Machine Learning for Link Prediction between Nodes in Complex Networks

Elaf Adel Abbas [1], Nisreen Abbas Hussein[2], Raaid Alubady[3, 4,*]

[1]College of Computer Science and Information Technology, University of Karbala, Karbala, Iraq

[2]Babylon Education Directorate, Ministry of Education, Babil, Iraq

[3]Al-Ayen Iraqi University, Thi-Qar, Iraq

[4]College of Information Technology, University of Babylon, Babil, Iraq

Emails: elaf1982adil@gmail.com; nasreenabbas2013@gmail.com; alubadyraaid@alayen.edu.iq

Abstract

Recently, the complex network has become popular use as it can transfer huge amounts of multimedia, text, ideas, and other information, encouraging many participant connections. Social media is one of these networks that make the most connections. Predicting the formation or dissolution of links between nodes presents a problem for social network analysis researchers. Since social networks are dynamic, this task is exciting as it may also forecast lost network links with less information. On the other way, current link prediction methods use simply node similarity to find links. This study proposes a new technique that relies on node attributes and similarity measures. Nodes are labeled by their centrality and similarity. The network's edges are negative and positive samples. A well-defined dataset for link prediction comprises the features of the nodes at the edges labeled either positive or negative. The dataset is passed to multiple machine learning classifiers. On several real-world networks. The experiments conducted during the research show that Gradient Boosting gave the highest accuracy of 99% compared with other methods.

Keywords: Complex networks; Social networks; Link prediction; Machine learning Techniques

1. Introduction

Internet technology has matured significantly in recent decades, and the digital world has influenced many aspects of human activity and lifestyle. Different methods have been developed for information dissemination, news dissemination, trading, and communication. Social networks are now operational tools for disseminating information due to their vast user bases and quick information flow among them [1]. By joining these networks, users can communicate, share data, or even exchange opinions. Consequently, several types of social networking platforms have gained popularity in society. However, many of these networks, including biological networks, transportation networks, and online social networks, can be considered to some extent as a model of complex networks [2]. These networks contain different nodes or entities and complex connections between them. Many researchers from different communities have been interested in analysing complex networks [3] [4].

The complex network has evolved into a popular medium for exchanging multimedia, texts, ideas, and other types of information, which encourages the development of numerous connections between the parties. The majority of connections are made through social media platforms, particularly when people are drawn to one another because of a shared interest. One example is a pleasant suggestion that appeared in real-time when someone shared a friend on a social media website like Facebook. Similar product ads on an e-commerce website may appear when you are interested in them [5]. One of the difficulties faced by social network analysis researchers is link prediction, which relates to whether new relationships between nodes are expected to emerge or vanish in the near future.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

This challenge is particularly intriguing since social networks are dynamic. Additionally, missing links in the network can be predicted with insufficient information [6].

The prediction of social network links has many widespread applications. It can be used first with recommender systems to help people find friends [7] [8]. Second, it can be used by tourists and travelers to locate or suggest premium hotels [9]. Third, contact between co-authors in any field of study and expert results is essential [10]. In both biology and biomathematics, the link prediction method is unquestionably proper [11]. Therefore, in this study, a new strategy has been proposed for predicting links based on the structural importance of the network, taking into account the importance of nodes in their immediate surroundings to calculate their similarity score.

Hence, the major contributions of this study are listed as follows:

1. To design a technique for solving the problem of link prediction in computationally complex social networks using machine-learning algorithms to predict future links.
2. To develop machine-learning models that incorporate structural elements and node features to predict links, avoiding inferences based on specific similarity measures.
3. To validate and evaluate the proposed technique over complex networks.

The manuscript is organized as follows: Section 2 covers the related works. The Preliminary research is explained in Section 3. Section 4 investigates the most important machine learning techniques used. The proposed method is described in Section 5. Section 6 evaluates the proposed method. Section 7 begins with the experimental results and analysis, followed by conclusions in Section 8.

2. Related Work

According to the network nodes' features, the link prediction problem can primarily anticipate the presence of new linkages between them. Many link prediction strategies have been presented in the literature over the years, and these techniques can be broadly categorized into two main groups: feature learning techniques and feature extraction techniques [12].

Zhou *et al.,* [13] proposes a comprehensive analysis of the prediction problem on two important subclasses of similarity measures: local measures, which consider only local data about the target link, and global measures, which consider data from both sources. Shows that it is very difficult to compute optimal attacks on any of these measures. In addition, many iterations of the overall problem are very difficult for both local and global measures; however, several manageable and well-motivated special cases are presented. Furthermore, efficient and temporary heuristic solutions to intractable cases are presented, as well as worst-case approximation guarantees in certain scenarios.

To develop the latest methods, which are based on deep learning techniques, Hajani and Kiyvanpour reviewed the oldest registration-based methods for the link prediction problem [14]. In addition to classifying ink prediction methods according to their technical approach and discussing the advantages and disadvantages of different methods, the review took into account the dynamic behavior of a social network, which is primarily determined by two important features: node information and link information about the relationship between two nodes.

