single framework for variants of general approaches, such as incorporating the edge features in addition to the node features.

Graph Auto-Encoders (GAE) and Variational Graph Auto-Encoders (VGAE) [123] are another category of graph neural networks that aim to learn the node representations in an unsupervised manner. The majority of models based on GAE and its derivations employ Graph Convolutional Networks (GCNs) for the node encoding procedure. Next, these algorithms employ a decoder in order to reconstruct the graph's adjacency matrix A. This procedure can be formally represented as:

$$Z = GCN(X, A), \tag{63}$$

where Z is the convolved attribute matrix. GAEs can learn the graph structures while using deep neural network architectures, and reduce the graph dimensionality in accordance with the number of channels of the auto-encoder hidden layers [126]. Additionally, GAE-based models are able to embed the nodes into sequences with diverse lengths. This benefits the auto-encoders not only to achieve high performances for testing over the unseen node embeddings, but also to aggregate the node attributes in order to improve their prediction accuracy [101]. GC-MC [124] and Adversarially Regularized Graph Auto-Encoders (ARGA) are examples of representation models with auto-encoder architectures [127]. Auto-encoders are also being used without neural network architectures, for instance, LINE [108], DNGR [126], and SDNE [128]. The algorithm in LINE consists of a combination of two encoder-decoder structures to study and optimize the first and second node proximities in the vector space. Both of the DNGR [126] and SDNE [128] algorithms embed the node local neighborhood information while using a random surfing method and approach single embeddings through auto-encoders than pairwise transformations.

Although the graph representation learning models that are based on GNNs consider both graph structures and node features to embed the graph, they suffer from computational complexity and inefficiency in iterative updating of the hidden states. Furthermore, GNNs use the same parameters for all layers, which limits their flexibility. These architectures are always designed as shallow networks with no more than three layers, and including a higher number of layers is still being considered to be a challenge for CNNs [115].

The introduction of neural networks, specially convolutional neural networks, in order to graph structures, has led to extract features from complex graphs flexibly. Graph Convolutional Networks (GCNs) [122] tackle the problem of high computational complexity and shallow architectures via defining a convolution operator for the graph. Furthermore, a rich class of convolutional filter functions can be achieved through stacking many convolution layers. The iterative aggregation of a node's local neighborhood is being used in GCNs to obtain graph embeddings, where this aggregation method leads to higher scalability besides learning graph global neighborhoods. The features for these models include the information from the topology of the network aggregated by the node attributes, when the node features are available from the data domain [115]. Additionally, GCNs can be utilized for node embeddings, as well as subgraph embeddings [101]. Varying convolutional models have been derived from GCNs that employ different convolutional filters in their architecture. These filters can be designed as either spatial filters or spectral filters. The former type of convolutional filters can be directly operated on the original graph and its adjacency matrix; however, the latter type is being utilized on the spectrum of the graph Laplacian [114].

In [129], the problem of link prediction is studied while using a combination of two convolutional neural networks for the graph network of molecules. The molecules are represented as having a hierarchical structure for their internal and external interactions. The graph structure transformation to a low dimensional vector space is obtained from an internal convolutional layer that is randomly initialized for each node representation and trained by backpropagation. The external convolutional layer receives the embedded nodes as input to learn over the external graph representations. Finally, the link prediction algorithm consists of a multilayer neural network, which was accepting the final representations in order to predict the molecule-molecule interactions by a softmax function.

The algorithms that belong to the family of neighborhood aggregation methods, are also being referred to as convolutional models. An example is GraphSAGE [130], which aggregates the information from local neighborhoods recursively, or iteratively. This iterative characteristic leads the model to be generalizable to unseen nodes. The node attributes for this model might include simple node statistics, such as node degrees, or even textual data for profile information on online social networks.

Graph convolutional neural networks for relational data analysis is proposed in [104], which introduces Relational Graph Convolutional Networks (R-GCNs) for the task of link prediction and node classification. Because of relational models referring to directed associations, the node relationships in this models for the graph $G = \langle V, E, R \rangle$ are represented as $(v_x, r, v_y) \in E$, where $r \in R$ is a relation type for both canonical and inverse directions. This model can be considered to be a special case of simple differentiable message-passing model. In this model, the forward-pass update for entity v_x in a relational multigraph can be propagated by:

$$h_{v_x}^{(l+1)} = \sigma\left(\sum_{r \in R} \sum_{v_y \in \Gamma_{v_x}^{r}} \frac{1}{c_{v_x,r}} W_r^{(l)} h_{v_y}^{(l)} + W_0^{(l)} h_{v_x}^{(l)} \right), \tag{64}$$

where $\sigma(.)$ is an element-wise activation function, l denotes the layer of the neural network, h_{v_x} is the hidden state of node v_x, $c_{v_x,r}$ refers to a problem-specific normalization constant, W is the weight matrix, and $\Gamma_{v_x}^{r}$ denotes the set of neighbor indices of vertex v_x under relation $r \in R$. Thus, this model is different from normal GCNs as the accumulation of transformed feature vectors of neighboring nodes are relation-specific. For this model, using multi-layer neural networks instead of simple linear message transformation is also possible. The task of link prediction by this model can be viewed as computing node representations with an R-GCN encoder and DistMult factorization [131] as the scoring function, which is a known score function for relation representation with low number of relation parameters. The triple (s, r, o) for (subject, relation, object) is being calculated in order to determine the likelihood of possible edges as:

$$f(s, r, o) = v_{x_s}^{\prime T} R_r v_{x_o}, \tag{65}$$

in which $R_r \in \mathbb{R}^{d \times d}$ is a diagonal matrix for every relation r. The model can be trained with negative sampling via randomly corrupting the subject or object of positive examples.

