Artificial Intelligence Review (2024) 57:348
https://doi.org/10.1007/s10462-024-10998-7

A survey on feature extraction and learning techniques for link prediction in homogeneous and heterogeneous complex networks

Puneet Kapoor[1] · Sakshi Kaushal[1] · Harish Kumar[1] · Kushal Kanwar[2]

Accepted: 4 October 2024 / Published online: 28 October 2024
© The Author(s) 2024

Abstract
Complex networks are commonly observed in several real-world areas, such as social, biological, and technical systems, where they exhibit complicated patterns of connectedness and organised clusters. These networks have intricate topological characteristics that frequently elude conventional characterization. Link prediction in complex networks, like data flow in telecommunications networks, protein interactions in biological systems, and social media interactions on platforms like Facebook, etc., is an essential element of network analytics and presents fresh research challenges. Consequently, there is a growing emphasis in research on creating new link prediction methods for different network applications. This survey investigates several strategies related to link prediction, ranging from feature extraction based to feature learning based techniques, with a specific focus on their utilisation in dynamic and developing network topologies. Furthermore, this paper emphasises on a wide variety of feature learning techniques that go beyond basic feature extraction and matrix factorization. It includes advanced learning-based algorithms and neural network techniques specifically designed for link prediction. The study also presents evaluation results of different link prediction techniques on homogeneous and heterogeneous network datasets, and provides a thorough examination of existing methods and potential areas for further investigation.

Keywords Complex networks · Link prediction · Graph neural networks · Feature learning · Predictive analytics

1 Introduction

The field of network theory and its practical applications have experienced an immense shift, moving from a focus on simple networks to a more subtle examination of complicated ones. At first, network analysis primarily concentrated on relatively simple systems, where

S. Kaushal, H. Kumar and K. Kanwar have contributed equally to this work.

Extended author information available on the last page of the article

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

interactions were restricted and foreseeable. Initial research focused on linear and clearly identifiable connections, neglecting the complex dynamics that characterise real-world systems. Nevertheless, as the research advanced, it became increasingly clear that the majority of real-world systems are not simple and instead display a significant level of complexity. This change in viewpoint resulted in the development of intricate network theory, a discipline that aims to comprehend systems with several interconnected components that display non-linear, dynamic, and frequently unpredictable behaviour. Complex networks are widely acknowledged as essential structures that underlie numerous crucial real-world systems (Zhou et al. 2023b; Nasiri et al. 2023; Dong et al. 2021). These entities are distinguished by their complex network of connections, extending beyond basic one-on-one interactions to embrace a wide array of relationships. Graph structures are commonly used to depict these networks, with nodes representing things and edges representing their links or interactions. Complex networks are crucial in various sectors, including social media platforms, biological creature interactions, and large-scale information storage frameworks. Social media platforms consist of individuals who are connected through various relationships, such as friendships, followers, and common interests. Within biological systems, interactions can span from basic food chains to complex symbiotic connections. Conversely, information storage structures encompass intricate data connections and interdependencies. These networks provide deep insights into the behaviours and functionalities of the systems they depict, uncovering patterns and dynamics that are crucial for comprehending and efficiently controlling these systems (Wu et al. 2022; Dong et al. 2018). The intricate and varied nature of these networks requires sophisticated analytical methods in order to comprehensively comprehend and use their capabilities. Fig. 1 depicts the transition from traditional network analysis to the more advanced field of graph representation learning. It highlights the adaptability of latent space mappings in tackling various predicting problems within network topologies. These mappings form the basis for link prediction, node categorization, and graph-level inference, expanding our range of predictive skills. The diagram illustrates how networks can be represented in a lower-dimensional latent space, allowing for the prediction of new connections, categorization of individual nodes, and characterisation of network segments. By employing a multifaceted predictive strategy, we acquire a full set of

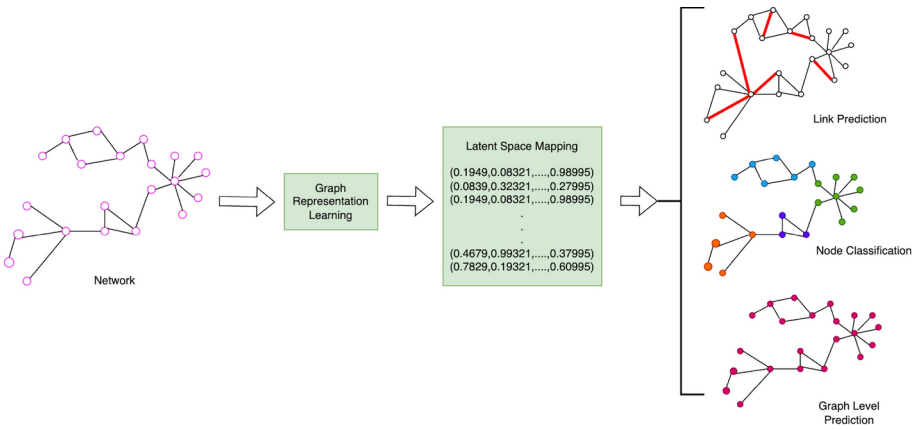


Fig. 1 Networks analysis using graph representation learning: converting networks into a latent space for use in predictive modelling of links, nodes, and graphs

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

tools to transform theoretical network models into practical insights that guide real-world applications and strategies (Daud et al. 2020; Zhao et al. 2023). As we explore the potential of graph representation learning, we come across the emerging topic of link prediction, which is a highly influential area of research in network analysis. This method not only corresponds to, but also greatly improves our capacity to see developing patterns and potential future connections in intricate systems.

Within the realm of intricate networks, the importance of link prediction is growing exponentially. Link prediction is an analytical approach that aims to find and predict probable future relationships within a network. These connections may not be obvious at first, but they can form gradually and offer important insights about how the network changes over time (Zeb et al. 2022). Predicting these connections is not just a theoretical exercise; it is crucial for understanding the existing condition of the network and provides a strategic tool for planning and decision-making. By proactively predicting future connections, stakeholders can more effectively negotiate the intricacies of these networks, enabling them to make well-informed decisions that coincide with the projected path of the network's development (Trouillon et al. 2016).

Link prediction has its roots in early research on complex networks and social dynamics. The principle of triadic closure, initially formulated by mathematician Jaccard in 1901 (1901), states that if two individuals share a mutual friend, there is a greater probability that they will establish a friendship. Jaccard's investigation on the spatial distribution of plant life in the Alps and Jura areas resulted in the development of the Jaccard Index, a metric that quantifies the degree of similarity between two sets. This fundamental concept laid the groundwork for comprehending the process by which connections are established and develop in different types of networks. Physicist M. E. Newman (Newman 2001) investigated the impact of triadic closure on the development of complex networks in 2001. His research on clustering and preferred attachment emphasized the tendency of nodes in a network to establish connections depending on their current ties and shared neighbors. This study established a mathematical framework for examining the expansion of networks and facilitated more targeted investigations into link prediction. In their 2003 study, computer scientists David Liben-Nowell and Jon Kleinberg expanded on Newman's research by introducing the formal problem of link prediction. The Common Neighbor Index (CN) was introduced as a straightforward yet effective approach to estimate the probability of a link being established between two nodes, by considering the number of common neighbors they possess. This approach, based on the concept of triadic closure, has become a fundamental aspect in the domain of link prediction (Liben-Nowell and Kleinberg 2003). By 2007, network complexity had expanded dramatically, needing more efficient methods of connection analysis and prediction. Traditional one-mode networks, in which all nodes are of the same kind, were discovered to be less useful when dealing with more complicated systems such as social networks, where interactions frequently involve diverse sorts of entities (e.g., users and items). The necessity to appropriately depict these interactions prompted the study of bipartite networks, in which nodes are separated into two distinct sets and connections are created only between nodes from different sets. In 2007, Zhou et al. (2007) proposed the concept of bipartite network projection and used it to personal recommendation systems. This study tackled the challenge of compressing bipartite networks, which are made up of two separate sets of nodes, into one-mode networks while retaining critical structural information. The proposed weighting method, which is motivated by resource-allocation dynam-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

ics, greatly enhanced recommendation performance when compared to standard methods such as global ranking and collaborative filtering. This study highlighted the significance of specialized methodologies for various network types and cleared the way for future research on bipartite graphs. Shang et al. (2017b) focused on how bi-directional links (interactions in both directions between nodes) are more relevant for link prediction and network connectivity than one-directional links. This work used a phase-dynamic algorithm to investigate the role of link directions and created a directional randomized algorithm to test their hypothesis. They discovered that bidirectional links are more likely to connect to shared neighbors and are essential for creating and maintaining network structure. This emphasis on the directional element of links paved the way for more sophisticated analyses in subsequent studies, highlighting how knowing the directionality of connections may considerably improve the accuracy of link prediction models. This study emphasized the need of considering the direction of interactions in a variety of applications, ranging from social networks to biological systems, in order to better forecast future connections and understand the underlying dynamics of such networks. In 2017, (Shang et al. 2017c) highlighted the importance of taking into account the evolution of direct connections over time in order to improve link prediction accuracy. This technique addresses the dynamic character of real-world networks, in which node interactions can change and evolve. Tracking these changes improves the accuracy and reliability of prediction algorithms for future links. Building on this (Shang et al. 2019a), investigated the role of weak ties in link prediction. Weak ties are connections that are uncommon or have low strength yet play an important function in linking distinct clusters within a network. The study discovered that these weak ties are critical for facilitating the flow of knowledge and creativity. It stressed that weak linkages are more important than strong ties for ensuring network connectivity and improving prediction accuracy. This unanticipated finding challenged the usual emphasis on strong connections, demonstrating that even weak relationships have a significant impact on network dynamics and link prediction. Together, these findings indicated a shift in emphasis from simply discovering links to studying their strengths and evolutionary patterns. This change gave a better knowledge of network activity and increased the precision of link prediction algorithms. It became clear that typical techniques, which rely on assumptions such as triangle topologies or preferred attachment, frequently fail in increasingly complex or sparse networks. Tree-like topologies, such as genealogical or organizational networks, provide unique link prediction issues due to their hierarchical structure and lack of triangular interactions. In 2019, Shang et al. (2019b) demonstrated that traditional link prediction approaches are ineffective for these hierarchical networks because they rely heavily on features such as triangular structures and preferential attachment, which are not common in tree-like topologies. Instead, the study suggested link prediction methods that were suited to the unique topologies of hierarchical networks. These techniques accounted for the depth of nodes in the hierarchy, parent-child relationships, and the distinct growth patterns inherent in tree topologies. The study found that by tailoring link prediction methodologies to these structural attributes, a better understanding of the network's specific characteristics might result in more accurate and meaningful predictions. This versatility is essential for ensuring accuracy and relevance in a variety of applications, including genealogical research and organizational network analysis. The findings underscored the need of taking into account a network's specific topology when creating link prediction algorithms, guaranteeing that the unique dynamics and evolutionary patterns of hierarchical networks are adequately recorded for better forecasts and

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

insights into their future expansion. The growth of link prediction saw notable breakthroughs in multilayer networks, which are made up of numerous layers that represent different types of linkages. In 2022, additional developments were made to address the constraints of traditional techniques in sparse networks, notably those with long-line or circular designs. Shang et al. (Shang and Small 2022) introduced unique link prediction methods tailored to the specific topologies of hierarchical and sparse networks. These methods took into account both hierarchical depth and parent-child relationships, as well as the structural properties of long-circle and long-line networks. By addressing these distinct characteristics, the new techniques considerably increased the accuracy of link predictions in complicated settings. Furthermore, the study underscored the need of taking into account both homogeneity and heterogeneity in network structures. Link prediction algorithms were significantly improved in precision by including community detection methods and adjusting to the individual structural patterns of distinct subnetworks. This method is critical for applications in a variety of domains, including biology, social sciences, and engineering, where understanding the complex dynamics of network evolution is required.

In addition to these Improvements several categorization techniques for link prediction algorithms have made major contributions to the field. Network structure analysis, for example, uses network topology or information theory to anticipate linkages. Lü et al. (2015) proposed the structural consistency index, which quantifies a network's regularity by comparing the consistency of its structural properties before and after random link removal. This score is a decent estimator of link predictability and emphasizes the significance of understanding network structural aspects in order to predict links effectively. Network embedding techniques are another modern approach to link prediction. These methods use embeddings from random walks to capture node interactions in a network. Zhang et al. (2021), exemplifies this method by embedding nodes in a continuous vector space, these approaches efficiently capture node structural traits and linkages, considerably improving link prediction accuracy. Machine and deep learning algorithms have also contributed significantly to the evolution of link prediction. These methods use node attributes and network structure to learn complex patterns and relationships, improving the accuracy of link prediction models. These algorithms, which use advanced learning techniques, may uncover complex relationships and interactions within networks, making them useful tools for link prediction (Hamilton et al. 2017a). These categorization techniques complement the conventional methods outlined above, providing a comprehensive toolkit for handling the various issues of link prediction in complex networks. Researchers can improve the robustness and versatility of link prediction models by combining insights from network structure analysis, network embedding techniques, and machine learning algorithms. These methodologies ensure that the dynamic and evolutionary patterns of different network types are adequately captured, resulting in more accurate insights and projections.

Fig. 2 highlights the significance of link prediction in predicting connections that are not yet apparent inside a network. The concealed interconnections, depicted as dotted lines, provide a stark contrast to the apparent solid-line connections and play a crucial role in revealing the underlying structures and dynamics that define intricate networks. The practical application of link prediction extends beyond academic interest and is applicable in several real-world contexts (Cao et al. 2020). Online merchants such as Amazon utilise link prediction to reliably identify consumer-product affinities, hence improving the accuracy of "you might also like..." recommendations (Ying et al. 2018; Liu and Duan 2021). Within the cor-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

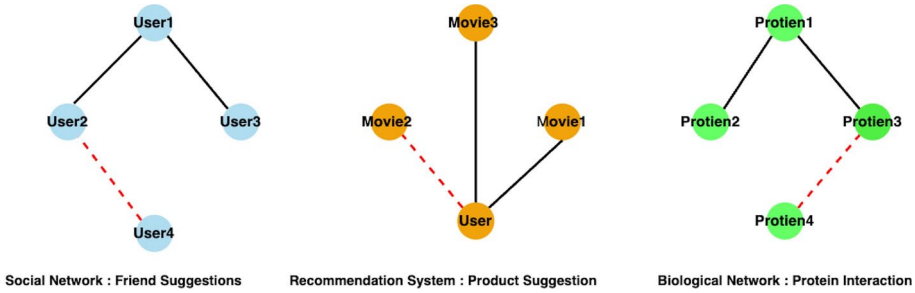


Fig. 2 Link Prediction scenarios in different networks: within a social network to anticipate potential connections between users, specifically User2 and User4; in a recommendation system to propose related products, such as Movie2, to a user according to their previous interactions; in a biological network to forecast future relationships between proteins, specifically Protein3 and Protein4

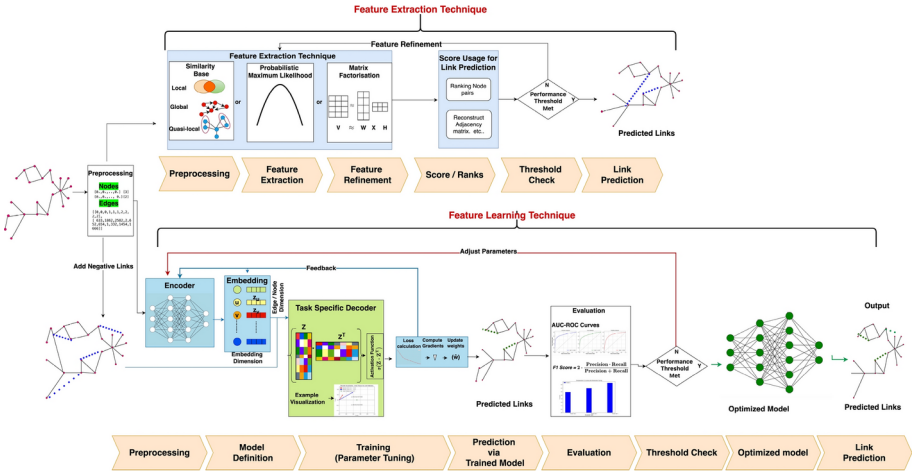


Fig. 3 General steps for feature extraction and feature learning techniques

porate domain, predicting future interactions can enhance synergy and promote teamwork. Law enforcement authorities utilise link prediction to analyse intricate terrorist networks, obtaining crucial knowledge from the interconnected web of relationships (Kumar et al. 2020; Kumari et al. 2022). These many instances jointly demonstrate the significant consequences of link prediction, effortlessly integrating its predictive ability into everyday life.

While employing comprehensive approach to anticipate link development in complex networks, it combines feature extraction and feature learning approaches in a two-pronged strategy, as shown in Fig. 3. Feature extraction begins with thorough preprocessing to define and enhance node and edge features, using various techniques like as similarity-based methods and probabilistic maximum likelihood estimates to get structural information from the network. Matrix factorization is used to uncover hidden properties that are crucial for predictive modelling by examining the inherent connections in the network. Transitioning to the feature learning stage, an encoder is employed, typically implemented as a graph neural network or path based encoders, to convert the extracted features into a compact embedding

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

space. The embeddings are decoded in a task-specific way to recreate network features or predict new connections. The encoder-decoder framework is refined by an iterative process guided by a loss function to adjust model parameters. The evaluation of this process involves metrics like Micro-F1, Macro-F1, AUC-ROC curves etc. The iterative process of training and evaluating enhances the accuracy of the model's predictions, resulting in a proficient system for link prediction (Qiu et al. 2018; Zhang et al. 2019a; Derrow-Pinion et al. 2021; Kumar et al. 2020).

The focus of this paper is on the field of feature learning, highlighting algorithms that utilise sequential data to create embeddings, while acknowledging the creative application of path and walk-based tactics (Zhang et al. 2019a). A specific section is focused on neural network techniques, specifically highlighting the abilities of Graph Neural Networks (GNNs) such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GANs)(Fig. 4), which are particularly effective in understanding the intricate relationship between network structure and node characteristics (Wang et al. 2023b). Furthermore, the discussion encompasses techniques that are skilled at managing the intricacies inherent in diverse networks and knowledge graphs, demonstrating models specifically designed to navigate the multiple structure of these networks (Yadati et al. 2020; Derrow-Pinion et al. 2021). The highlights of our paper are as follows:

- It provides a taxonomy which classifies link prediction techniques into two main categories: feature extraction and feature learning methods. This classification helps to organise and understand the methodology used in the field, and allows for a deeper understanding of both the theoretical and practical aspects of link prediction techniques.
- An in-depth investigation of homogeneous datasets, Cora, CiteSeer, PubMed (Sen et al. 2008), Ego-Facebook, and WikiVote (Leskovec and Krevl 2014), as well as heterogeneous datasets DBLP (Tong et al. 2006), MovieLens (Nasiri et al. 2023), IMDb (Fu et al. 2020a), OGBN-MAG, and OGBL-BIOKG (Hu et al. 2020b) has been provided. Information regarding nodes, edges, node types, and edge types has been included to improve understanding of the elements involved in link prediction.
- It presents the evaluation of different learning-based techniques, such as matrix-based, path-based, and neural network-based approaches, on homogeneous datasets including PubMed, CiteSeer, Cora, Ego-Facebook, and Wiki-Vote. The results for each dataset and technique are presented in separate sections, which include Macro-F1 and Micro-F1

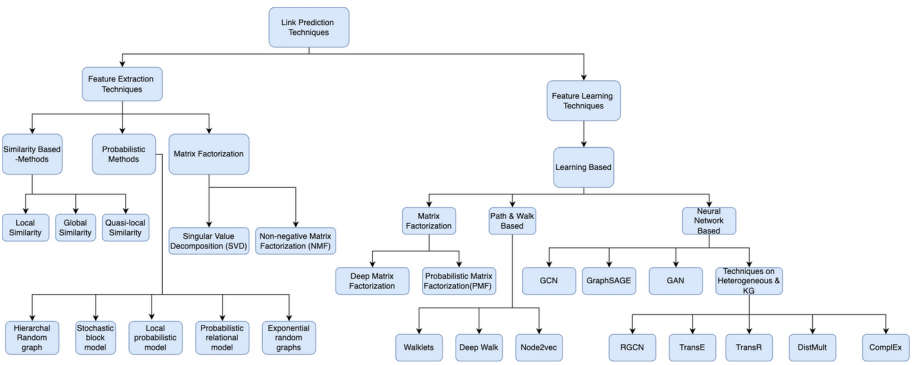


Fig. 4 Link prediction taxonomy

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

scores presented in tables, and AUC curves depicted through graphical representations. A thorough examination of the results has been performed to determine the performance of each technique on every homogeneous dataset.

- It presents a specialised segment that discusses methodologies relevant to heterogeneous networks. The heterogeneous techniques have been implemented on considered heterogeneous datasets (DBLP, MovieLens, IMDb, OGBN-MAG, OGBL-BIOKG), and the outcomes have been measured using macro F1, micro F1, and AUC metrics. An evaluation and examination of the performance of each technique based on the dataset have been presented.
- The research sheds new light on the utility of Graph Neural Networks (GNNs) by assessing their performance on the aforementioned datasets, with a focus on the flexibility of GCN, GAN, and Relational-GCN models. The findings demonstrate GNNs' revolutionary impact in link prediction, particularly in learning from dynamic networks' intricate topology.The subsequent sections of this paper are structured in the following manner: Sect. 2 explores the domain of link prediction, specifically emphasising on feature extraction techniques. Section 3 focuses on techniques for link prediction that are based on feature learning. The literature is analysed to classify different methods, ranging from path and walk-based algorithms to sophisticated neural network models. Section 4 is devoted to the examination of research direction, and indicates potential possibilities for further research and exploration. Finally, the last section concludes the study.

2 Link prediction methods-feature extraction techniques

This section carefully looks at the different Feature Extraction Techniques used in Link Prediction. The techniques are divided into three main categories: Similarity-Based Methods, Probabilistic Methods, and Matrix Factorization, each with its own subcategories and methodologies.

2.1 Similarity based methods

Similarity-based metrics are among the most basic in link prediction, calculating a similarity score S(x, y) for every pair x and y. S(x,y) is a measure obtained from the structural qualities or attributes of the nodes in the analysed pair. The unseen connections links i.e., U-E(U: Universal set, E: Link set) are scored based on their similarities. The higher-scoring pair of nodes shows the expected link between them. The similarity measures between each pair can be determined using many network attributes, which include the structural property. Scores depending on this attribute can be classified as local or global, path-dependent or node-dependent, parameter-free or parameter-dependent (Kumar et al. 2020). There are further similarity indices as described below:

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

2.1.1 Local similarity indices

In general, local indices are computed using details about common neighbours and node degree. These indices take into account a node's immediate neighbours. The following are examples of local similarity indices.

2.1.1.1 Common neighbours In a graph or network, the number of common neighbours shared by a node pair x and y is calculated by computing the amount of the overlap between the neighbourhoods of these two nodes (Liben-Nowell and Kleinberg 2003). The level of similarity between both nodes x and y can be calculated using the following Eq. (1).

$$S(x, y) = |N(x) \cap N(y)| \tag{1}$$

in which $N(x)$ and $N(y)$ are the neighbours of nodes x and y. The greater the number of neighbours that x and y share, the more likely a relationship exists between them. Kossinets and Watts (Kossinets and Watts 2009) analysed a big social network and concluded that two students who have many shared friends are much more likely to be buddies. It has been observed that the common neighbour strategy outperforms other complicated strategies on most real-world networks.

2.1.1.2 Resource allocation Resource allocation index utilizes the notion of frequent neighbors to forecast connections. This approach is based on the concept of resource allocation, where it takes into account the common neighbors of two nodes and distributes a "resource" value that is inversely proportional to the degree of these shared neighbors. This index quantifies the likelihood of a direct relationship between two nodes, x and y, based on their common neighbors (Eq. 2). The variable k(u) denotes the degree of the common neighbor u, which corresponds to the number of connections for node u. It assigns greater importance to common neighbors with fewer connections. The reason for this is that nodes with less connections (lower degree) are deemed more influential in establishing fresh links with the nodes they are connected to (Shang et al. 2019b; Ou et al. 2007).

$$S(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{k(u)} \tag{2}$$

2.1.1.3 Hub promoted index (HPI) The HPI (Regan et al. 2002; Shang et al. 2019b) approach predicts network links by boosting the influence of hub nodes, which encourages the establishment of links with high-degree nodes. The system runs based on the idea that nodes with a high number of connections (referred to as hubs) are more inclined to establish ties with one another. The HPI assigns a higher score to pairs of nodes that have a large

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

number of common neighbors, especially if one of the nodes is a hub. The Eq. 3 calculates the Hub promoted index, here $h(x)$ and $h(y)$ represents node degree of x and y.

$$S(x, y) = \frac{|N(x) \cap N(y)|}{\min\{h(x), h(y)\}} \tag{3}$$

2.1.1.4 Hub depressed index (HDI) The HDI (Lü and Zhou 2010; Shang et al. 2019b) is a link prediction technique designed to decrease the probability of establishing connections between hub nodes and nodes with low degrees. This index allows the establishment of connections between nodes with similar degrees, hence preventing the generation of links that may result in excessively centralized network architectures (Eq. 4).

$$S(x, y) = \frac{|N(x) \cap N(y)|}{\max\{h(x), h(y)\}} \tag{4}$$

2.1.1.5 Leicht–Holme–Newman index (LHNI) The LHNI (Leicht et al. 2006) is a link prediction algorithm that computes the probability of a link between two nodes by normalizing the number of shared neighbors by the product of their degrees. This means it divides the number of common neighbors by both nodes' degrees, taking into consideration their connectedness (Eq. 5).

$$S(x, y) = \frac{|N(x) \cap N(y)|}{h(x) \cdot h(y)} \tag{5}$$

2.1.1.6 Jaccard coefficient This metric is comparable to the common neighbour metric. It also normalises the above score, as seen below in Eq. 6.

$$S(x, y) = \left| \frac{N(x) \cap N(y)}{N(x) \cup N(y)} \right| \tag{6}$$

The chance of selecting common neighbours of paired vertices from all neighbours of either vertex is described as the Jaccard coefficient. The count of common neighbours in-between the two vertices investigated raises the pairwise Jaccard score (Kumar et al. 2020).

2.1.1.7 Adamic/adar index Adar and Adamic proposed a measure for calculating a similarity score among two web pages using common traits, which was later modified and utilized

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

in link prediction by Liben-Nowell et al. (Liben-Nowell and Kleinberg 2003). The following equation can be used to calculate the degree of similarity between nodes x and y (Eq. 7).

$$S(x,y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{\log d_z} \tag{7}$$

where d_z is the node z's degree. The equation clearly shows that common neighbours with lower degrees are given more weight. This is also visible in real-life circumstances; for example, a person with a higher friend count often devotes less time or resources to each individual friend than someone with fewer friends.

2.1.1.8 Preferential attachment The concept of preferential attachment is used to build a developing scale-free network. Growing refers to the progressive nature of nodes in a network over time. The chances of adding a new connection to a node x is proportional to g_x, the node's degree (Daud et al. 2020). The preferential attachment score between two nodes x and y can be calculated as shown in Eq. (8):

$$S(x,y) = g_x \cdot g_y \tag{8}$$

Al Hasan et al. (2006) demonstrated in a controlled learning environment how to compute the feature value of a connection by applying aggregation functions such as multiplication or sum on the feature values of vertices. Summation serve as an aggregate function instead of multiplication in the above equation, and it has been shown to be highly useful. Al Hasan et al. (2006) shown that preferred attachment using the aggregate function "sum" works effectively for link prediction in a co-authorship network.

2.1.1.9 Cosine similarity In a vector space similarity between documents is computed using the Cosine similarity. The Cosine of the angle between two records (documents) is used to calculate the similarity index. The metric is entirely about direction rather than magnitude.

2.1.2 Global similarity indices (GSI)

GSI indices are calculated using a network's whole topological information. Such approaches have significant computing difficulties and appear to be infeasible for big networks. The following are examples of global similarity indices.

2.1.2.1 Katz index The Katz index (Katz 1953) compares the similarity of two nodes based on the number of paths of varying lengths that connect them. The index gives paths of increasing length exponentially decreasing weights, with longer paths accounting a lower similarity score than smaller paths.The Katz index calculation for a pair of nodes u and v is represented by the Eq. (9) as

$S_{u,v}$.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

$$S_{uv} = \sum_{k=1}^{\infty} \beta^k A_{uv}^k \tag{9}$$

where $\beta(0 < \beta < 1)$is a scaling parameter that determines the relevance of longer paths, and A^k counts number of path of length k between node u and v. The Eq. (10) can be used to figure out the Katz index matrix.

$$S = \sum_{i=1}^{\infty} \beta^i A^i = (I - \beta A)^{-1} - I \tag{10}$$

here I represents identity matrix. Higher Katz index values imply that nodes are more comparable and likely to be connected in the future.

2.1.2.2 Rooted page rank To rank web sites according to their importance, the concept of PageRank (Page and Brin 1998) was first put forth. The technique is predicated on the idea that a random walker will randomly access a website with probability

α and follow an embedded hyperlink with probability (1- α). This idea was applied by Chung et al. (Chung and Zhao 2010) in their framework for link prediction along with a random walk. In a random walk, static distribution can take the place of web page relevance. The static probability of y via x in a random walk, in which the walker walks to any random neighbouring vertex with probability α and come back to x with probability (1– α), can be used to determine how similar two vertices x and y are to one another. This score can be calculated mathematically using the following Eq. (11). M indicates an adjacency matrix that has been normalised using the diagonal degree in this case.

$$r = (1 - \alpha)(I - \alpha M)^{-1} \tag{11}$$

2.1.3 Quasi local similarity indices (QLSI)

QLSI (Kumar et al. 2020) indices were originally developed as a compromise between regional and global perspectives, or between complexity and performance. These measurements can be calculated just as quickly as local indices. Some of these indexes capture the network's whole topological data. When compared to global techniques, these indexes' time complexities are still lower. Local directed path (LDP), Local random walk index, Local path index (Wang et al. 2015) , and other related indices examples are discussed below.

2.1.3.1 Local path index (LP) The LP-based metric is taken into consideration in order to provide a suitable trade-off among computational complexity and accuracy (Lü et al. 2009). The Eq. (12) can be used for mathematical representation of the metric.

$$S_{LP} = M^2 + \varepsilon M^3 \tag{12}$$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

where ε stands for an open parameter. When $\varepsilon = 0$, it is evident that the scale converges to a common neighbour. If x and y don't directly relate to one another, $(M^3)_{xy}$ represents all possible pathways between x and y that are 3 lengths long. The index can be generalised as follows (refer to Eq. 13)

$$S_{LP} = M^2 + \varepsilon M^3 + \varepsilon^2 M^4 + \ldots + \varepsilon^{n-2} M^n \tag{13}$$

where n denotes the highest order. As n increases, it gets more difficult to calculate this index.

2.1.3.2 Local random walk (LRW) and superposed random walk (SRW) Liu and Lu (2010) developed novel similarity measures based on random walks on graphs with a restricted number of steps. In comparison to other random walk-based approaches, they used less computationally intensive random walks to establish node similarity (Tong et al. 2006). Equation (14) expresses the likelihood of a random walker reaching node y in t steps, beginning at node n.

$$\pi_n(t) = M^T \pi_n(t-1) \tag{14}$$

where M^T represents the transpose of the matrix of transition probability M and $\pi_n(0)$ represents a vector of columns with the nth element as 1 and the remaining elements as 0s. The M_{ny} element of this matrix indicates the likelihood that a random walker from node n will proceed to node y. $M_{ny} = b_{ny}/k_n$. here b_{ny} is 1 when n and y are connected and 0 otherwise.The authors used the above idea to figure out the similarity score (LRW) between two nodes as shown in Eq. 15

$$S^{lrw} = \frac{k_n}{2E}\pi_{ny}(t) + \frac{k_y}{2E}\pi_{ny}(t) \tag{15}$$

This similarity metric focuses on the random walker's early movements, making it quasi-local, rather than on the stationary state, unlike other methods.

Similarity-based approaches mostly rely on the structural characteristics of networks to compute similarity scores. Local techniques typically utilise knowledge about the immediate neighbourhood or neighbours of neighbours to expedite calculation. Local techniques are suited for big real-world network datasets due to their efficiency. Global techniques consider the complete network structure, necessitating longer computation times. In a decentralised setting, full topological data not be accessible during computation, hindering the parallelization of global strategies in comparison to local and quasi-local methods.

2.2 Probabilistic maximum likelihood methods

The probabilistic model optimises an objective function to establish a model with multiple parameters for a particular network G(V, E), where V represents vertices and E represents edge connections. This model is capable of accurately estimating the observed network data. The likelihood of the existence of a non-existent link (i, j) is then assessed using con-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

ditional probability. To predict missing connections in networks, numerous probabilistic models and maximum likelihood models (Goyal and Ferrara 2018) are suggested in the literature. Typically, probabilistic models necessitate additional information, such as edge or node attribute information, with structural data. Not only is it difficult to extract this attribute information, however optimizing parameter is also a significant factor in models which restrict their applicability. Because maximum likelihood techniques are intricate and labor-intensive, such models are unsuitable for truly big networks. The following are examples of probabilistic maximum likelihood methods.