Wu *et al.,* [15] suggested performing a study to develop a method that uses the influential node identification approach to show the influence at the macro level and combine it with a similarity framework based on co-neighbors at the micro level. This study developed an approach that uses the influencing node identification method. Identifying influential nodes, often known as INI, is another important issue in complex network research. By giving each node in a network a ranking score, it attempts to determine how significant each node is concerning the other nodes in the network. It would appear that the ranking function operates on a global scale and that it can be readily interpreted as the similarity score for in-link prediction metrics. Hence, Influential Nodes Identification Link Prediction (INILP) is a general approach to modeling global and local information for link prediction. This method includes influential node identification, which is another common technique.

Using supervised machine learning techniques [5], the research aimed to predict the potential for future connection establishment. The main contribution is the features generated from the topological structure of the network, which predicted missing links by considering the structure-based properties of social networks. Several potential approaches to predicting links based on structure-based similarity measures have been found in the literature. Failure to accurately predict links in an empirical analysis of any single similarity measure has been observed. However, links are predicted more accurately when considered as features in a machine learning system.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

In [16], Abdel Samad postulated that soon highly similar nodes with the same centrality status would be able to establish connections with one another. It has been noted that the vast majority of link prediction techniques make use of the topological information of nodes contained inside the social graph. This information is used to locate similar pairs of nodes to forecast the link. The estimated degree of similarity is used to generate a score, which is then assigned to the respective pair of nodes (X, Y). In this regard, a high similarity score generates a greater number of opportunities for **X** to be linked to **Y** in the near or far future. In addition, a low similarity score indicates that there is a considerable possibility that **X** and **Y** will not be connected in the near future.

Sanjay Kumar in [4] were presented a strategy for predicting links using a range of node centralities in conjunction with several machine-learning classifiers. The method was developed as part of this study. One may better represent the network's global, quasi-local, and local structures using several well-known and more recently developed node centralities. The values assigned to the various degrees of node centrality serve as the attribute labels for the individual nodes that make up the network. The edges that do exist in the network are referred to as positive samples and negative samples. A clearly defined dataset for the link prediction task can be constructed using the characteristics of the nodes located at the edges' ends, in addition to the positive or negative label. After that, the dataset is used as input for several different machine-learning classifiers; however, the Light Gradient Boosted Machine (LGBM) classifier produces the best overall results.

3. Preliminaries

Real-world networks are useful for depicting the intricate interconnections between people, places, and things. "Nodes" or "vertices" refer to the users or entities, while "links" or "*edges*" describe the relationships or interactions between them [17]. The graph **G(V, E)**, where **V** is the collection of nodes and **E** is the collection of edges connecting them, represents a directed, unweight network. A complete graph on n vertices specifies the largest number of edges present in a chart [4]. In other words, precisely one edge connects every two vertices, and each vertex is connected to **n-1** additional vertices. If the edges that are self-loop and multiple edges are ignored, the edges between any two nodes can be calculated as follows:

$$E=\frac{n(n-1)}{2} \quad (1)$$

Whereas, a set $\ddot{E}$ can be used to show the missing links or the links that will show up soon:

$$\tilde{E}=\frac{n(n-1)}{2}-E \quad (2)$$

Furthermore, the problem of predicting links in an undirected network has been formally defined. So, predicting links in a network, given a snapshot of the network at time t, can be thought of as indicating the missing links from set **E**, i.e., links that are present but not observed in the current snapshot but will be noticed at time **t + 1**. Several heuristics for predicting links between users presume that the greater the number of their mutual neighbors, the greater the likelihood of a future link between those users. Based on this hypothesis, several metrics have been proposed to measure the unseen connections between the nodes.

These are some examples of key performance indicators:

- **Common Neighbors (CN)**

The core tenet of common neighbors is that a greater number of shared neighbors increases the likelihood that two node pairs will build a link in the future [18]. The transitive features of the social network provide the basis for this index. As such, it can be described as follows:

$$S_{x,y}^{CN}=N_{(x)}\cap N_{(y)} \quad (3)$$

Where the $\boldsymbol{N_{(x)}}$ refers to the neighbors of node **x**.