7. Network Data Sets

One of the challenging tasks in network research is the implementation and validation of the proposed methods and models. In the majority of the network research, the popular collections of data sets are used as common sense: a friendship network of 34 members of a Karate Club and 78 interactions among them [132], the power network of an electrical grid of western US with 4941 nodes and 6594 edges [133], an internet-based router network with 5022 nodes and 6258 edges [134], a protein–protein interaction network that contains 2617 proteins and 11855 interactions [135], a collaboration network of 1589 authors with 2742 interactions [136], an airline network of 332 nodes and 2126 edges that show the connection between airports (http://vlado.fmf.uni-lj.si/pub/networks/data/), a social network of 62 dolphins in New Zealand with 159 interactions [137], a biological network of the cerebral cortex of Rhesus macaque with 91 nodes and 1401 edges [138].

Data set collection is time-consuming and labor-intensive work. While some studies build their own data set, the researchers mostly prefer to employ an existing data set. Some popular collections of network data sets that might be used in link prediction studies are as follows:

- SNAP [139]: a collection of more than 90 network data sets by Stanford Network Analysis Platform. With biggest data set consisting of 96 million nodes.
- BioSNAP [140]: more than 30 Bio networks data sets by Stanford Network Analysis Platform
- KONECT [141]: this collection contains more than 250 network data sets of various types, including social networks, authorship networks, interaction networks, etc.
- PAJEK [142]: this collection contains more than 40 data sets of various types.
- Network Repository [143]: a huge collection of more than 5000 network data sets of various types, including social networks.
- Uri ALON [144]: a collection of complex networks data sets by Uri Alon Lab.
- NetWiki [145]: more than 30 network data sets collection of various types.
- WOSN 2009 Data Sets [146]: a collection of Facebook data provided by social computing group.
- Citation Network Data set [147]: a collection of citation network dat aset extracted from DBLP, ACM, and other sources.
- Grouplens Research [148]: a movie rating network data set.
- ASU social computing data repository [149]: a collection of 19 network data sets of various types: cheminformatics, economic networks, etc.
- Nexus network repository [150]: a repository collection of network data sets by iGraph.
- SocioPatterns [151]: a collection of 10 network data sets that were collected by SocioPatterns interdisciplinary research collaboration.
- Mark Newman [152]: a collection of Network data sets by Mark Newman.
- Graphviz [143]: an interactive visual graph mining and analysis.

8. Taxonomy

According to the methods that were explained earlier in this paper, we propose a taxonomy to better categorize the link prediction models. In our proposed taxonomy, the link prediction techniques are mainly categorized under two sections: feature learning and feature extraction techniques (Figure 7).

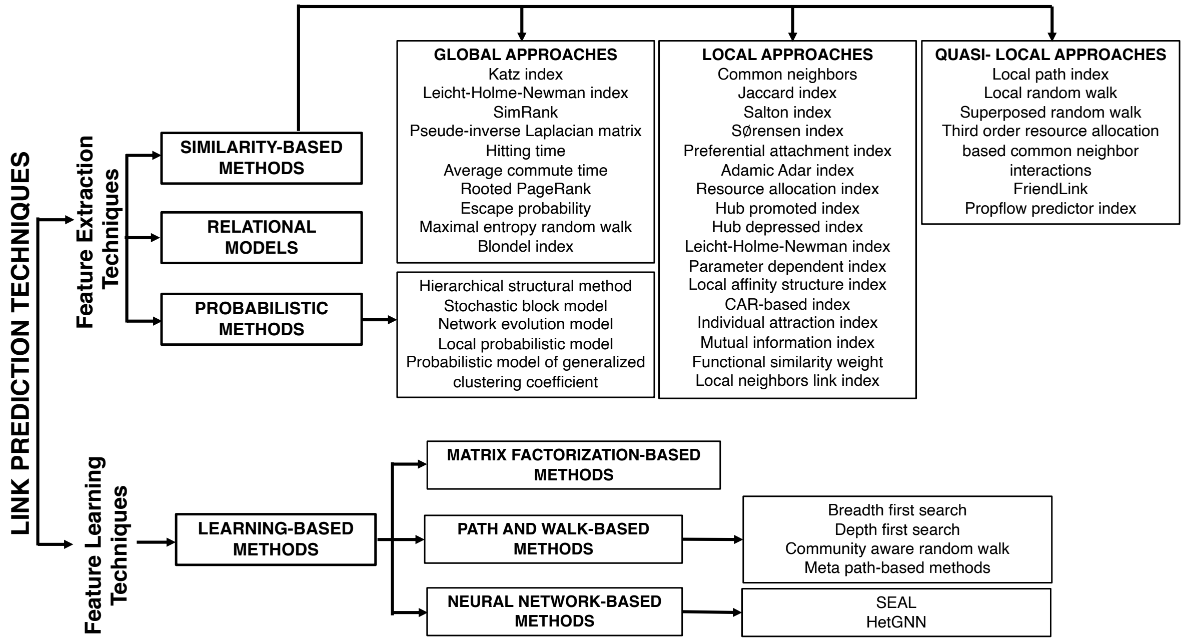
LINK PREDICTION TECHNIQUES
Feature Extraction Techniques
SIMILARITY-BASED METHODS
RELATIONAL MODELS
PROBABILISTIC METHODS
GLOBAL APPROACHES
Katz index
Leicht-Holme-Newman index
SimRank
Pseude-inverse Laplacian matrix
Hitting time
Average commute time
Rooted PageRank
Escape probability
Maximal entropy random walk
Blondel index
LOCAL APPROACHES
Common neighbors
Jaccard index
Salton index
Sørensen index
Preferential attachment index
Adamic Adar index
Resource allocation index
Hub promoted index
Hub depressed index
Leicht-Holme-Newman index
Parameter dependent index
Local affinity structure index
CAR-based index
Individual attraction index
Mutual information index
Functional similarity weight
Local neighbors link index
QUASI- LOCAL APPROACHES
Local path index
Local random walk
Superposed random walk
Third order resource allocation based common neighbor interactions
FriendLink
Propflow predictor index
Hierarchical structural method
Stochastic block model
Network evolution model
Local probabilistic model
Probabilistic model of generalized clustering coefficient
Feature Learning Techniques
LEARNING-BASED METHODS
MATRIX FACTORIZATION-BASED METHODS
PATH AND WALK-BASED METHODS
NEURAL NETWORK-BASED METHODS
Breadth first search
Depth first search
Community aware random walk
Meta path-based methods
SEAL
HetGNN