2.2.1 Local probabilistic link prediction model (LPLM)

Wang et al. (2007) introduced a LPLM approach to undirected networks. They utilised three distinct categories of extracted features, namely semantic, topological, as well as co-occurrence probability from various information sources. Authors proposed the concept of a central neighbourhood set, which is based on the local topology of the analysed node pair and is essential for identifying a probable relationship between them. They calculated non-derivable frequent itemset (items for which frequency data cannot be deduced from patterns found in other itemets) from the network's activities log data, which was then used to train the model. Using local event log data, the central neighbourhood set between x and y is first calculated. Finding the shortest path among two vertices of a certain length is a common method for locating the central neighbourhood set; vertices lying on this route can be incorporated into the required set. In the next phase, non-derivable sets of frequent items are utilized to learn LPLM for a given central neighbourhood set. Calders et al. (Calders and Goethals 2005) suggested a depth-first search approach and identical algorithm utilised by the authors (Wang et al. 2007) to compute non-derivable item sets. They discover all such item sets that are totally contained within the centre neighbourhood set. A markov random field (Kumar et al. 2020) is acquired using these itemsets. The iterative scaling algorithm (Wang et al. 2007) is employed in the final phase to learn a local MRF for the provided central neighbourhood set. This procedure continues to enforce constraints on the entire set of items and constantly adjusts the model until convergence. Once the process of model learning is complete, the concurrent occurrence probability can be inferred by calculating the minor probability across the created model. The junction tree prediction algorithm (Kumar et al. 2020) is employed for determining the probability of co-occurrence.

2.2.2 Probabilistic relational link prediction model (PRM)

PRM was originally created to predict attributes in relational data, yet it was eventually expanded to predict links (Kumar et al. 2020). To anticipate links, the authors used the attribute prediction framework.Contemplate the link prediction problem in a co-authorship network. Non-relational prediction of links frameworks accept only one entity type "person" as a node and one relationship; though, relational frameworks (PRMs) contain more entity kinds such as article, conference location, institution, and so on. Each object include attributes such as a person (affiliation institute,name,, rank (professor,student), article (publication year, kind (review,regular), and so on. Many relational links between these entities are possible, such as an advisor-advisee/research scholar association among two people, an author relationship among a paper and a person entity, and a paper can be linked to the

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

venue of conference through a publish paper relationship. Furthermore, links between these entities can contain qualities such as link exists (if there is a relationship between the two entities) or does not exist (there is no relation between the two entities). This reduces the link prediction framework to a simple attribute prediction system. During model training, a singular link graph containing the aforementioned diverse entities and their relationships is constructed. Model parameters are estimated with discrimination in order to maximise the chances of link existence along with other parameters considering graph attribute data (Getoor et al. 2003).

2.2.3 Hierarchical structure model (HSM)

HSM models assume that many actual networks have a hierarchical structure, with nodes organised into groups, which are then segmented into subgroups, and so on, at varying levels of granularity. In order to construct a model in which model parameters are estimated statistically, some exemplary work (Kumar et al. 2020) routinely encodes such structures from network data. The chance of newly formed, unseen linkages is then estimated using these values.

2.2.4 Stochastic block model (SBM)

The majority of networks may not be represented by hierarchical structures. The block model is a larger way to representing these networks, in which vertices are classified as blocks or communities. The probability of a link between two vertices is determined by their respective blocks. Guimera et al. (Guimerà and Sales-Pardo 2009) proposed a unique method for discovering missing and incorrect links in a network using a stochastic block model representation of the network.Given an observed network, the authors compute the probability of the existence of links, which is then used to identify missing links (non-existent links that have greater probabilities) and existing links that have lower probabilities.

2.2.5 P-star model or exponential random graph model (ERGM)

Holland and Leinhardt (1981) were the first to investigate exponential random graphs; (Frank and Strauss 1986) investigated them further; and (Pattison and Wasserman 1999) employed them practically. ERGM is an composite model in which it is defined as a collection of all basic undirected graphs and a probability is specified for every graph in ensemble. The ERGM's properties are calculated by aggregating over the ensemble. Pan et al. (2016) also proposed a comparable framework (ERGM) to identify absent and spurious network links.

2.3 Matrix factorization for feature extraction

Matrix Factorization is a crucial technique in feature extraction, renowned for its capacity to condense extensive and intricate datasets into more comprehensible and controllable formats. Essentially, it entails breaking down a huge matrix, like an adjacency matrix in network research, into a multiplication of smaller matrices. The process of decomposition uncovers latent elements or aspects that are inherent in the original data, offering useful

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

insights into the concealed patterns and linkages (Keyvanpour and Moradi 2014). Matrix Factorization, when applied to network analysis, can effectively detect intricate linkages and clusters inside the network, revealing its structural complexities. This approach excels at reducing dimensionality and minimising noise, improving the clarity and usefulness of the retrieved features. Matrix Factorization simplifies data analysis by decomposing large datasets into their fundamental components (Haghani and Keyvanpour 2019).

2.3.1 Singular value decomposition (SVD)

The SVD (Sadek 2012) is a crucial matrix factorization technique employed in diverse domains, such as network research for predicting links between entities. The SVD of an adjacency matrix $\mathbf{A}$ is expressed by Eq. (16)

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}} \tag{16}$$

where:

- $\mathbf{U}$ is a $m \times m$ orthogonal matrix, with its columns representing the left singular vectors of $\mathbf{A}$. The given vectors constitute a set of mutually perpendicular unit vectors that span the range of matrix $\mathbf{A}$. In the context of networks, they can be understood as representing the connection patterns based on the emission of nodes.
- $\mathbf{D}$ is a diagonal matrix of size $m \times n$ that contains the singular values of $\mathbf{A}$. These numbers are genuine and non-negative. They are arranged in decreasing order based on their magnitude and indicate the "strength" of the singular vectors they belong to. The non-zero elements in matrix $\mathbf{D}$ correspond to the square roots of the non-zero eigenvalues of both the matrices $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$.
- $\mathbf{V}^T$ is the transpose of a $n \times n$ orthogonal matrix $\mathbf{V}$, where each column of $\mathbf{V}$ represents a right singular vector of $\mathbf{A}$. The right singular vectors constitute an orthonormal basis for the range of matrix $\mathbf{A}^T$ and represent the connection patterns as perceived by the receiving nodes.The goal of link prediction is to deduce missing connections or anticipate new interactions among nodes in diverse domains such as social networks, cooperation networks, and biological networks. To effectively handle this problem, we can employ truncated Singular Value Decomposition (SVD). In this approach, we replace the matrix $\mathbf{D}$ with a matrix that only keeps the greatest singular values. Additionally, we maintain the associated vectors in $\mathbf{U}$ and $\mathbf{V}$. The condensed representation captures the most important connections in the network, enabling the estimation of unseen linkages by applying a threshold to the reconstructed adjacency matrix. The primary singular values and vectors play a vital role in anticipating missing connections, as they include fundamental information about the structure of the network. By utilising SVD to uncover underlying patterns and similarities, it becomes possible to make predictions regarding missing linkages or future connections. The SVD is widely employed across various domains to infer missing links and anticipate future interactions. It offers a computationally efficient and effective approach for link prediction (Peng et al. 2022; Trouillon et al. 2016).

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Nevertheless, conventional techniques for extracting features such as SVD, while useful, are limited in their ability to capture intricate and non-linear relationships within data. Frequently, they depend on predetermined characteristics, which might not sufficiently capture all the complex patterns and dynamics found in actual networks. Learning-based strategies are crucial in this context. Feature learning, particularly deep learning techniques, have the ability to autonomously uncover the fundamental characteristics and complicated patterns from unprocessed data, providing a more robust and adaptable strategy for comprehending and forecasting difficult network structures. These methods employ adaptive learning to identify the most pertinent properties for the task at hand, resulting in more precise and resilient link prediction models. This is particularly advantageous in situations when the structure and connections of the network are ill-defined or constantly changing.

2.3.2 Non-negative matrix factorization (NMF)

NMF is a specific method of breaking down a matrix into its constituent parts, commonly used to extract features in predicting links in network research. NMF seeks to break down a non-negative adjacency matrix of a network into two non-negative matrices, typically referred to as X and Y. In the context of link prediction, the variable X represents a matrix that measures the degree to which a node is connected to certain latent factors. On the other hand, the variable Y represents a matrix that illustrates the interactions between these factors (Nasiri et al. 2023). NMF utilises the decomposition process to reveal underlying features or elements that significantly influence the structure of the network. The matrix resulting from the multiplication of X and Y can be utilised to forecast potential connections, particularly in networks where non-negativity is an important limitation, such as social or biological networks. Enforcing non-negativity in NMF improves the interpretability of the components, hence boosting the intuitive comprehension of link prediction results (Chen et al. 2022b, 2023).

The capacity of NMF to do low-rank approximation, which is a fundamental aspect of matrix factorization, allows it to effectively represent the key features of the network in a space with fewer dimensions. NMF enables the prediction of missing or future connections by analysing the interactions between nodes and latent factors. The application of this method extends to several fields, such as recommender systems and biological networks, where its capacity to impose non-negativity constraints results in simply understandable and significant outcomes. NMF is a powerful technique for gaining a thorough understanding and making accurate predictions about connections in complicated networks (Chen et al. 2022a). Moreover, by making specific adjustments to its implementation, such as using regularisation terms or incorporating it into a wider machine learning pipeline, NMF can evolve from solely extracting features to adopting a more flexible feature learning strategy. The versatility of NMF enables it to extract and learn features, hence increasing its effectiveness in difficult data analysis situations.

In this section, we explored the complexities of Feature Extraction Techniques essential to the field of Link Prediction Methods. The techniques can be classified into three main types: Similarity Based Methods, Probabilistic Maximum Likelihood Methods, and Matrix Factorization Techniques. Similarity Based Methods,utilise data from immediate surroundings, the overall network structure, or a combination of local and global data to predict possible links. Then there are, Probabilistic Maximum Likelihood Methods, which

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

involve models that use a statistical technique to determine the likelihood of link development, adding a level of stochastic analysis to the prediction process. Finally, Matrix Factorization Techniques have been analysed, which involve using methods like Singular Value Decomposition and Non-negative Matrix Factorization to break down and extract structural patterns from the network's adjacency matrix. This thorough examination of Feature Extraction Techniques improves the precision of Link Prediction Methods and highlights the delicate equilibrium between methodological complexity and practical applicability in intricate network settings.

3 Link prediction—learning based approaches

Previously described methods (such as similarity and probabilistic methods) compute a score for each unobserved link using a similarity function.The link prediction problem can be approached using a learning-based model that takes into account both the topological aspects of the graph and attribute data. Learning-based approaches for link prediction utilise machine learning techniques to predict the existence or probability of links between entities in a network (Balvir et al. 2023). These methods include data representation, feature extraction, training data preparation, choosing the model, training, evaluation, and forecasting. Key aspects of learning-based link prediction methods include following points :

- Learning-based approaches employ machine learning models to identify complex network patterns and interdependencies (Waikhom and Patgiri 2023).
- The network is represented as a graph, and pertinent information is extracted from the graph using graph features (Zhao et al. 2022).
- Positive and negative examples are used to train the model using labelled data (Barros et al. 2021).
- Different machine learning models, such as supervised classifiers, neural networks, and probabilistic models, can be utilised. Some of these models, known as graph embedding-basedor representation learning models for link prediction, seek to learn graph encoding, node, and/or domain-related properties into low-dimensional space (Waikhom and Patgiri 2023).
- The model is trained using labelled data to discover network characteristic patterns and dependencies (Barros et al. 2021).
- Utilising metrics such as F1-score, precision, recall, and AUC-ROC, the trained model is evaluated.
- Once the model has been trained, it can be utilised to predict the chances of connections between entities in new, unseen data (Kumar et al. 2020; Zeng et al. 2013).
- The benefits of learning-based methods include scalability, the ability to manage complex patterns, and the incorporation of a wide range of features (Waikhom and Patgiri 2023).Link prediction strategies based on learning can be classified into three main types: Matrix Factorization, Path and Walk-Based, and Neural Network-Based methods, each with its own subclasses and specialised approaches.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3.1 Matrix factorization/decomposition for feature learning

Matrix Factorization, when used for feature learning, goes beyond the conventional boundaries of feature extraction by providing a flexible technique that adjusts and acquires knowledge from the data itself. Contrary to traditional methods of extracting features, which mainly aim to decrease dimensionality and eliminate noise, feature learning using Matrix Factorization is focused on uncovering and capturing the fundamental structure of the data in a manner that is advantageous for predictive modelling (Salakhutdinov and Mnih 2008). This sophisticated method enables the detection of hidden variables that may not be readily observable in the original data but are essential for comprehending intricate patterns and connections. Matrix Factorization approaches, such as Probabilistic Matrix Factorization and Deep Matrix Factorization, are employed in the domain of feature learning. These techniques have the ability to reveal concealed patterns and frameworks in extensive datasets, such as finding underlying themes in collections of written texts or discovering inherent user inclinations in recommendation systems. The primary benefit is in their capacity to acquire and depict non-linear connections, which are frequently overlooked by conventional extraction techniques, so offering a more intricate and all-encompassing comprehension of the data (Lyu et al. 2017).

Moreover, the shift from extracting features to learning features through Matrix Factorization indicates a move towards more advanced and adaptable techniques for analysing data. Although feature extraction is highly beneficial for simplifying data and uncovering its fundamental structure, it frequently depends on pre-established assumptions and lacks the adaptability to capture the dynamic and intricate characteristics of real-world data. Conversely, feature learning via Matrix Factorization exhibits dynamic adaptability to the data, acquiring knowledge from it and evolving as additional data is made accessible. This flexibility renders it very efficient in situations where data is not fixed and continues to expand or alter, such as in dynamic social networks or shifting market trends (De Handschutter et al. 2021). The acquired features are not merely compressed representations of the data, but rather sophisticated, data-driven insights that can greatly improve forecast accuracy and the overall excellence of machine learning models. Matrix Factorization for feature learning is an advanced and forward-thinking method that can effectively handle the complexity and dynamism found in modern data sets. This allows for more precise predictions and a deeper understanding of the data.

3.1.1 Deep matrix factorization (DMF)

DMF is a substantial advancement over traditional matrix factorization methods inside learning-based methodologies. DMF utilises deep learning techniques to effectively reveal intricate and multi-layered hidden structures in data. This makes it an essential tool in fields that demand sophisticated feature extraction and pattern identification. This is especially apparent in advanced learning situations where comprehending complex data linkages is essential. The discerning capacity of DMF to identify these nuanced relationships significantly improves learning models, resulting in more precise and anticipatory insights across a wide range of domains, including advanced recommendation systems and complex analysis of image and bioinformatics data (Chen et al. 2022b). DMF is a powerful feature learning tool, particularly for link prediction tasks in complicated networks. DMF excels at portray-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

ing complicated network features, which are necessary for anticipating potential link forms, by exposing latent hierarchical structures. This method is especially useful in scenarios such as social network analysis or biological network mapping, where correct modelling of linkages and interactions is critical. DMF's capacity to learn and express these deep properties not only improves link prediction accuracy but also leads to a deeper understanding of network dynamics (Fan et al. 2019; Ying et al. 2018). Following are the examples of DMF techniques :-

3.1.1.1 Deep orthogonal non-negative matrix factorization (Deep ONMF) Deep ONMF is a sophisticated method of decomposing matrices, which plays a vital role in machine learning for tasks such as extracting features and predicting links (Lyu et al. 2017). The method builds upon the conventional Non-negative Matrix Factorization (NMF) by incorporating hierarchical and orthogonal restrictions. This allows for the extraction of intricate and layered characteristics from the given data.

The first stage entails decomposing a data matrix M into two matrices A_1 and W_1 (Eq. 17)

$$M \approx A_1 W_1^T \tag{17}$$

with the condition $W_1^T W_1 = I$ to assure orthogonality, here I is an identity matrix. The orthogonality present in W_1 is crucial, since it facilitates the preservation of separate and non-overlapping characteristics. In the next layers, A_{i-1} is divided into A_i and W_i, with each layer following the orthogonality principle ($W_i^T W_i = I$). The hierarchical method employed by Deep ONMF enables it to effectively capture various levels of abstraction present in the data. This is especially advantageous when dealing with intricate tasks such as link prediction in networks. The process of fine-tuning the model entails minimising the cost function ($Deep\_c$) while adhering to the restriction $W_i^T W_i = I$ for each layer (Eq. 18).

$$Deep\_c = \frac{1}{2}\|M - A_1 W_1^T \dots W_i^T\|_F^2 \tag{18}$$

The process of minimising is essential in order to enhance the factor matrices and accurately depict the underlying patterns and relationships within the data. In order to modify the hidden layers, the cost function is modified as Eq. 19

$$Deep\_c = \frac{1}{2}\|MW_i \dots W_{l+1} - A_1 W_1^T \dots W_l^T\|_F^2 \tag{19}$$

Equation 19 allows for updates of W_l while preserving orthogonality. This guarantees that every layer makes a valuable contribution to the overall structure, accurately representing hierarchical characteristics in the data. In the given Eq. 19, the subscript F denotes the Frobenius norm. It is a metric that quantifies the size or magnitude of a matrix and is computed by taking the square root of the sum of the absolute squares of all the elements in the matrix. Within this particular framework, the term is employed to measure the discrepancy or discrepancy between the initial matrix M and its estimation obtained through the process

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

of factorization. Minimising this norm is crucial for optimising the factor matrices in Deep ONMF.

Deep ONMF is a matrix factorization technique that improves the processing of complex data. The process begins by breaking down a data matrix into separate layers, each representing various degrees of features. This is done while ensuring that the layers are orthogonal to one another, in order to preserve the distinctiveness of each feature. The procedure progressively improves these layers, with the goal of minimising the discrepancy between the original and reconstructed data. This approach is particularly efficient for jobs such as link prediction, where comprehending hierarchical data relationships is essential. The layered and orthogonal structure of Deep ONMF enables the identification of complex data patterns, rendering it a potent tool for sophisticated machine learning tasks (Chen et al. 2022b). Through this layered and orthogonal approach, Deep ONMF effectively unravels complex data structures, making it highly suitable for link prediction and similar tasks that require understanding multi-level relationships and attributes in data (Chen et al. 2022a).

3.1.1.2 Sparse deep nonnegative matrix factorization (SDNMF) It is an advanced method that is crucial in the fields of feature learning and link prediction (Guo and Zhang 2020). This technique is highly skilled at revealing concealed structures and patterns within extensive datasets, which is essential for extracting features and predicting connections or links within the data. SDNMF method expands upon classic matrix factorization techniques by using a multi-layer hierarchical approach. Each layer plays a role in enhancing the comprehension and refining of features. The stratified framework is especially beneficial in delicate situations like social network analysis, recommendation systems, and bioinformatics, where comprehending the complex network of connections is crucial. The main objective of SDNMF is to break down a matrix M into a sequence of weight matrices W_b and coefficient matrices C at various layers represented by k. The incorporation of nonlinear transformations via a function f at each layer significantly improves the model's ability to capture non-linear relationships and patterns, rendering it a resilient tool for feature acquisition and link prediction.

The initial phase in SDNMF involves matrix factorization, where the main matrix M containing the data is broken down into a weight matrix W_b and a coefficient matrix C. The objective is to precisely depict M by multiplying W_b and C, while ensuring that both matrices consist solely of nonnegative values. The following equation 20 represents a crucial technique for extracting latent characteristics from the dataset.

$$M \approx W_b C \tag{20}$$

This stage is essential for initially revealing latent patterns and characteristics within the data. Growing on this, SDNMF adds the concept of hierarchical decomposition as shown in equation 21

$$f(C_{k-1}) \approx W_{bk} C_k \tag{21}$$

This stage employs a stratified method to enhance the feature extraction procedure. Each layer k goes further into the data structure, with C_{k-1} being the coefficient matrix obtained

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

from the previous layer, and W_{bk} and C_k being the weight and coefficient matrices of the current layer, respectively. The function f, which may be either linear or nonlinear, is applied to the coefficient matrix of each layer. This feature guarantees that nonnegativity is maintained across the matrices, while also allowing the model to capture more detailed and complicated data structures. The shift from the initial matrix factorization to hierarchical decomposition indicates a gradual improvement in the model's capacity to extract and represent more complex and subtle characteristics in the data. The model incorporates a crucial element known as the Transformation Function f, following a hierarchical decomposition. The function f is crucial in the model's architecture as it is responsible for converting the coefficient matrices at each layer while guaranteeing that their nonnegative feature is preserved. The main purpose of this is to adjust and illustrate complex connections within the data, which is essential in the fields of feature learning and link prediction. Linear models frequently fail to capture intricate patterns present in real-world data. However, the incorporation of f in SDNMF enables a more sophisticated representation, particularly in situations when data structures display nonlinear attributes. The capacity of f to alter and preserve the integrity of the data is what allows SDNMF to explore data analysis at a higher level, uncovering more significant and complex characteristics that linear methods may fail to consider.

Expanding upon this advanced method of transformation, SDNMF applies sparsity constraints to further enhance the feature representation. This is accomplished by applying L1-norm penalties on the matrices W_b and C. By employing this approach, the model promotes a greater level of sparsity in these matrices, favouring the presence of zeros or elements that are close to zero. The sparsity serves not only as a mathematical convenience, but also as a practical tool for improving the interpretability and simplification of the data format. Within intricate datasets, where understanding the fundamental arrangement and relationships might be overwhelming, this induced sparsity aids in isolating the most important characteristics, thus streamlining the process of extracting features and predicting linkages. The Optimisation phase of SDNMF aims to align the factorised matrices with the original matrix M while adhering to the sparsity requirements. The objective is to minimise the reconstruction error, which is defined as the discrepancy between M and the product W_bC. It is essential to perform this step as it ensures a proper balance between precisely representing the original data and considering the limitations and goals of the model. SDNMF optimises the relationship to ensure that the extracted features and patterns are both sparse and interpretable, while also closely aligned with the dataset's underlying structure. This makes it a powerful tool for learning features and predicting links in diverse and complex data environments. Through these steps, SDNMF emerges as a powerful tool in feature learning and link prediction, capable of extracting nuanced and interpretable features from complex datasets.

3.1.2 Probabilistic matrix factorization (PMF)

Salakhutdinov et al. (Salakhutdinov and Mnih 2008) introduced a sophisticated method for collaborative filtering. PMF is a widely utilised probabilistic approach within the field of collaborative filtering and recommendation systems. The fundamental premise of this approach entails breaking down a matrix that represents the interactions between users and items into the multiplication of two matrices with low ranks, which correspond to the charac-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

teristics of users and items. The values within the resulting approximation matrix represent the anticipated levels of interaction between users and items. The PMF method employs a probabilistic framework in which it assumes a Gaussian distribution for the observed elements included in the original matrix. This approach treats the task as a maximum likelihood estimation problem. The PMF's ability to incorporate probabilistic elements enables it to effectively represent and account for uncertainty within the data, hence offering a measurable degree of confidence in the accuracy of forecasts. The model is designed to optimise the probability of the observed data, resulting in reduced-dimensional representations that capture underlying characteristics of individuals and things. This improves the performance of recommendation and link prediction tasks, especially when dealing with limited user interactions in extensive datasets (Berahmand et al. 2023; Han et al. 2022).

One notable benefit of employing the PMF approach is its capacity to effectively strike a balance between accurately representing data and mitigating the risk of overfitting. This is particularly advantageous for users who possess limited interaction histories. The goal function incorporates the sum of squared errors and quadratic regularisation in order to achieve a trade-off between accuracy and regularisation. The introduction of a modified probability mass function version, which integrates restrictions derived from the premise that users who exhibit similar interaction patterns also have similar preferences, yields notable enhancements in predicting user preferences with limited interactions. This modified approach outperforms conventional matrix factorization techniques (Jain et al. 2023). In general, the PMF model demonstrates a high level of proficiency in accurately forecasting user preferences across large datasets that have limited data points. This highlights its durability and promise as a reliable tool in recommendation systems. In the subsequent part, we will analyse the efficacy of matrix factorization methodologies on different data sets. Five widely used academic databases in the field of research are Cora, Citeseer, PubMed, Ego-Facebook and Wiki-Vote. Initially, we will examine the datasets individually, with a specific emphasis on their nodes and links.

3.1.3 Homogeneous datasets

Within the domain of network science and computational research, databases like as Cora (Veličković et al. 2017), Citeseer (Veličković et al. 2017), PubMed (Veličković et al. 2017) , Ego-Facebook (Leskovec and Krevl 2014), and Wiki-Vote (Leskovec and Krevl 2014) play a crucial role in comprehending intricate network architecture and dynamics.There are many homogeneous datasets available (Leskovec and Krevl 2014); we have chosen five of them based on their unique characteristics and their extensive use in state-of the-art.

- Cora :- The Cora dataset is a well-known benchmark in the field of machine learning, serving as a platform for testing various graph-based learning methods that go beyond conventional models (Fig. 5). The dataset comprises 2708 scientific papers represented as nodes, connected by 5429 undirected citation linkages. Within the Cora dataset, each node corresponds to a scientific publication or document. The publications are categorised into seven classes representing specific fields of computer science study, including Neural Networks, Probabilistic Methods, Genetic Algorithms, and more. The papers are represented as binary word vectors with 1433 features indicating the presence or absence of specific terms from a predetermined vocabulary. The network's intricate cita-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

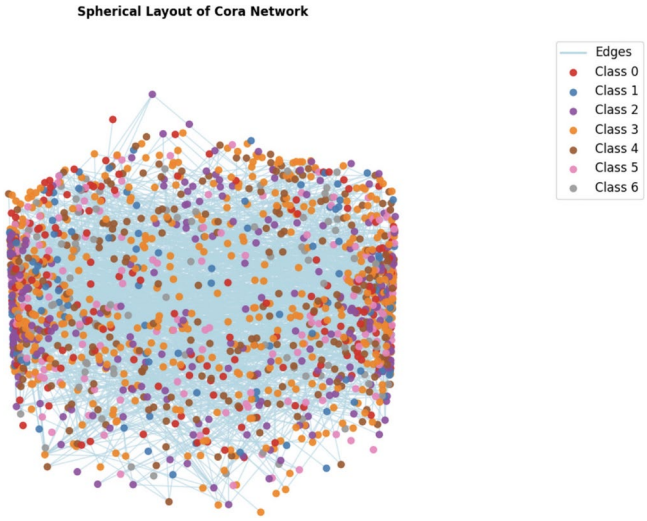


Fig. 5 The Cora dataset is shown using a spherical architecture, where nodes are categorized into seven distinct groups, each represented by a unique color. The nodes are linked by edges to represent the links between the documents

tion structure, along with the sparse and high-dimensional feature representation, poses a difficult situation for learning algorithms, especially in tasks such as node categorization, link prediction, and community discovery. The complex organisation and extensive features of the Cora dataset allow for a thorough evaluation of learning methods, especially those capable of utilising the inherent relational information in graph data. This makes it a crucial tool for assessing progress in learning approaches in computer science research.

- Citeseer :- The CiteSeer dataset is a helpful tool for assessing learning-based methods, especially in the realm of graph-based models. The dataset comprises 3327 scientific publications classified into six unique categories: Agents, Artificial Intelligence, Database, Human-Computer Interaction, Machine Learning, and Information Retrieval. These classes cover many research areas in computer science. The collection contains a citation network with 9104 edges, each representing a citation connection between two documents (Fig. 6). The network's average node degree is 5.47, highlighting the interconnectedness of the scientific papers. The description of each document is represented by a binary word vector with a length of 3703, indicating the presence or absence of particular terms from a predefined vocabulary. The high-dimensional feature space, along with the network topology, poses a challenging challenge for graph-based learning algorithms. The CiteSeer dataset is frequently used as a standard for evaluating different graph-based learning methods. It allows academics to evaluate and improve their strategies for tasks such as node categorization, connection prediction, and community detection.
- PubMed :- The PubMed dataset is a significant resource for learning-based techniques in graph-based models, especially in the field of biomedical research (Fig. 7). The dataset consists of 19,717 scientific papers, each depicted as a node in a citation network with 88,648 edges. The papers are classified into three categories: Diabetes Mellitus

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

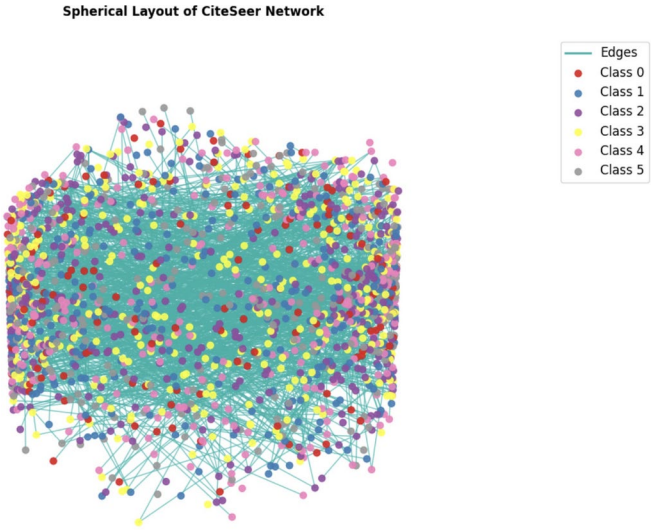


Fig. 6 CiteSeer dataset is shown using a spherical arrangement, where nodes are classified into six categories and highlighted with different colors. The nodes are connected by edges, which reflect citation linkages between publications

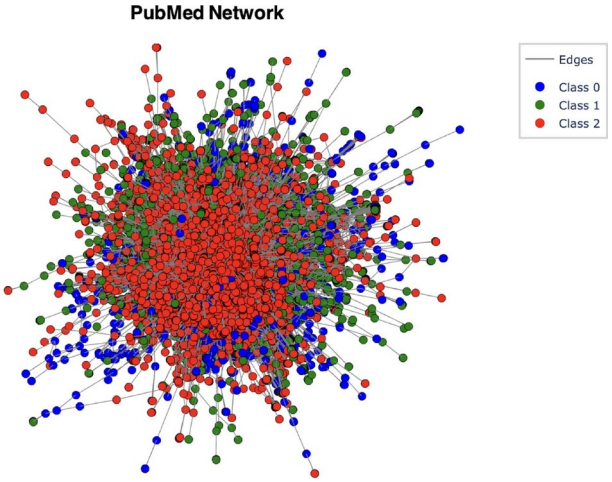


Fig. 7 PubMed dataset is visualized as a network graph, with nodes representing scientific papers that are classified into three distinct groups (Class 0, Class 1, Class 2). The connections between nodes are represented by edges, which indicate citations similarities

Experimental, Diabetes Mellitus Type 1, and Diabetes Mellitus Type 2, indicating a concentration on research connected to diabetes. Every publication is represented by a 500-dimensional feature vector, which indicates the presence or absence of particular terms from a specified vocabulary. The average node degree in the network is around 8.99, suggesting a moderately dense citation network. The PubMed dataset is commonly utilised as a standard for testing graph-based learning algorithms. It is frequently employed to evaluate models' effectiveness in tasks like node classification, link prediction, and community detection within the field of biomedical literature.

- Ego-Facebook :- The ego-Facebook dataset, obtained from the Stanford Network Anal-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

ysis Project (SNAP), is a great resource for analysing the complex dynamics of social media networks. The network comprises 4039 nodes representing individual Facebook users and 88,234 edges indicating friendships between these users. This dataset demonstrates the dense connection typical of social networks, with an average node degree of 43.69. It is commonly utilised in research to analyse social network architecture, including community detection, centrality metrics, and information spread. The dataset is essential for evaluating the effectiveness of network analysis algorithms and gaining insights into the organisational structures of online social platforms. Figure 8 displays the ego-Facebook dataset visualisation, illustrating the intricate network of relationships inside a social media context. This visualisation is essential for scholars and professionals in the field of network science.

- Wiki-Vote :- The Wiki-Vote dataset is a significant resource for analysing the dynamics of social influence and authority in online communities. The network consists of 7115 nodes and 103,689 directed edges, representing the voting interactions between Wikipedia users during the election of administrators. This dataset has a high average node degree of 29.15, indicating a dense network structure that allows researchers to study trust, leadership, and reputation trends in a collaborative environment. Scholars can analyse the directed edges of users' votes to understand how agreement is formed and power is distributed in a peer-driven setting. The Wiki-Vote dataset is used to demonstrate graph-based learning techniques and social network analysis. It provides insight into how individual actions and collective outcomes interact on online platforms, as seen in Fig. 9.

The Cora, Citeseer, PubMed, Ego-Facebook, and Wiki-Vote datasets are examples of homogenous networks commonly used in graph-based learning. Cora and Citeseer are citation networks that consist of scientific articles connected by citation linkages. PubMed is a network of biomedical literature where research articles are represented as nodes and citations as edges. The Ego-Facebook dataset records interactions among Facebook users, while the Wiki-Vote network depicts voting behaviour of Wikipedia users. These datasets listed in Table 1 are used as benchmarks to assess the effectiveness of different learning models in tasks including node categorization, connection prediction, and community discovery.

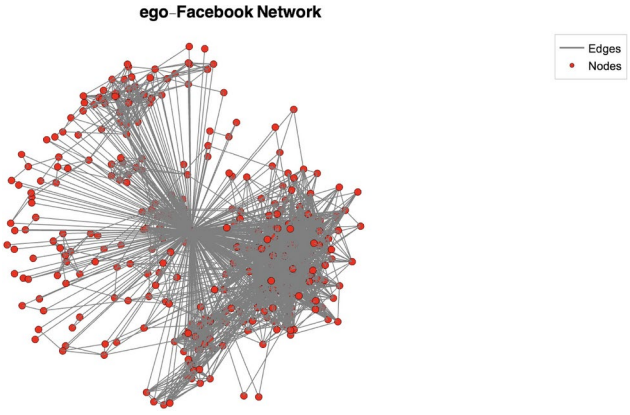


Fig. 8 The Ego-Facebook dataset depicts a network graph with nodes representing particular Facebook users and edges indicating friendships or connections between them

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

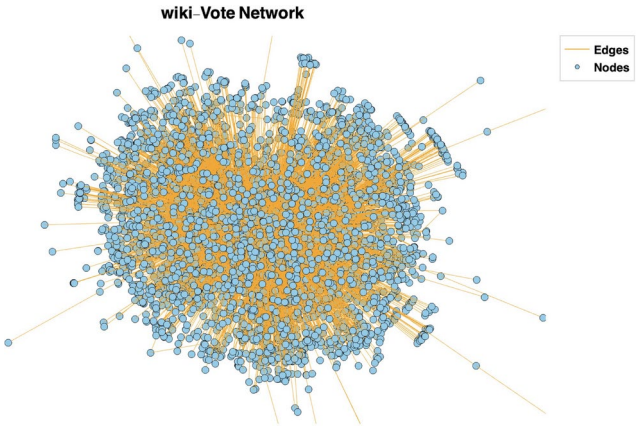


Fig. 9 The Wiki-Vote dataset is a representation of a network of Wikipedia users. In this network, each person is depicted as a node, and their edges indicate that one person voted for another in administrative elections

Table 1 Specifics regarding the Cora, Citeseer, PubMed, Ego-Facebook, and Wiki-Vote databases

Dataset	Nodes	Edges	Average degree
Cora	2708	5429	4.01
Citeseer	3327	9104	5.48
PubMed	19,717	88,648	8.98
Ego-Facebook	4039	88,234	43.69
Wiki-Vote	7115	103,689	29.15

Upon obtaining a thorough comprehension of the datasets, it is now significant to assess the efficacy of the corresponding methodologies on these datasets.