- **Jaccard's Coefficient**

Jaccard's coefficient is a normalized form of the common neighbor metric that considers the entire number of neighbors for both vertices [19]. This Metric was named after the mathematician who first developed it. Given the following, which illustrates the node intersection divided by the union:

$$S_{x,y}^{Jaccard}=\frac{\boldsymbol{N(x)\cap N(y)}}{\boldsymbol{N(x)\cup N(y)}} \quad (4)$$

- **Preferential Attachment**

It has been seen that links are more likely to be made between two nodes with a higher degree than between two nodes with a lower degree [20]. The idea behind this measure is that network nodes with many connections are

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

likelier to make new connections with other network nodes. It can figure out the predicted score for hidden links by multiplying the degrees of both nodes. It can be shown mathematically like this:

$$S_{x,y}^{PA}=N_{(x)}*N_{(y)} \quad (5)$$

- **Resource Allocation (RA)**

When a pair of nodes do not share any neighbors with any other nodes, the score given to the pair of nodes increases [18]. The idea behind this metric is that two nodes are more likely to hook up in the future if they share a neighbor that does not have any other connections. Having more highly valued neighbors increases the probability of a connection being formed between two nodes. The corresponding mathematical expression is shown in Equation:

$$S_{x,y}^{RA}=\sum_{z\in\{N(x)\cap N(y)\}}\frac{1}{N(z)} \quad (6)$$

- **Adamic–Adar Index**

The Adamic–Adar (AA) index is comparable to the RA one [21]. The large-degree node still receives a punishment, although the severity of it varies. More importantly, the difference between RA and AA is negligible when the average degree is low but large otherwise.

$$S_{x,y}^{AA}=\sum_{z\in\{N(x)\cap N(y)\}}\frac{1}{log\,N(z)} \quad (7)$$

Connectivity in different networks can show different kinds of relationships between the nodes, which can be inferred based on the problem being looked at. It uses node centrality as a user feature. It can also consider the various topological and structural particulars that these relationships provide to show the critical nodes based on the studied problem [22].

Different node centralities have different rules about what properties a node should have in order to be called central. All node centralities work by giving other nodes a score of their importance based on the centrality definition used. In this study, we looked at the following centralities of nodes.

- **Betweenness centrality**

An importance value is assigned to a node based on the number of times it appears on the shortest path between two other nodes in the network. This measure of centrality is known as betweenness centrality [4]. It is predicated on the notion that the node that appears in the shortest path between two other nodes has the potential to function as a bridge between those nodes and significantly affect the flow of information between them.

$$Betweenness(z)=\sum_{i,j=1,i\neq j\neq z}^{n}\frac{\partial_{i,j}(z)}{\partial_{i,j}} \quad (8)$$

Where the $\partial_{i,j}$ is the total number of pathways from i to j, and $\partial_{i,j}$ (z) is the shortest path from **i** to **j** with **u** as an intermediate point along the way. It is derived from this idea.

- **PageRank centrality**

The PageRank centrality that underpins the Google search engine has been a driving force behind the company's meteoric rise to the top of the search engine rankings [4]. It expands on the idea of providing more weight to nodes that have received a greater number of links from other nodes. It is possible to formulate it in mathematical terms as follows:

$$PageRank(z)=\alpha\sum_{x=1}^{n}\frac{PageRank(x)}{degree_x^{out}}+\frac{1-\alpha}{n} \quad (9)$$

Here, the α damping factor with a value less than one was devised to mimic the effect that the closer nodes tend to exert a greater influence on the considered node. The value represents the node's out-degree.

- **Eigenvector centrality**

As a generalization of degree centrality, eigenvector centrality measures a node's significance relative to its neighbors [23][24] Connecting with influential individuals is a proven way to increase one's social standing. Taking equation 10 as a constant, $\mathbf{X_i}$ as the centrality score of node **i,** and if and only if node **z** is linked to node **i**, otherwise, the math formula for the eigenvector centrality score is as follows:

$$Eigenvector(z)=\frac{1}{\delta}\sum_{i=1}^{n}a_{z,i}\,X_i \quad (10)$$

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

4. Machine Learning Techniques

After converting this problem into a binary elegance classification problem [25][26], and setting up some capabilities for our data, we moved on to the supervised and unsupervised models. Our proposed model incorporated the following machine-learning techniques:

Support Vector Machine

This is a method of classification that does not rely on probabilities in any way. Additionally, it is sometimes referred to as a support vector network [19]. Support Vector Machine (SVM) is a mixture of techniques in supervised learning that are employed in the conduct of classification and regression tests. In other words, the algorithm creates a structure that divides fresh examples into two groups, each of which is based on the features of the corresponding training data. This framework classifies a new example based on the training data.

Decision Tree

The decision tree is an example of a class of machine learning algorithms known as supervised learning algorithms [5]. This type of method uses a tree-like structure as a splitting model to proceed from the root (the item's observations) to the leaves (the conclusion). In this scenario, the values for the dependent variables are all discrete.

Artificial Neural Network

It is a computer system that is built on the concept of a biological neural network [19]. The basic unit of this system is a nerve cell, and it is utilized to process information. Neurons are the technical term for these nerve cells. Each layer of the ANN model is partitioned into different layers based on the requirements. Each layer has a set of nodes that are linked to nodes from the layer above it. ANN stands for an artificial neural network. Different amounts of importance are placed on each relationship. There are several distinct varieties of neural networks, two of which are feed-forward and feedback networks. However, the loop is present in the feedback network so that, in response to error, the system parameters are continuously adjusted until it reaches a state of equilibrium. This process continues until the system is stable.