Figure 7. A taxonomy for the feature extraction techniques and feature learning methods in link prediction literature.

9. Discussion

This paper presents a comprehensive and state-of-the-art literature review on link prediction analysis, in which the emerging links or missed associations are predicted in a complex network, through a custom taxonomy. We classified link prediction techniques under two main categories. Firstly, feature extraction techniques consist of the methods that start with an initial set of features and build the required resources by using these raw features in order to describe the structural similarity. We discussed these methods under three different titles due to their strategy for addressing link prediction problems; namely, similarity-based, relational, and probabilistic methods. Among these methods, similarity-based techniques are the simplest and relatively less computationally intensive. These methods aim to explore missing links by assigning similarity scores between node pairs while using the structural properties of the graphs. According to the required topological information from a network, these methods are further divided into three subcategories. Global approaches require the complete topological information of the graph; therefore, they provide relatively more accurate results. However, the whole network may not be observable, or the large size of the network may require less time-consuming methods. In such cases, local approaches, in which a maximum second or third-degree neighborhood relationship is taken into consideration, rather than a whole network, are suggested to be applied instead. This trade-off triggered the emergence of the so-called quasi-local approaches. These methods are generally more favored and applied among similarity-based methods, since they are as efficient as global approaches due to the use of additional topological information, but less time-consuming. Other feature extraction techniques used in link prediction problems covered in this study are relational and probabilistic methods. Using maximum likelihood calculations in probabilistic methods makes them relatively time-consuming and expensive to deploy. Another major drawback of these models is the lack of the concept of an object, which is addressed in relational models. Thus, these models are able to use the logical structure of underlying data that is helpful for more complex problems. Accordingly, employing relational methods in a link prediction problem requires a massive computation of marginal probability distributions for each node in the network. Although these methods are considered to be powerful, the nonexistence of the compact closed form of these distributions due to mutual dependencies in the correlated networks makes their utilization challenging [24]. Secondly, feature learning-based techniques consist of methods that allow for a system to automatically learn the necessary set of features before building the required resources to further address the link prediction problems. These high-performance approaches enable the integration of extra information that is related to the network that might be effective in predicting the existence of links, such as community structure [153], users' behavior [154], common interests [99], etc. Additionally, machine learning models are useful in picking the right combination of features by optimizing an objective function, which renders these methods more preferable when compared to the previously discussed approaches in many cases.

Getoor and Diehl [155] categorized link prediction problems under four main sections: (i) the link existence, in which the likelihood of forming a connection between two nodes in the future is questioned, (ii) the link load, in which the weight of the associated links are analyzed, (iii) the link cardinality, in which the question of whether more than one link between a node pair exists or not is inspected, and (iv) the link type, in which the role of the link between node pairs are evaluated. Although the methods that are discussed in this survey mainly address the link prediction problem in networks, they can be easily prolonged to the problems of link load and link cardinality, since they both require a similar computational approach [156]. Some learning-based methods and probabilistic models are being deployed for link prediction in temporal and dynamic networks. Whereas, the problem of link type differ since the prediction methods foTr multi-object type links may require special attention and the deployment of different methods. To obtain more detailed information regarding the commonly used approaches for link prediction problem in weighted, bipartite, temporal, dynamic, signed, and heterogeneous networks, please visit [72,157–161], respectively.