3.1.4 Results of matrix decomposition techniques on homogeneous datasets

This section addresses the experimental setup for learning-based Matrix Factorization/ Decomposition Techniques. We employ and implement three advanced methodologies: Deep ONMF , Sparse Deep NMF, and PMF on various established datasets. The databases that fall under this category are Cora, Citeseer, Pubmed, Ego-Facebook, and Wiki-Vote. Before discussing the exact details of each technique, we begin by giving a thorough summary of the datasets, emphasising their distinct characteristics and the particular difficulties they pose. Having a solid grasp of the datasets provides a basis for a more sophisticated implementation and assessment of the matrix factorization/decomposition techniques. The efficacy of Deep ONMF, Sparse Deep NMF, and PMF is thoroughly assessed using a comprehensive set of metrics: Macro F1, Micro F1 scores, and AUC curve plots.

The AUC (Area Under the ROC Curve) is a fundamental parameter for assessing the performance of link prediction algorithms. It assesses the model's ability to differentiate between positive examples (real linkages) and negative examples (non-existent links). Specifically, the AUC shows the likelihood that a randomly chosen positive example will be ranked higher than a randomly picked negative example. The AUC score ranges from 0.5 (random performance) to 1.0 (perfect prediction accuracy). A high AUC score implies that

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

the link prediction algorithm is effective at differentiating between positive and negative examples. This metric is particularly valuable because it provides a single scalar value summarizing the model's overall performance across all threshold levels, making it ideal for comparing models (ke Shang et al. 2017a).

The Macro-F1 Score computes the F1 score for each class separately and then averages these results to produce a complete assessment of performance across all classes. In this application, a class denotes a specific category or label to which an instance might be given, such as "cat," "dog," or "bird" in a creature classification task. The Macro-F1 result assures that every class participates equally to the overall result, regardless of how many instances exist in each class. This strategy is especially beneficial in settings with imbalanced datasets because it assesses how well the model performs on each class separately, providing a more balanced perspective of performance.

In contrast, the Micro-F1 Score calculates a single F1 score by aggregating false positives (FP), false negatives (FN) and true positives (TP) across all classes. Here, TP stands for the number of properly anticipated positive cases, FP for the number of mistakenly predicted positive instances, and FN for the number of missed positive instances. By adding these values from all classes, the Micro-F1 Score indicates the model's overall ability in properly categorizing instances, providing a global picture of accuracy that is especially relevant for datasets with class imbalance.

This methodology guarantees a comprehensive evaluation of the capability of each technique to precisely capture and understand the intricate network patterns that are present in these varied datasets. This establishes a reliable standard for future experimental comparisons with strategies based on paths.

3.1.4.1 Deep ONMF The Table 2 and Fig. 11 illustrates the effectiveness of the Deep ONMF (Lyu et al. 2017) technique in the field of link prediction across multiple datasets. Link prediction is an important task in network research that involves predicting probable links inside a graph-based representation. It is critical for gaining insights into the underlying structure of networks in several domains, including social media, citation networks, and biological data.

The performance metrics consist of the Micro-F1 and Macro-F1 scores, which are calculated as the harmonic means of precision and recall. The Micro-F1 score provides an overall assessment of performance, while the Macro-F1 score offers an unweighted average that takes into account class imbalance. In addition, the report includes the Area Under the Receiver Operating Characteristic (AUC) Score, which indicates the model's capacity to differentiate between different classes. The recorded scores for the Micro-F1 and Macro-F1 range from 0.5935 to 0.8822, while for the AUC they range from 0.6415 to 0.9453. These results indicate differing levels of accuracy in predicting outcomes across different datasets. The F1 ratings for citation networks such as Cora, Citeseer, and PubMed are slightly higher

Table 2 Deep ONMF performance for Link Prediction on datasets

Dataset	Micro-F1	Macro-F1	AUC score
Citeseer	0.5745	0.5935	0.6415
Cora	0.6185	0.6166	0.6499
PubMed	0.6293	0.6291	0.6781
Ego-Facebook	0.6877	0.6868	0.7586
Wiki-Vote	0.8822	0.8822	0.9453

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

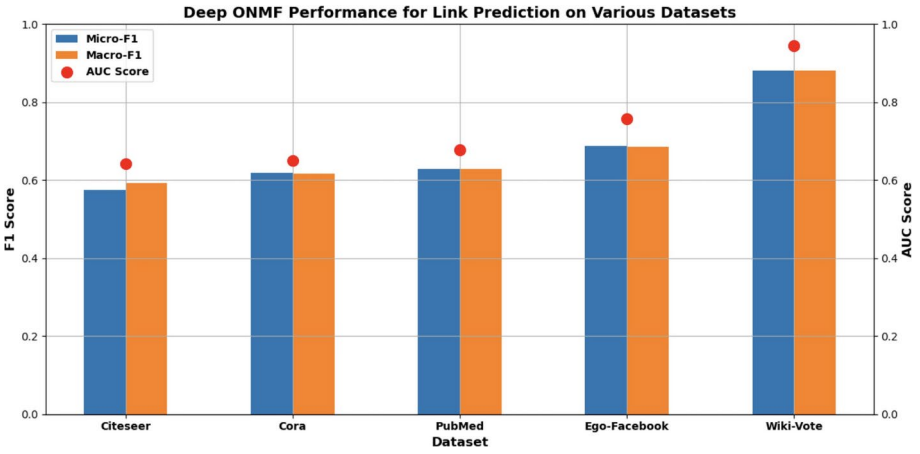


Fig. 11 Deep ONMF performance for link prediction across multiple datasets, including CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote. The bar graph depicts the Micro-F1 and Macro-F1 scores for each dataset, and the red circles indicate the AUC values, offering a full comparison of the model's efficacy. (Color figure online)

than the baseline. This indicates that the Deep ONMF model performs moderately well in predicting citation relationships between scholarly publications. The AUC ratings, although significantly higher, also suggest that the model's ability to distinguish between existing and non-existent linkages might be improved. The middling results indicate the intricate structure of citation patterns, which could be affected by aspects that the model does not fully account for. In contrast, the model has exceptional performance on datasets that represent social interactions, specifically Ego-Facebook and Wiki-Vote. The Ego-Facebook dataset, which is believed to represent social ties, has enhanced performance, achieving an AUC of 0.7586, indicating a greater level of prediction accuracy. The model attains its highest level of performance on the Wiki-Vote dataset, with F1 scores that approach 0.9 and an AUC that comes close to the ideal threshold of 1. This demonstrates the model's strong ability to accurately forecast administrative endorsements, which have more clear and easily identifiable patterns compared to the citation networks.

The Deep ONMF model's performance in link prediction differs across different datasets due to various fundamental dataset variables, as seen in Fig. 10. The Wiki-Vote dataset likely excels because to its higher average degree and denser network structure, which offer more information for accurate link prediction in the algorithm. Conversely, the Citeseer and Cora datasets, which have lower scores, exhibit sparser structures and lower average degrees, making link prediction more challenging. The Ego-Facebook dataset achieves a reasonable performance by balancing density and sparsity, resulting in improved outcomes compared to Citeseer and Cora, but not as high as Wiki-Vote. PubMed's network topology is quite thick, which enables reasonably good link predictions. The density and connectedness of the network are essential factors in determining the success of link prediction algorithms.

3.1.4.2 Sparse deep NMF Sparse Deep NMF (Guo and Zhang 2020) exhibits exceptional performance on multiple datasets such as Cora, Citeseer, PubMed, Ego-Facebook, and Wiki-Vote, as shown in Table 3 and Fig. 13. The AUC values are remarkable, with the model

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

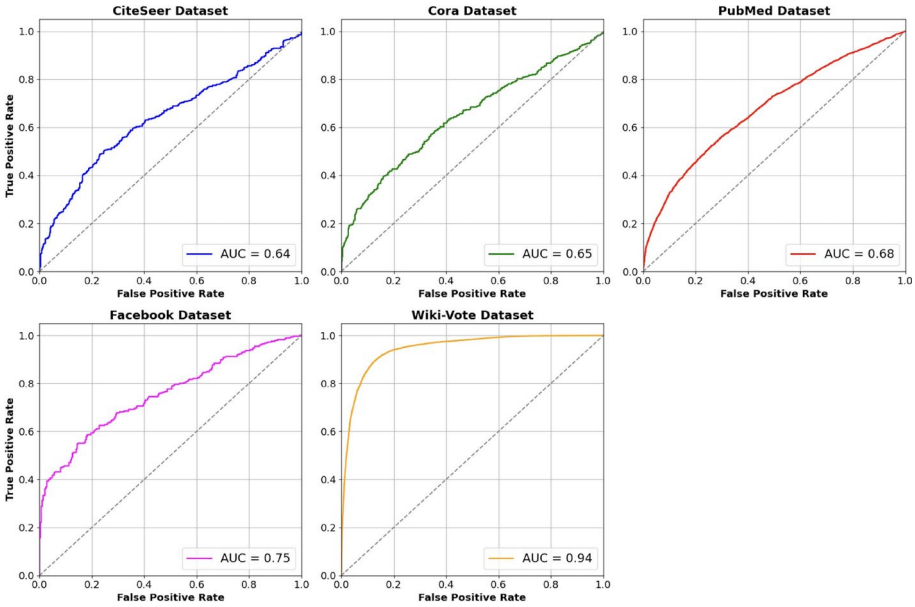


Fig. 10 Deep ONMF ROC Curves for CiteSeer, Cora, PubMed, Ego-Facebook and Wiki-Vote Datasets

Table 3 Sparse deep NMF performance for link prediction on datasets

Dataset	Micro-F1	Macro-F1	AUC score
Citeseer	0.658242	0.642492	0.668695
Cora	0.672676	0.667698	0.696051
Pubmed	0.713109	0.708514	0.737336
ego-Facebook	0.698246	0.697407	0.775894
Wiki-Vote	0.500000	0.333333	0.939840

attaining a score of 0.775894 on the Ego-Facebook dataset and a score of 0.939840 on the Wiki-Vote dataset. The results demonstrate the model's ability to accurately represent the complex relationships within networks. The Micro-F1 and Macro-F1 ratings emphasise the model's balanced performance across many classes, particularly in dense networks such as PubMed and Ego-Facebook.

Nevertheless, there are discernible discrepancies in performance among various datasets. The model excels on the Wiki-Vote dataset but achieves a lower Macro-F1 score on the Cora dataset. The variation indicates that the effectiveness of Sparse Deep NMF can be affected by the unique characteristics and intricacy of individual networks. The inferior performance on the Cora dataset is due to its comparatively lower average degree and smaller size in comparison to other datasets. Sparse Deep NMF is a versatile and powerful tool for link prediction tasks in various network types. Its capacity to effectively collect and utilise intricate patterns in network data makes it a vital resource in the growing field of graph-based learning and network analysis (Fig. 12). The model's flexibility with various network structures and its reliable performance with varied datasets highlight its potential to enhance the comprehension of intricate network dynamics.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

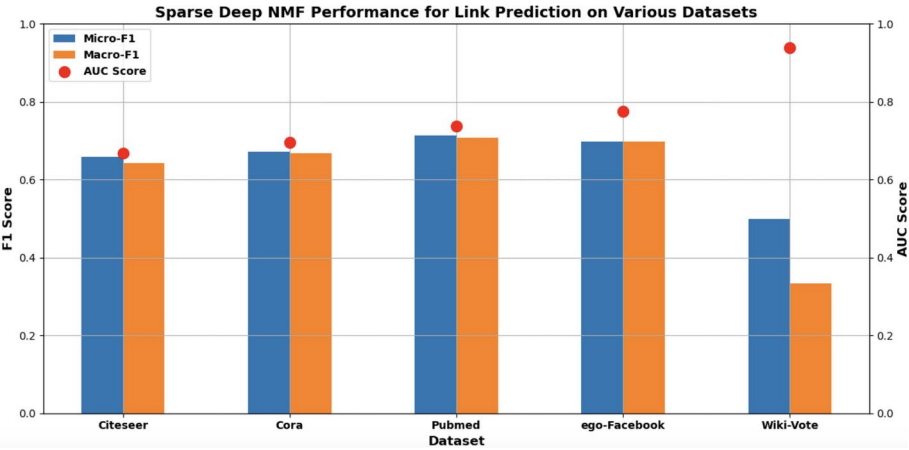


Fig. 13 CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki- Vote: Sparse Deep NMF performance for link prediction across datasets. While the red circles show the AUC scores, the bar graph shows the Micro-F1 and Macro-F1 scores for every dataset, therefore offering a whole picture of the model's performance on many distinct datasets. (Color figure online)

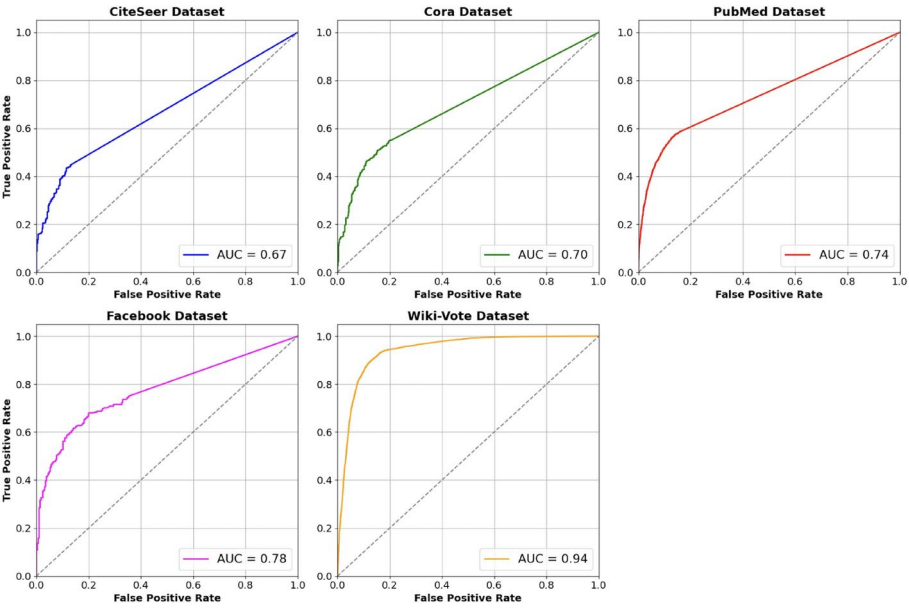


Fig. 12 Sparse Deep NMF ROC Curves for CiteSeer, Cora, PubMed , Ego-Facebook and Wiki-Vote Datasets

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 4 PMF performance for link prediction on datasets

Dataset	Micro-F1	Macro-F1	AUC score
Citeseer	0.6418	0.6341	0.6732
Cora	0.6129	0.5537	0.7113
Pubmed	0.7163	0.7007	0.7880
Ego-Facebook	0.8018	0.8002	0.8971
Wiki-Vote	0.9165	0.9165	0.9664

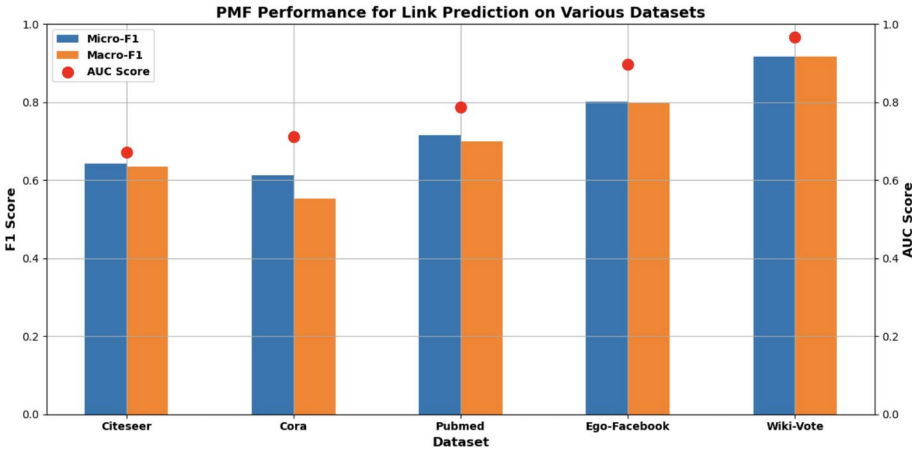


Fig. 15 PMF effectiveness for link prediction on several datasets, including CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote. The bar graph displays the Micro-F1 and Macro-F1 scores for every dataset, and the red circles indicate the AUC scores, allowing comparison of the model's performance measures across datasets. (Color figure online)

3.1.4.3 PMF The PMF model (Berahmand et al. 2023; Han et al. 2022), a well-regarded technique in link prediction, has been assessed on several datasets, as shown in Table 4 and Fig. 15. This assessment offers a comprehensive study of the model's efficacy using its Micro-F1, Macro-F1, and AUC ratings. The PMF model shows varying levels of effectiveness in different datasets. The model shows outstanding performance on the Wiki-Vote dataset, achieving a Micro-F1 score of 0.9165, a Macro-F1 score of 0.9165, and an impressive AUC score of 0.9664. The high scores indicate the PMF model's effective predictive capacity in forecasting relationships within this unique network characterised by voting and suggestion interactions.The model shows limited performance on the Citeseer and Cora datasets, with AUC scores of 0.6732 and 0.7113, respectively. These datasets, mostly consisting of citation networks, provide various obstacles for link prediction techniques. The lower results indicate that the PMF model is influenced by the specific structural traits of these networks, including their sparsity and the way nodes interact. The model achieved notable AUC scores of 0.7880 and 0.8971 on the Pubmed dataset and the Ego-Facebook dataset, respectively. The PMF model demonstrates a strong capacity to generalise and predict connections accurately in other network types, such as biomedical literature networks (Pubmed) and social networks (Ego-Facebook). The evaluation of the PMF model on diverse datasets demonstrates its potential as an adaptable tool for forecasting connections in different types of networks. The model demonstrates exceptional performance in particular networks like Wiki-Vote and Ego-Facebook (see Fig. 14). Its effectiveness differs

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

when used in various networks such as Citeseer and Cora. It highlights the importance of considering the distinct features of each network when using link prediction models. This discovery can offer useful insights for enhancing the PMF model and other related techniques in the field of graph-based learning.

Matrix factorization is a technique that offers concise representations of items within a matrix, rendering it well-suited for situations involving explicit ratings and collaborative filtering tasks. Nevertheless, the interpretability of the model is constrained by the inherent characteristics of latent features. The algorithm demonstrates proficiency in properly managing sparse data, however it impose significant processing demands when applied to large networks. Matrix factorization is highly proficient in capturing global relationships inside the network (Chen et al. 2022b). In contrast, we have walk/path-based methodologies which prioritise the examination of nearby graph structures and their interconnections, hence offering enhanced interpretability rooted in local structures. These techniques are particularly suitable for scenarios with intricate graph structures, since they provide efficient ways for capturing localised structural information and discovering trends within communities. Walk/path-based algorithms encounter difficulties in addressing sparsity in specific contexts, despite their ability to scale more effectively, particularly when utilising sampling strategies. The implementation of these algorithms necessitate the utilisation of more complex graph traversal techniques in contrast to the comparatively uncomplicated implementation of matrix factorization. Hence lets discuss path and walk based techniques.

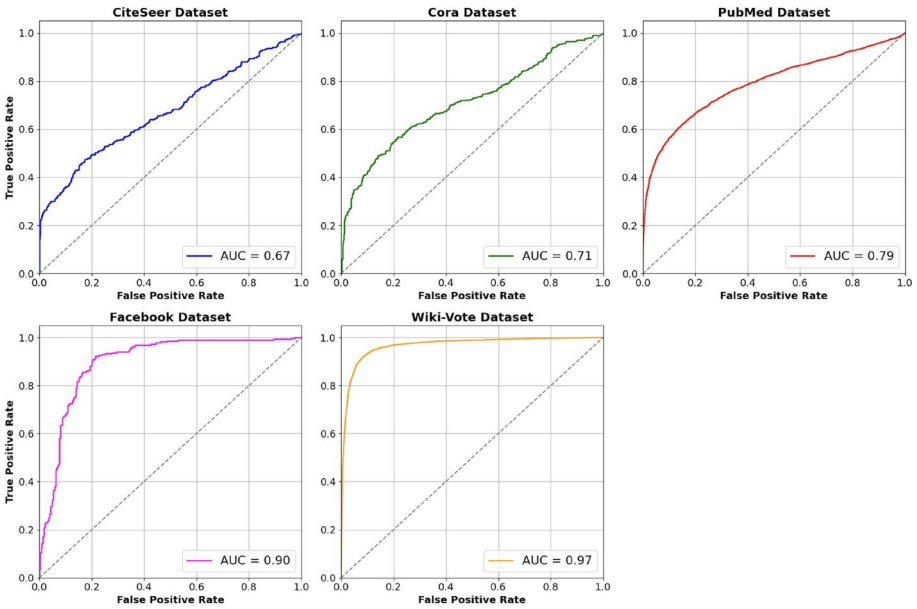


Fig. 14 ROC Curves for PMF Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3.2 Path and walk based techniques

Path-based and walk-based link prediction techniques are commonly employed in network analysis to forecast the presence of absent or forthcoming connections among nodes within a network. These methodologies utilise the encoded information within the structure of the network to generate predictions.

3.2.1 Basics of path and walk based techniques

In order to comprehend and analyse path and walk-based learning approaches, it is imperative to first gain a comprehensive understanding of the fundamentals of embedding. The utilisation of the embedding concept is prevalent in path-walk based learning methodologies (Wu et al. 2022; Liu et al. 2023).

3.2.1.1 Dimensionality reduction and embedding The phenomenon of the curse of dimensionality is widely recognised and acknowledged within the field of machine learning. Certain researchers have utilised dimension reduction techniques to address the aforementioned issue and implement them within the context of link prediction. In contemporary times, a significant number of scholars are actively engaged in the exploration of matrix decomposition and network embedding methodologies, both of which are widely acknowledged as approaches for dimensionality reduction (Kumar et al. 2020; Zhou et al. 2024; Dettmers et al. 2018; Kapoor et al. 2022).

Network embedding is a widely recognised technique for reducing the dimensionality of graphs. It involves mapping higher-dimensional nodes D (also known as vertices) in the graph to a lower-dimensional representation space$d$ ($d << D$). This mapping is achieved by preserving the neighbourhood structures of the nodes (Ou et al. 2016; Liu et al. 2024; Li et al. 2014; Ying et al. 2018). In essence, the objective is to discover the transformation of nodes into a reduced set of dimensions, such that nodes that are similar in the original network exhibit similar embeddings within the representation space. Figure 17 illustrates

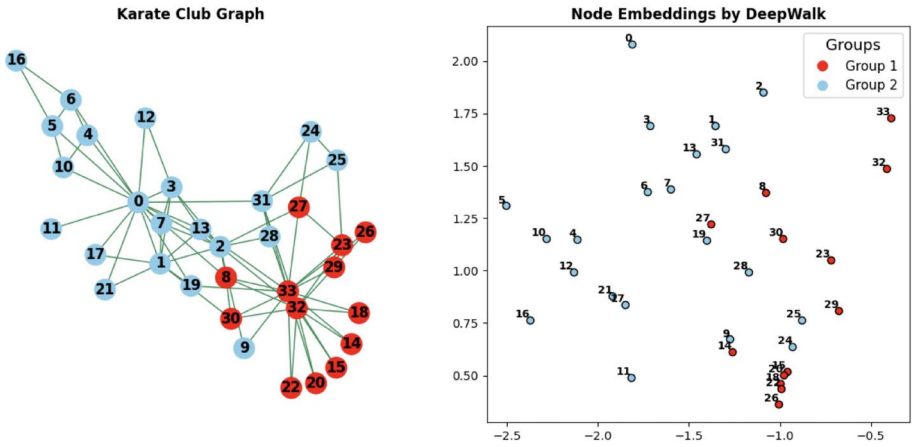


Fig. 17 Network of Karate club (left) and associated embedding space depiction using DeepWalk

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

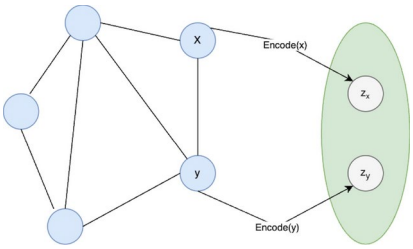


Fig. 18 Nodes x and y are embedded in the embedding space

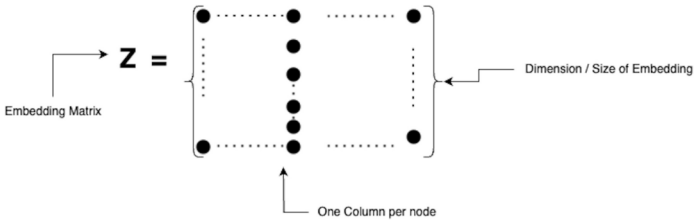


Fig. 16 Basic node embedding matrix

the configuration of the social network of the Zachary Karate club on the left, while on the right it presents the depiction of nodes in the embedding space through the utilisation of DeepWalk (Perozzi et al. 2014). The colouring of the nodes is determined by the communities to which they belong, as depicted in Fig. 17. The primary element of the network embedding technique is the encoding function, also known as the encoder f_{encode} (Eq. 22), which is responsible for mapping each node to the embedding space, as depicted in Fig. 18.

$$f_{encode}(x) = z_x \tag{22}$$

where z_x represents x's d-dimensional embedding. The embedding matrix, denoted as $Z \in R^{d \times |V|}$ (Fig. 16), is composed of columns that correspond to embedding vectors representing individual nodes (Grover and Leskovec 2016).

A similarity function, denoted as $S(x, y)$ (Eq. 23), is developed to represent the linear vector space interactions that match the original network's links.

$$S(x, y) \approx z_x^T z_y \tag{23}$$

The function S(x, y) is utilised to reconstruct pairwise value of similarity from the produced embedding. The variable S(x, y) is the term that exhib its variation based on the function employed in various factorization-based embedding methodologies.

The subsequent bullet points delineate the characteristics of the embedding techniques.

- The degree of similarity in network can be inferred by examining the similarity of embedding between nodes (Goyal and Ferrara 2018).
- Embeddings are a means of encoding network information (Zhou et al. 2023a).
- Embeddings have the potential to be utilised for downstream prediction (Goyal and Ferrara 2018).In the following part, we will look at learning based methods of acquiring

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

node embedding.

3.2.1.2 Random walk embedding The random walk algorithm is a widely utilised graph-based method employed in network analysis for the purpose of link prediction. The process entails the simulation of random walks on a graph in order to make predictions about the presence of missing or future connections between nodes.Node similarity is described in Random Walk embedding as how two nodes are linked. In random walk embedding, it is not necessary to consider all possible pairs of nodes during the training process. Instead, only focus on the pairs of nodes that co-occur in the random walk (Chung and Zhao 2010; Nie et al. 2017).

3.2.1.3 Random walk strategy Given a graph G and a starting point, randomly select a neighbour of the starting point and move to that neighbour. Then proceed to randomly select a neighbour of this new point (Kumar et al. 2020). The aforementioned procedure persists for a predetermined number of iterations. The encoder can be described as a simple embedding lookup mechanism (Barros et al. 2021). For example in Eq. 24, *ENC*(*u*) represents node $u's$ encoding.

$$ENC(u) = z_u \tag{24}$$

The similarity measure $S(u, v) \approx z_u^T z_v$ represents the probability that nodes u and v co-occur during a random walk on a graph (Fig. 19) . Greater probability indicates an increased likelihood of a connection between the nodes.

3.2.1.4 Embedding method In order to learn the coordinate *z*, the following points are taken into account:

- Using the random walk strategy *R*, calculate the probability of visiting node *v* on a random walk beginning at node *u* ($P_R(v/u)$) (Perozzi et al. 2014).
- To efficiently encode these random walk statistics, optimize embedding .

3.2.1.5 Unsupervised feature learning The objective is to discover the representation of the node in a d-dimensional space that maintains similarity. Therefore, the fundamental

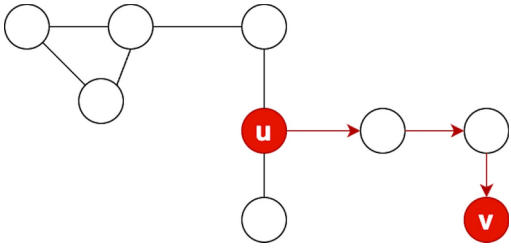


Fig. 19 Random walk from node u to node v

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

concept revolves around acquiring node embedding knowledge in such a way that nodes in close proximity exhibit a higher degree of proximity within the network. The term "neighbourhood" of a node u, as defined by a random walk strategy R, is denoted as $N_R(u)$. For example, a neighbourhood can be a random walk of nodes beginning with u. One approach to learn features involves formulating learning as an optimisation problem. Our goal is to learn mapping $f : u \longrightarrow \mathbb{R}^d$ for any node u in a given graph $G(V, E)$ to their embedding. Coordinates of node u in embedding space can be defined via Eq. 25.

$$f(u) = z_u \tag{25}$$

Functional mapping can be achieved by the following Eq. 26.

$$max_f \sum_{u \in V} \log(P(N_R(u)|z_u)) \tag{26}$$

In Eq. 26 the embedding coordinate (z_u) is sought in order to maximise the sum of log probabilities associated with the nodes that appear in the neighbourhood of u ($N_R(u)$). Here maximize the sum means we want to make nodes that are visited in same random walk are embedded closely (Grover and Leskovec 2016). Since the aim is to learn feature representations which are predictive of nodes in its random walk neighbourhood .The following points can help with optimization.

- Conduct a short fixed-length random walk beginning at each node u, using the random walk strategy R.
- Collect the neighbourhood ($N_R(u)$) of every node u.
- Optimise embedding based on the following criteria: given node u predict neighbouring $N_R(u)$ using Eq. 26 . Equivalently Eq. 26 can be rewritten as η (Eq. 27). Here our goal is to maximize the log probability such that it predicts node v is in neighbourhood of u (Qiu et al. 2018).

$$\eta = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|z_u)) \tag{27}$$

-
- Here probability P is parameterized using softmax function (refer Eq. 28). Equation 28 denotes exponential value of dot product of starting node u and neighbourhood node v divided by exponential value of sum of dot products with all the other nodes in network.

$$P(v|z_u) = \frac{\exp(z_u^T z_v)}{\sum_{n \in V} \exp(z_u^T z_n)} \tag{28}$$

-
- Hence random walk embedding can be optimized by finding embedding z where η is

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

minimum. However, the problem here is two nested summation, which increases the time complexity of the operation to $O(v^2)$. Computing sum over all nodes of the network in denominator of Eq. 28, will be a costly operation to accomplish.

3.2.1.6 Negative sampling Normalising softmax, as shown in Eq. 28, is the culprit for increased time complexity. As a result, negative sampling is used to approximate Eq. 28. Instead of normalising across all nodes, normalise across *k* negative samples. Hence Eq. 27 and Eq. 28 can be rewritten as Eq. 29 where α is baised random distribution over nodes and σ represents sigmoid function. Nodes with higher degree are more likely to be chosen in *k*.

$$\eta = \log(\sigma(z_u^T z_v)) - \sum_{i=1}^{k} \log(\sigma(z_u^T z_\alpha)) \tag{29}$$

After converting the optimization problem in form of η (Eq. 29) solve it by Stochastic Gradient Descent (SGD) (Grover and Leskovec 2016).

3.2.2 DeepWalk

The DeepWalk algorithm was introduced by Perozzi, Bryan, et al. (Perozzi et al. 2014) in 2014. The DeepWalk algorithm is widely utilised for the purpose of network modelling learning or graph embedding. The purpose of this approach is to acquire low-dimensional vector depictions, also known as embeddings, for nodes within a network. This is achieved by considering the network as an undirected graph. DeepWalk utilises methodologies derived from the domain of natural language processing, particularly word2vec, in order to produce significant embeddings for nodes by considering the network's topology. It is a method that acquires social representations of the vertices in a graph through the modelling a series of short random walks. Social representations refer to underlying characteristics of individuals or groups that capture similarities within a neighbourhood and indicate membership within a community. The latent forms encode social relations within a continuous vector space characterised by a limited number of dimensions. DeepWalk leverages truncated random walks to gather local knowledge and learn latent representations. It considers these walks as analogous to sentences refer Fig. 20, it depicts the transformation process in the DeepWalk algorithm, which begins with a random walk sequence starting node 1 and progresses via nodes 3, 5, and back to 3. This sequence is then turned into a series of embeddings (W1–W4), demonstrating how DeepWalk transforms random network walks into vector embeddings which capture node interactions.