K-Nearest Neighbor

It is one of the simpler and more straightforward supervised machine learning methods. The K-Nearest Neighbor (KNN) algorithm can be used to resolve both regression and classification issues [27]. The KNN algorithm has as its premise the idea that related items tend to congregate together. It is calculating the Euclidean or other mathematical distance between the data points. Here, the initial value of **K** can be assigned by specifying a certain number of neighbors.

Random Forest

A random forest is created by collecting the results of classification and regression problems using decision trees. This type of forest can be applied to both classification and regression issues [28]. If there are more trees in the forest, the results will be more accurate. When working with large datasets, the number of variables increases and it can be challenging to cluster the data. Because of this, the RF approach can provide improved precision when analyzing data associated with a specific category. The different factors used in the decision-making process were chosen randomly for each tree. Using the testing dataset, one may predict which tree provides the most accurate classification.

Gradient Boosting

Gradient boosting is a method that can make learning easier from classification and regression models, which are usually not linear and are more often called "*decision trees*" or "*regression trees*" [29]. By slowly and orderly adding new learners to the model, a model can be built for a range of inaccurate prediction models, such as regression decision trees. It is made up of nodes and leaves, and the decision nodes allow it to figure out what will happen next. Individually, regression trees are not perfect models, but when they were looked at as a group, their accuracy was much better. Therefore, the groups were put together gradually, systematically, so that each one fixed the mistakes made in the previous one.

5. Proposed Technique

The steps necessary to construct our technique are outlined in Figure (1). The network is assumed to be undirected for every network **G**(**V**, **E**), where **V** is the collection of nodes and **E** is the collection of links. More specifically, the higher the value of $\mathbf{S_{x,y}}$, the greater the similarity between nodes **x** and **y** in this study.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

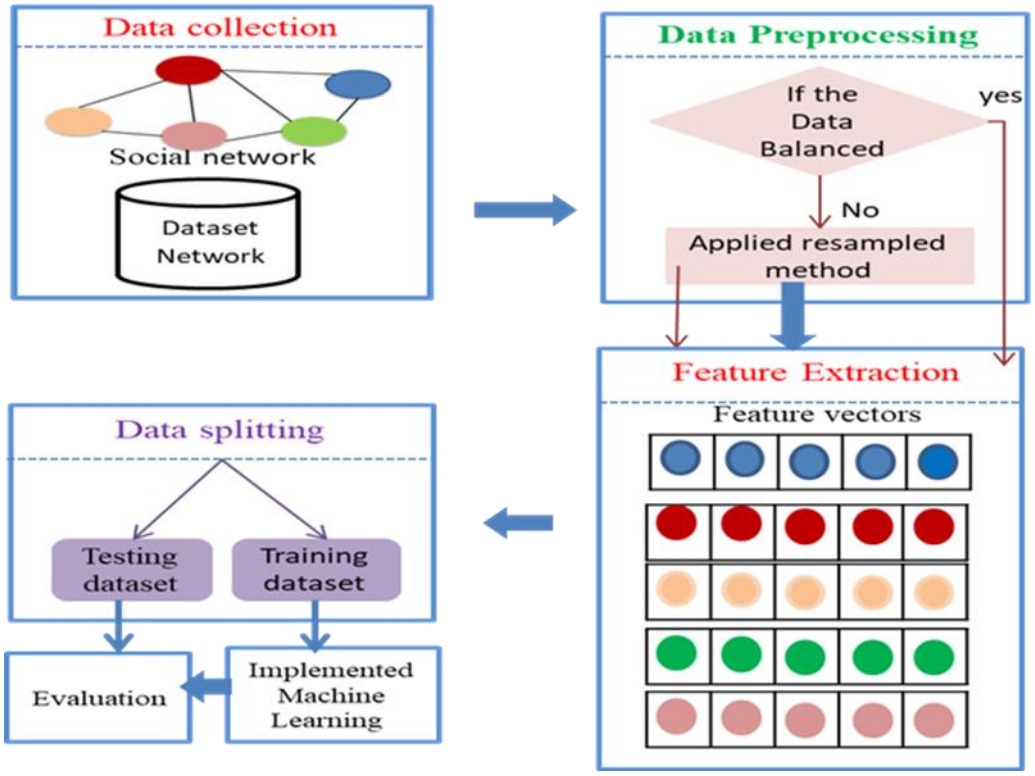

Figure 1. The block diagram of the proposed technique

It is worth noting that there are two ways to compute the degree of similarity between two nodes: comparing their local structures and their attributes, in addition to various similarity measures based on local structure. In our work, we relied on node attributes by integrating different types of node centrality to find the ideal centrality distribution, which accounted for the network's general structure, topological characteristics, and each node's weight.