Although the link prediction problem is an established field of research, several problems are yet to be explored in this domain. In general, the available methods in the literature produce new methods or compare the extant ones by assuming that the network is noise-free; however, some links might be missing, substituted, or fake, which is called noisy networks. While, Zhang et al. [162] compared a few numbers of similarity-based methods, but there is no detailed study that compare the robustness of different approaches. Besides, each network has its own characteristics, i.e., domain/network problem, and this makes transferring knowledge or generalizing the superiority of the link prediction algorithms challenging. Still, there are a few works that consider the effects of varying topological properties on the performance of different link prediction approaches. Furthermore, most of the real-world networks are shown to be sparse. The resulting unbalanced dataset obstructs the handling of link prediction problems, especially with the utilization of supervised techniques. Lastly, limited studies address the link prediction problem in multiplex/multilayer methods, and these studies are generally constrained with two layers. Further studies may consider this problem on multiplex networks with more than two layers.

Funding: This work was supported by the Defense Advanced Research Projects Agency (DARPA) under grant number FA8650-18-C-7823.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahuja, R.; Singhal, V.; Banga, A. Using hierarchies in online social networks to determine link prediction. In *Soft Computing and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 67–76.
2. Sulaimany, S.; Khansari, M.; Masoudi-Nejad, A. Link prediction potentials for biological networks. *Int. J. Data Min. Bioinform.* **2018**, *20*, 161–184. [CrossRef]
3. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
4. Zhou, W.; Gu, J.; Jia, Y. *h*-Index-based link prediction methods in citation network. *Scientometrics* **2018**, *117*, 381–390. [CrossRef]
5. Ebrahimi, F.; Golpayegani, S.A.H. Personalized recommender system based on social relations. In Proceedings of the 2016 24th Iranian Conference on Electrical Engineering (ICEE), Shiraz, Iran, 10–12 May 2016; pp. 218–223.
6. Martínez, V.; Berzal, F.; Cubero, J.C. A survey of link prediction in complex networks. *ACM Comput. Surv. (CSUR)* **2017**, *49*, 69. [CrossRef]
7. Song, H.H.; Cho, T.W.; Dave, V.; Zhang, Y.; Qiu, L. Scalable proximity estimation and link prediction in online social networks. In Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, Chicago, IL, USA, 4–6 November 2009; ACM: New York, NY, USA, 2009; pp. 322–335.
8. Hristova, D.; Noulas, A.; Brown, C.; Musolesi, M.; Mascolo, C. A multilayer approach to multiplexity and link prediction in online geo-social networks. *EPJ Data Sci.* **2016**, *5*, 24. [CrossRef] [PubMed]
9. Jalili, M.; Orouskhani, Y.; Asgari, M.; Alipourfard, N.; Perc, M. Link prediction in multiplex online social networks. *R. Soc. Open Sci.* **2017**, *4*, 160863. [CrossRef] [PubMed]
10. Lei, C.; Ruan, J. A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. *Bioinformatics* **2013**, *29*, 355–364. [CrossRef] [PubMed]
11. Iakovidou, N.; Symeonidis, P.; Manolopoulos, Y. Multiway spectral clustering link prediction in protein-protein interaction networks. In Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine, Corfu, Greece, 3–5 November 2010; pp. 1–4.
12. Airoldi, E.M.; Blei, D.M.; Fienberg, S.E.; Xing, E.P.; Jaakkola, T. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In Proceedings of the International Biometrics Society Annual Meeting, Hong Kong, China, 5–7 January 2006; Volume 15.
13. Zhang, M.; Cui, Z.; Jiang, S.; Chen, Y. Beyond link prediction: Predicting hyperlinks in adjacency space. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

14. Ma, Y.; Liang, X.; Huang, J.; Cheng, G. Intercity transportation construction based on link prediction. In Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 6–8 November 2017; pp. 1135–1138.
15. Desmarais, B.A.; Cranmer, S.J. Forecasting the locational dynamics of transnational terrorism: A network analytic approach. *Secur. Inform.* **2013**, *2*, 8. [CrossRef]
16. Heidari, M.; Jones, J.H.J. Using BERT to Extract Topic-Independent Sentiment Features for Social Media Bot Detection. In Proceedings of the IEEE 2020 11th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON 2020, New York, NY, USA, 28–31 October 2020.
17. Xiao, H.; Huang, M.; Zhu, X. From one point to a manifold: Knowledge graph embedding for precise link prediction. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 1315–1321.
18. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Zhao, Z.; Papalexakis, E.; Ma, X. Learning Physical Common Sense as Knowledge Graph Completion via BERT Data Augmentation and Constrained Tucker Factorization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November, 2020; pp. 3293–3298.
20. Heckerman, D.; Geiger, D.; Chickering, D.M. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.* **1995**, *20*, 197–243. [CrossRef]
21. Friedman, N.; Getoor, L.; Koller, D.; Pfeffer, A. Learning probabilistic relational models. In Proceedings of the IJCAI, Stockholm, Sweden, 31 July–6 August 1999; Volume 99, pp. 1300–1309.
22. Getoor, L.; Friedman, N.; Koller, D.; Pfeffer, A.; Taskar, B. Probabilistic relational models. In *Introduction to Statistical Relational Learning*; The MIT Press: Cambridge, MA, USA, 2007; Volume 8, pp. 128–174.
23. Getoor, L.; Friedman, N.; Koller, D.; Taskar, B. Learning probabilistic models of link structure. *J. Mach. Learn. Res.* **2002**, *3*, 679–707.
24. Al Hasan, M.; Zaki, M.J. A survey of link prediction in social networks. In *Social Network Data Analytics*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 243–275.
25. Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. *arXiv* **2018**, arXiv:1802.09691.
26. Kumar, A.; Singh, S.S.; Singh, K.; Biswas, B. Link prediction techniques, applications, and performance: A survey. *Phys. A Stat. Mech. Appl.* **2020**, *553*, 124289. [CrossRef]
27. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **2015**, *104*, 11–33. [CrossRef]
28. Lü, L.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A Stat. Mech. Appl.* **2011**, *390*, 1150–1170. [CrossRef]
29. Wasserman, S.; Faust, K. *Social Network Analysis: Methods and Applications*; Cambridge University Press: Cambridge, UK, 1994; Volume 8.
30. Wang, X.W.; Chen, Y.; Liu, Y.Y. Link Prediction through Deep Learning. *bioRxiv* **2018**, 247577. [CrossRef]
31. Manning, C.D.; Schütze, H.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
32. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36. [CrossRef]
33. Chen, H.; Li, X.; Huang, Z. Link prediction approach to collaborative filtering. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05), Denver, CO, USA, 7–11 June 2005; pp. 141–142.
34. Gray, P.M.D. P/FDM. In *Encyclopedia of Database Systems*; Liu, L., ÖZSU, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 2011–2012.
35. Voorhees, E.M.; Harman, D. Overview of the sixth text retrieval conference (TREC-6). *Inf. Process. Manag.* **2000**, *36*, 3–35. [CrossRef]
36. Zhang, E.; Zhang, Y. Average Precision. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 192–193. [CrossRef]
37. Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **2007**, *58*, 1019–1031. [CrossRef]
38. Newman, M.E. Clustering and preferential attachment in growing networks. *Phys. Rev. E* **2001**, *64*, 025102. [CrossRef]

39. Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.
40. Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; McGraw-Hill, Inc.: New York, NY, USA, 1986.
41. Wagner, C.S.; Leydesdorff, L. Mapping the network of global science: Comparing international co-authorships from 1990 to 2000. *Int. J. Technol. Glob.* **2005**, *1*, 185–208. [CrossRef]
42. Sørensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* **1948**, *5*, 1–34.
43. McCune, B.; Grace, J.B.; Urban, D.L. *Analysis of Ecological Communities*; MjM Software Design: Gleneden Beach, OR, USA, 2002; Volume 28.
44. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [CrossRef]
45. Adamic, L.A.; Adar, E. Friends and neighbors on the web. *Soc. Netw.* **2003**, *25*, 211–230. [CrossRef]
46. Zhou, T.; Lü, L.; Zhang, Y.C. Predicting missing links via local information. *Eur. Phys. J. B* **2009**, *71*, 623–630. [CrossRef]
47. Wang, P.; Xu, B.; Wu, Y.; Zhou, X. Link prediction in social networks: The state-of-the-art. *Sci. China Inf. Sci.* **2015**, *58*, 1–38. [CrossRef]
48. Ravasz, E.; Somera, A.L.; Mongru, D.A.; Oltvai, Z.N.; Barabási, A.L. Hierarchical organization of modularity in metabolic networks. *Science* **2002**, *297*, 1551–1555. [CrossRef]
49. Leicht, E.A.; Holme, P.; Newman, M.E. Vertex similarity in networks. *Phys. Rev. E* **2006**, *73*, 026120. [CrossRef]
50. Zhu, Y.X.; Lü, L.; Zhang, Q.M.; Zhou, T. Uncovering missing links with cold ends. *Phys. A Stat. Mech. Appl.* **2012**, *391*, 5769–5778. [CrossRef]
51. Sun, Q.; Hu, R.; Yang, Z.; Yao, Y.; Yang, F. An improved link prediction algorithm based on degrees and similarities of nodes. In Proceedings of the Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference, Wuhan, China, 24–26 May 2017; pp. 13–18.
52. Cannistraci, C.V.; Alanis-Lobato, G.; Ravasi, T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci. Rep.* **2013**, *3*, 1613. [CrossRef] [PubMed]
53. Dong, Y.; Ke, Q.; Wang, B.; Wu, B. Link prediction based on local information. In Proceedings of the Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference, Kaohsiung, Taiwan, 25–27 July 2011; pp. 382–386.
54. Tan, F.; Xia, Y.; Zhu, B. Link prediction in complex networks: A mutual information perspective. *PLoS ONE* **2014**, *9*, e107056. [CrossRef] [PubMed]
55. Chua, H.N.; Sung, W.K.; Wong, L. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics* **2006**, *22*, 1623–1630. [CrossRef] [PubMed]
56. Brun, C.; Chevenet, F.; Martin, D.; Wojcik, J.; Guénoche, A.; Jacq, B. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biol.* **2003**, *5*, R6. [CrossRef]
57. Yang, J.; Yang, L.; Zhang, P. A New Link Prediction Algorithm Based on Local Links. In Proceedings of the International Conference on Web-Age Information Management, Qingdao, China, 8–10 June 2015; pp. 16–28.
58. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **1953**, *18*, 39–43. [CrossRef]
59. Jeh, G.; Widom, J. SimRank: A measure of structural-context similarity. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 538–543.
60. Liben-Nowell, D. An Algorithmic Approach to Social Networks. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005.
61. Spielman, D.A. Spectral graph theory and its applications. In Proceedings of the Foundations of Computer Science, FOCS'07, 48th Annual IEEE Symposium, Providence, RI, USA, 21–23 October 2007; pp. 29–38.
62. Fouss, F.; Pirotte, A.; Renders, J.M.; Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 355–369. [CrossRef]
63. Pearson, K. The problem of the random walk. *Nature* **1905**, *72*, 342. [CrossRef]