Algorithm 1 illustrates steps of Deepwalk technique.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

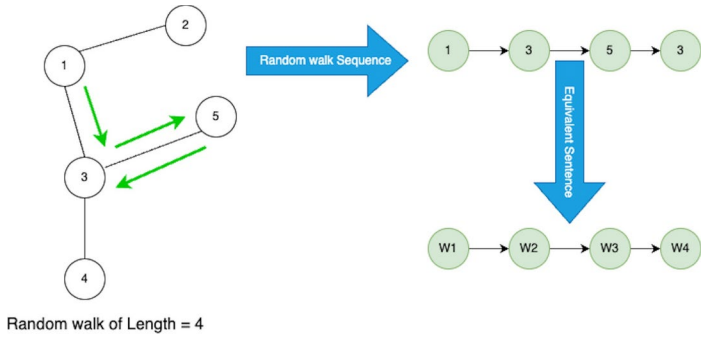


Fig. 20 DeepWalk social representation

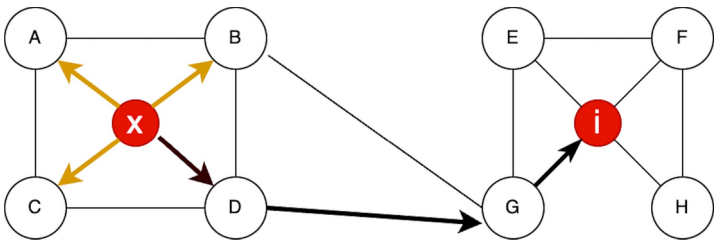


Fig. 21 BFS and DFS for node x

1: Conduct random walks across the network. Each random walk is a sequence of nodes where the next node is chosen based on a random probability distribution. These random walks capture the neighborhood structure of the graph's nodes.
2: Construct sentences by treating each random walk as a separate "sentence." In this analogy, each node is considered a "word" in the sentence.
3: Employ the Skip-gram model, a popular word2vec-based model for natural language processing, to learn node representations. The Skip-gram model is trained to predict surrounding nodes in the random walk sequence from a target node, thereby learning to capture the context of each node.
4: Train the Skip-gram model on sequences of nodes from random walks to obtain node embeddings. These embeddings are low-dimensional vector representations that aim to preserve the structural similarities between nodes, as reflected by the node sequences in random walks.

Algorithm 1 DeepWalk

Several uses, including link prediction, node categorization, and visualisation, are possible with the DeepWalk-generated node embeddings. Standard machine learning methods can be applied to the embeddings, allowing for a deeper dive into the analysis and comprehension of large-scale networks (Kumar et al. 2020). Social network analysis, recommendation systems, bioinformatics, and citation networks are just a few of the areas where DeepWalk has showed promise (Liu and Lü 2010).

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3.2.3 Node2vec

In 2016, Grover, Aditya et al. (Grover and Leskovec 2016) introduced the Node2vec algorithm. The Node2vec algorithm is widely utilised for the purpose of network representation learning or graph embedding. The algorithm being referred to is an expansion of the DeepWalk technique, which aims to capture the structural characteristics of a network by representing it in the form of low-dimensional vector embeddings . The Node2Vec algorithm utilises a biassed random walk approach in order to traverse the network and produce node embeddings that maintain the characteristics of local as well as global network properties. The incorporation of a flexible concept of network neighbourhood surrounding a node contributes to the generation of rich node embeddings.The distinction between node2vec and deep walk lies in the manner in which the set of neighbours is determined and the characterization of random walk.

Consider the graph depicted in Fig. 21, and let's examine the neighbourhood approach of node2vec sampling. Limit the size of the neighbourhood set N_x to k nodes in order to meaningfully compare various sampling procedures S, and then sample several sets for a single node x. For the purpose of creating neighbourhood set(s) N_x of k nodes, there exist commonly two extreme sampling techniques (Grover and Leskovec 2016):

- Depth-first Sampling (DFS) The locality comprises of nodes that have been systematically sampled in a sequential manner, with each subsequent node being at an increasing distance from the source node. As depicted in Fig. 21, when the size of the locality is denoted as k = 3, the nodes sampled using the Depth-First Search (DFS) algorithm are identified as *D*, *G*, and *I*. This sampling is good for detecting communities.
- Breadth-first Sampling (BFS) The neighbourhood is restricted to solely encompass nodes that are in immediate proximity to the source.As exemplified in Fig. 21, when considering a neighbourhood of magnitude k = 3, the nodes sampled by the breadth-first search algorithm are denoted as *A*, *B*, and *C*. This sampling is efficient for structural equivalences.

Both breadth-first and depth-first sampling reflect extreme cases with interesting consequences for the learnt representations in terms of the search space they examine. In example, homophily and structural equivalence are frequently toggled during prediction tasks on network nodes. Highly connected nodes that cluster together in the same way in the network should be physically close to one another, as proposed by the homophily hypothesis. For example, nodes *A* and *x* in Fig. 21 pertain to a specific network community. The structural equivalency idea suggests that nodes in networks with similar structural responsibilities should be anchored more closely together.For example, nodes *x* and *I* in Fig. 21 act as hubs for their corresponding communities. Nodes in a network may be physically separated yet play the same structural function if they are equivalent according to structural equivalence, which is a key difference from homophily. Both notions of equivalence coexist in practise, with some nodes displaying homophily and others showing structural equivalence in networks. Models reflecting any of the aforementioned correspondences can be generated using the BFS and DFS methods, as claimed by Node2vec. When it comes to yield, the BFS-sampled neighbourhoods provide embeddings that are quite near to structural equivalence. Intuitively, node2vec observes that in order to determine structural equivalence, accu-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

rately characterising the immediate neighbourhoods is frequently sufficient. For example, structural equivalency based on network responsibilities such as bridges and hubs might be deduced simply by examining each node's immediate surroundings. In contrast, the Depth-First Search (DFS) algorithm exhibits the capacity to traverse more extensive portions of the network by venturing farther from the initial node u, while maintaining a fixed sample size k. The nodes that are sampled in the DFS algorithm provide a more accurate representation of the larger-scale neighbourhood. This is of utmost importance when it comes to making inferences about communities based on the principle of homophily.The Breadth-First Search algorithm accomplishes this categorization task by effectively restricting the search to nearby nodes, thereby enabling a detailed examination of the immediate surroundings of each node. Moreover, it is worth noting that nodes within the sampled neighbourhoods exhibit a tendency to recur multiple times in the breadth-first search algorithm. This holds particular significance as it serves to diminish the variability in characterising the distribution of 1-hop nodes in relation to the source node. For every designated value of k, it is worth noting that merely a minute proportion of the graph is subjected to scrutiny.

However, the problem with DFS is that it is necessary to characterise not just which node-to-node relationships exist in a network, but also the precise form of these dependencies. This is difficult since it have a sample size constraint and a big neighbourhood to explore, resulting in significant variance. Second, moving to larger depths introduces complex dependencies because a sampled node may be a long way from its origin and thus potentially less representative.

3.2.3.1 Sampling strategy and parameters Expanding upon the aforementioned observations, the node2vec algorithm incorporates a versatile approach for sampling neighbourhoods, enabling a seamless transition between BFS and DFS strategies. Node2vec accomplishes this objective by designing a versatile assessed random walk algorithm capable of exploring neighbourhoods in both BFS and DFS manners.A bias random walk of fixed length allows node x to develop its neighbourhood. Biased random walk uses two hyper parameters, return parameter p and in-out parameter q. The parameter p determines the likelihood that a node will be revisited promptly. Setting it to a high value reduces the likelihood that will sample a previously visited node in the next two steps (as long as the subsequent node in the walk has no neighbours). This strategy promotes moderate searching and avoids redundant sampling over two hops. Alternatively, if p is small, the walk would backtrack a step refer Fig. 22, which would maintain the walk "local" and near to the initial node x. The parameter q distinguishes between "inward" as well as "outward" nodes during the search. Referring back to Fig. 22, if $q > 1$, the random walk favours nodes close to node x. Such walks achieve a local view of the actual graph relative to the walk's starting

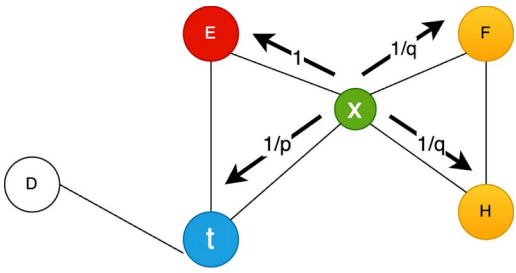


Fig. 22 Biased second order random walk

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

node and emulate BFS behaviour in the sense that our samples consist of nodes within a small locality. Alternatively, if $q < 1$, the walk is more likely to visit nodes that are further from node x. This behaviour exemplifies DFS, which promotes outward exploration. In this case, however, accomplish DFS-like exploration within the framework of the random walk.

3.2.3.2 Biased random walk example Refer to Fig. 22 in order to gain a better understanding of the concept of biased random walk. Let us consider a scenario where a walker arrives at the edge (t, x) and is currently located at position x. The walker has unnormalized probabilities of $\frac{1}{p}$, $\frac{1}{q}$, and 1. The walker has the option to either return to the starting point, remain in the same orbit, or continue moving away from the starting point. The nodes E and t are equidistant for a given value of x, indicating that they are located in the same orbit. As a result, the probability of their occurrence remains constant. The probability of transitioning from node x to node t is equal to $\frac{1}{p}$. Conversely, the probability of moving away from node x equal to $\frac{1}{q}$. Here, p is return parameter and q is walk away parameter. Unnormalized probability distribution can be defined by Eq. 30, where d_{tn} represents distance of node n from node t. A low p value ensures a BFS-like walk, while a low q value ensures a DFS-like walk.

$$P(t,n) = \begin{cases} 1/p, & if\ d_{tn} = 0 \\ 1, & if\ d_{tn} = 1 \\ 1/q, & if\ d_{tn} = 2 \end{cases} \tag{30}$$

1: Calculate the probabilities of random walks.
2: Perform r random walks of length l starting at node u.
3: Optimize the node2vec objective using Stochastic Gradient Descent.

Algorithm 2 Node2vec

Algorithm 2 is having linear time complexity. The node2vec algorithm offers a semi-supervised approach for acquiring comprehensive feature representations for nodes within a network. However, there is often a greater interest in prediction tasks that involve pairs of nodes rather than individual nodes. In the context of link prediction, the objective is to forecast the presence or absence of a connection between two nodes within a given network. Given that our random walks are inherently dependent on the connectivity structure among nodes in the underlying network, one can employ a bootstrapping technique to expand them to pairs of nodes by leveraging the feature representations of the individual nodes. When considering two nodes, x and y, a binary operator $*$ is introduced to operate on the feature vectors $f(x)$ and $f(y)$ associated with these nodes. This operation results in the generation of a representation $g(x, y)$, which satisfies the Eq. $g : V \times V \rightarrow R^d$. Here, d represents the size of the representation for the pair (x, y). The objective is to establish a general definition for operators that can be applied to any combination of nodes, regardless of the presence of an edge between them. This approach is valuable for link prediction, as it allows for the evaluation of both true and false edges in the test set, including cases where the edges do not exist. The binary operator $*$ can encompass various operations such as averaging, hadamard product, weighted-L1, and others.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3.2.4 Walklets

Walklets (Perozzi et al. 2017) greatly enriches the field of network analysis by expanding on the foundation built by DeepWalk. It presents an advanced approach to acquiring multi-scale network embeddings, which is essential for comprehending intricate network architectures. This is accomplished by employing a distinctive methodology that entails bypassing nodes while conducting random traversals of the network. Through this approach, Walklets adeptly record and depict diverse levels of relationships inside the network, ranging from local neighbours to more remote connections. The utilisation of a multi-scale method enables a more intricate and all-encompassing comprehension of the network's overall structure, which is especially advantageous in activities that necessitate a profound understanding of network dynamics. The mathematical foundation of walklets is expressed by the objective function shown in Eq. 31

$$\eta = -\sum_{(u,v)\in G_h} \log Pr(u|v) \tag{31}$$

This function aims to maximise the value of η, where (u, v) represents pairs of nodes in the network. The collection G_h comprises these pairings of nodes, with h determining the scale of observation and determining the length of the 'skip' in the random walks. The parameter *h* regulates the number of nodes bypassed in each step of the walk, allowing the algorithm to record relationships at different distances. The expression $\log Pr(u|v)$ represents the natural logarithm of the conditional probability of detecting node u given node v. This transition to a logarithmic scale improves the algorithm's sensitivity to otherwise modest probability shifts. Walklets increases the significance of uncommon but information-rich transitions throughout the optimization process by turning their probabilities into logarithmic values. Furthermore, this logarithmic translation helps to alleviate numerical stability difficulties, which are common when working with very small probabilities, ensuring that the optimization process stays stable across network scales. Through the process of optimising this function, Walklets acquires embeddings that accurately capture the complex multi-scale relationships within the network, providing a more comprehensive representation in contrast to conventional single-scale approaches. Walklets' multi-scale embeddings have significant value in the context of link prediction. Link prediction is the process of predicting the probability of a future connection between two nodes in a network. This task can be significantly improved by analysing the topology of the network at different levels. Walklets' capacity to record this information at several scales enables more precise estimates of these potential connections. The efficacy of the technique has been proven on many datasets, highlighting its resilience and adaptability in varied network settings.

3.2.5 Results of path based techniques homogeneous datasets

This section examines the implementation of our learning-based path techniques - DeepWalk, Node2vec, and Walklets-on different real-world datasets, such as Cora, Citeseer, Pubmed, Ego-Facebook, and Wiki-Vote. The efficacy of these strategies is thoroughly assessed through the utilisation of Micro F1, Macro F1 scores, and AUC curve graphs. This methodology enables us to evaluate the flexibility and precision of our approaches in vari-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 5 Deepwalk performance for Link Prediction on datasets

Dataset	Deepwalk settings (p,q)	Micro F1	Macro F1	AUC score
Citeseer	p = 1, q = 1	0.8250	0.8249	0.8970
Cora	p = 1, q = 1	0.8060	0.8059	0.8797
Pubmed	p = 1, q = 1	0.7881	0.7880	0.8571
Ego-Facebook	p = 1, q = 1	0.7231	0.7231	0.7938
Wiki-Vote	p = 1, q = 1	0.8433	0.8433	0.9161

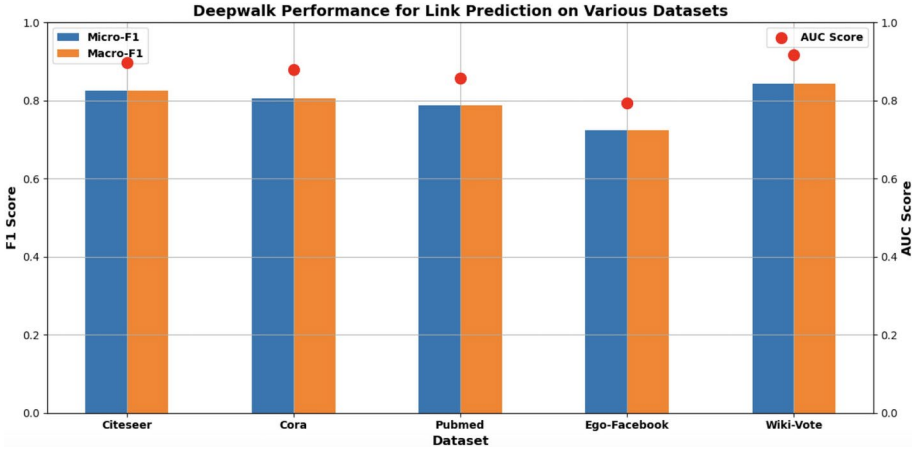


Fig. 24 Deepwalk Effectiveness for Link Prediction on CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote Datasets. Having the AUC Score given by red markers, this graph shows the Micro-F1 and Macro-F1 scores as bars, consequently offering a whole picture of the model's performance over several datasets. (Color figure online)

ous network contexts, while also aiding in the comprehension of the subtle behaviours and structures present in these distinct datasets.

3.2.5.1 Deepwalk DeepWalk (Perozzi et al. 2014; Grover and Leskovec 2016), when used on different datasets with p = 1, q = 1 parameters in the Node2Vec framework, proves to be effective in link prediction tasks as shown in Table 5 and Fig. 24. This setup guarantees an equitable examination of localities in network embeddings, a crucial aspect of DeepWalk. DeepWalk performs exceptionally well in the Citeseer and Cora datasets, achieving AUC scores of 0.8970 and 0.8797, respectively. The high results demonstrate its ability to accurately forecast links by capturing the complex network patterns in citation networks.

DeepWalk's effectiveness in networks containing voting and endorsement interactions is highlighted by its performance on the Wiki-Vote dataset, achieving an AUC score of 0.9161. The model's consistent Micro F1 and Macro F1 scores across these datasets indicate its dependable link classification.

The AUC score of 0.7938 on the Ego-Facebook dataset indicates that DeepWalk may have challenges when dealing with networks that include distinctive structural characteristics, such social networks with intricate interaction patterns (Fig. 23). This emphasises the significance of taking into account the unique attributes of each network when utilising link

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

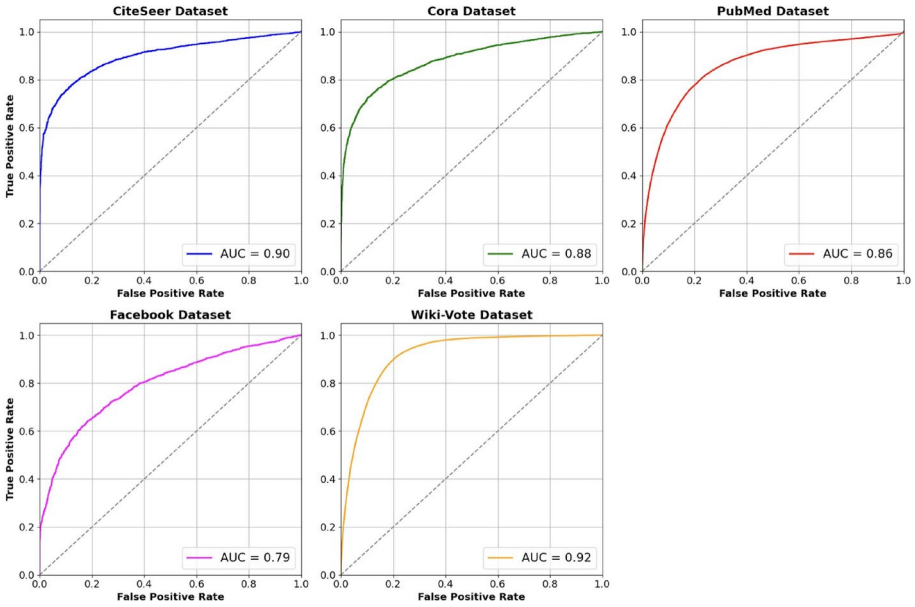


Fig. 23 ROC Curves for Deepwalk Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

Table 6 Node2vec performance for Link Prediction on datasets

Dataset	node2vec settings (p,q)	Micro F1	Macro F1	AUC score
Citeseer	p = 4, q = 1	0.8137	0.8134	0.8779
Cora	p = 4, q = 1	0.7971	0.7969	0.8677
Pubmed	p = 4, q = 1	0.7916	0.7916	0.8599
Ego-Facebook	p = 4, q = 1	0.7351	0.7351	0.8072
Wiki-Vote	p = 4, q = 1	0.8433	0.8433	0.9135

prediction algorithms. The variability in performance among various datasets highlights the necessity for flexible models capable of efficiently managing a wide range of network configurations in the field of graph-based learning.

3.2.5.2 Node2vec The Node2Vec (Grover and Leskovec 2016) model's performance on several datasets is shown in Table 6 and Fig. 26, using set of parameters, $p = 4$ and $q = 1$. This parameter selection indicates a well-rounded approach that balances exploration and exploitation in the random walks employed for network embeddings. A greater value of p promotes the random walk to investigate more distant sections of the network, while setting $q = 1$ ensures an equal chance of examining local and remote nodes. The model's robustness in predicting linkages across various network configurations is demonstrated by its performance metrics such as AUC, Micro F1, and Macro F1 scores. Node2Vec demonstrates impressive performance on the Wiki-Vote dataset, with an AUC score of 0.9135, highlighting its capability to capture intricate interaction patterns in voting networks. The model excels on the Citeseer dataset, with an AUC score of 0.8779, showcasing its efficacy

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

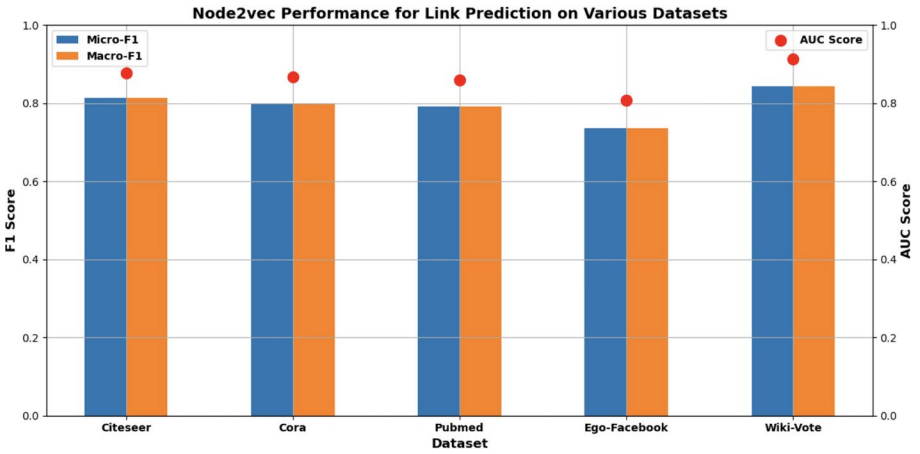


Fig. 26 Performance measures for Node 2Vec Link Prediction across CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote datasets. Micro-F1 and Macro-F1 scores are shown by the bar graphs; red dots show the AUC scores for every dataset

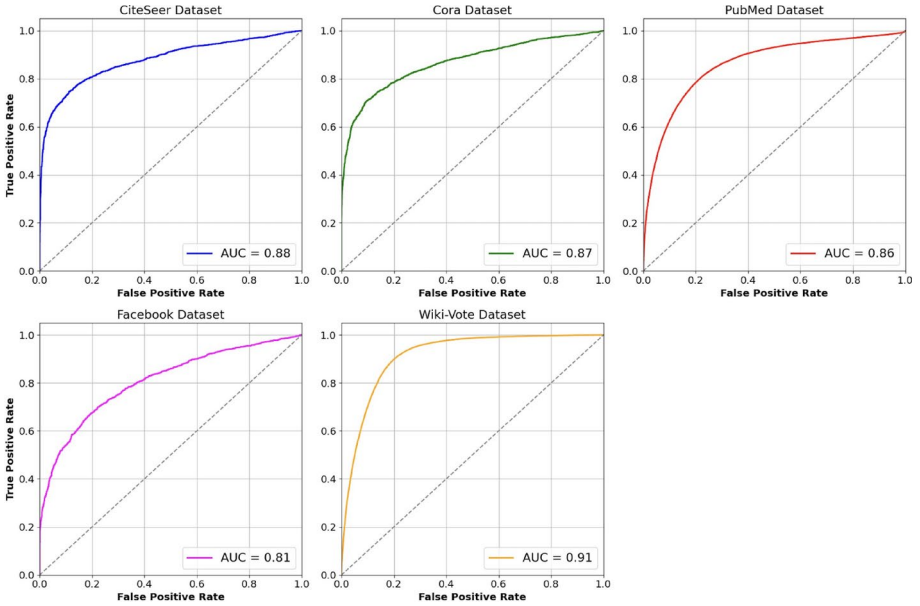


Fig. 25 ROC Curves for Node2Vec Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

in academic citation networks. The model shows significantly reduced performance on the Ego-Facebook dataset, achieving an AUC score of 0.8072 (see Fig. 25). This is due to the distinctive structural characteristics of social networks, like strong clustering coefficients and community structures, which create difficulties during the embedding process. The disparity in performance among various datasets underscores the impact of network-specific

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 7 Walklets performance for link prediction on datasets

Dataset	Skip-Length	Micro F1	Macro F1	AUC score
Citeseer	2	0.5832	0.5820	0.6118
Cora	2	0.6003	0.5971	0.6334
Pubmed	2	0.7406	0.7403	0.8168
Ego-Facebook	2	0.7540	0.7537	0.8255
Wiki-Vote	2	0.8297	0.8297	0.8964

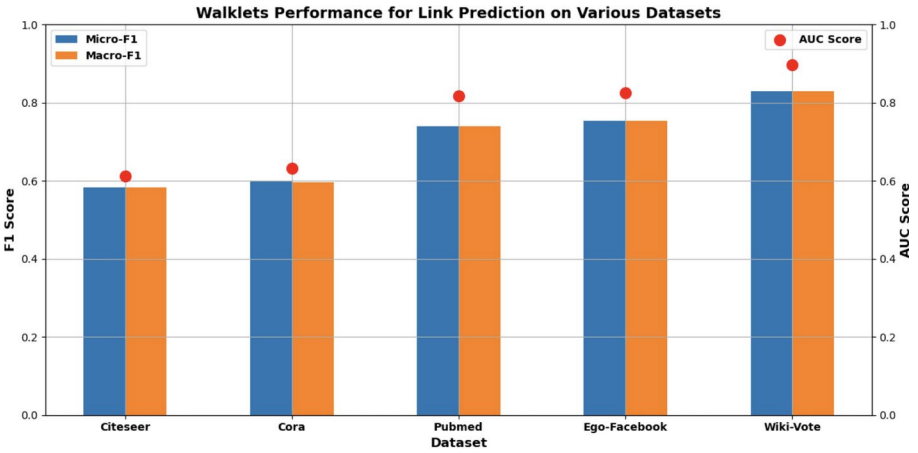


Fig. 28 Walklets for Link Prediction on CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki- Vote Datasets. Micro-F1 and Macro-F1 scores are shown on the bar graph; AUC scores for every dataset are shown by the red dots. (Color figure online)

traits on the efficiency of the Node2Vec model in link prediction tasks. The appropriate calibration of hyperparameters, such as the values of p and q, might result in additional enhancements in outcomes, hence requiring their careful alteration.

3.2.5.3 Walklets The Walklets model (Perozzi et al. 2017) demonstrates its effectiveness in link prediction on various datasets using a fixed skip-length of 2, as shown in Table 7 and Fig. 28. This parameter allows the model to skip nearby nodes during random walks, effectively capturing distant connections. This feature is essential for understanding the broader network structure beyond immediate areas. The model shows modest performance for Citeseer and Cora with AUC scores of 0.6118 and 0.6334, respectively. Academic networks with complex citation patterns require a more detailed methodology to capture their connections adequately. While a skip-length of 2 helps identify distant connections, it may not suffice for extensive linkages in these networks. Adjusting skip-length according to requirements can improve performance. The model performs exceptionally well in datasets like Pubmed, Ego-Facebook, and Wiki-Vote, with AUC scores of 0.8168, 0.8255, and 0.8964, respec-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

tively. Higher performance in these cases indicates that a skip-length of 2 is more effective in networks with sparser or simpler structures, highlighting the model's adaptability.

The varied results of the Walklets model on different datasets underscore the substantial influence of network-specific features on link prediction effectiveness (Fig. 27). Selecting the appropriate skip-length is crucial for optimizing the model's accuracy in predicting links across different network structures

3.3 Graph neural network based

A Graph Neural Network (GNN) is a deep learning model specifically developed for the purpose of processing and analysing data that is represented in the form of graph structures. Graphs are composed of nodes, also known as vertices, which are interconnected by edges, also referred to as edges or links (Matsunaga et al. 2019; Fey and Lenssen 2019). Each node has the capability to store information, while the edges symbolise relationships or connections between the nodes. GNNs exhibit significant utility in domains where the underlying data's relational structure and interactions assume the greatest significance. These domains encompass but are not limited to social networks, recommendation systems, molecular chemistry, citation networks, and other related areas (Zhou et al. 2020; Hu et al. 2020a).

The primary concept underlying GNN involves the utilisation of message-passing algorithms to iteratively modify node representations by incorporating information propagated from adjacent nodes. During each iteration, a node performs the task of aggregating information from its neighbouring nodes. This aggregated information is then combined with the node's own features. Subsequently, a neural network layer is applied to generate an updated representation. The aforementioned procedure is commonly iterated multiple times in order

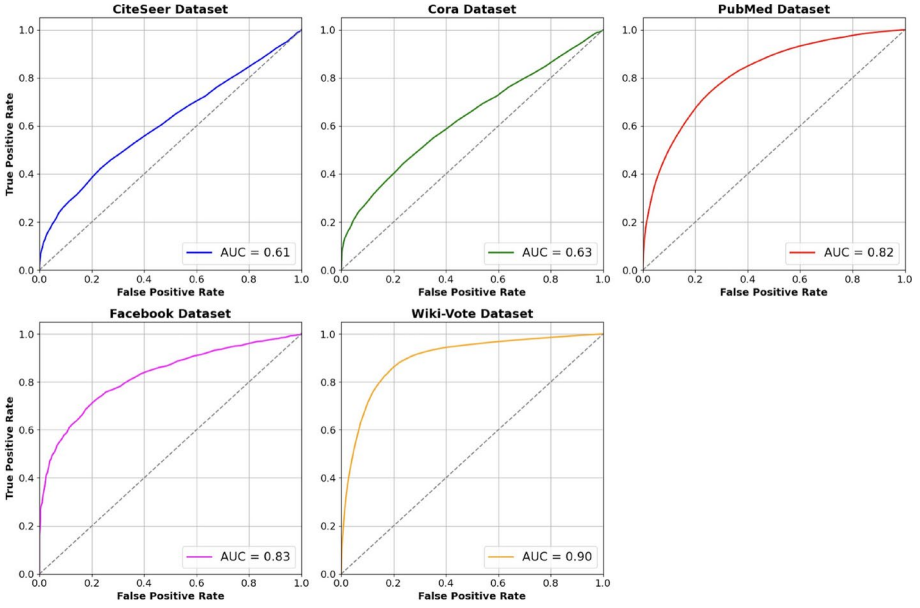


Fig. 27 ROC Curves for Walklets Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

to capture multi-hop relationships within the graph (Hamilton et al. 2017c; Riesen et al. 2007; Wang et al. 2024).

3.3.1 Basics of graph neural network techniques

This section will focus on the basics terminologies of GNN.

3.3.1.1 Computational graph Consider the simple input graph shown in the Fig. 29. For the target node A, the structure of the neural network will do calculations to make a prediction for node A. Here, node A will get information from its neighbours in the network, which are nodes B, C and D. Node A will take message from B, C and D and then aggregates it. We can spread it out into more than one layer. For example, node D can get information from node A in layer 2. This implies that under the GNN architecture, we must learn not just the aggregation operators but also the message transformation operators along the edges (Bronstein et al. 2021). For example in Fig. 29, node A will transform message from node B, then node A will transform message from C, and finally node A will transform a message from D. The messages will then be aggregated and sent to the next tier. Transformation and aggregation will be learned hence parameterised. And these will be the parameters for our model (Hamilton et al. 2017c).

3.3.1.2 Deep model The key distinction between GNN and conventional NN is that each node is given the opportunity to design its own computation graph based on its neighbourhood. Hence structure of neural network depends upon structure of graph. The GNN model has the flexibility to have a depth that can vary according to the requirements (Kipf and Welling 2016; Cai et al. 2018). At each layer of the model, the nodes are equipped with an embedding. The embedding of node x at layer 0 (refer Fig. 29) is represented by the feature vector V_x, which captures the characteristics of node x. The layer-h embedding retrieves information from nodes that are h hops distant. Which means the embedding of node x at layer 0 is different from embedding of node at layer 2. This procedure to embed nodes will run for limited number of steps. Figure 29 depicts the progressive feature aggregation and

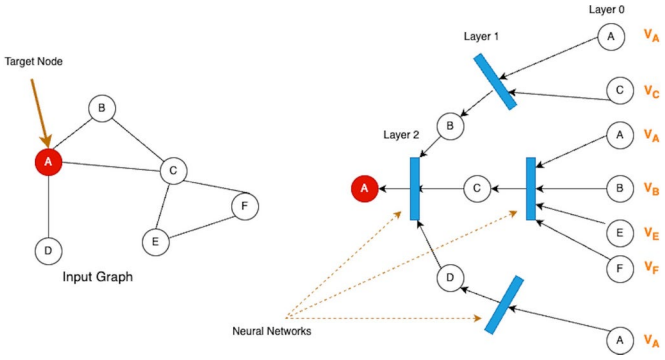


Fig. 29 Aggregation process for node A

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

embedding augmentation of node A across various layers of a graph neural network. The procedure is outlined as follows:

- Layer 0: Node A starts with its own feature vector V_A . It also captures initial feature vectors (V_C, V_B, V_E, V_F) from directly linked nodes.
- Layer 1: Using features from the initial set, node A aggregates embeddings from each of the associated nodes (*B*, *C*, *E*, *F*) to create a new, richer embedding that includes local structural data from its immediate neighborhood.
- Layer 2: The method expands to incorporate second-hop links. Node A collects additional features from nodes B, C, and D, improving the embedding with more detailed neighborhood knowledge from nodes two hops away.This sequence of aggregation demonstrates graph neural networks' capacity to gradually include greater contextual data into node embeddings, enhancing the representation with every single layer as nodes collect information from their growing neighborhoods.