5.1 Feature Vector Extraction

To produce a more general algorithm and treat it as a black box, one can ignore the details of the network context. This indicates that there are no attributes associated with the nodes that can be used as potential attributes. By taking into account the network structure, the connectivity between the nodes, and other network topological properties, local structures and node centralities typically provide important information about a node. This is true regardless of the context and the type of network being considered. Yet, diverse node centralities cover a variety of informative facets that are covered by the network. This demonstrates the need to use a mix of the network's node centralities and local structural similarities to analyze the data correctly.
Betweenness centrality, PageRank centrality, and Eigenvector centrality have been employed in addition to other measures of local network structure that capture different aspects of the network, such as Common Neighbors, Resource Allocation, Jaccard's Coefficient, Preferential Attachment, Eigenvector Centrality, and Adamic-Adar Index. There were three measurements of centrality for each node and five attributes based on commonalities in the local structure. Then, these collected features were fed into various machine-learning methods to perform the link prediction task. The characteristics were computed over a historical version of the network and then used to forecast links in a new version. This is warranted because, in the real world, we also predict future links by analyzing the structure of the network now.

5.2 Dataset Splitting Creation

To function properly, supervised machine learning techniques require a labeled collection that contains both positive and negative samples. To do this, we made a dataset with demonstrations that are both positive and negative. Here, the positive and negative illustrations showed edges that were currently there and those that did not exist. Existent nodes are those that can be found in the network at present; non-existent nodes are those that cannot be found in the network but could become so shortly. For an undirected graph, **G**(**V**, **E**) represents a complex network, and **E** and $\bar{\mathbf{E}}$ are the sets of existing edges and non-existing edges, respectively. The positive samples were labeled with a 1, and the negative samples were labeled with a 0. As a result, label 1 denoted the

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

existence of edges; however, label 0 represented the absence of edges between two nodes. For both types of samples, the process of feature extraction, as outlined in Section 5.1 was carried out by computing the values of local structures and node centralities.

A dataset is considered unbalanced if the data with a positive category is much smaller than the data with a negative category. This problem arises when machine learning distributes the classes inside a dataset. This class imbalance problem can be solved using a variety of techniques, such as resampling (over and under sampling), ensemble methods, or grouping the most common class. If the dataset is inherently unbalanced, we have thought about using the resampling method to balance it. The dataset is prepared for binary classification tasks like link prediction because it contains positive and negative samples. However, real-world networks are frequently sparse, and the proportion of connected edges to unconnected edges is generally lower. As a result, the dataset that was produced might be very unbalanced. We chose the same number of samples from edges with positive and negative values to solve this issue. Additionally, this is consistent with the findings of [30]. With the same number of positive and negative samples, this provided us with a balanced dataset. Additionally, it lessened the model's bias toward any specific sample type. To create a single dataset, these edge samples were then combined. For a randomly chosen sample to have an equal chance of being positive or negative, this dataset was shuffled. The final dataset was made up of the shuffled dataset.

5.3 Dividing the Dataset

To divide a well-labeled dataset into training and testing portions; first create the dataset. The testing set was used to measure the classifier's performance, while the training set was used to train the classifier model. We found an 80:20 split ratio works best for training and testing the model. As a result, the dataset was split into two parts, with 80% of it serving as a training set and the other 20% serving as a testing set, in an 80:20 ratio. The decision to divide the dataset in this way was made to ensure there are sufficient data reserves for testing the classifier's performance and training the classifier effectively while avoiding overfitting.

6. Evaluation of the Proposed Technique

6.1 The Dataset

Looking at networks from different domains, and taking into account their unique characteristics. Ignoring multiple edges and self-loops in those networks; we tested the proposed model on four real networks in Table (1).

Table 1: The fundamental statistical characteristics of the employed network datasets

Datasets	Nodes	Edges	Clustering coefficient	Average degree	Directed/undirected
Hamsterster	1858	12534	0.0904	13.494	Undirected
Facebook NIPS	2888	2981	0.0359	2.0644	Undirected
Caltech	769	16656	0.409	43	Undirected
Ca-GrQc	4158	13422	0.529	5	Undirected
DBLP	12591	49743	0.119	7	Directed
Epinions	26588	100120	0.135	7	Directed

The datasets used have different sizes and are different in how sparse they are. These datasets were picked to show how our proposed algorithm could be used and scaled up. Here are the different datasets we used for our study:

- ✓ **Hamsterster Friendship:** People who use the website hamsterster.com have made friends with each other, which are represented by this dataset. The nodes in this network represent the users, while the edges indicate the friendships among those people [31].
- ✓ **Facebook NIPS:** A friendship between users is represented in this network, which is both undirect and unweighted. A node represents each user in this network, and the friendship network allows authors to work together. It exemplified the authors' teamwork on works belonging to the Quantum Cosmology and General

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

Relativity subfields. The nodes represented the writers, and the co-authorship between them is shown by the edges [4].