64. Tong, H.; Faloutsos, C.; Faloutsos, C.; Koren, Y. Fast direction-aware proximity for graph mining. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; pp. 747–756.
65. Lü, L.; Zhou, T. Link prediction in weighted networks: The role of weak ties. *EPL Europhys. Lett.* **2010**, *89*, 18001. [CrossRef]
66. Hetherington, J. Observations on the statistical iteration of matrices. *Phys. Rev. A* **1984**, *30*, 2713. [CrossRef]
67. Duda, J. Extended Maximal Entropy Random Walk. Ph.D. Thesis, Jagiellonian University, Krakow, Poland, 2012.
68. Li, R.H.; Yu, J.X.; Liu, J. Link prediction: The power of maximal entropy random walk. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; pp. 1147–1156.
69. Blondel, V.D.; Gajardo, A.; Heymans, M.; Senellart, P.; Van Dooren, P. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.* **2004**, *46*, 647–666. [CrossRef]
70. Lü, L.; Jin, C.H.; Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Phys. Rev. E* **2009**, *80*, 046122. [CrossRef] [PubMed]
71. Liu, W.; Lü, L. Link prediction based on local random walk. *EPL Europhys. Lett.* **2010**, *89*, 58007. [CrossRef]
72. Zhang, J.; Zhang, Y.; Yang, H.; Yang, J. A link prediction algorithm based on socialized semi-local information. *J. Comput. Inf. Syst.* **2014**, *10*, 4459–4466.
73. Papadimitriou, A.; Symeonidis, P.; Manolopoulos, Y. Fast and accurate link prediction in social networking systems. *J. Syst. Softw.* **2012**, *85*, 2119–2132. [CrossRef]
74. Lichtenwalter, R.N.; Lussier, J.T.; Chawla, N.V. New perspectives and methods in link prediction. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 243–252.
75. Ravasz, E.; Barabási, A.L. Hierarchical organization in complex networks. *Phys. Rev. E* **2003**, *67*, 026112. [CrossRef]
76. Clauset, A.; Moore, C.; Newman, M.E. Hierarchical structure and the prediction of missing links in networks. *Nature* **2008**, *453*, 98. [CrossRef]
77. Goldenberg, A.; Zheng, A.X.; Fienberg, S.E.; Airoldi, E.M. *A Survey of Statistical Network Models*; Now Publishers Inc: Delft, The Netherlands, 2010.
78. Guimerà, R.; Sales-Pardo, M. Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 22073–22078. [CrossRef]
79. Peixoto, T.P. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **2014**, *4*, 011047. [CrossRef]
80. Vallès-Català, T.; Peixoto, T.P.; Sales-Pardo, M.; Guimerà, R. Consistencies and inconsistencies between model selection and link prediction in networks. *Phys. Rev. E* **2018**, *97*, 062316. [CrossRef]
81. Guimera, R.; Sales-Pardo, M. A network inference method for large-scale unsupervised identification of novel drug-drug interactions. *PLoS Comput. Biol.* **2013**, *9*, e1003374. [CrossRef]
82. Rovira-Asenjo, N.; Gumí, T.; Sales-Pardo, M.; Guimera, R. Predicting future conflict between team-members with parameter-free models of social networks. *Sci. Rep.* **2013**, *3*, 1–6. [CrossRef] [PubMed]
83. Godoy-Lorite, A.; Guimerà, R.; Moore, C.; Sales-Pardo, M. Accurate and scalable social recommendation using mixed-membership stochastic block models. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 14207–14212. [CrossRef] [PubMed]
84. Holland, P.W.; Laskey, K.B.; Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **1983**, *5*, 109–137. [CrossRef]
85. Bhanot, G. The metropolis algorithm. *Rep. Prog. Phys.* **1988**, *51*, 429. [CrossRef]
86. Kashima, H.; Abe, N. A parameterized probabilistic model of network evolution for supervised link prediction. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 340–349.
87. Wang, C.; Satuluri, V.; Parthasarathy, S. Local probabilistic models for link prediction. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 322–331.
88. Huang, Z. Link prediction based on graph topology: The predictive value of generalized clustering coefficient. *SSRN Electron. J.* **2010**. [CrossRef]