3.3.1.3 Neighbourhood aggregation and deep encoder In neighbourhood aggregation, the key idea is to create the node embedding based on the local network area. Nodes use neural networks to gather knowledge from their partners. Key difference in distinct GNN architectures is how this aggregation is done. Note that since the ordering of nodes in a graph is arbitrary, the aggregation operator is required as a permutation invariant. It implies that regardless of the sequence in which we aggregate, the aggregation will always be the same. The basic strategy to aggregation is to average out the data and then apply a neural network. Following equation represents mathematics for deep encoder. Equation 32 shows feature vector at layer 0.The initial feature vector of node *x* is represented by E_x^0, whereas the raw feature values are denoted by V_x. Equation 33 can be run in several layers to get average for embedding. Here k_x^h represent embedding of node *x* at layer *h* , *H* represents total number of layers/steps considered for embedding , σ represents non linearity function (eg: ReLU). The Eq. 33 explains how W_h and B_h are trainable parameters for layer *h*, used in transformation of node characteristics. The non-linear activation function (σ) enhances the model's capacity to capture complicated patterns. This iterative technique incorporates averaging the embeddings of surrounding nodes (k_y^h), adjusted by their count *N*(*x*), and integrating the node's original embedding from the prior layer (k_x^h). Equation 34 represents the final embedding of node after total number of layers *H* under consideration (Bronstein et al. 2021). This multi-layer aggregation efficiently captures complicated patterns from both immediate neighbors and the wider network structure, rendering it very efficient for diverse downstream processes such as classification or link prediction.

$$E_x^0 = V_x \tag{32}$$

$$k_x^{(h+1)} = \sigma\left(W_h \sum_{y \in N(x)} \frac{k_y^h}{|N(x)|} + B_h k_x^h\right), \ \forall h \in \{0, ..., H-1\} \tag{33}$$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

$$Z_x = k_x^{(H)} \tag{34}$$

3.3.1.4 Model training The method by which we train the model involves determining the parameters of the model. In Eq. 33, W_h and B_h are trainable weight matrices. For every layer we have different W and B . W_h is weight matrix for neighbourhood aggregation. B_h is weight matrix for transforming hidden vector representation of node itself. The provided embeddings should be inputted into a chosen loss function, and the weight parameters should be trained using SGD. Also Eq. 33 can be rewritten as Eq. 35 by using matrix formulation (Kipf and Welling 2016), here A^1 is equal to $D^{-1}A$. The model can be trained in either a supervised or unsupervised setting. In unsupervised learning the discrepancy between similarity in graph and similarity in embedding to be small. Hence we can run this as optimization problem. In supervised learning directly train the model for supervised task (e.g. node classification). We can run cross entropy loss for node classification.

$$K^{(h+1)} = \sigma(A^1 K^h W_h^T + K^h B_h^T) \tag{35}$$

As previously established, the same aggregate parameters are common across all nodes for a specific layer, therefore we can generalise them to unseen nodes. Means we can train parameters on one graph and apply it on new graph (Hamilton et al. 2017c). Now we can engage in a discussion on the generalised (GNN) framework.

3.3.1.5 General GNN framework A general GNN framework consist of following components (You et al. 2020; Jin et al. 2020) :-

- GNN Layer : A single GNN layer typically consists of two fundamental components: the message passing mechanism and the aggregation process (Fig. 30). Message computation takes the representation of node(y) from previous layer and transforms it, each node creates a message and send it to other nodes in the next layer refer Eq. 36. In Eq. 36 megssage transformation(MT) will vary according to architechure of GNN. mt_y^h denotes the"message transformation" produced by node y at layer h. $MT^{(h)}$ denotes the Message Transformation function at layer h. This function is exclusive to the architecture of the GNN and is accountable for converting node characteristics or embeddings from one layer to the subsequent layer. $k_y^{(h-1)}$ is embedding of node y at the preceding layer $(h-1)$. It serves as the input for the function MT. Then in aggregation step each node(x) will aggregate message from its neighbour($N(x)$) refer Eq. 37. All the aggregation function (AF) are order invariant for example sum, min, max etc. After message transformation and aggregation we pass it to non linear activation (ReLU, Sigmoid, etc.) to add expressiveness.

$$mt_y^{(h)} = MT^{(h)}(k_y^{(h-1)}) \tag{36}$$

$$k_x^{(h)} = AF^{(h)}(mt_y^{(h)}, y \in N(x)) \tag{37}$$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

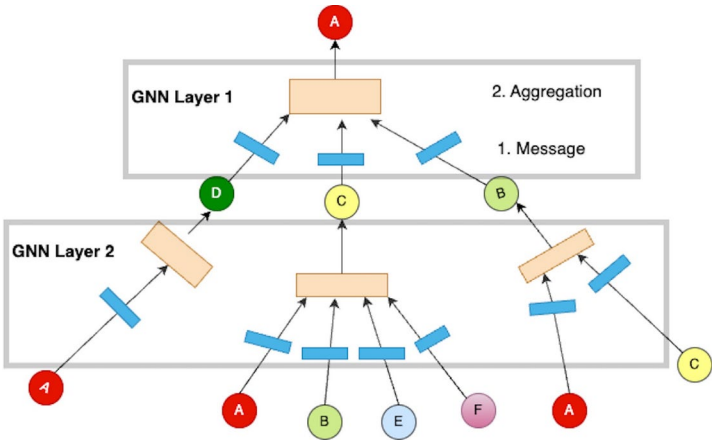


Fig. 30 A two-layer GNN framework. In Layer 1, node A collects features from its immediate neighbors B, C, and D. Layer 2 deepens aggregation to encompass second-hop neighbors A, B, E, and F, demonstrating how GNN layers gradually improve node embeddings through more neighborhood integration

-
- Layer Stacking : The second step pertains to the manner in which many layers of Graph Neural Networks (GNNs) are combined. This includes the sequential stacking of layers, the incorporation of skip connections, and other relevant considerations
- Design Selection : Design decision pertains to the methodology employed in constructing the computation graph. It involves determining whether to equate the input graph with the computation graph or to employ any form of augmentation.
- Learning Method : We must determine the appropriate objective function and task(edge level, node level or graph level prediction) that will enable us to effectively learn the parameters of our GNN. In order to train a model, one can employ various methodologies such as supervised learning, unsupervised learning, or objective-based learning.Different architectural models, such as GCN, GraphSAGE, GAN, etc. demonstrate variations in their methodologies for defining aggregation and message propagation systems.The subsequent sections will address these topics.

3.3.2 Graph convolutional networks (GCN)

The GCN was introduced by Kipf et al (Kipf and Welling 2016) in the year 2016. GCN is a type of multi-layer feedforward neural network that is designed to distribute and modify node properties across a given graph structure. In GCN layer-wise propagation(transformation and aggregation) is defined by Eq. 38 , here $N = D^{-\frac{1}{2}}A^1D^{-\frac{1}{2}}$ is normalized adjacency matrix. The matrix $A = A^1 + I_N$ represents the adjacency matrix of an undirected graph G, with self-connections. I_N is identity matrix. $D_{ii} = \sum_j A_{ij}$ and W^h are weight matrices that can be trained and are specific to each layer. $K^0 = K$, while $K^h = [k_1^h, ..., k_n^h]^T$ specifies the node representations of the $h - th$ layer. σ denotes an activation function, such as ReLU. Each layer transforms the message by taking previous layer embedding (K^h

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

) multiplying with W and normalizing it. Then the aggregation function utilised in GCN is the summation over all the neighbours.

$$K^{(h+1)} = \sigma(NW^{(h)}K^{(h)}) \tag{38}$$

3.3.3 GraphSAGE (Graph sample and aggregated)

The GraphSAGE was proposed by Hamilton et al (Hamilton et al. 2017b) in the year 2017. GraphSAGE is based on GCN but extends on it in several key areas. The first point is that it realizes that aggregation function (AGG) is arbitrary. As a result, it provides a variety of aggregate function options. Second, for a particular layer-h take message from the node, alter it, and concatenating it with aggregated messages. So the GraphSAGE equation (refer Eq. 39) includes both message transformation and aggregation. It is a two-stage procedure in which aggregation(AGG) from node neighbours ($N(x)$) is concatenated with message from node (x) itself, multiplied by transformation matrix(W^h), and then passed via non linearity.

$$K_x^{(h)} = \sigma(W^{(h)} \cdot CONCAT(K_x^{(h-1)}, AGG(K_y^{(h-1)}, \forall y \in N(x)))) \tag{39}$$

There exist different kinds of aggregating functions that can be utilised. Various aggregation functions provide distinct theoretical features. The selection of an aggregation function has an impact on the expressive power of the learning model. The aggregation method of utilising a weighted average of neighbouring data points can be employed (refer Eq. 40).

$$AGG = \sum_{y \in N(x)} \frac{k^{(h-1)}}{|N(x)|} \tag{40}$$

Instead of utilising linear transformation, an alternative approach might involve employing a multi-layer perceptron followed by aggregation using the mean operation (refer Eq. 41).

$$AGG = MEAN(MLP(k_x^{(h-1)}, \forall y \in N(x))) \tag{41}$$

Sequence models, such as Long Short-Term Memory (LSTM), have the potential to be utilised for the purpose of aggregation (refer Eq. 42).

$$AGG = LSTM(k_x^{(h-1)}, \forall y \in \pi(N(x)))) \tag{42}$$

3.3.4 Graph attention networks (GAT)

The GAT is a neural network architecture specifically developed to effectively process data structured as graphs, including but not limited to social networks, citation networks, and molecular graphs. The paper titled “Graph Attention Networks” authored by Petar Velikovi et al. (Veličković et al. 2017) and published at the Conference on Neural Information Processing Systems (NeurIPS) in 2018, established the concept of GAT. The GAT has notable

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

efficacy in tasks that involve graphs with nodes exhibiting varying levels of significance and inter-relatedness. Graph data can be visually depicted using nodes and edges, wherein nodes symbolise various entities such as users, documents, or molecules, while edges denote the interactions or connections that exist between these items. In the context of graph neural networks, it is common practice to employ a uniform transformation across all adjacent nodes , means all nodes are given equal importance while aggregation. However, this approach may not be ideal for effectively representing the diverse levels of impact that exist within a graph structure. The limitation is addressed by GAT by the incorporation of attention processes into graph neural networks. In GAT, when aggregating messages from neighbours, each neighbour is assigned a weight, so for each neighbour y of node x, attention weight α_{xy} is assigned (Eq. 43).

$$k_x^h = \sigma\left(\sum_{y \in N(x)} \alpha_{xy} W^h k_y^{h-1}\right) \tag{43}$$

3.3.4.1 Attention weight α_{xy} α indicates the level of priority given to messages from a specified node. The word attention is inspired by the cognitive attention. As a result, attention α focuses on the most important component of the incoming data and ignores the rest. Hence, NN should allocate more computational power to the little but important part of input. It's dependent on the scenario as to which part of data is the most crucial. We are going to learn what part of data is important through model training process. As a result, we allow the model to learn the importance of various pieces of input that it receives. So in GAT we want to learn α_{xy} , that will tell us how important a message coming from node y to node x.

3.3.4.2 Attention mechanism The attention mechanism produces α. The attention mechanism will provide us with attention weight scores (Fig. 31). Based on the messages, compute attention coefficients I_{xy} across pairs of nodes x and y. Define a function f that takes the embedding of node y at the previous layer and the embedding of node x at the previous layer, transforms these embedding and give weight I_{xy} as output (refer Eq. 44). The function f

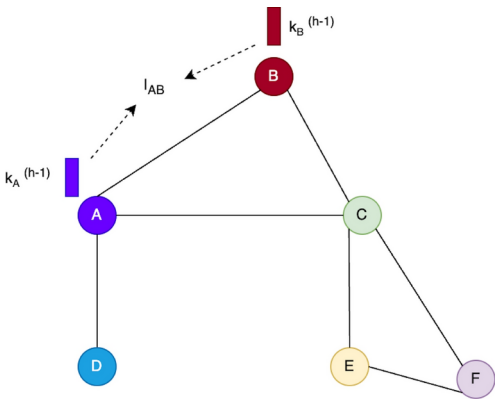


Fig. 31 Attention mechanism representation for node A and B

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

has the potential to exhibit various forms such as sigmoid, concatenation, linear etc. Hence, weight I_{xy} will tell importance of y's message on x.

$$I_{xy} = f(W^h k_x^{h-1}, W^h k_y^{h-1}) \tag{44}$$

In reference to Fig. 31 the attention coefficient between node A and node B can be written as $I_{AB} = f(W^h k_A^{h-1}, W^h k_B^{h-1})$. Once we get attention coefficients for all the neighbors of particular node we will normalize I_{xy} with softmax (refer Eq. 45) to get attention weights (α), such that $\sum_{y \in N(x)} \alpha_{xy} = 1$.

$$\alpha_{xy} = \frac{\exp(I_{xy})}{\sum_{z \in N(x)} \exp(I_{xz})} \tag{45}$$

Hence the aggregation for node A in Fig. 32 will look like Eq. 46 , here $k_B^{(h-1)}$ represents node embedding of previous layer.

$$k_A^h = \sigma(\alpha_{AB} W^h k_B^{(h-1)} + \alpha_{AC} W^h k_C^{(h-1)} + \alpha_{AD} W^h k_D^{(h-1)}) \tag{46}$$

The attention mechanism exhibits computational efficiency, storage efficiency and possesses inductive capabilities.

GNNs have discovered applications in a variety of fields for link prediction tasks. GCNs are commonly used to classify academic papers and predict citations by using structural information from citation networks. For example, in the Cora and Citeseer datasets, GCNs can accurately categorize papers and forecast citations. Furthermore, GCNs are used in social network research to forecast new friendships or interactions based on existing network topologies. This is especially useful in datasets such as Ego-Facebook and Wiki-Vote, where GCNs look at the social network structure to uncover potential new connections. GraphSAGE is optimized for dynamic and large-scale networks, making it ideal for inductive learning applications (Zhou et al. 2023b). In academic settings, GraphSAGE is used to classify publications by sampling and aggregating information from a subset of neighbors, as seen in the Cora and Citeseer datasets. It is also used to anticipate new relationships in biomedical research networks with datasets such as PubMed. In social network analysis, GraphSAGE predicts new links in huge networks by effectively sampling and aggregating

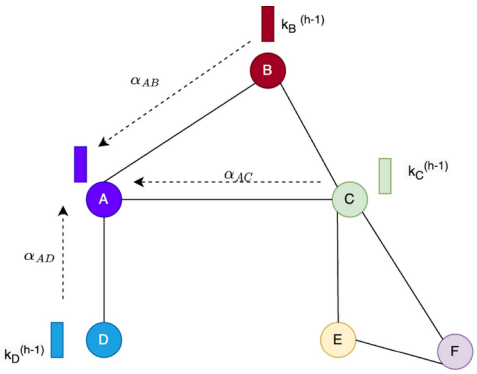


Fig. 32 GAT Aggregation example for node A

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 8 GCN performance for link prediction on datasets

Dataset	GCN Layers	Micro F1	Macro F1	AUC score
Citeseer	2	0.6746	0.6735	0.7382
Cora	2	0.6886	0.6865	0.7498
Pubmed	2	0.8213	0.8213	0.9055
Ego-Facebook	2	0.7456	0.7456	0.8126
Wiki-Vote	2	0.8772	0.8772	0.9413

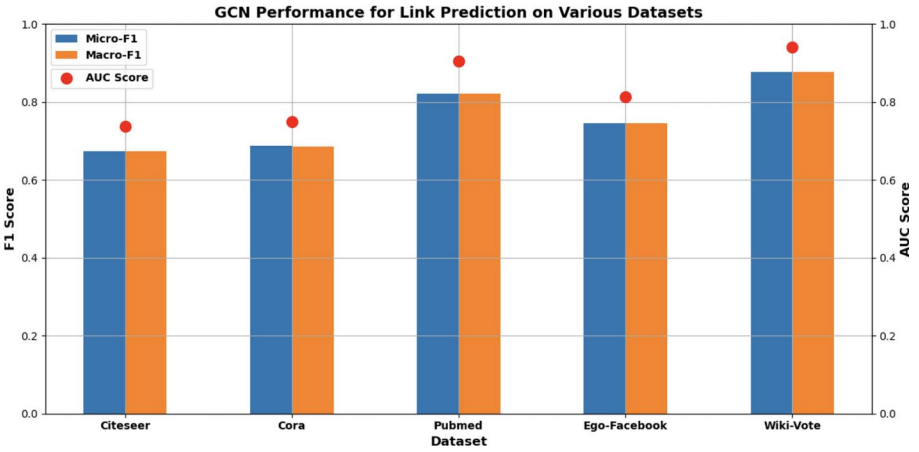


Fig. 34 Performance measures for GCN in link prediction problems spanning several datasets including CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki- Vote. While the red markers indicate the AUC scores, the bar graph displays Micro-F1 and Macro-F1 scores, therefore offering a complete comparison of the model's performance over several criteria and datasets. (Color figure online)

information from dynamic settings, as demonstrated by the Ego-Facebook and Wiki-Vote datasets. GAT are used in situations where the relevance of individual nodes varies, as they can assign varying weights to neighbors. GATs are used to classify publications and predict citations by evaluating the importance of various citations, making them useful for datasets like as Cora and Citeseer. GATs discover prominent users and prospective new connections in social networks by weighting current relationships differently (Schlichtkrull et al. 2018).

3.3.5 Results of graph neural network based techniques on homogeneous graphs

The experimental setting for Neural Network-Based Techniques focuses on advanced graph neural network models, namely GCN , GraphSAGE, and GAT . These link prediction techniques are utilised on the identical collection of datasets that have been the primary focus of our previous sections: The datasets include Cora, Citeseer, Pubmed, Ego-Facebook, and Wiki-Vote.

3.3.5.1 GCN The Graph Convolutional Network (GCN) model (Kipf and Welling 2016) showcases its adaptability in link prediction across different datasets, as outlined in Table 8 and Fig. 34. The datasets consist of Citeseer, Cora, Pubmed, Ego-Facebook, and Wiki-Vote, each characterised by unique network architectures. The GCN model's performance is assessed based on Micro F1, Macro F1, and AUC values, employing a standardised con-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

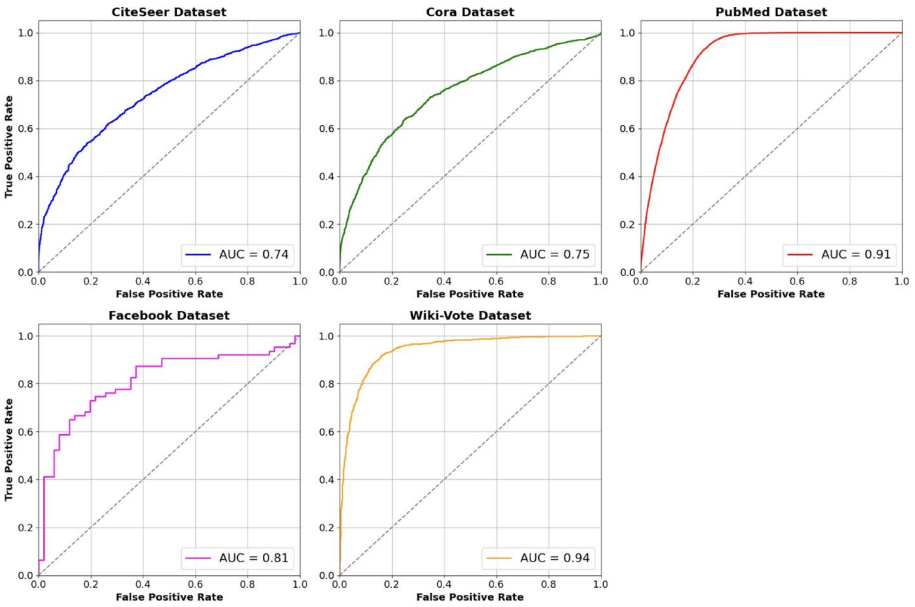


Fig. 33 ROC Curves for GCN Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

Table 9 GraphSAGE performance for Link Prediction on datasets

Dataset	Graph-SAGE Layers	Micro F1	Macro F1	AUC score	Aggregator
Citeseer	2	0.6143	0.5831	0.7189	Mean
Cora	2	0.6622	0.6516	0.7252	Mean
Pubmed	2	0.6787	0.6657	0.8021	Mean
Ego-Facebook	2	0.6737	0.6586	0.8137	Mean
Wiki-Vote	2	0.7246	0.7097	0.9282	Mean

figuration of 2 GCN layers across all tests. The model has reasonable performance on the Citeseer and Cora datasets, achieving AUC scores of 0.7382 and 0.7498, respectively. This demonstrates the GCN model's capacity to comprehend citation links while also indicating room for enhancement. Conversely, the model performs exceptionally well on the Pubmed dataset, attaining an AUC score of 0.9055, showcasing its proficiency in modelling biomedical citation networks. The Ego-Facebook dataset poses a unique challenge with an AUC value of 0.8126, indicating the model's effectiveness in tracking social interactions.The Wiki-Vote dataset achieved the highest AUC score of 0.9413, demonstrating the remarkable effectiveness of the GCN model in modelling voting behaviour and influence patterns. This achievement highlights the significance of taking into account the distinct attributes of each network when utilising the GCN model for link prediction tasks (Fig. 33).

3.3.5.2 GraphSAGE GraphSAGE (Hamilton et al. 2017b) is a leading graph neural network model tailored for inductive learning on graph-structured data. The Table 9 presents

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

a complete evaluation of GraphSAGE's performance in link prediction tasks on different datasets. The model utilises neighbourhood sampling and aggregation methods to create low-dimensional vector representations (embeddings) of nodes, which are then applied for tasks like link prediction. The model consistently utilises two GraphSAGE layers with a mean aggregator function across many datasets including Cora, Citeseer, Pubmed, Ego-Facebook, and Wiki-Vote in our assessment. The mean aggregator is required for combining node information from a node's nearby neighbours to capture the local network structure, which is crucial for link prediction.

The Table 9 and Fig. 36 displays performance measures such Micro F1 Score, Macro F1 Score, and AUC Score (Area Under the ROC Curve). The Micro F1 Score calculates an overall score that takes into account all classes, which is beneficial for datasets with uneven class distribution. The Macro F1 Score computes the F1 Score for individual classes and then calculates the average, assigning equal importance to all classes. The AUC score, as shown in Fig. 35, evaluates the model's capacity to differentiate across classes, with a score nearing 1 signifying outstanding performance. The results show that GraphSAGE is both robust and successful in predicting links across various datasets. The model performs quite well on the Wiki-Vote dataset, with an AUC value of 0.9282, demonstrating a high ability to distinguish between present and absent links. Yet, its efficacy is slightly diminished when applied to the Cora and Citeseer datasets, potentially because of the distinctive attributes and intricacies of these datasets, such as the graph's type (e.g., citation networks) and the node connectivity patterns. The model's adaptability and effectiveness in capturing crucial graph features for link prediction are emphasised by the uniform use of two GraphSAGE layers and a mean aggregator across all datasets.

3.3.5.3 GAT The GAT model (Veličković et al. 2017) demonstrates different levels of effectiveness when used with a range of datasets including Cora, Citeseer, Pubmed, Ego-Facebook, and Wiki-Vote. The model consistently uses two GAT layers with varying numbers of attention heads in each layer: 8 heads in the first layer and either 1 or 8 heads in the

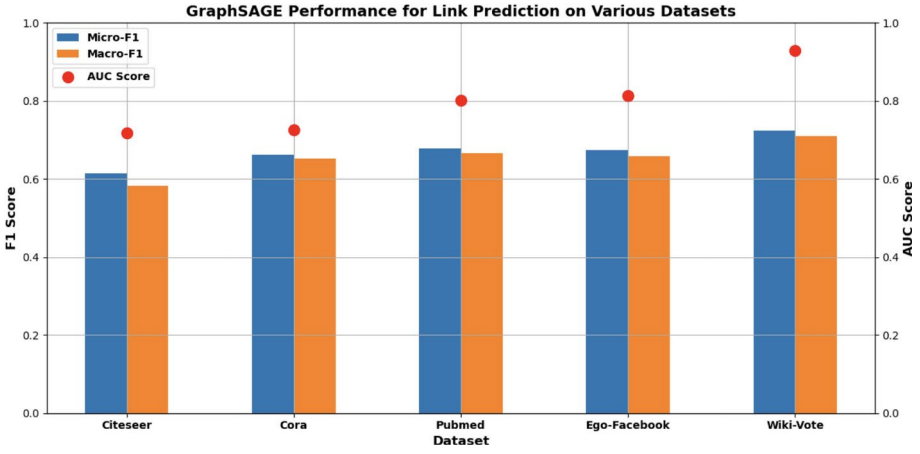


Fig. 36 GraphSAGE's effectiveness in link prediction on CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote datasets is evaluated, demonstrating the scores for Micro-F1, Macro-F1, and AUC

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

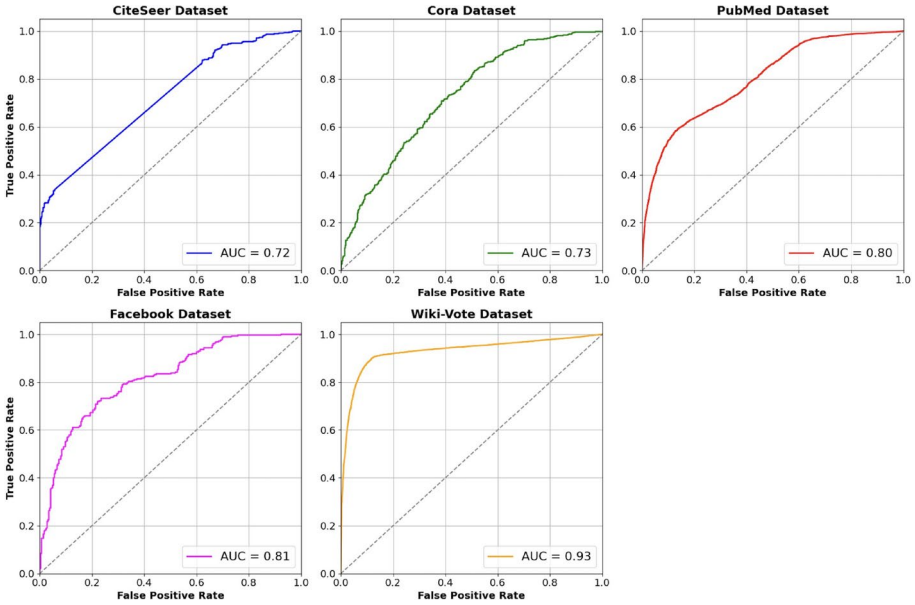


Fig. 35 AUC Curves for GraphSAGE Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

Table 10 GAT performance for Link Prediction on datasets

Dataset	GAT Layers	Heads(L1,L2)	Micro F1	Macro F1	AUC score
Citeseer	2	8,1	0.7308	0.7152	0.8040
Cora	2	8,1	0.7002	0.6892	0.7571
Pubmed	2	8,8	0.7262	0.7094	0.8106
Ego-Facebook	2	8,8	0.6211	0.6114	0.7225
Wiki-Vote	2	8,8	0.6514	0.6373	0.6940

second layer, as indicated in Table 10. The performance indicators, such as Micro F1, Macro F1, and AUC scores, demonstrate the model's capacity to adjust to various graph structures (Fig. 38).

The model demonstrates a robust performance on the Citeseer dataset, with a Micro F1 score of 0.7308 and an AUC score of 0.8040. This indicates that GAT is highly efficient at representing the structure of the citation network, emphasising the importance of information and influence transfer across papers. However, the model's performance is poorer on the Ego-Facebook and Wiki-Vote datasets, with Micro F1 scores of 0.6211 and 0.6514, respectively. The lower scores in these social network datasets is result of the intricate and ever-changing nature of social interactions, which could present difficulties for the model in accurately forecasting connections. The fluctuation in the quantity of attention heads in the second layer across various datasets highlights the need of tailoring the model. Using one attention head in the second layer for the Cora and Citeseer datasets, and several heads for

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

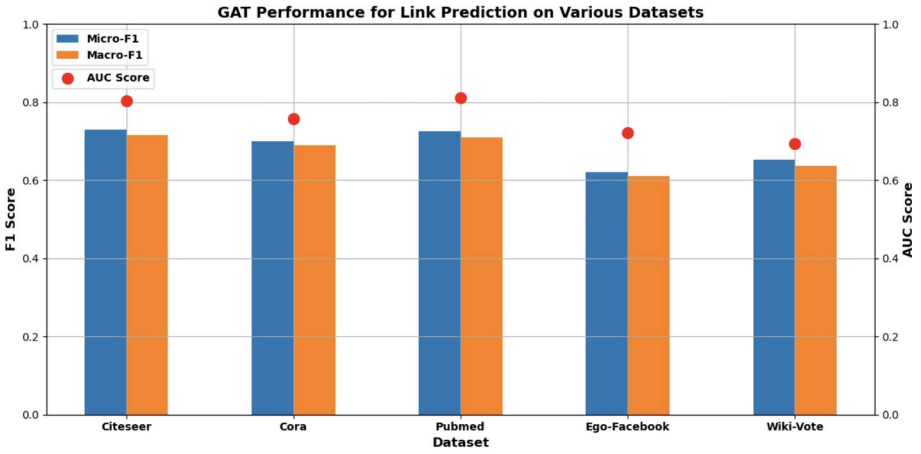


Fig. 38 Measurements of GAT's performance in link prediction on the CiteSeer, Cora, PubMed, Ego-Facebook, and Wiki-Vote datasets, showing Micro-F1, Macro-F1, and AUC values

the Pubmed, Ego-Facebook, and Wiki-Vote datasets, indicates that the ideal setup of attention heads might differ considerably depending on the dataset's attributes.

The GAT model shows flexibility in processing different forms of graph-structured data, but the results highlight the importance of adjusting the model parameters to match the characteristics of each dataset. These insights are essential for researchers and practitioners to choose and set up GAT models for various link prediction tasks, ensuring that the model architecture is best suited to the dataset's distinctive characteristics (Fig. 37).

GNN have considerable advantages in a wide range of network-based applications, including node classification and link prediction. Their capacity to learn from network structures and node properties makes them ideal for jobs that need relational data. For example, GCN are particularly advantageous in applications involving homogeneous networks, such as node categorization and link prediction. They are simple to set up and use, with node characteristics and graph architectures that perform well on medium-sized datasets. GCNs excel on citation networks like Cora and Citeseer, as well as biomedical datasets like PubMed, because they efficiently gather information from nearby nodes. However, GCNs suffer difficulties when applied to heterogeneous graphs with a variety of node and edge types and features. Their architecture is mostly targeted toward homogeneous data, which limits their flexibility and efficacy in capturing the complexity of heterogeneous networks (Zhang et al. 2019b). GraphSAGE tackles scalability difficulties by leveraging local sampling, making it suitable for large-scale and dynamic graphs. It effectively generalizes to new, previously unseen nodes, which is crucial for applications involving large amounts of data and dynamic networks. Despite these advantages, GraphSAGE suffers with heterogeneous graphs since it uses sophisticated algorithms to manage different node and edge types. This constraint limits its utility for tasks involving complex, heterogeneous networks. Similarly, GAT use an attention mechanism to apply different weights to neighbors, capturing complex interactions and allowing for a variety of homogeneous network configurations. However, GATs face increased computing costs and difficulty when dealing with heterogeneous data because their architecture does not always support the diversity present in such networks. Overall, GCNs, GraphSAGE, and GATs are useful tools for homogeneous data. However, their

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

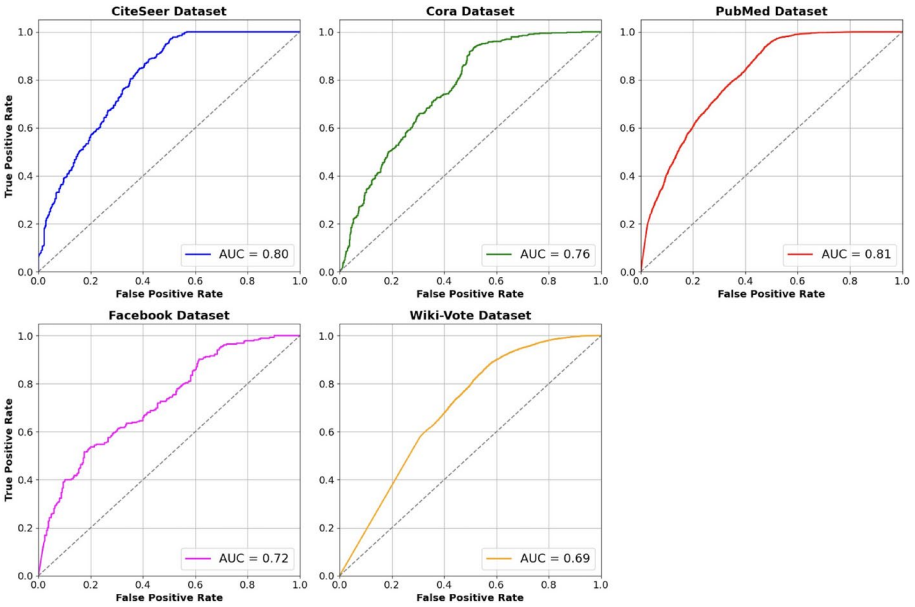


Fig. 37 AUC Curves for GAT Link Prediction on CiteSeer, Cora, PubMed,ego-Facebook and Wiki-vote Datasets

shortcomings when dealing with heterogeneous graphs show the need for specific models or adaptations. To fully grasp the potential of different and complex datasets, graph neural network models tailored to heterogeneous are developed (Lou et al. 2024).

3.3.6 GNN-based techniques vs. traditional link prediction techniques

The performance discrepancies between classic link prediction algorithms and GNN-based techniques have sparked significant research interest. Traditional heuristics, such as Common Neighbors (CN), Adamic-Adar (AA), Shortest Path, Resource Allocation (RA) and Katz, are straightforward approaches that use graph topology to evaluate the likelihood of a link between two nodes. These methods often leverage graph structure to detect patterns and forecast relationships, but their simplicity can restrict their capacity to capture complicated dependencies in graph data (Huang et al. 2023; Grover and Leskovec 2016).