- ✓ **Caltech:** It is a friend graph on Facebook created by the students attending Caltech in Pasadena, California. Each individual is represented as a node in the graph, and the links between nodes indicate friendships or other types of social ties that exist inside the network. It is comprised of 769 nodes and approximately 16656 edges [4][32].
- ✓ **Ca-GrQc:** For the arXiv e-print collection, Ca-GrQc serves as a network for authors to work together. It exemplifies the authors' teamwork on works belonging to the Quantum Cosmology and General Relativity subfields. The nodes represent the writers, and the co-authorship between them is shown by the edges [32].
- ✓ **DBLP:** It is a DBLP computer science reference co-citation network; here a node represents a research paper while an edge represents a citation between the papers [33]. The network contains 12591 research papers, which are represented by nodes, and 49743 citations, which are represented by edges.
- ✓ **Epinions**: It is a social network based on the trust principle between users of the site Epinions. This means that the members of the network, which are represented by nodes, trust each other. This means that the edges represent the trust between the members [34]. This network has 26,588 nodes and 100,120 edges.

6.2 Evaluation Metrics

The proposed algorithm was contrasted with common link prediction algorithms. Accuracy, F-score, precision, and recall had all been used to gauge the models' performance. We also used the AUC value to gauge performance. Additionally, we compared the performance of our algorithm to a benchmark link prediction algorithm based on information gain. All these evaluation metrics clearly showed the predictive performance of various link prediction algorithms. Below is a brief description of these metrics:

- ✓ **Accuracy:** The ratio of the number of links that are correctly predicted to the total number of non-existent links is used to measure the model's accuracy [29].

$$Accuracy = \frac{number\ of\ correct\ prediction}{total\ number\ of\ data} \quad (11)$$

- ✓ **Precision:** Precision measures how many correctly classified positive classifications there were overall [35]. In mathematics, it can be written as:

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

- ✓ **Recall:** This performance metric seeks to indicate the proportion of actual positives correctly identified. Recall for a binary classification task can be mathematically expressed as [14]:

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

- ✓ **F1 Score:** This performance F1 score is a metric used to evaluate the performance of classification models. This metric combines precision and recall into a single score [36].

$$F1\text{-}score = \frac{2*Precision*Recall}{Precision+Recall} \quad (14)$$

- ✓ **Information gain:** The improvement in performance metrics that an algorithm achieves for the task at hand when compared to other algorithms is represented by the information gain attained by any algorithm. Formally, it can be said as follows:

$$Information\ gain = \frac{Metric(proposed) - Metric(Existing)}{Metric(Existing)} \quad (15)$$

Equation (15) represents the performance metric under consideration; metric (Proposed) defines the metric computed for the proposed algorithm; and metric (Existing) defines the metric calculated for the existing algorithm [4].

7. Experimental Results and Analysis

In particular, we provided a comprehensive evaluation of our own inquiry. We conducted our tests using a split of the entire dataset commonly used for training and testing machine learning models, focusing on the latter. In our practice, we trained our model on 80% of the data and then put it to the test on the remaining 20%. The dataset

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

was partitioned to prevent overfitting while providing sufficient data for training the classifier. In addition, Section 5.3 provides a thorough evaluation of the optimal splitting ratio. Python was used for all of the code development. Several public GitHub repositories and Python libraries, such as sklearn, numpy, pandas, etc., were used to obtain results on existing methodologies. We used a personal computer with a 2.00GHz Intel Core I7-2630 processor and 4.00GB of random-access memory for our simulations. We compared the efficacy of our proposed approach to that of other well-established ones for link predictions, including the Adamic-Adar (AA), the Jaccard Coefficient (JC), and the Preferred Attachment (PA). In addition to these baseline techniques, we also conducted experiments with a variety of additional machine learning classifiers using our extracted node centrality-based features. The machine learning methods included in Section 4; Multi-layer preceptor, Support vector machine, Decision trees, Random forest, KNN, and gradient boosting are some of the different machine learning classifiers that are applied in our experiments. In this section, we offer a brief description of the value achieved by varying evaluation metrics for each dataset based on the various link prediction algorithms that were taken into consideration.

7.1 Splitting Ratio Option Selection

This portion looks at the best way to divide the dataset into training and testing sets. To figure this out, we looked at how well different machine learning classifiers, like Multi-layer preceptors, Support vector machines, Decision trees, Random forests, KNN, and gradient boosting, do on accuracy measures for each dataset. The study was done while the value of the splitting ratio was changed in steps of 5% from 50% to 95%. The results that were found are shown in Figure (2).