89. Bilgic, M.; Namata, G.M.; Getoor, L. Combining collective classification and link prediction. In Proceedings of the Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, NE, USA, 28–31 October 2007; pp. 381–386.
90. Taskar, B.; Abbeel, P.; Koller, D. Discriminative probabilistic models for relational data. *arXiv* **2012**, arXiv:1301.0604.
91. Taskar, B.; Wong, M.F.; Abbeel, P.; Koller, D. Link prediction in relational data. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2004; pp. 659–666.
92. Taskar, B.; Abbeel, P.; Wong, M.F.; Koller, D. Relational markov networks. In *Introduction to Statistical Relational Learning*; The MIT Press: Cambridge, MA, USA, 2007; pp. 175–200.
93. Heckerman, D.; Meek, C.; Koller, D. Probabilistic entity-relationship models, PRMs, and plate models. In *Introduction to Statistical Relational Learning*; The MIT Press: Cambridge, MA, USA, 2007; pp. 201–238.
94. Heckerman, D.; Meek, C.; Koller, D. *Probabilistic Models for Relational Data*; Technical Report, Technical Report MSR-TR-2004-30; Microsoft Research: Redmond, WA, USA, 2004.
95. Yu, K.; Chu, W.; Yu, S.; Tresp, V.; Xu, Z. Stochastic relational models for discriminative link prediction. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; The MIT Press: Cambridge, MA, USA, 2007; pp. 1553–1560.
96. Neville, J.; Jensen, D. Relational dependency networks. *J. Mach. Learn. Res.* **2007**, *8*, 653–692.
97. Heckerman, D.; Chickering, D.M.; Meek, C.; Rounthwaite, R.; Kadie, C. Dependency networks for collaborative filtering and data visualization. *arXiv* **2013**, arXiv:1301.3862.
98. Xu, Z.; Tresp, V.; Yu, K.; Yu, S.; Kriegel, H.P. Dirichlet enhanced relational learning. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 1004–1011.
99. Al Hasan, M.; Chaoji, V.; Salem, S.; Zaki, M. Link prediction using supervised learning. In Proceedings of the SDM06: Workshop on Link Analysis, Counter-Terrorism and Security, Bethesda, MD, USA, 22 April 2006.
100. Duan, L.; Ma, S.; Aggarwal, C.; Ma, T.; Huai, J. An ensemble approach to link prediction. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2402–2416. [CrossRef]
101. Hamilton, W.L.; Ying, R.; Leskovec, J. Representation learning on graphs: Methods and applications. *arXiv* **2017**, arXiv:1709.05584.
102. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 855–864.
103. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. *K-BERT: Enabling Language Representation with Knowledge Graph*; Cornrll University: Ithaca, NY, USA, 2019.
104. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 3–7 June 2018; pp. 593–607.
105. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
106. Khosla, M.; Leonhardt, J.; Nejdl, W.; Anand, A. Node representation learning for directed graphs. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Würzburg, Germany, 16–20 September 2019; pp. 395–411.
107. Lichtenwalter, R.N.; Chawla, N.V. Vertex collocation profiles: Subgraph counting for link analysis and prediction. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 1019–1028.
108. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
109. Cui, P.; Wang, X.; Pei, J.; Zhu, W. A survey on network embedding. *arXiv* **2017**, arXiv:1711.08752.
110. Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; Zhu, W. Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 1105–1114.
111. Menon, A.K.; Elkan, C. Link prediction via matrix factorization. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bristol, UK, 23–27 September 2011; pp. 437–452.

112. Ahmed, A.; Shervashidze, N.; Narayanamurthy, S.; Josifovski, V.; Smola, A.J. Distributed large-scale natural graph factorization. In Proceedings of the 22nd International Conference on World Wide Web, Janeiro, Brazil, 13–17 May 2013; ACM: New York, NY, USA, 2013; pp. 37–48.
113. Cao, S.; Lu, W.; Xu, Q. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; ACM: New York, NY, USA, 2015; pp. 891–900.
114. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* **2018**, *151*, 78–94. [CrossRef]
115. Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *arXiv* **2018**, arXiv:1812.08434.
116. Wang, H.; Wang, J.; Wang, J.; Zhao, M.; Zhang, W.; Zhang, F.; Xie, X.; Guo, M. Graphgan: Graph representation learning with generative adversarial nets. *arXiv* **2017**, arXiv:1711.08267.
117. Keikha, M.M.; Rahgozar, M.; Asadpour, M. Community aware random walk for network embedding. *Knowl.-Based Syst.* **2018**, *148*, 47–54. [CrossRef]
118. Sun, Y.; Han, J.; Yan, X.; Yu, P.S.; Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **2011**, *4*, 992–1003. [CrossRef]
119. Fu, G.; Ding, Y.; Seal, A.; Chen, B.; Sun, Y.; Bolton, E. Predicting drug target interactions using meta-path-based semantic network analysis. *BMC Bioinform.* **2016**, *17*, 160. [CrossRef]
120. Zhang, C.; Song, D.; Huang, C.; Swami, A.; Chawla, N.V. Heterogeneous graph neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 793–803.
121. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef]
122. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
123. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308.
124. Berg, R.v.d.; Kipf, T.N.; Welling, M. Graph convolutional matrix completion. *arXiv* **2017**, arXiv:1706.02263.
125. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. *arXiv* **2017**, arXiv:1704.01212.
126. Cao, S.; Lu, W.; Xu, Q. Deep Neural Networks for Learning Graph Representations. In Proceedings of the AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 1145–1152.
127. Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; Zhang, C. Adversarially regularized graph autoencoder for graph embedding. *arXiv* **2018**, arXiv:1802.04407.
128. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1225–1234.
129. Harada, S.; Akita, H.; Tsubaki, M.; Baba, Y.; Takigawa, I.; Yamanishi, Y.; Kashima, H. Dual Convolutional Neural Network for Graph of Graphs Link Prediction. *arXiv* **2018**, arXiv:1810.02080.
130. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; pp. 1024–1034.
131. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
132. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [CrossRef]
133. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world'networks. *Nature* **1998**, *393*, 440–442. [CrossRef]
134. Spring, N.; Mahajan, R.; Wetherall, D. Measuring ISP topologies with Rocketfuel. *ACM SIGCOMM Comput. Commun. Rev.* **2002**, *32*, 133–145. [CrossRef]
135. Von Mering, C.; Krause, R.; Snel, B.; Cornell, M.; Oliver, S.G.; Fields, S.; Bork, P. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* **2002**, *417*, 399–403. [CrossRef]
136. Newman, M.E. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **2006**, *74*, 036104. [CrossRef] [PubMed]

137. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. [CrossRef]
138. Markov, N.T.; Ercsey-Ravasz, M.; Ribeiro Gomes, A.; Lamy, C.; Magrou, L.; Vezoli, J.; Misery, P.; Falchier, A.; Quilodran, R.; Gariel, M.; et al. A weighted and directed interareal connectivity matrix for macaque cerebral cortex. *Cereb. Cortex* **2014**, *24*, 17–36. [CrossRef] [PubMed]
139. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. 2014. Available online: http://snap.stanford.edu/data (accessed on 16 December 2020).
140. Zitnik, M.; Sosič, R.; Maheshwari, S.; Leskovec, J. BioSNAP Datasets: Stanford Biomedical Network Dataset Collection. 2018. Available online: http://snap.stanford.edu/biodata (accessed on 16 December 2020).
141. Kunegis, J. KONECT: The Koblenz Network Collection. In Proceedings of the 22Nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; ACM: New York, NY, USA, 2013; pp. 1343–1350.
142. Batagelj, V.; Mrvar, A. Pajek Datasets. 2006. Available online: http://http://vlado.fmf.uni-lj.si/pub/networks/data/ (accessed on 16 December 2020).
143. Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
144. Lab, U.A. Collection of Complex Networks. 2006. Available online: http://www.weizmann.ac.il (accessed on 16 December 2020).
145. Mucha, P.; Porter, M. Netwiki Shared Data. 2013. Available online: http://netwiki.amath.unc.edu (accessed on 16 December 2020).
146. Viswanath, B.; Mislove, A.; Cha, M.; Gummadi, K.P. On the Evolution of User Interaction in Facebook. In Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09), Barcelona, Spain, 16–21 August 2009.
147. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceedings of the KDD'08, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
148. Grouplens. Movielens Rating Dataset. Available online: https://grouplens.org/datasets/movielens/ (accessed on 16 December 2020).
149. Zafarani, R.; Liu, H. Social Computing Data Repository at ASU. 2009. Available online: https://www.re3data.org/repository/r3d100010959 (accessed on 16 December 2020).
150. Nexus Network Repository. 2015. Available online: https://igraph.org/r/doc/nexus.html (accessed on 16 December 2020).
151. SocioPAttern Research Collaboration. Available online: http://www.sociopatterns.org/datasets/ (accessed on 16 December 2020).
152. Newman, M. Mark Newman Network Datasets Collection. 2013. Available online: http://www-personal.umich.edu/~mejn/netdata (accessed on 16 December 2020).
153. Mohan, A.; Venkatesan, R.; Pramod, K. A scalable method for link prediction in large real world networks. *J. Parallel Distrib. Comput.* **2017**, *109*, 89–101. [CrossRef]
154. Xiao, Y.; Li, X.; Wang, H.; Xu, M.; Liu, Y. 3-HBP: A three-level hidden Bayesian link prediction model in social networks. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 430–443. [CrossRef]
155. Getoor, L.; Diehl, C.P. Link mining: A survey. *Acm Sigkdd Explor. Newsl.* **2005**, *7*, 3–12. [CrossRef]
156. Kushwah, A.K.S.; Manjhvar, A.K. A review on link prediction in social network. *Int. J. Grid Distrib. Comput.* **2016**, *9*, 43–50. [CrossRef]
157. Wind, D.K.; Mørup, M. Link prediction in weighted networks. In Proceedings of the 2012 IEEE International Workshop on Machine Learning for Signal Processing, Santander, Spain, 23–26 September 2012; pp. 1–6.
158. Kunegis, J.; De Luca, E.W.; Albayrak, S. The link prediction problem in bipartite networks. In Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Dortmund, Germany, 28 June–2 July 2010; pp. 380–389.
159. Bu, Z.; Wang, Y.; Li, H.J.; Jiang, J.; Wu, Z.; Cao, J. Link prediction in temporal networks: Integrating survival analysis and game theory. *Inf. Sci.* **2019**, *498*, 41–61. [CrossRef]
160. Marjan, M.; Zaki, N.; Mohamed, E.A. Link prediction in dynamic social networks: A literature review. In Proceedings of the 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 21–27 October 2018; pp. 200–207.

161. Daud, N.N.; Ab Hamid, S.H.; Saadoon, M.; Sahran, F.; Anuar, N.B. Applications of link prediction in social networks: A review. *J. Netw. Comput. Appl.* **2020**, *166*, 102716. [CrossRef]

162. Zhang, P.; Wang, X.; Wang, F.; Zeng, A.; Xiao, J. Measuring the robustness of link prediction algorithms under noisy environment. *Sci. Rep.* **2016**, *6*, 1–7. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.