In contrast, GNN-based approaches such as GCN, GAT, and GraphSAGE use node characteristics and graph structure to learn more sophisticated embeddings that better represent the complex relationships between nodes. This enhanced modeling capabilities enables GNNs to outperform older approaches, especially in datasets with rich feature information and complicated graph structures (Huang et al. 2023). Table 11 shows that GNN-based models frequently outperform heuristic techniques in terms of accuracy. For example, a study comparing different approaches on the Cora, Citeseer, and Pubmed datasets discovered that GNN models outperformed classical heuristics in terms of AUC values. GCN achieved an AUC score of 95.01 on the Cora dataset, compared to CN's 70.85, indicating a significant improvement. Similar findings are observed in other datasets, with GAT and GraphSAGE outperforming older approaches like as AA and RA.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 11 Comparison of Traditional Link Prediction Techniques vs. GNN-based Techniques (Li et al. 2024)

Technique	Cora(AUC)	Citeseer(AUC)	Pubmed(AUC)
Heuristic			
Common Neighbors (CN)	70.85	67.49	63.9
Adamic-Adar (AA)	70.96	67.49	63.9
Resource Allocation (RA)	70.96	67.48	63.9
Shortest Path	81.08	75.5	74.64
Katz	81.17	75.37	74.86
GNN-based			
Graph Convolutional Network (GCN)	95.01	95.89	98.69
Graph Attention Network (GAT)	93.90	96.25	98.20
GraphSAGE	95.63	97.39	98.87

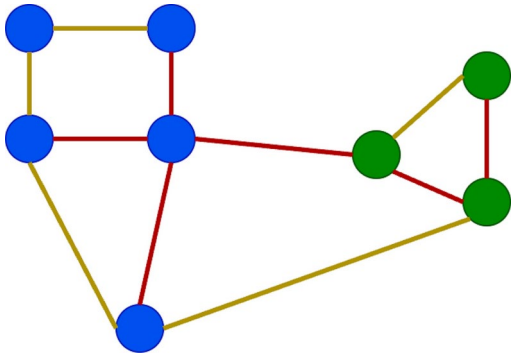


Fig. 39 Heterogeneous Graph with two types of nodes and two types of edges

This detailed comparison highlights GNNs' advantages in catching more complicated and nuanced patterns in graph data, which classical heuristics frequently overlook. These findings support the use of GNN-based techniques for link prediction tasks, especially when the graph data is complicated and rich in characteristics.

Subsequently, we shall assess heterogeneous data. In order to accomplish this task, distinct datasets are necessary, given the variations in node and edge types that exist within heterogeneous graphs. As a result, we shall independently establish and analyse the heterogeneous data.

3.3.7 Techniques on heterogeneous graphs

A heterogeneous graph refers to a specific form of graph data structure in which the nodes and edges possess distinct types or features. In contrast to homogeneous graphs, which consist of nodes and edges that are uniform in type, heterogeneous graphs offer increased versatility in capturing and depicting intricate and diverse interactions among different entities (refer Fig. 39). In the context of a heterogeneous graph, it is possible for nodes to exhibit membership in distinct categories or classes, each characterized by its unique type (Ding et

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

al. 2024; Sun et al. 2011; Kaibiao et al. 2024). For example in social network, it is common to see many categories of nodes, such as users, posts, comments, and hashtags. In a heterogeneous graph, it is possible for edges to possess distinct types or labels(Wang et al. 2023a; Fu et al. 2020b; Leskovec 2017). The aforementioned edge types serve as representations of the connections or exchanges that occur between nodes. Within the context of a social network, the various sorts of connections between individuals can be classified as edge types. These edge types encompass relationships such as "friendship," "likes," "follows," or "mentions". In a heterogeneous network, nodes and edges have the potential to possess distinct features or properties that are unique to their respective types (Hu et al. 2020c; Yang et al. 2020; Fang et al. 2021; Shi et al. 2017a). An example can be given where a user node possesses attributes such as name, age, and location, whereas a post node possesses attributes such as timestamp and content. Heterogeneous graphs are utilized in multiple sectors such as social network analysis, recommendation systems, e-commerce, biology, and other scenarios where data exhibits varied forms and entities possess intricate and diversified relationships. Heterogeneous graphs possess significant utility in the representation and examination of real-world data that intrinsically exhibits diversity and complexity, characterized by entities and interactions that do not adhere to a singular, uniform structure. Heterogeneous graphs enable a heightened level of flexibility and expressiveness in the representation of data, rendering them a crucial instrument in the domains of data science, machine learning, and network analysis (Nguyen et al. 2023; Zhao et al. 2021; Wang et al. 2019).

Mathematically heterogeneous graph can be defined as $G = (V, E, R, T)$, here node with node types $x_i \in V$, edges with relation types $(x_i, r, x_j) \in E$, node type can be represented as $T(x_i)$, relation type $r \in R$. Here *R* represents relation set. Different nodes are connected by relation type *r*. Hence we have nodes and edges where every node is labeled by a type and every relation is labeled by a type (Yang et al. 2023; Hong et al. 2020). Relation type and node type varies according to application. If we have this type of graph then our model should handle heterogeneity. As a result, in order to handle it, we must extend our GCN, which we refer to as the R-GCN (Relational GCN).

3.3.7.1 Relational GCN In order to effectively manage a heterogeneous graph, the concept of R-GCN was proposed by Michael Schlichtkrull et al. (Schlichtkrull et al. 2018). As we have seen in GCN the weight matrix *W* is same for a particular layer (Eq. 38) in message transformation. In R-GCN since relation at every edge is different, we will not apply same weight matrix for incoming edge. For example if we have two different relation types we will use two different matrices *W* (Fig. 40). Hence in R-GCN we can write propagation as Eq. 47. The details of Eq. 47 are as following ,to compute the message at node *x* at level $(h + 1)$, we need to sum up over all the relation types *R*, for every relation type check the neighbors N_x^r of node *x* that are connected with this relation type *r*. We have transformation matrix *W* indexed by relation type. Then transformation matrix (W_r^h) specific to relation type *r* is taken with each neighbor *y* that are connected to *x*. Embedding of node *x* from

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

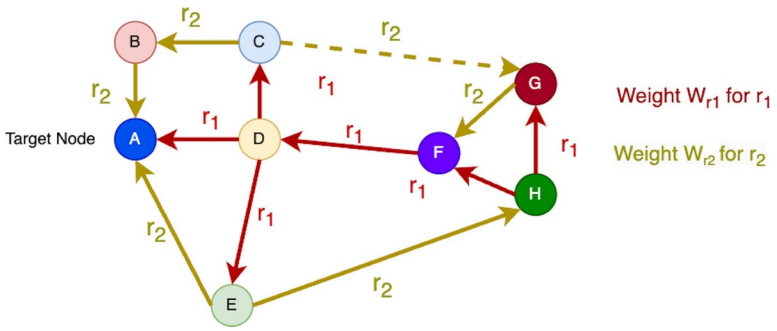


Fig. 40 Weight matrix for two different relationship types

previous layer is also taken into consideration. Here $\frac{1}{c_{x,r}}$ is number of incoming relation of given type r to node x.

$$K_x^{(h+1)} = \sigma\left(\sum_{r\in R}\sum_{y\in N_x^r}\frac{1}{c_{x,r}}W_r^h k_y^h + W_0^h k_x^h\right) \tag{47}$$

Link prediction in heterogeneous graphs involves the task of predicting various sorts of links. Since some relation types are common and others are uncommon, random splitting of links in a heterogeneous graph is inefficient. Hence for each relation type, divide the relationship into training message edges, training supervision edges, validation edges, and testing edges.This should be done for each relation type, and then all of the message edges for the training set, training supervision edge set, validation edges set, and testing edges set should be merged. This means for very rare relation type also some of the instances will be in training, validation and testing edge set. There are different methods to predict links in heterogeneous graph, for example refer Fig. 40 , suppose we want to predict link of type r_2 between nodes C and G. Means whats the probability of edge between C and G represented by (C, r_2, G). Assume this edge is training supervision edge and all other edges are training message edges. We can use R-GCN to score the edges , take final layer embedding of node C (k_C^h) and final layer embedding of node G (k_G^h), the have a relation specific scoring function $f_{r_2}(k_C, k_G) = k_C^T W_{r_2} k_G$ that will take embedding and transform it into a real value. For interpreting the function f_{r_2} send it through activation function like sigmoid etc. to get link probability. In the subsequent section, we will explore an attractive instance of heterogeneous graphs known as a knowledge graph.

3.3.7.2 Knowledge graph A Knowledge Graph (KG) refers to a methodical depiction of knowledge that use nodes, edges, and properties to represent things, relationships, and semantic information. Knowledge graphs play a crucial role in the fields of artificial intelligence and knowledge management by facilitating the organisation and interconnection of information (Guu et al. 2015; Yang et al. 2015; Chen et al. 2020; Shi et al. 2017b). KG enable the execution of intricate queries and provide support for a wide range of applications. Entity-relationship diagrams possess significant value in the representation and

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

comprehension of relationships between things or concepts. Consequently, they serve as a potent instrument for data integration, semantic search, recommendation systems, and knowledge-based artificial intelligence. Knowledge graphs frequently remain unfinished as a result of the extensive nature of knowledge in the actual world. However, they offer a structure that enables the effective capture and utilization of related information (Nayyeri et al. 2021; Neelakantan et al. 2015; Yang and Liu 2021; Bastos et al. 2023) . The representation of a knowledge graph can be expressed as triples, denoted as (*h*, *r*, *t*). The head entity *h* is in a relation *r* with the tail entity *t*, e.g.,(*Delhi*, *CapitalOf*, *Bharat*). While considering the automated completion of knowledge graph relations, it can be conceptualized as the task of link prediction within a knowledge graph. To do so, entities and relations are represented as points and vectors in an embedding space, which is the basic idea behind modelling a knowledge graph via shallow embedding. Hence concept of link prediction involves the objective of generating embeddings such that the embedding of the head entity and relation, denoted as (*h*, *r*), is in close proximity to the embedding of the tail entity, denoted as *t*, when provided with a true triple (*h*, *r*, *t*) (Lin et al. 2015; Nguyen et al. 2016). Two fundamental questions arise: how to incorporate the (*h*, *r*) embedding and how to provide a definition of proximity. To express this, various models with various geometric intuitions are presented.

3.3.7.3 TransE Bordes et al (Bordes et al. 2013) suggested a canonical model that is simple to train, has few parameters, and is scalable even for massive data sets. In the TransE model, the objective is to create embedding for the head entity, relation, and tail entity in a manner that allows the addition of the head entity embedding and the relation embedding to provide the embedding of the tail entity ($h + r \approx t$). Hence the scoring function is defined as $f_r(h, t) = -||h + r - t||$. Scoring function should be capable of entertaining different types of relations in heterogeneous graphs, for example (*h*, *roommate*, *t*) then (*t*, *roommate*, *h*) should also exist called as symmetric relation. Means the embedding according to relation type should also meet. TransE model is capable of effectively representing antisymmetric, inverse, and compositional interactions. However, it is not suitable for modeling symmetric relations and 1-to-N relations.

3.3.7.4 TransR Lin et al. (2015) proposed a different method that will allow to fix some issues present in TransE. TransR is a derivative of the first TransE model, designed with the objective of representing entities and relations within a knowledge network using a shared vector space. In TransE, relations are depicted as translation vectors that move entities from one entity's vector to another to express the semantics of the relationship.TransR enhances the TransE model by incorporating the concept of distinct embedding spaces for entities and relations. Instead of acquiring knowledge through a unified vector space for all entities and relations, TransR adopts a methodology that involves the creation of several vector spaces, each specifically linked to a distinct relation type. This enables the model to effectively capture intricate connections between items, as diverse associations exhibit distinct attributes. TransR models entities as points, and for each relationship, two things must be learned: the translation relation vector and the projection matrix. Which makes TransR model capable of

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

effectively representing symmetric, 1-to-N, antisymmetric and inverse relations. However, it is not suitable for modeling composition relation.

3.3.7.5 DistMult The TransE and TransR models employ a scoring function that functions as a distance metric in the embedding space. However, Yang et al. (2015) introduced a new method called DistMult, which utilises a bilinear model for the scoring function. The fundamental concept underlying DistMult is to employ a straightforward yet efficient scoring mechanism for evaluating the compatibility between an entity, a relationship, and another entity. The scoring function is derived from the element-wise multiplication of the embeddings. The scoring function for a triple (*h*, *r*, *t*) including three entities, namely head, relation, and tail, is computed as $f_r(h,t) = \sum_i h_i \cdot r_i \cdot t_i$. The idea is if (*h*, *r*, *t*) is true then score is high otherwise score should be low. The DistMult model successfully describe 1-to-N and symmetric data. It is, however, unsuitable for modeling antisymmetric, composition and inverse relations.

3.3.7.6 ComplEx Trouillon et al. (2016) developed ComplEx embedding based on DistMult, which embeds entities and relations in complex vector space(C^k). The ComplEx model is utilised to represent entities and relations by employing complex numbers. A complex-valued embedding is assigned to each item and relation, comprising a real component and an imaginary component. The sophisticated nature of this representation allows ComplEx to effectively capture intricate and subtle interactions inside heterogeneous/knowledge graphs.The fundamental aspect of ComplEx resides in its scoring function, which quantifies the degree of compatibility between entities and relations. The scoring function employs the utilisation of complex conjugates and element-wise multiplications of the embeddings in order to calculate the score for a given triple, consisting of a head entity, a relation, and a tail entity. The score function of ComplEx an be represented as $f_r(h,t) = Re(\sum_i h_i \cdot r_i \cdot t_i)$. Here *Re* represents real part of the complex function. The ComplEx model has demonstrated efficacy in accurately representing data with antisymmetric, 1-to-N, inverse, and symmetric characteristics. However, it is not ideal for modelling the composition relation.

Different knowledge graphs can exhibit significantly distinct patterns, indicating that there is no universal embedding method that can effectively capture the characteristics of all knowledge graphs. The appropriateness of the modelling approach and the ability to predict relationships depend on the specific sorts of relationships the user wishes to model and the specific type of relationship the user aims to predict.

GNN are useful for studying and harnessing the complex architecture of heterogeneous networks. The models discusssed above RGCN, TransE, TransR, DistMult, and ComplEx have shown effective in a variety of applications, including recommender systems, social network analysis, and fraud detection. These models, by design, can reflect the numerous relationships inherent in diverse networks, improving their analysis and use. RGCN is particularly useful in recommender systems because it takes advantage of the heterogeneity of data, which includes users, objects, and interactions (Bing et al. 2023). For example, in movie recommendation systems that leverage datasets such as MovieLens (Kumar et al. 2020) and IMDB (Fu et al. 2020b), RGCN can mimic complex user-item interactions. These

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

datasets cover a wide range of interactions, including user ratings and movie reviews, allowing RGCN to create highly tailored suggestions. TransE and TransR improve these systems by embedding these heterogeneous relationships into continuous vector spaces, allowing for more precise predictions of user preferences. DistMult and ComplEx models are extremely useful for social network analysis. DistMult's bilinear diagonal model excels at handling symmetric relationships, making it ideal for studying datasets like DBLP (Fu et al. 2020b). The DBLP, which provides data on academic publications, authors, and conferences, aids in the discovery of intellectual collaboration and influence trends. complicated expands on this feature by managing asymmetric relationships, making it excellent for evaluating complicated interactions in networks like IMDB's actor-director-movie relationships. These models provide insights into social structures and interactions, making it easier to identify major influencers and community structures. Fraud detection benefits substantially from the use of models such as TransR and ComplEx, which can represent the complex, multi-relational connections found in datasets like OGBN-MAG (Hu et al. 2020b) and OGBL-BIOKG (Hu et al. 2020b). These datasets include precise information about intellectual and biological entities, as well as their interconnections. TransR's capacity to project entities and relations into separate spaces enables a more sophisticated understanding of these interactions, whilst ComplEx's handling of both symmetric and asymmetric relationships provides a full picture of probable fraud situations. These models help detect fraudulent actions by finding patterns that depart from the norm, hence improving system security and trust.

3.3.8 Heterogeneous dataset

This section will examine diverse datasets such as DBLP (Fu et al. 2020b), MovieLens (Kumar et al. 2020), IMDb (Fu et al. 2020b), OGBN-MAG (Hu et al. 2020b), and OGBL-BIOKG (Hu et al. 2020b). These datasets are used as standards for assessing the effectiveness and resilience of various graph-based modelling methods. Each dataset corresponds to a distinct real-world situation: DBLP is an academic database, MovieLens is a dataset for movie recommendations, IMDb is centred on the entertainment sector, OGBN-MAG is a detailed academic network, and OGBL-BIOKG is a biological knowledge graph.

- MovieLens :- The MovieLens dataset is an excellent example of a heterogeneous graph, distinguished by its various types of nodes and edges. It consists of 10,352 nodes categorised as 'movie' and 'user'. The 'movie' nodes represent specific films, while the 'user' nodes reflect individuals who have rated these films. The graph is enhanced by an additional 100,836 edges, all representing the relationship 'user-rates-movie', which signifies a user's rating of a movie. The diverse composition of the MovieLens dataset makes it a significant asset for academic research, especially in the fields of recommendation systems and graph neural networks (Fig. 41). It offers a comprehensive and authentic framework for assessing algorithms created to manage intricate graph topologies containing various sorts of nodes and edges.
- IMDb :- The IMDb dataset is a diverse graph that illustrates the links between actors, directors, and movies . The network comprises 11,616 nodes and 34,212 edges. Nodes represent movies, directors, and actors, each with 3,066 features. The edges represent connections between directors and films, as well as actors and films (Fig. 42). The dataset is organised with different sorts of edges, such as 'movie-to-director', 'movie-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fig. 41 MovieLens Dataset

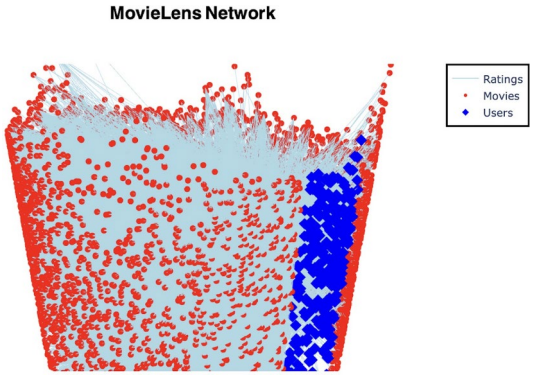
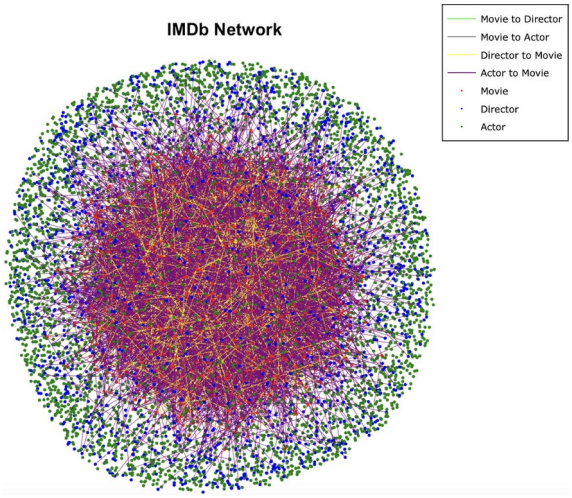


Fig. 42 IMDb Dataset



to-actor', 'director-to-movie', and 'actor-to-movie', demonstrating the reciprocal aspect of these connections. The dataset is highly interconnected and important for analysing trends and insights in the film business. It is a great resource for applications such as recommendation systems, network analysis, and machine learning research that focuses on diverse graphs.

- DBLP :- The DBLP dataset is an extensive scholarly network that includes a diverse range of academic connections. The system comprises 26,128 nodes categorised into four main groups: 'author', 'document', 'term', and 'conference'. The dataset has been enhanced with an additional 239,566 edges representing other types of links like authorship, subject significance, and conference affiliations. This dataset is crucial for analysing and comprehending the complex network of academic relationships and thematic links within the scholarly community. Figure 43 depicts the DBLP dataset visually, where nodes indicate various elements in the academic network such as authors, articles, phrases, and conferences. Up to 300 nodes of each type are shown to improve visualisation clarity and manage complexity. Each node type is designated a unique colour for clear differentiation: red for 'author' nodes, blue for 'paper' nodes, green for 'term'

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fig. 43 DBLP Dataset

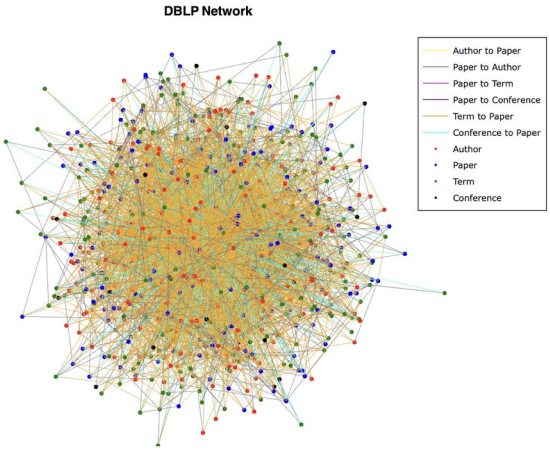
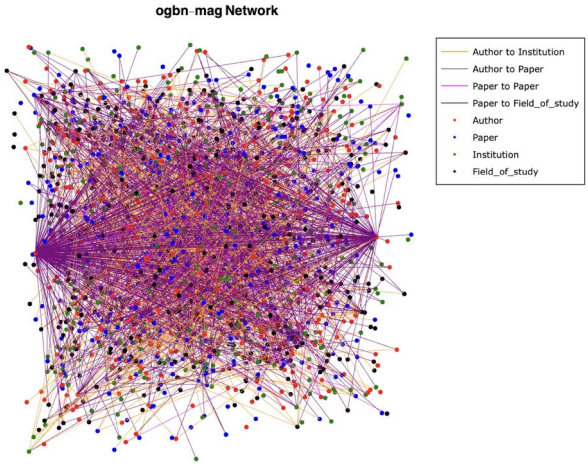


Fig. 44 OGBN-MAG Dataset



nodes, and black for 'conference' nodes. The graph's edges are coloured according on their types, with different shades indicating distinct associations including author-to-paper, paper-to-term, and paper-to-conference. The color-coded approach helps to rapidly identify the relationships within the scholarly network.

- OGBN-MAG :- The OGBN-MAG dataset is a large academic graph dataset that represents the complex relationships in the academic field. The dataset contains 1,649,743 nodes, divided into four types: 'author', 'field_of_study', 'institution', and 'article', with corresponding counts of 1,134,649, 59,965, 8740, and 736,389. The dataset contains 21,111,007 edges representing different relationships like author affiliations, authorship, citations, and topic associations. OGBN-MAG is a powerful tool for analysing academic networks, studying scholarly collaborations, and investigating the relationship between research themes and institutions. Figure 44 of the OGBN-MAG dataset visually represents several entities in the academic landscape, including authors, articles, institutions, and topics of research. The visualisation is restricted to a maximum of 500 nodes per type to provide clarity and manageability. Each node type is designated

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

a distinct colour for simple recognition: red for 'author' nodes, blue for 'paper' nodes, green for 'institution' nodes, and black for 'field_of_study' nodes. The graph's edges are coloured to show different associations, such as author affiliation with institution, author writing paper, paper citing paper, and paper having topic field of study. The color-coded system helps quickly distinguish the many relationships in the academic network.

- OGBN-BIOKG :- The OGBL-BioKG dataset is an extensive biological knowledge graph that represents intricate relationships in the biomedical field. The dataset consists of 93,773 nodes, classified into five specific categories: 'disease', 'drug', 'function', 'protein', and 'sideeffect'. The collection contains 5,088,434 edges representing various biomedical linkages, including disease-protein associations, drug-disease interactions, and protein-function relationships. This dataset is a significant resource for studying the complex relationships between different biological entities and understanding the mechanisms involved in illnesses and medication effects. Figure 45 depicts the OGBL-BioKG dataset visually, where nodes represent various things in the biomedical field such as diseases, medications, functions, proteins, and side effects. The visualisation is limited to a maximum of 200 nodes per type to maintain clarity and manageability. Each type of node is color-coded for easy identification: disease nodes are red, drug nodes are blue, function nodes are green, protein nodes are magenta, and side effect nodes are purple. The graph's borders are evenly coloured in light grey to indicate the various biological interactions. The color-coded approach helps quickly distinguish the many connections in the biological network.

Table 12 lists datasets explained above, which offer a strong basis for link prediction tasks in several fields. Each dataset has distinct node and edge types that represent different links, such as user-movie ratings, movie-director connections, author-paper collaborations, academic affiliations, and biomedical interactions. Heterogeneous networks provide a variety of structural and feature information, which is advantageous for creating and assessing link prediction models. Researchers can use the complex relationships and characteristics found

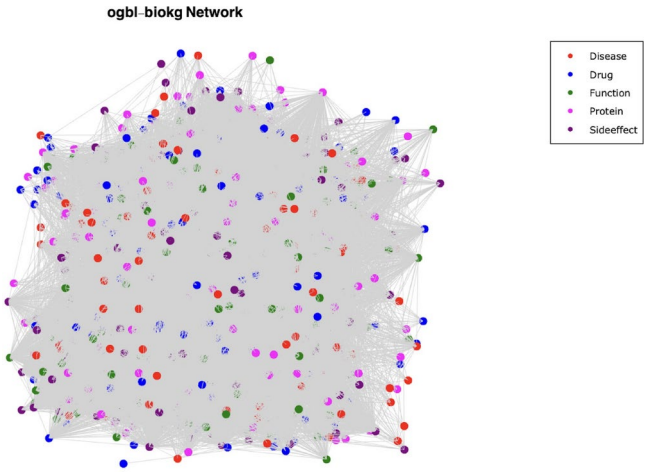


Fig. 45 OGBN-BIOKG Dataset

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 12 Dataset Details for MovieLens, IMDB,DBLP , OGBN-MAG, and OGBL-BIOKG

Dataset	Nodes	Edges	Node features	Edge features
MovieLens	10,352	100,836	Movie, user	User-rates-movie
IMDB	11,616	34,212	Movie, director, actor	Movie-to-director, movie-to-actor, etc.
DBLP	26,128	239,566	Author, paper, term, conference	Author-to-paper, paper-to-author, etc.
OGBN-MAG	1,939,743	21,111,007	Author, field of study, institution, paper	Affiliated with, writes, cites, has topic
OGBL-BIOKG	93,773	5,088,434	Disease, drug, function, protein, sideeffect	Various biomedical relations

Table 13 RGCN performance for Link Prediction on datasets

Dataset	RGCN Layers	Micro-F1 score	Macro-F1 score	AUC score
MovieLens	2	0.6616	0.6521	0.6616
IMDB	2	0.4162	0.4168	0.6764
DBLP	2	0.3403	0.5155	0.6035
OGBN-MAG	2	0.5874	0.5965	0.6200
OGBL-BIOKG	2	0.5718	0.6077	0.8167

in these datasets to improve link prediction tools, which can lead to breakthroughs in recommendation systems, academic network analysis, and biological knowledge discovery.

3.3.9 Results of neural network based techniques on heterogeneous datasets

Our objective is to acquire a comprehensive understanding of the strengths and limits of the approaches being reviewed by utilising these unique and substantial datasets. The purpose of this thorough examination is to enhance our understanding of how these techniques can be used and how effective they are in analysing many types of graphs. This will enable us to tackle complicated data-driven problems in a more informed manner.

3.3.9.1 RGCN Table 13 anf Fig. 47 summarises the performance of the RGCN (Schlichtkrull et al. 2018) for link prediction on different datasets. RGCN is a neural network model specifically created to process heterogeneous graphs, showing varied levels of effectiveness depending on the dataset. The model's performance is assessed using measures including the Micro-F1 score, Macro-F1 score, and the Area Under the Receiver Operating Characteristic Curve (AUC) score. The model attains a Micro-F1 score of 0.6616, a Macro-F1 score of 0.6521, and an AUC score of 0.6616 on the MovieLens dataset. The performance on the

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

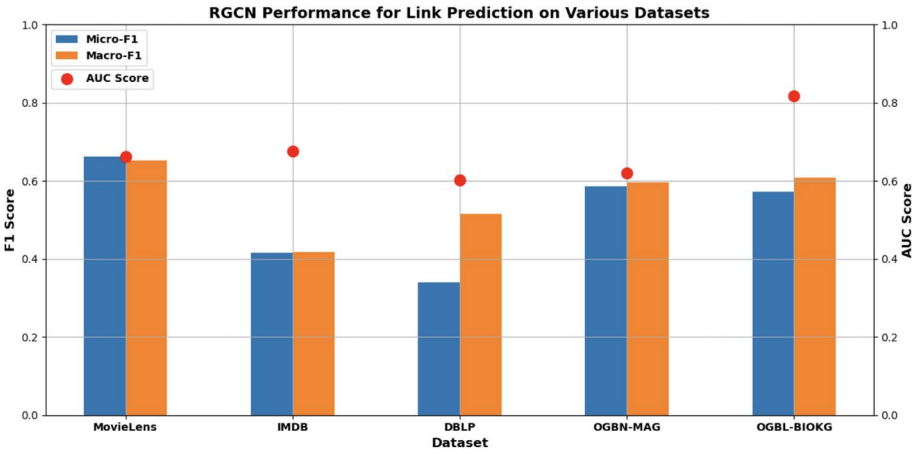


Fig. 47 The effectiveness of RGCN for link prediction on the MovieLens, IMDB, DBLP, OGBN-MAG, and OGBL-BIOKG datasets is illustrated by the Micro-F1, Macro-F1, and AUC scores

DBLP dataset is inferior compared to other datasets, with a Micro-F1 score of 0.3403 and a Macro-F1 score of 0.5155, but a somewhat higher AUC score of 0.6035.

The differences in RGCN's performance on various datasets can be explained by the unique features of each dataset, including the graph structure, node and edge type complexity, and label distribution. The OGBL-BIOKG dataset, containing diverse biological relationships, achieved a higher AUC value of 0.8167 (Fig. 46), demonstrating that RGCN is more adept at capturing the intricate interactions within this dataset. Conversely, the inferior performance on the IMDB and DBLP datasets could be attributed to the sparse connections or the intricate interconnections between entities in these academic networks. Figure 1 displays AUC curves that demonstrate the model's performance on several datasets, offering insights into its predictive capabilities for links in heterogeneous graphs.

3.3.9.2 TransE The TransE (Bordes et al. 2013) model is well regarded in the field of knowledge graph embeddings for its exceptional performance in link prediction tasks on different graph-structured datasets. TransE has remarkable versatility when used with datasets like MovieLens, IMDB, DBLP, ogbn-mag, and ogbl-biokg, covering diverse categories such as movie recommendations, academic publications, and biological knowledge graphs. The model's capacity to adapt to various datasets demonstrates its robustness and applicability in numerous disciplines. The table labelled "TransE performance comparison" offers vital insights into the effectiveness of TransE in link prediction by demonstrating its performance on various datasets.

The TransE model reliably generates Micro-F1 and Macro-F1 scores that are steady across all datasets, typically around 0.4 and 0.67, respectively. The model consistently performs well in making predictions, independent of the dataset's properties. The model may be improved to better distinguish between distinct classes, as shown by the middling Macro-F1 scores. The AUC values, which measure the model's capacity to distinguish between positive and negative class labels, show more heterogeneity among datasets. TransE outper-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

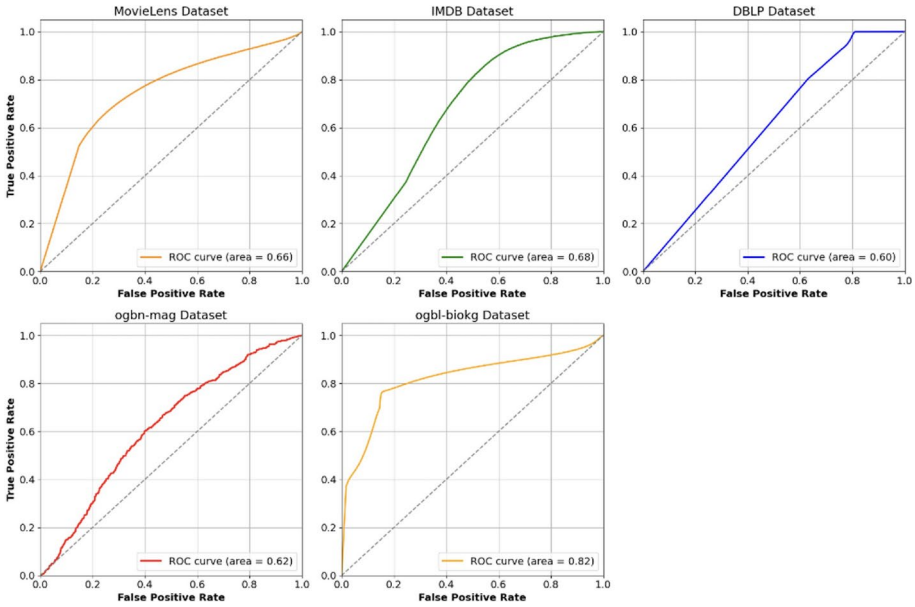


Fig. 46 AUC Curves for RGCN Link Prediction on MovieLens, IMDB, DBLP,OGBN-MAG and OGBL-BIOKG

forms on the ogbn-mag and DBLP datasets, attaining AUC scores of 0.8796 (Fig. 48) and 0.8617, respectively. The outstanding performance can be credited to specific characteristics of datasets, including the graph's structure, link type, or connection density, which are well-suited for TransE's embedding approaches. The lower AUC score on the ogbl-biokg dataset suggests difficulties in handling complex network architectures with fewer connections (Table 14 and Fig. 49).

TransE consistently performs well in achieving satisfactory Micro-F1 and Macro-F1 scores on various datasets, demonstrating its reliability for link prediction tasks in heterogeneous graphs. This is significant as it indicates that TransE can effectively capture and describe the underlying patterns, even with variations in data format, types of links, and domain-specific complexities. This capacity is especially beneficial in real-world scenarios where datasets are frequently intricate and varied. These investigations show that TransE is very effective in managing diverse graph data, especially in detecting complex connections across different domains.