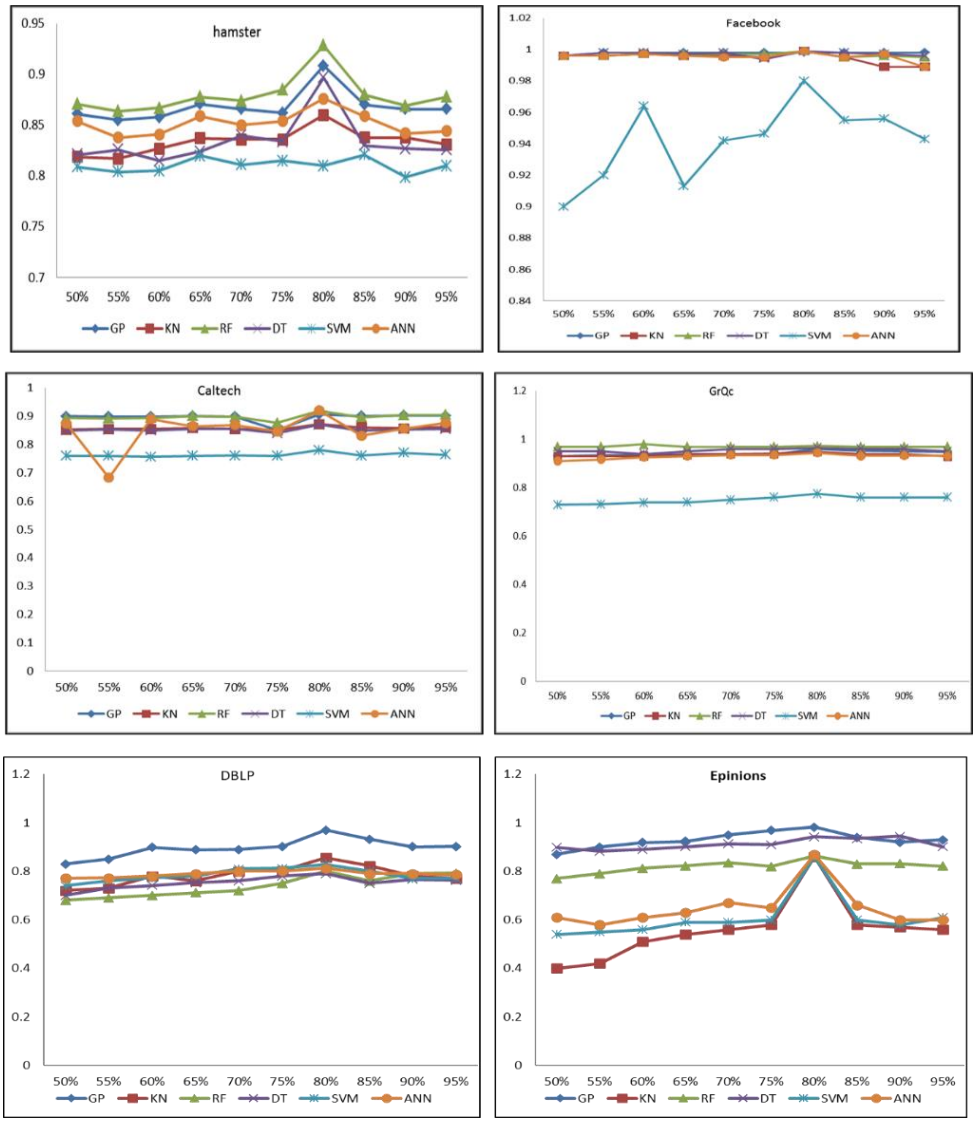

Figure 2. Accuracy scores for different datasets and classifiers were found when the splitting ratio was changed from 50% to 95% in 5% increments

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

Figure (2) shows that for each dataset, the accuracy value grows with increasing splitting ratio, reaching a maximum of almost 80%. The data were being underfitted in this case since the splitting ratio was close to 50%. This shows that the technique used could not correctly learn the network's features at a smaller splitting ratio because there was not enough data to train the dataset. The results showed that increasing the split ratio from 50% to 80% yielded better accuracy performance, with the best result being achieved at 80%. It was also shown that the accuracy value decreased as the splitting ratio increased by over 80%. This shows that overfitting increased as the size of the training set grew. According to the analysis results, the dataset was best divided by 80/20.

7.2 Comparative Performance Study

Here, we evaluated our proposed approach against other algorithms that were examined for each dataset using a variety of criteria. To gain a more accurate analysis, we conducted our experiments ten times and then took the average of the findings obtained for each evaluation metric, as most link prediction methods are stochastic. From Figure (3) to (6), we saw the comparative analysis in terms of several evaluation metrics. The accuracy comparison analysis of six real-world network datasets is presented in Figure (2). Compared to competing techniques, Gradient Boosting performed exceptionally well on the Caltech, Facebook Friendships, DBLP, and Epinions datasets. However, the Ca-GrQc and hamster network results showed that random forests provided higher accuracy. Although the decision tree's link prediction accuracy was lower than that of the random forest, it was still greater than that of other models.