3.3.9.3 TransR The Table 15 and Fig. 51 displays the performance of the TransR (Lin et al. 2015) on different datasets, using three key evaluation metrics: Micro-F1 Score, Macro-F1 Score, and AUC Score. The measures provide important insights into the accuracy, class distribution, and discriminative capability of the model. Each occurrence of TransR follows a uniform structure consisting of three layers: Entity, Relation, and Projection. This archi-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

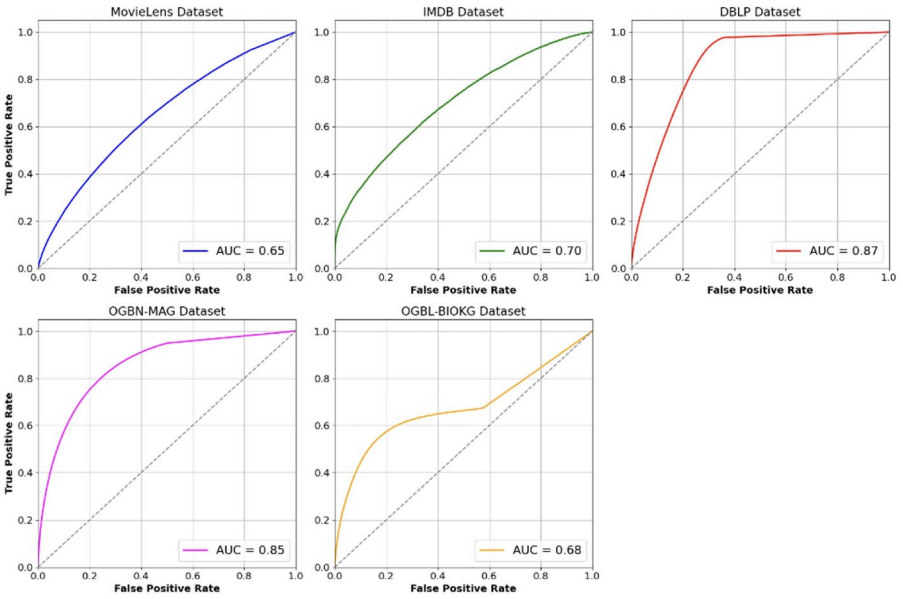


Fig. 48 AUC Curves for TransE Link Prediction on MovieLens, IMDB, DBLP,OGBN-MAG and OGBL-BIOKG

Table 14 TransE performance for Link Prediction on datasets

Dataset	TransE layers	Micro-F1 score	Macro-F1 score	AUC score
MovieLens	1	0.4286	0.7500	0.6464
IMDB	1	0.4000	0.6667	0.7019
DBLP	1	0.4286	0.7500	0.8652
OGBN-MAG	1	0.4000	0.6667	0.8547
OGBL-BIOKG	1	0.4000	0.6667	0.6757

tecture allows for the implementation of a standardised assessment system across different datasets.

The Micro-F1 Score varies from 0.3333 in the MovieLens dataset to 0.4000 in the IMDB, DBLP, OGBN-MAG, and OGBL-BIOKG datasets. It is a statistic that takes into account the individual contributions of all classes to compute an average. This variation showcases the model's ability to accurately classify relationships on a case-by-case basis. The constant score of 0.4000 across multiple datasets demonstrates a dependable performance in precision and recall for various types of relational data. However, the existence of the lowest score in the MovieLens dataset could indicate challenges in handling its distinct features or structure, potentially requiring adjustments to the model or data preparation. The Macro-F1 Score, which computes the F1 score for each class individually and then calculates the average, shows a similar trend, ranging from 0.5000 in the MovieLens dataset to 0.6667 in the other datasets. The higher Macro-F1 scores in most datasets suggest a balanced performance of the model across several classes. Ensuring balance in datasets with uneven

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

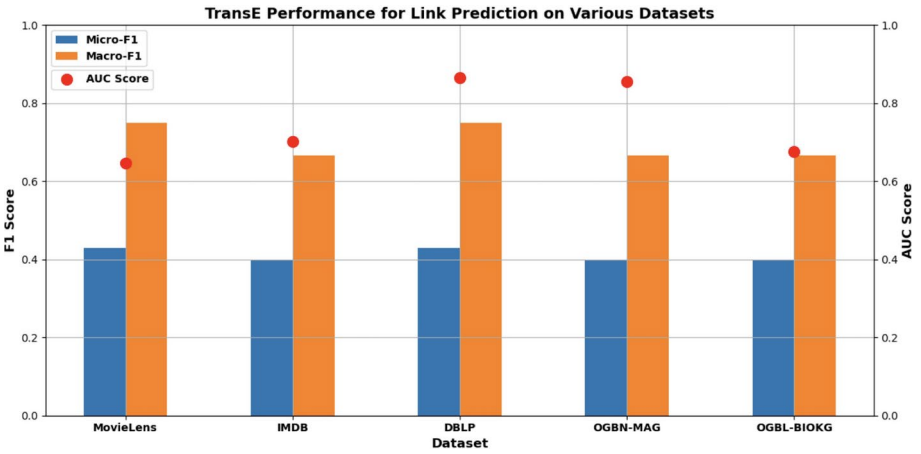


Fig. 49 The effectiveness of TransE in predicting links across a variety of datasets is illustrated by a bar and scatter plot. The AUC scores for each dataset are denoted by the red circles, while the Micro-F1 and Macro-F1 scores are represented by the bars. (Color figure online)

Table 15 TransR performance for link prediction on datasets

Dataset	TransR layers	Micro-F1 score	Macro-F1 score	AUC score
MovieLens	3 (Entity, Relation, Projection)	0.3333	0.5000	0.7525
IMDB	3 (Entity, Relation, Projection)	0.4000	0.6667	0.8535
DBLP	3 (Entity, Relation, Projection)	0.4000	0.6667	0.8860
OGBN-MAG	3 (Entity, Relation, Projection)	0.3333	0.5000	0.6319
OGBL-BIOKG	3 (Entity, Relation, Projection)	0.4000	0.6667	0.7698

class distributions is crucial to prevent any single class from disproportionately affecting the model's overall accuracy. The AUC Score of 0.8860 in the DBLP dataset indicates that the model has a high ability to distinguish between positive and negative linkages, showing excellent performance in identifying relevant ties in academic citation networks (Fig. 50). The OGBL-BIOKG dataset's low score of 0.7698 highlights the challenges the model faces in dealing with intricate and implicit linkages often present in biological knowledge graphs. The variation in AUC ratings across datasets emphasises the model's inconsistent performance in various relationship scenarios. The high AUC score in the DBLP dataset is likely due to the structured form of academic citations, which aligns well with the TransR model's ability to capture distinct relational patterns. The complex and varied linkages in biological datasets, such as those in OGBL-BIOKG, pose a substantial challenge that require advanced or specialised algorithms to effectively identify the underlying patterns.

The performance metrics of these datasets offer valuable insights into the abilities and limitations of the TransR model in link prediction tasks. The model regularly shows great accuracy and maintains a balanced class distribution, as indicated by the Micro-F1 and

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

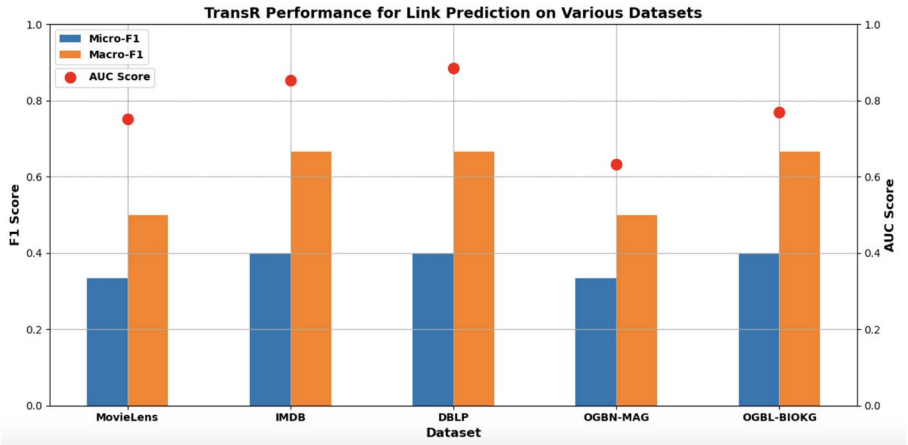


Fig. 51 Performance evaluation of TransR model for link prediction. The AUC scores are denoted by red circles, while the Micro-F1 and Macro-F1 scores are represented by bars

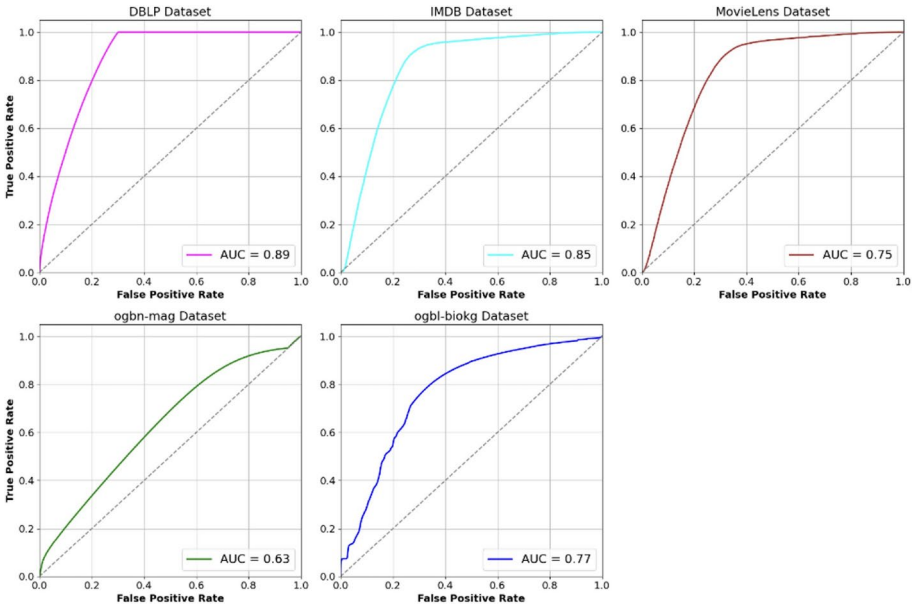


Fig. 50 AUC Curves for TransR link prediction on MovieLens, IMDB, DBLP,OGBN-MAG and OGBL-BIOKG

Macro-F1 scores. This validates its effectiveness as a diverse method for forecasting connections. The variety of AUC scores highlights the model's differing efficacy in handling different forms of relational data. The data indicates that TransR is skilled at managing structured relational datasets but may require enhancements to effectively manage complex or unstructured data types, like those seen in biological knowledge graphs.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 16 DistMult performance for link prediction on datasets

Dataset	DistMult layers	Micro-F1 score	Macro-F1 score	AUC score
MovieLens	2	0.5227	0.5787	0.7709
IMDB	2	0.7171	0.7330	0.8086
DBLP	2	0.7260	0.7404	0.8371
OGBN-MAG	2	0.7134	0.7283	0.9150
OGBL-BIOKG	2	0.7682	0.7685	0.8056

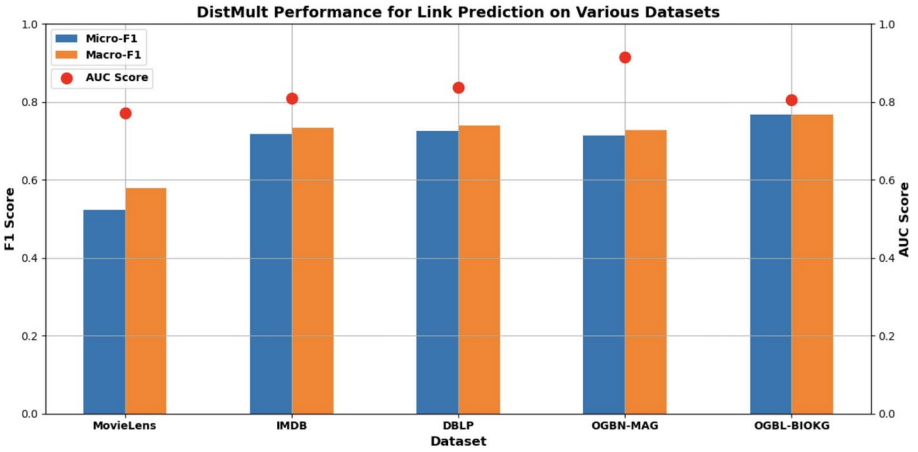


Fig. 53 Performance of distmult for Link Prediction across different data sets, highlighting Micro-F1, Macro-F1 scores, and AUC values

3.3.9.4 DistMult The DistMult (Yang et al. 2015) model's performance in link prediction across several datasets is thoroughly examined, showcasing its properties in different scenarios. Refer to Table 16 for more details. Each dataset has unique characteristics that provide a specific structure for evaluating the model's efficiency and accuracy. The DistMult model achieves an AUC Score of 0.7709 with the MovieLens dataset, indicating a decent ability to differentiate between different types of links (Fig. 53). The Micro-F1 score is 0.5227 and the Macro-F1 score is 0.5787, both relatively low, suggesting challenges in handling the specific complexities of user-movie interactions in this dataset. The model's performance improves in the IMDB dataset with an AUC Score of 0.8086. The Micro-F1 score is 0.7171 while the Macro-F1 score is 0.7330. These values suggest a high level of understanding of the complex relationships between films, performers, and directors. The model's performance on the DBLP dataset, which signifies academic collaborations, has been enhanced. The DistMult model attains an AUC score of 0.8371, along with Micro-F1 and Macro-F1 ratings of 0.7260 and 0.7404, respectively. These scores indicate a strong ability to predict academic relationships, despite the complex nature of academic networks. The OGBN-MAG dataset shows a notable enhancement in performance, with the DistMult model obtaining an AUC Score of 0.9150. The model's exceptional performance is demonstrated by its overall score, with Micro-F1 and Macro-F1 scores of 0.7134 and 0.7283, respectively. The results emphasise the model's ability to navigate the complex academic network, especially in predicting links between authors and publications. In the OGBL-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

BIOKG dataset, consisting of biological knowledge graphs, the model achieves an AUC Score of 0.8056. The Micro-F1 score is 0.7682 and the Macro-F1 score is 0.7685. These ratings show competency but also reveal the challenges of simulating complex biological systems and their interactions (Fig. 52).

In conclusion, the DistMult model has shown its versatility and effectiveness in link prediction tasks based on the performance on various datasets. The model has outstanding performance, particularly in datasets with complex linkages and vast structures, such as OGBN-MAG. This showcases its capacity to efficiently manage a variety of complex and challenging graph-based datasets.

3.3.9.5 ComplEx The ComplEx (Trouillon et al. 2016) model's performance in link prediction has been assessed on many datasets, each with unique features and structural qualities. The table provides a detailed comparison of the model's performance on five datasets: MovieLens, IMDB, DBLP, OGBN-MAG, and OGBL-BIOKG. It is labelled as Table 17. The ComplEx model, with its four-layer structure consisting of two actual and two imaginary layers, has demonstrated different levels of effectiveness in forecasting connections within these networks.

The ComplEx model obtained a Micro-F1 score of 0.4463, a Macro-F1 score of 0.5194, and an AUC score of 0.7130 when applied to the MovieLens dataset. The results suggest a moderate level of predictive accuracy, with the model showing a reasonable ability to differentiate between positive and negative link classes. The Micro-F1 score evaluates the model's performance in predicting individual links, while the Macro-F1 score assesses its capability to forecast various sorts of links with a balanced approach. The model's perfor-

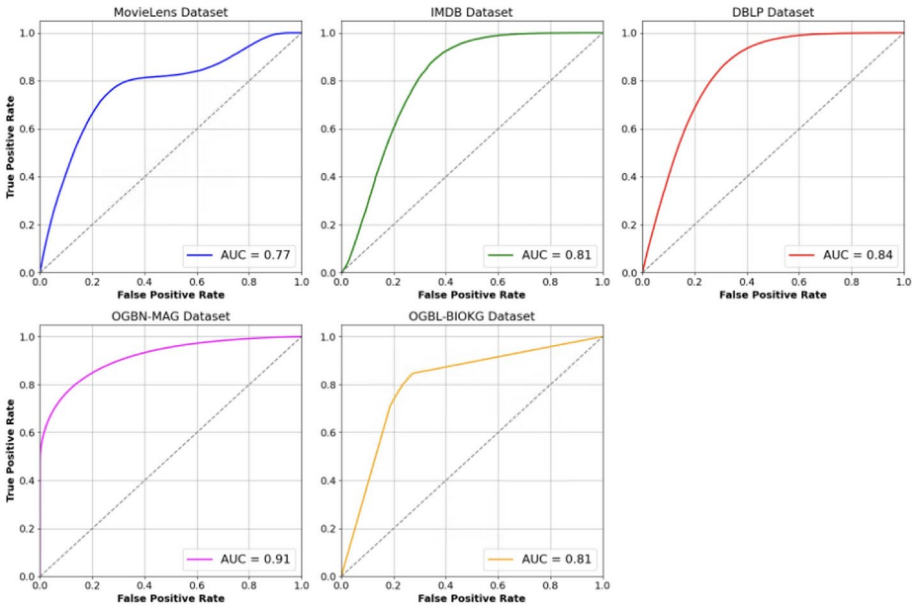


Fig. 52 AUC Curves for DistMult link prediction on MovieLens, IMDB, DBLP,OGBN-MAG and OGBL-BIOKG

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 17 ComplEx performance for link prediction on datasets

Dataset	ComplEx layers	Micro-F1 score	Macro-F1 score	AUC score
MovieLens	4 (2 Real, 2 Imaginary)	0.4463	0.5194	0.7130
IMDB	4 (2 Real, 2 Imaginary)	0.7316	0.7482	0.6964
DBLP	4 (2 Real, 2 Imaginary)	0.7313	0.7479	0.7065
OGBN-MAG	4 (2 Real, 2 Imaginary)	0.5736	0.5737	0.6070
OGBL-BIOKG	4 (2 Real, 2 Imaginary)	0.8008	0.7996	0.9180

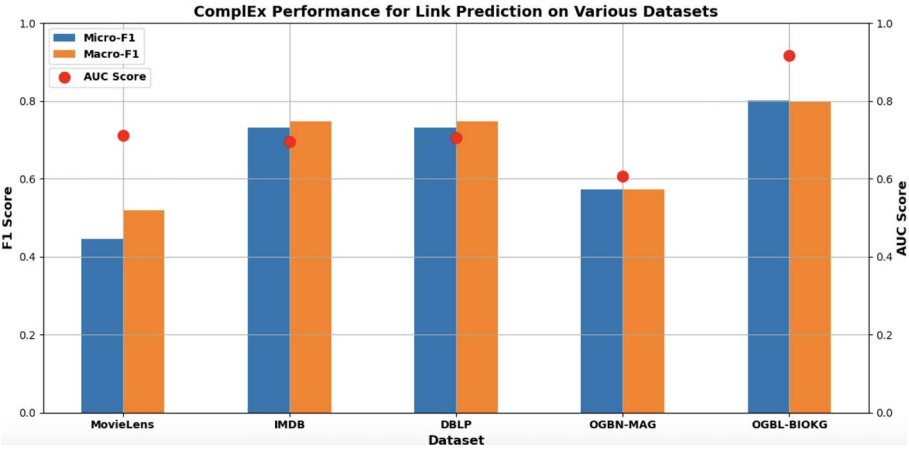


Fig. 55 The performance of ComplEx for link prediction across multiple datasets is demonstrated. The bar graphs show the Micro-F1 and Macro-F1 scores, while the red circles show the AUC scores. This mix of bar and scatter charts gives a complete picture of the model's performance in link prediction tasks on the MovieLens, IMDB, DBLP, OGBN-MAG, and OGBL-BIOKG datasets. The thorough performance data illustrate both the model's strengths and places for improvement

mance has significantly improved on the IMDB and DBLP datasets, achieving Micro-F1 scores of 0.7316 and 0.7313, and Macro-F1 values of 0.7482 and 0.7479, respectively. The AUC scores for these datasets are 0.6964 and 0.7065, demonstrating a robust discriminatory capability of the model in these scenarios. The elevated F1 scores indicate that the ComplEx model excels at precisely forecasting connections in these datasets, because of their intrinsic structural characteristics or the types of relationships seen in the data (Fig. 55). The model's performance on the OGBN-MAG dataset is subpar, achieving a Micro-F1 score of 0.5736, a Macro-F1 score of 0.5737, and an AUC score of 0.6070. The algorithm struggles to reliably predict relationships in complex and diverse networks like academic citation networks in the OGBN-MAG dataset. The OGBL-BIOKG dataset demonstrates the model's highest performance, achieving a Micro-F1 score of 0.8008, a Macro-F1 score of 0.7996, and an AUC score of 0.9180. The ComplEx model is particularly effective in predicting

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

links within organised biological knowledge networks, especially when the interactions are explicit and well-defined.

The link prediction performance of the ComplEx model, which consists of 4 layers with 2 real and 2 imaginary layers, varies depending on the features and organisation of each dataset, as shown in Fig. 54. The model exhibits potential in structured and clearly defined networks; yet, it faces difficulties in complex and diverse scenarios. This emphasises the need of taking into account the unique attributes of datasets and the layer configuration of the model when utilising link prediction approaches. It emphasises the need for customised strategies to successfully handle various link prediction jobs.

This section thoroughly assesses Feature Learning Techniques for link prediction on many types of graphs, including Matrix Factorization, Path and Walk-Based Methods, GNN-based tactics, and techniques tailored for heterogeneous datasets. The section on Matrix Factorization highlights the effectiveness of DMF and PMF. These methods use deep learning and probabilistic frameworks to extract predictive link features from the graph's adjacency matrix. Path and Walk-Based Techniques , i.e, Random Walk Embedding, DeepWalk, Node2vec, and Walklets are investigated for their efficiency in embedding node sequences, utilising unsupervised learning and random walk techniques. The discussion shifts to Graph Neural Network-Based techniques, emphasising the computational frameworks supporting GCN, GraphSAGE, and GAT techniques. The methods are highly effective in capturing neighbourhood connectedness via new aggregation and attention processes, providing accurate predictions even in intricate network topologies. The discussion concludes by examining Techniques on Heterogeneous Graphs, focusing on the unique issues presented by multi-typed nodes and edges in knowledge graphs. Relational GCN, TransE, TransR, DistMult, and ComplEx techniques are evaluated based on their capacity to traverse complex semantic relationships inside graphs. The results subsection demonstrates

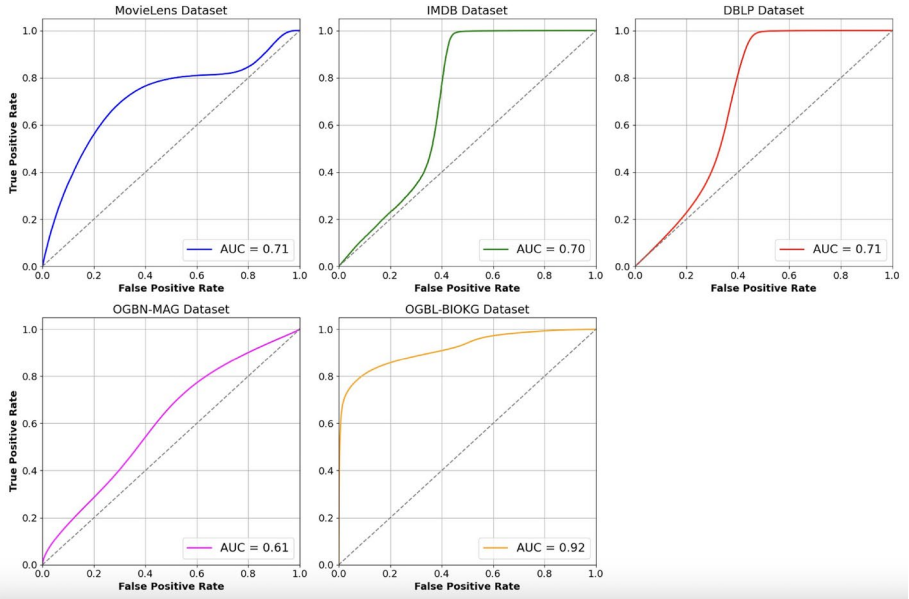


Fig. 54 AUC Curves for ComplEx Link Prediction on MovieLens, IMDB, DBLP,OGBN-MAG and OGBL-BIOKG

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

implementation of techniques to homogeneous datasets, i.e, Cora, CiteSeer, PubMed, Ego Facebook, and WikiVote, and heterogeneous datasets, DBLP, MovieLens, IMDb, OGBN-MAG, and OGBL-BIOKG. Comparative analysis is performed on the datasets,focusing on the parameters like macro-F1, micro-F1, and AUC.

4 Transfer learning on graphs

Transfer learning on graphs is a rapidly evolving field that focuses on using knowledge from one graph to enhance learning and performance on another. This strategy is especially effective in circumstances when labeled data is low since it allows models to transfer insights and patterns learned from large, well-annotated source graphs to new, often smaller, target graphs. Transfer learning can greatly decrease training time, improve model accuracy, and allow the use of GNN in different disciplines, including social network research, bioinformatics, and recommendation systems (Lee et al. 2017).Transfer learning techniques increase link prediction performance by leveraging structural information shared between the source and destination graphs. For example, approaches that discover and use common subgraphs can transfer meaningful connection information from a data-rich source graph to a sparse target graph. This method helps to efficiently disseminate link information and improves the accuracy of link predictions in the target graph.Adaptive learning approaches, such as transformed label propagation and knowledge distillation with multi-layer perceptrons, have demonstrated promising outcomes in transferring knowledge for link prediction tasks. These strategies aid in generalizing link prediction models trained on the source network to perform successfully on the target graph by focusing structural similarities and shared data between the graphs (Zheng et al. 2023). The following are the key methods employed in transfer learning for link prediction on graphs:

4.1 Label propagation

Label propagation is a semi-supervised learning technique that distributes labels in a network based on node structure and proximity. This technique makes the assumption that nodes joined by an edge are probable to have the same label. Modified label propagation can be especially useful in transfer learning for link prediction. Label propagation strategies typically include the following steps:

- Initialization: Begin by labeling a small subset of nodes.
- Propagation: Progressively update the labels of unlabeled nodes using the labels of their neighbors.
- Convergence: Repeat the propagation as long as the labels no longer differ considerably between iterations.In transfer learning, this technique can be modified to use the structure and information obtained from a well-annotated source network to improve the learning process on a less-annotated target graph. The goal is to leverage known link information from the source network to help forecast missing links in the target graph (Zheng et al. 2023; Yang et al. 2016).

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

4.2 Knowledge distillation

Knowledge distillation is a transfer learning strategy in which a simpler model (the student) is trained to reproduce the output of a more complicated model (the teacher). This method is especially effective for link prediction in GNN. By transferring learned interpretations from a well-trained GNN (teacher) to a simpler model, such as a Multi-Layer Perceptron (MLP) (student), the latter can successfully use the GNN's complicated knowledge to perform well on the target task (Hinton et al. 2014). In the context of link prediction, the procedure typically includes the following steps:

- Teacher Model Training: First, a complicated GNN is trained on a source graph containing a large amount of labeled data. This GNN extracts detailed relational and structural data from the graph.
- Student Modeling Training: A simpler model, such as an MLP, is then trained to reproduce the GNN's output. The goal is for the learner to learn the patterns and insights collected by the GNN while not requiring the same computational resources.
- Knowledge Transfer: The knowledge is transferred using strategies such as logit matching, in which the student model learns to reproduce the teacher model's output logits, and feature matching, in which the student attempts to copy the teacher model's intermediate representations.Knowledge distillation improves efficiency by transferring knowledge from a sophisticated GNN to a smaller model, allowing for the deployment of models with lower computational overhead. This technique ensures that the student model, after learning from the complicated patterns captured by the teacher, can perform well even with a simpler design. Furthermore, knowledge distillation improves scalability, allowing for the deployment of more scalable and deployable models in real-world applications. By incorporating knowledge distillation approaches into transfer learning for link prediction, researchers can capitalize on the characteristics of sophisticated GNNs while keeping the efficiency and scalability of simple models. This finally improves the prediction performance on target graphs, making the method useful and powerful for a variety of applications (Guo et al. 2023).

4.3 Meta-learning

Meta-learning, sometimes known as "learning to learn," is a strategy that allows models to swiftly adapt to new tasks by utilizing prior learning experiences. In the setting of transfer learning for graph link prediction, meta-learning emphasizes training models that can generalize across different graph structures, making them highly versatile and effective for detecting missing connections with minimal additional data.

4.3.1 Gradient-based meta-learning

Gradient-based meta-learning techniques, such as Model-Agnostic Meta-Learning (MAML), have been successfully applied to link prediction in graph domains. These techniques improve a model such that it can quickly adapt to new jobs with only a few gradient modifications. Also it ensures that the model may quickly converge to an optimal result

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

when presented with new graph data by learning an effective initialization for model parameters (Finn et al. 2017).

4.3.2 Meta-graph framework

The Meta-Graph framework, developed by Bose et al. (2020), is a remarkable example of gradient-based meta-learning for few-shot link prediction. To conditionally construct a GNN initialization, this system uses higher-order gradients and a previously learnt graph signature function. Meta-Graph effectively transfers information across various graphs using a small sample of known edges, resulting in quick adaptation and increased convergence on new tasks. The system showed considerable improvements in few-shot link prediction benchmarks, demonstrating its capacity to handle sparse data and efficiently transfer learned information.

4.3.3 Meta relational learning (MetaR)

Another popular approach is MetaR, which uses meta-learning concepts to anticipate few-shot links in knowledge networks. MetaR, developed by Chen et al. (2019), focuses on conveying relation-specific meta-information, allowing the model to rapidly adapt to new tasks by learning the most critical relational patterns. This framework dramatically improves performance on few-shot link prediction benchmarks by leveraging relation meta and gradient meta-learning techniques.

4.4 Subgraph-based models

Subgraph-based models (Zheng et al. 2021) use the local subgraph structure of a target connection to improve link prediction. Such models are very useful for transfer learning on graphs since they capture the relational patterns inside the subgraph, making them highly adaptive to new, unknown graphs. Recent research strongly supports the viability and usefulness of subgraph-based techniques for transfer learning for link prediction. Steps to implement:

- Subgraph Extraction: Subgraph-based models begin by extracting a local subgraph around the target link, usually using k-hop neighborhoods. This subgraph represents the target link's immediate relational context.
- Encoding: The retrieved subgraph is encoded with GNN to capture structural and relational information. The encoding procedure converts the subgraph into a feature representation suitable for link prediction.
- Meta-Learning Integration: Some subgraph-based models use meta-learning techniques to improve flexibility to few-shot conditions. For example, the Meta-iKG framework combines subgraph-based modeling and meta-learning to rapidly transfer subgraph-specific information and learn transferable patterns using meta gradients. This method enables the model to adapt to new tasks using minimum training data.Transfer learning on graphs is a rapidly developing field that uses knowledge from one graph to improve learning and performance in another. This method is particularly useful in situations where labeled data is scarce, allowing models to transfer insights and patterns acquired

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

from large, well-annotated source graphs to new, often smaller, target graphs. Label propagation, meta-learning, knowledge distillation and subgraph-based models have all showed promise for improving link prediction results. Integrating these techniques allows researchers to leverage the strengths of complex GNN while preserving the speed and flexibility of simpler models, ultimately improving link accuracy for prediction across a wide range of applications, including social network analysis, bioinformatics, and knowledge graph completion.

5 Research directions

Learning based techniques such as GNNs are poised to revolutionise link prediction methodologies. Integrating them into dynamic systems presents novel opportunities for modelling and comprehending networks.

- Learning based techniques exhibit great potential for integration into dynamic systems, surpassing the constraints of static network models. This adjustment enables the creation of more accurate depictions of the progression of networks throughout time.The utilisation of temporal dynamics is a distinguishing feature of GNNs. Unlike conventional approaches, which may disregard the temporal dimension of data, GNNs are capable of capturing and leveraging temporal shifts within networks. This feature allows the models to not only capture and document changes over a period of time but also to actively utilise this knowledge to improve the accuracy of predictions.
- The demand for the creation of link prediction approaches that are both technically solid and specifically designed for the intricate topologies of real-world networks is growing. Learning based techniques are essential for addressing these technological and structural requirements.
- The integration of theoretical advancements and practical implementations in GNNs is expected to expedite problem-solving in the field of network analysis. This collaboration is expected to result in substantial advancements in addressing the challenges that arise from the ongoing expansion and evolution of networks. Resolving real-world network issues is of utmost importance as digital networks expand in both scale and intricacy. The ability of GNNs to adapt and react to these developments will play a vital role. This will lead to more efficient solutions that are in line with the rapid advancements in technology.
- Creating hybrid models that combine similarity-based approaches and latent feature models can provide a more complete picture of the network structure. These models can strike a compromise between computing expense and the depth of insights derived from network data. Recent research have shown the efficacy of this combined technique in enhancing link prediction accuracy by harnessing the characteristics of both methodologies (Sharma et al. 2023).
- One crucial area for advancement is the creation of algorithms with reduced complexity and computational resource requirements. As network datasets become larger and more complex, it is critical to develop algorithms that can manage these computing needs efficiently. This can include experimenting with more efficient optimization methodolo-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

gies, establishing scalable model designs, and creating algorithms capable of effectively managing large-scale networks. By focusing on these factors, we may create more solid and practical solutions for real-world applications.

- Incorporating multimodal data into GNN models can improve the accuracy and robustness of link prediction. This entails merging information from many sources, such as text, photos, and structured data, to provide more detailed and useful network representations. According to research, combining multiple data sources can increase prediction accuracy by exploiting the complementing strengths of distinct modalities (Zhang et al. 2023).
- Exploring transfer learning and domain adaption methodologies might help greatly when extending GNN models trained on one type of network to different but related networks, saving computational resources by lowering the requirement for huge labeled datasets in new domains. Researchers can improve the effectiveness and accuracy of link prediction by applying pre-trained algorithms to new, related tasks. Furthermore, combining transfer learning with advanced approaches like reinforcement learning and generative models may improve link prediction capabilities and broaden the models' application to real-world scenarios (Wang et al. 2023b).

6 Conclusion

This paper presented a thorough categorization and comprehensive examination of link prediction strategies using feature extraction and learning-based methods. Through the analysis of similarity-based methods categorized into local, global, and quasi-local similarities, as well as the examination of probabilistic methods, a detailed analysis has been provided, offering a solid understanding of the principles and complexities involved with each approach. A comprehensive analysis of feature learning techniques emphasizes their proficiency in identifying complex network patterns have been presented. In this survey, we picked datasets from a variety of heterogeneous and homogeneous datasets. The homogeneous datasets are PubMed, CiteSeer, Cora, Ego-Facebook, and Wiki-vote, whereas the heterogeneous datasets are DBLP, MovieLens, IMDb, OGBN-MAG, and OGBL-BIOKG, each chosen for their distinct properties. The study provides a detailed description of these datasets to ensure a thorough understanding of link prediction on these networks. On homogeneous datasets, advanced matrix factorization, path-based algorithms, and techniques like GCN, GraphSAGE, and GAT have been implemented, and a comparative evaluation has been provided based on parameters such as Macro-F1, Micro-F1, and AUC-ROC plots. Similarly, on heterogeneous datasets, techniques such as RGCN, TransE, TransR, DistMult, and ComplEx have been implemented, and results are compared with respect to parameters such as Macro-F1, Micro-F1, and AUC. The findings demonstrate the flexibility and resilience of these approaches in handling complex, interconnected data. The results confirm the significant impact of GNNs in predicting links, especially their ability to effectively learn from the changing structure of networks. This analysis not only presents visual and factual evidence of the current level of advancement but also sets the foundation for creating standards for future research efforts.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

The link prediction approaches presented the in paper utilise temporal dynamics to enhance prediction accuracy and effectively manage the intricacies of real-world network architectures. The combination of theoretical developments and practical applications in learning based techniques is expected to greatly enhance the field of network analysis, allowing for the creation of solutions that can adjust to the increasing complexities and expansion of digital networks.These findings not only challenge present approaches, but also pave the door for more adaptable and intelligent network analysis tools. Future research should focus on improving these strategies to handle larger and more diverse datasets while maintaining scalability and robustness. Furthermore, multidisciplinary techniques that include insights from other disciplines may offer novel views and answers, propelling the field forward. Emphasizing real-time flexibility and integration with developing technologies will be critical to ensuring that these methods remain relevant and applicable. These developments are expected to have a substantial impact on a variety of applications, including bio-informatics, social network analysis, and recommendation systems.

Acknowledgements This research is supported by University Institute of Engineering and Technology, Panjab University, Chandigarh, India,160014.

Author contributions Puneet Kapoor: Conceptualization, Methodology, Software, Data curation, Writing-original draft, Visualization, Investigation, Validation, Writing-review & editing. Sakshi Kaushal: Conceptualization, Methodology, Formal analysis, Supervision, Writing-review & editing. Harish Kumar: Conceptualization, Methodology, Formal analysis, Supervision, review & editing. Kushal Kanwar: Conceptualization, Methodology, Software, Supervision, Investigation, review & editing.

Data availibility The data utilized in this study is available from the PyTorch Geometric Datasets repository, accessible at https://pytorch-geometric.readthedocs.io/en/latest/index.html.

Code availability Code will be made available upon request.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

Al Hasan M, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. Sdm06: workshop on link analysis, counter-terrorism and security (Vol. 30, pp. 798–805)

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Balvir SU, Raghuwanshi MM, Singh KR (2023) A comprehensive survey on learning based methods for link prediction problem. 2023 6th International conference on information systems and computer networks, ISCON 2023, pp. 1–7, https://doi.org/10.1109/ISCON57294.2023.10112010

Barros CD, Mendonça MR, Vieira AB, Ziviani A (2021) A survey on embedding dynamic graphs. ACM Comput Sur. https://doi.org/10.1145/3483595

Bastos A, Singh K, Nadgeri A, Hoffart J, Singh M, Suzumura T (2023) Can persistent homology provide an efficient alternative for evaluation of knowledge graph completion methods? Acm web conference 2023—proceedings of the world wide web conference, www 2023 (pp. 2455–2466). New York, NY, USA: ACM. Accessed from https://doi.org/10.1145/3543507.3583308

Berahmand K, Mohammadi M, Sheikhpour R, Li Y, Xu Y (2023) WSNMF: weighted symmetric nonnegative matrix factorization for attributed graph clustering. Neurocomputing 566:127041. https://doi.org/10.1016/j.neucom.2023.127041

Bing R, Yuan G, Zhu M, Meng F, Ma H, Qiao S (2023) Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. Artif Intell Rev 56(8):8003–8042. https://doi.org/10.1007/s10462-022-10375-2

Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Adv Neural Inform Proc Syst 26:1–9

Bose AJ, Jain A, Molino P, Hamilton WL (2020) Meta-graph: few shot link prediction via meta learning. Accessed from https://arxiv.org/abs/1912.09867

Bronstein MM, Bruna J, Cohen T, Veličković P (2021) Geometric deep learning: grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478, Accessed from http://arxiv.org/abs/2104.13478

Cai H, Zheng VW, Chang KCC (2018) A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans Knowl Data Eng 30(9):1616–1637. https://doi.org/10.1109/TKDE.2018.2807452

Calders T, Goethals B (2005) Depth-first non-derivable itemset mining. Proceedings of the 2005 siam international conference on data mining (pp. 250–261). Philadelphia, PA: Society for Industrial and Applied Mathematics. Accessed from https://doi.org/10.1137/1.9781611972757.23

Cao W, Yan Z, He Z, He Z (2020) A comprehensive survey on geometric deep learning. IEEE Access 8:35929–35949. https://doi.org/10.1109/ACCESS.2020.2975067

Chen M, Zhang W, Zhang W, Chen Q, Chen H (2019) Meta relational learning for few-shot link prediction in knowledge graphs. Accessed from https://arxiv.org/abs/1909.01515

Chen Z, Wang Y, Zhao B, Cheng J, Zhao X, Duan Z (2020) Knowledge graph completion: a review. IEEE Access 8:192435–192456. https://doi.org/10.1109/ACCESS.2020.3030076

Chen G, Wang H, Fang Y, Jiang L (2022a) Link prediction by deep non-negative matrix factorization. Expert Syst Appl 188:115991

Chen WS, Zeng Q, Pan B (2022b) A survey of deep nonnegative matrix factorization. Neurocomputing 491:305–320. https://doi.org/10.1016/j.neucom.2021.08.152

Chen WS, Xie K, Liu R, Pan B (2023) Symmetric nonnegative matrix factorization: a systematic review. Neurocomputing 557:126721. https://doi.org/10.1016/j.neucom.2023.126721

Chung F, Zhao W (2010) PageRank and random walks on graphs. Fete of combinatorics and computer science (Vol. 20, pp. 43–62), Accessed from http://link.springer.com/10.1007/978-3-642-13580-4_3

Daud NN, Ab Hamid SH, Saadoon M, Sahran F, Anuar NB (2020) Applications of link prediction in social networks: a review. J Netw Comput Appl. https://doi.org/10.1016/j.jnca.2020.102716

De Handschutter P, Gillis N, Siebert X (2021) A survey on deep matrix factorizations. Comput Sci Rev 42:100423. https://doi.org/10.1016/j.cosrev.2021.100423

Derrow-Pinion A, She J, Wong D, Lange O, Hester T, Perez L, Velickovic P (2021) ETA prediction with graph neural networks in google maps. International conference on information and knowledge management, Proceedings, pp. 3767–3776, https://doi.org/10.1145/3459637.3481916arXiv:2108.11482

Dettmers T, Minervini P, Stenetorp P, Riedel S (2018) Convolutional 2D knowledge graph embeddings. 32nd AAAI conference on artificial intelligence, AAAI 2018, pp. 1811–1818, https://doi.org/10.1609/aaai.v32i1.11573 , Accessed from http://arxiv.org/abs/1707.01476arXiv:1707.01476

Ding Y, Lai Z, Mok P, Chua T-S (2024) Computational technologies for fashion recommendation: a survey. ACM Comput Surv 56(5):1–45. https://doi.org/10.1145/3627100

Dong G, Fan J, Shekhtman LM, Shai S, Du R, Tian L, Havlin S (2018) Resilience of networks with community structure behaves as if under an external field. Proc Natl Acad Sci USA 115(27):6911–6915. https://doi.org/10.1073/pnas.1801588115

Dong G, Wang F, Shekhtman LM, Danziger MM, Fan J, Du R, Havlin S (2021) Optimal resilience of modular interacting networks. Proc Natl Acad Sci USA 118(22):e1922831118. https://doi.org/10.1073/pnas.1922831118

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation. The web conference 2019—proceedings of the world wide web conference, www 2019 (pp. 417–426). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3308558.3313488

Fang Y, Lin W, Zheng VW, Wu M, Shi J, Chang KCC, Li XL (2021) Metagraph-based learning on heterogeneous graphs. IEEE Trans Knowl Data Eng 33(1):154–168. https://doi.org/10.1109/TKDE.2019.2922956

Fey M, Lenssen JE (2019) Fast Graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428, Accessed from http://arxiv.org/abs/1903.02428

Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks, Accessed from https://arxiv.org/abs/1703.03400

Frank O, Strauss D (1986) Markov graphs. J Am Stat Assoc 81(395):832–842. https://doi.org/10.1080/01621459.1986.10478342

Fu X, Zhang J, Meng Z, King I (2020a) MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. The web conference 2020—proceedings of the world wide web conference, WWW 2020, pp. 2331–2341, https://doi.org/10.1145/3366423.3380297arXiv:2002.01680

Fu, X., Zhang J, Meng Z, King I (2020b) MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. Proceedings of the web conference 2020 (pp. 2331–2341). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3366423.3380297

Getoor L, Friedman N, Koller D, Taskar B (2003) Learning probabilistic models of link structure. J Mach Learn Res 3(4–5):679–707

Goyal P, Ferrara E (2018) Graph embedding techniques, applications, and performance: a survey. Knowl-Based Syst 151:78–94. https://doi.org/10.1016/j.knosys.2018.03.022

Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. Proceedings of the ACM SIGKDD International conference on knowledge discovery and data mining, 13-17-Aug, pp. 855–864, https://doi.org/10.1145/2939672.2939754arXiv:1607.00653

Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. Proc Natl Acad Sci USA 106(52):22073–22078. https://doi.org/10.1073/pnas.0908366106

Guo Z, Zhang S (2020) Sparse deep nonnegative matrix factorization. Big Data Min Anal 3(1):13–28. https://doi.org/10.26599/BDMA.2019.9020020

Guo Z, Shiao W, Zhang S, Liu Y, Chawla NV, Shah N, Zhao T (2023) Linkless link prediction via relational distillation. Proceedings of the 40th international conference on machine learning. JMLR.org

Guu K, Miller J, Liang P (2015) Traversing knowledge graphs in vector space. arXiv preprint arXiv:1506.01094, Accessed from http://arxiv.org/abs/1506.01094

Haghani S, Keyvanpour MR (2019) A systemic analysis of link prediction in social network. Artif Intell Rev 52(3):1961–1995. https://doi.org/10.1007/s10462-017-9590-2

Hamilton WL, Ying R, Leskovec J (2017a). Inductive representation learning on large graphs. Proceedings of the 31st international conference on neural information processing systems (pp. 1025–1035). Red Hook, NY, USA: Curran Associates Inc

Hamilton WL, Ying R, Leskovec J (2017b) Inductive representation learning on large graphs. I. Guyon et al. (Eds.), Advances in neural information processing systems (Vol. 2017-Dec, pp. 1025–1035). Curran Associates, Inc., Accessed from https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

Hamilton WL, Ying R, Leskovec J (2017c) Representation learning on graphs: methods and applications. arXiv preprint arXiv:1709.05584, pp. 1–24, https://doi.org/10.48550/arXiv.1709.05584, Accessed from http://arxiv.org/abs/1709.05584

Han L, Chen L, Shi X (2022) Recommendation model based on probabilistic matrix factorization, integrating user trust relationship, interest mining, and item correlation. IEEE Access 10:132315–132331. https://doi.org/10.1109/ACCESS.2022.3230351

Hinton G, Dean J, Vinyals O (2014) Distilling the knowledge in a neural network. (p.1-9)

Holland PW, Leinhardt S (1981) An exponential family of probability distributions for directed graphs. J Am Stat Assoc 76(373):33–50. https://doi.org/10.1080/01621459.1981.10477598

Hong H, Guo H, Lin Y, Yang X, Li Z, Ye J (2020) An attention-based graph neural network for heterogeneous structural learning. AAAI 2020—34th AAAI conference on artificial intelligence, 34(04):4132–4139, https://ojs.aaai.org/index.php/AAAI/article/view/5833, Accessed from arXiv:1912.10832

Hu L, Xu S, Li C, Yang C, Shi C, Duan N, Zhou M (2020a) Graph neural news recommendation with unsupervised preference disentanglement. Proceedings of the annual meeting of the association for computational linguistics (pp. 4255–4264). Stroudsburg, PA, USA: Association for Computational Linguistics, Accessed from https://www.aclweb.org/anthology/2020.acl-main.392

Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Leskovec J (2020b) Open graph benchmark: datasets for machine learning on graphs. Advances in neural information processing systems, 2020-Dec(NeurIPS), pp. 1–34, , Accessed from http://arxiv.org/abs/2005.00687

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Hu Z, Dong Y, Wang K, Sun Y (2020c) Heterogeneous graph transformer. The web conference 2020—proceedings of the world wide web conference, WWW 2020, pp. 2704–2710, https://doi.org/10.1145/3366423.3380027arXiv:2003.01332

Huang Z, Kosan M, Silva A, Singh A (2023) Link prediction without graph neural networks, Accessed from https://arxiv.org/abs/2305.13656

Jaccard P (1901) Etude de la distribution florale dans une portion des alpes et du jura. Bulletin de la Societe Vaudoise des Sciences Naturelles 37:547–579. https://doi.org/10.5169/seals-266450

Jain S, Chouzenoux E, Kumar K, Majumdar A (2023) Graph regularized probabilistic matrix factorization for drug-drug interactions prediction. IEEE J Biomed Health Inform 27(5):2565–2574. https://doi.org/10.1109/JBHI.2023.3246225

Jin W, Ma Y, Liu X, Tang X, Wang S, Tang J (2020) Graph structure learning for robust graph neural networks. Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp. 66–74, https://doi.org/10.1145/3394486.3403049arXiv:2005.10203

Kaibiao L, Chen J, Ruicong C, Fan Y, Yang Z, Min L, Ping L (2024) Adaptive neighbor graph aggregated graph attention network for heterogeneous graph embedding. ACM Trans Knowl Discov Data 18(1):1–21. https://doi.org/10.1145/3616377

Kapoor P, Kaushal S, Kumar H (2022) A review on architecture and communication protocols for electric vehicle charging system. ACM international conference proceeding series (pp. 1–6). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3590837.3590920

Katz L (1953) A new status I N D E X D E R I V E D from sociometric. Psychmetrika 18(1):39–43

Keyvanpour MR, Moradi SS (2014) A perturbation method based on singular value decomposition and feature selection for privacy preserving data mining. Int J Data Warehous Min 10(1):55–76. https://doi.org/10.4018/ijdwm.2014010104

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. 5th International conference on learning representations, ICLR 2017—conference track proceedings, pp. 1–14, Accessed from http://arxiv.org/abs/1609.02907arXiv:1609.02907

Kossinets G, Watts DJ (2009) Origins of homophily in an evolving social network. Am J Sociol 115(2):405–450. https://doi.org/10.1086/599247

Kumar A, Singh SS, Singh K, Biswas B (2020) Link prediction techniques, applications, and performance: a survey. Phys A: Stat Mech Its Appl 553:124289. https://doi.org/10.1016/j.physa.2020.124289

Kumari A, Behera RK, Sahoo KS, Nayyar A, Kumar Luhach A, Prakash Sahoo S (2022) Supervised link prediction using structured-based feature extraction in social network. Concurr Comput: Prac Exp. https://doi.org/10.1002/cpe.5839

Lee J, Kim H, Lee J, Yoon S (2017) Transfer learning for deep learning on graph-structured data. Proc AAAI Conf Artif Intell. https://doi.org/10.1609/aaai.v31i1.10904

Leicht EA, Holme P, Newman MEJ (2006) Vertex similarity in networks. Phys Rev E 73:026120. https://doi.org/10.1103/PhysRevE.73.026120

Leskovec J (2017) Large-scale graph representation learning. 2017 IEEE international conference on big data (big data) (pp. 4–4). IEEE, Accessed from https://ieeexplore.ieee.org/document/8257903/

Leskovec J, Krevl A (2014) {SNAP Datasets}: {Stanford} large network dataset collection. http://snap.stanford.edu/data

Li X, Du N, Li H, Li K, Gao J, Zhang A (2014) A deep learning approach to link prediction in dynamic networks. Proceedings of the 2014 siam international conference on data mining (pp. 289–297). Philadelphia, PA: Society for Industrial and Applied Mathematics, Accessed from https://doi.org/10.1137/1.9781611973440.33

Li J, Shomer H, Mao H, Zeng S, Ma Y, Shah N, Yin D (2024) Evaluating graph neural networks for link prediction: current pitfalls and new benchmarking. Proceedings of the 37th international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc

Liben-Nowell D, Kleinberg J (2003) The link prediction problem for social networks. Proceedings of the twelfth international conference on information and knowledge management (pp. 556–559)

Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015) Learning entity and relation embeddings for knowledge graph completion. Proc AAAI Conf Artif Intell 29(1):345–354. https://doi.org/10.1609/aaai.v29i1.9491

Liu W, Lü L (2010) Link prediction based on local random walk. EPL (Europhys Lett) 89(5):58007. https://doi.org/10.1209/0295-5075/89/58007

Liu J, Duan L (2021) A survey on knowledge graph-based recommender systems. IEEE advanced information technology, electronic and automation control conference (IAEAC) (pp. 2450–2453). IEEE, Accessed from https://ieeexplore.ieee.org/document/9390863/

Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J (2023) Self-supervised learning: generative or contrastive. IEEE Trans Knowl Data Eng 35(1):857–876. https://doi.org/10.1109/TKDE.2021.3090866

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Liu S, Liu S, Yang Z, Sun J, Shen X, Li Q, Du S (2024) Heterogeneous evolution network embedding with temporal extension for intelligent tutoring systems. ACM Trans Inform Syst 42(2):1–28. https://doi.org/10.1145/3617828

Lou X, Liu G, Li J (2024) Heterogeneous graph neural network with graph-data augmentation and adaptive denoising. Appl Intell 54(5):4411–4424. https://doi.org/10.1007/s10489-024-05363-8

Lü L, Zhou T (2010) Link prediction in weighted networks: the role of weak ties. Europhysics Letters. https://doi.org/10.1209/0295-5075/89/18001

Lü L, Jin CH, Zhou T (2009) Similarity index based on local paths for link prediction of complex networks. Phys Rev E—Stat Nonlinear Soft Matter Phys 80(4):1–9. https://doi.org/10.1103/PhysRevE.80.046122

Lü L, Pan L, Zhou T, Zhang Y-C, Stanley H (2015) Toward link predictability of complex networks. Proc Natl Acad Sci 112:201424644. https://doi.org/10.1073/pnas.1424644112

Lyu B, Xie K, Sun W (2017) A deep orthogonal non-negative matrix factorization method for learning attribute representations. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) (Vol. 10639 LNCS, pp. 443–452), Accessed from https://www.jstage.jst.go.jp/article/jsmemag/90/823/90_KJ00001464638/_article/-char/ja/ http://link.springer.com/10.1007/978-3-319-70136-3_47

Matsunaga D, Suzumura T, Takahashi T (2019) Exploring graph neural networks for stock market predictions with rolling window analysis. arXiv preprint arXiv:1909.10660, Accessed from http://arxiv.org/abs/1909.10660arXiv:1909.10660

Nasiri E, Berahmand K, Li Y (2023) Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks. Multimed Tools Appl 82(3):3745–3768. https://doi.org/10.1007/s11042-022-12943-8

Nayyeri M, Cil GM, Vahdati S, Osborne F, Rahman M, Angioni S, Lehmann J (2021) Trans4E: link prediction on scholarly knowledge graphs. Neurocomputing 461:530–542

Neelakantan A, Roth B, McCallum A (2015) Compositional vector space models for knowledge base inference. AAAI spring symposium—technical report, SS-15-03, pp. 31–34, Accessed from http://arxiv.org/abs/1504.06662arXiv:1504.06662

Newman M (2001) Newman mej: clustering and preferential attachment in growing networks. Phys Rev E 64:025102. https://doi.org/10.1103/PhysRevE.64.025102

Nguyen DQ, Sirts K, Qu L, Johnson M (2016) STransE: a novel embedding model of entities and relationships in knowledge bases. 2016 Conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL HLT 2016—proceedings of the conference, pp. 460–466, https://doi.org/10.18653/v1/n16-1054arXiv:1606.08140

Nguyen TK, Liu Z, Fang Y (2023) Link prediction on latent heterogeneous graphs (Vol. 1) (No. 1). Association for Computing Machinery

Nie F, Zhu W, Li X (2017) Unsupervised large graph embedding. 31st AAAI Conference on Artificial Intelligence, AAAI 2017, 31(1):2422–2428, https://doi.org/10.1609/aaai.v31i1.10814 , Accessed from https://ojs.aaai.org/index.php/AAAI/article/view/10814

Ou Q, Jin Y-D, Zhou T, Wang B, Yin B-Q (2007) Power-law strength-degree correlation from a resource-allocation dynamics on weighted networks. Phys Rev E Stat Nonlinear Soft Matter Phys 75:210102. https://doi.org/10.1103/PhysRevE.75.021102

Ou M, Cui P, Pei J, Zhang Z, Zhu W (2016) Asymmetric transitivity preserving graph embedding. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 1105–1114). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/2939672.2939751

Page L, Brin S (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30(1–7):107–117. https://doi.org/10.1016/S0169-7552(98)00110-X

Pan L, Zhou T, Lü L, Hu CK (2016) Predicting missing links and identifying spurious links via likelihood analysis. Sci Rep 6:1–10. https://doi.org/10.1038/srep22955

Pattison P, Wasserman S (1999) Logit models and logistic regressions for social networks: II. Multivariate relations. Br J Math Stat Psychol 52(2):169–193. https://doi.org/10.1348/000711099159053

Peng S, Sugiyama K, Mine T (2022) SVD-GCN: a simplified graph convolution paradigm for recommendation. international conference on information and knowledge management, Proceedings pp. 1625–1634. https://doi.org/10.1145/3511808.3557462

Perozzi B, Al-Rfou R, Skiena S (2014) DeepWalk. Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining (pp. 701–710). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/2623330.2623732

Perozzi B, Kulkarni V, Chen H, Skiena S (2017) Don't walk, skip! Proceedings of the 2017 ieee/acm international conference on advances in social networks analysis and mining 2017 (pp. 258–265). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3110025.3110086

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Qiu J, Dong Y, Ma H, Li J, Wang K, Tang J (2018) Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec. Wsdm 2018—proceedings of the 11th acm international conference on web search and data mining (Vol. 2018-Feb, pp. 459–467). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3159652.3159706

Regan E, Somera A, Mongru D, Oltvai Z, Barabasi A-L (2002) Mongru da, oltvai zn, barabasi al. hierarchical organisation of modularity in metabolic networks. Science (New York, N.Y.) 297:1551–1555. https://doi.org/10.1126/science.1073374

Riesen K, Neuhaus M, Bunke H (2007) Graph embedding in vector spaces by means of prototype selection. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) (Vol. 4538 LNCS, pp. 383–393). Berlin, Heidelberg: Springer, Accessed from http://link.springer.com/10.1007/978-3-540-72903-7_35

Sadek RA (2012) SVD based image processing applications: state of the art, contributions and research challenges. Int J Adv Comput Sci Appl 3(7):26–34. https://doi.org/10.14569/IJACSA.2012.030703

Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. Advances in neural information processing systems 20—proceedings of the 2007 conference, pp. 1–8,

Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2018) Modeling relational data with graph convolutional networks (Vol. 10843 LNCS) (No. 1). Springer, Accessed from https://doi.org/10.1007/978-3-319-93417-4_38

Sen P, Namata GM, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T (2008) Collective classification in network data. AI Mag 29(3):93–106. https://doi.org/10.1609/aimag.v29i3.2157

Shang K, Small M (2022) Link prediction for long-circle-like networks. Phys Rev E 105:24311. https://doi.org/10.1103/PhysRevE.105.024311

Shang K, Small M, Yan W (2017a) Fitness networks for real world systems via modified preferential attachment. Phys A: Stat Mech Its Appl 474:49–60. https://doi.org/10.1016/j.physa.2017.01.066

Shang K, Small M, Yan W (2017b) Link direction for link prediction. Phys A: Stat Mech Its Appl 469:767–776. https://doi.org/10.1016/j.physa.2016.11.129

Shang K, Small M, Xu X-K, Yan W-s (2017c) The role of direct links for link prediction in evolving networks. EPL (Europhys Lett) 117:28002. https://doi.org/10.1209/0295-5075/117/28002

Shang K, Small M, Yin D, Li T-C, Yan W (2019) The key to the weak-ties phenomenon. EPL (Europhys Lett) 127:48002. https://doi.org/10.1209/0295-5075/127/48002

Shang K-k, Li T-c, Small M, Burton D, Wang Y (2019b) Link prediction for tree-like networks. Chaos: Interdiscipl J Nonlinear Sci 29(6):061103

Sharma A, Yadav AK, Rai AK (2023) A novel and precise approach for similarity-based link prediction in diverse networks. Soc Netw Anal Min 14(1):11. https://doi.org/10.1007/s13278-023-01160-2

Shi C, Li Y, Zhang J, Sun Y, Yu PS (2017a) A survey of heterogeneous information network analysis. IEEE Tran Knowl Data Eng 29(1):17–37. https://doi.org/10.1109/TKDE.2016.2598561

Shi J, Gao H, Qi G, Zhou Z (2017b) Knowledge graph embedding with triple context. International conference on information and knowledge management, proceedings (Vol. Part F1318, pp. 2299–2302). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3132847.3133119

Sun Y, Han J, Yan X, Yu PS, Wu T (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks. Proc VLDB Endow 4(11):992–1003. https://doi.org/10.14778/3402707.3402736

Tong H, Faloutsos C, Pan JY (2006) Fast random walk with restart and its applications. Proceedings—IEEE international conference on data mining, ICDM, pp. 613–622. https://doi.org/10.1109/ICDM.2006.70

Trouillon T, Welbl J, Riedel S, Ciaussier E, Bouchard G (2016) Complex embeddings for simple link prediction. 33rd International conference on machine learning, ICML 2016, vol 5, pp. 3021–3032. arXiv:1606.06357

Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017) Graph attention networks. 6th International conference on learning representations, ICLR 2018—conference track proceedings, pp. 39–41, https://doi.org/10.1007/978-3-031-01587-8_7, Accessed from arXiv:1710.10903

Waikhom L, Patgiri R (2023) A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. Artif Intell Rev. https://doi.org/10.1007/s10462-022-10321-2

Wang C, Satuluri V, Parthasarathy S (2007) Local probabilistic models for link prediction. Proceedings—IEEE international conference on data mining, ICDM pp. 322–331. https://doi.org/10.1109/ICDM.2007.108

Wang X, Zhang X, Zhao C, Xie Z, Zhang S, Yi D (2015) Predicting link directions using local directed path. Phys A: Stat Mech Its Appl 419:260–267. https://doi.org/10.1016/J.PHYSA.2014.10.007

Wang X, Ji H, Cui P, Yu P, Shi C, Wang B, Ye Y (2019) Heterogeneous graph attention network. The web conference 2019—proceedings of the world wide web conference, WWW 2019 (pp. 2022–2032). New York, NY, USA: ACM, Accessed from https://doi.org/10.1145/3308558.3313562

Wang C, Gu Z, Wei JM (2024) Spectral clustering and embedding with inter-class topology-preserving. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2023.111278

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Wang R, Shi C, Zhao T, Wang X, Ye Y (2023a) Heterogeneous information network embedding with adversarial disentangler. IEEE Trans Knowl Data Eng 35(2):1581–1593. https://doi.org/10.1109/TKDE.2021.3096231

Wang X, Bo D, Shi C, Fan S, Ye Y, Yu PS (2023b) A survey on heterogeneous graph embedding: methods, techniques, applications and sources. IEEE Trans Big Data 9(2):415–436. https://doi.org/10.1109/TBDATA.2022.3177455

Wu H, Song C, Ge Y, Ge T (2022) Link prediction on complex networks: an experimental survey. Data Sci Eng 7(3):253–278. https://doi.org/10.1007/s41019-022-00188-2

Yadati N, Nitin V, Nimishakavi M, Yadav P, Louis A, Talukdar P (2020) NHP: neural hypergraph link prediction. International conference on information and knowledge management, Proceedings pp. 1705–1714. https://doi.org/10.1145/3340531.3411870

Yang H, Liu J (2021) Knowledge graph representation learning as groupoid. Proceedings of the 30th ACM international conference on information & knowledge management (pp. 2311–2320). New York, NY, USA: ACM. Accessed from https://doi.org/10.1145/3459637.3482442

Yang B, tau Yih W, He X, Gao J, Deng L (2015) Embedding entities and relations for learning and inference in knowledge bases. 3rd International conference on learning representations, ICLR 2015—Conference Track Proceedings, pp. 1–12, arXiv:1412.6575

Yang Z, Cohen WW, Salakhutdinov R (2016) Revisiting semi-supervised learning with graph embeddings. 33rd Int Conf Mach Learn 1:86–94

Yang C, Xiao Y, Zhang Y, Sun Y, Han J (2020) Heterogeneous network representation learning: a unified framework with survey and benchmark. IEEE Trans Knowl Data Eng 34(10):4854–4873. https://doi.org/10.1109/TKDE.2020.3045924

Yang X, Yan M, Pan S, Ye X, Fan D (2023) Simple and efficient heterogeneous graph neural network. Proc AAAI Conf Artif Intell 37(9):10816–10824. https://doi.org/10.1609/aaai.v37i9.26283

Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J (2018) Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 974–983). New York, NY, USA: ACM. , Accessed from https://doi.org/10.1145/3219819.3219890

You J, Ying R, Leskovec J (2020) Design space for graph neural networks. Adv Neural Inform Proc Syst 33:17009

Zeb A, Saif S, Chen J, Haq AU, Gong Z, Zhang D (2022) Complex graph convolutional network for link prediction in knowledge graphs. Expert Syst Appl 200:116796. https://doi.org/10.1016/j.eswa.2022.116796

Zeng Z, Chen K-J, Zhang S, Zhang H (2013) A link prediction approach using semi-supervised learning in dynamic networks. 2013 Sixth international conference on advanced computational intelligence (ICACI) (pp. 276–280). IEEE, Accessed from http://ieeexplore.ieee.org/document/6748516/

Zhang C, Song D, Huang C, Swami A, Chawla NV (2019a) Heterogeneous graph neural network. Proc ACM SIGKDD Int Conf Knowl Discov Data Mining. https://doi.org/10.1145/3292500.3330961

Zhang S, Tong H, Xu J, Maciejewski R (2019b) Graph convolutional networks: a comprehensive review. Comput Soc Netw 6(1):11. https://doi.org/10.1186/s40649-019-0069-y

Zhang C, Shang K, Qiao J (2021) Adaptive similarity function with structural features of network embedding for missing link prediction. Complexity 2021:1–15. https://doi.org/10.1155/2021/1277579

Zhang L, Li W, Guan H, He Z, Cheng M, Wang H (2023) MCPI: integrating multimodal data for enhanced prediction of compound protein interactions. https://doi.org/10.48550/ARXIV.2306.08907

Zhao J, Wang X, Shi C, Hu B, Song G, Ye Y (2021) Heterogeneous graph structure learning for graph neural networks. 35th AAAI Conf Artif Intell, AAAI 2021 5B(5):4697–4705. https://doi.org/10.1609/aaai.v35i5.16600

Zhao Z, Gou Z, Du Y, Ma J, Li T, Zhang R (2022) A novel link prediction algorithm based on inductive matrix completion. Expert Syst Appl 188:116033. https://doi.org/10.1016/j.eswa.2021.116033

Zhao Y, Sun Y, Huang Y, Li L, Dong H (2023) Link prediction in heterogeneous networks based on metapath projection and aggregation. Expert Syst Appl 227:120325. https://doi.org/10.1016/j.eswa.2023.120325

Zheng S, Mai S, Sun Y, Hu H, Yang Y (2021) Subgraph-aware few-shot inductive link prediction via meta-learning. Accessed from https://arxiv.org/abs/2108.00954

Zheng W, Huang EW, Rao N, Wang Z, Subbian K (2023) You only transfer what you share: intersection-induced graph transfer learning for link prediction. Accessed from https://arxiv.org/abs/2302.14189

Zhou T, Ren J, Medo M, Zhang Y-C (2007) Bipartite network projection and personal recommendation. Phys Rev E—Stat Nonlinear Soft Matter Phys 76:046115. https://doi.org/10.1103/PhysRevE.76.046115

Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhou J, Liu L, Wei W, Fan J (2023a) Network representation, learning from preprocessing feature extraction to node embedding. ACM Comput Surv. https://doi.org/10.1145/3491206

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Zhou M, Han Q, Li M, Li K, Qian Z (2023b) Nearest neighbor walk network embedding for link prediction in complex networks. Phys A: Stat Mech Appl 620:128757. https://doi.org/10.1016/j.physa.2023.128757
Zhou MY, Wang F, Chen Z, Wu J, Liu G, Liao H (2024) Weak link prediction based on hyper latent distance in complex network. Expert Syst Appl 238:121843. https://doi.org/10.1016/j.eswa.2023.121843

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Puneet Kapoor[1] · Sakshi Kaushal[1] · Harish Kumar[1] · Kushal Kanwar[2]

✉ Puneet Kapoor
puneetira@pu.ac.in

Sakshi Kaushal
sakshi@pu.ac.in

Harish Kumar
harishk@pu.ac.in

Kushal Kanwar
kushalneo@gmail.com

[1] University Institute of Engineering and Technology, Panjab University, Chandigarh 160014, India

[2] Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh 173215, India

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com