The comparative evaluation of several link prediction algorithms concerning their level of precision in real-world networks is shown in Figure (3). The precision value of Gradient Boosting was better in real-world datasets, particularly in friendship datasets such as those found on Facebook, Caltech, and the DBLP citation network. Figure (3) reveals that the SVM algorithm had a lower precision value when applied to the Ca-GrQc, Caltech, Facebook, Epinions and Hamster datasets. This is something that could be seen. Notwithstanding this, the Random Forest approach achieved the highest precision value in both the Ca-GrQc, Epinions, and Hamster datasets.

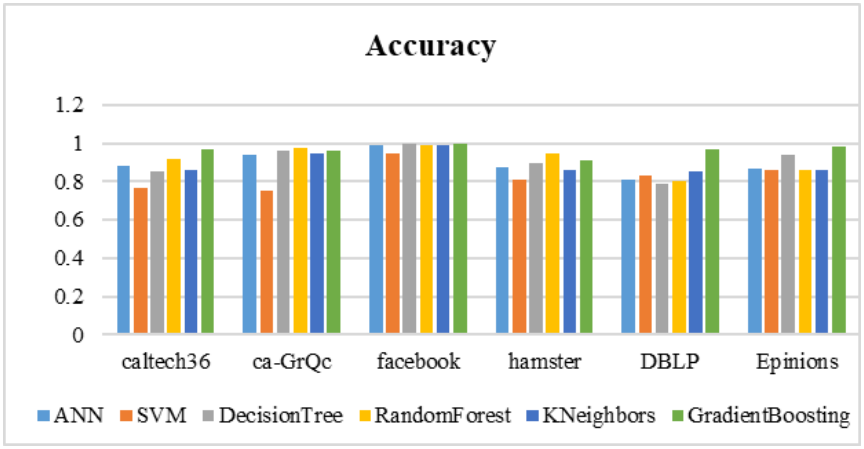

Figure 3. The accuracy comparison analysis of four real-world network datasets

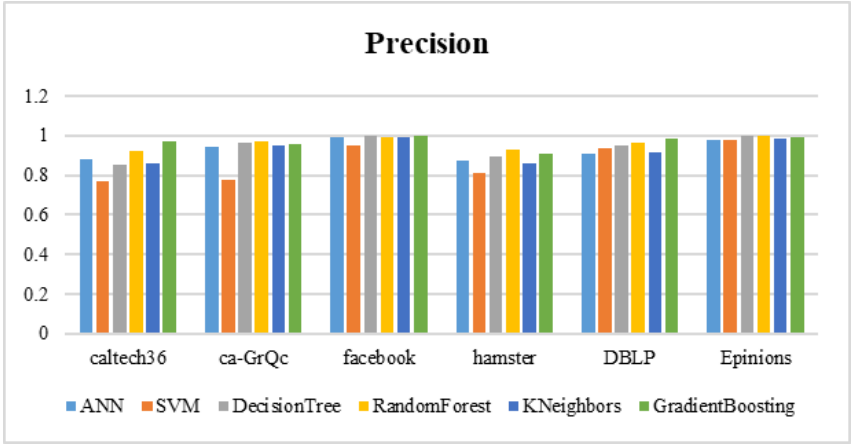

Figure 4. The precision result in four real-world networks

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

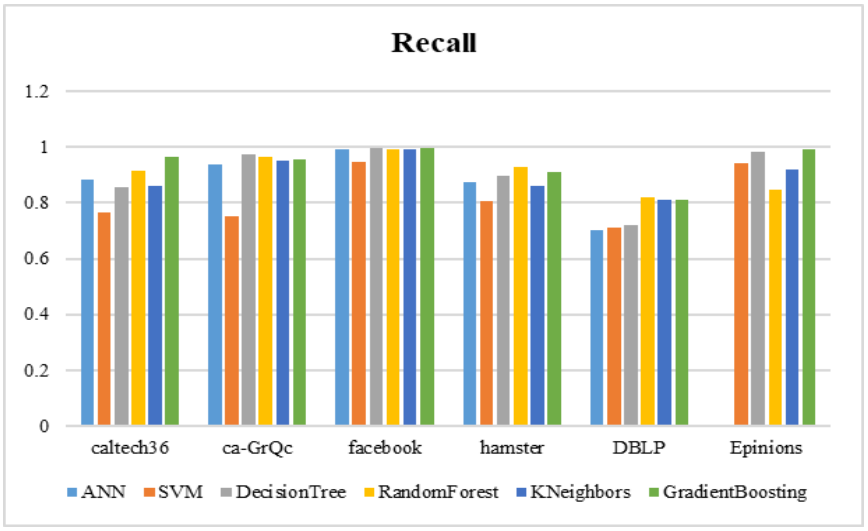

Figure 5. The recall comparative analysis

Figure (5) shows a comparison of different recall-based models for link prediction. Both the Caltech, Epinions and Facebook networks performed better when using the Gradient Boosting technique. Using the Ca-GrQc network dataset, a decision tree was determined to have the highest recall performance. The Hamsterster and DBLP networks dataset was where random forest shined.

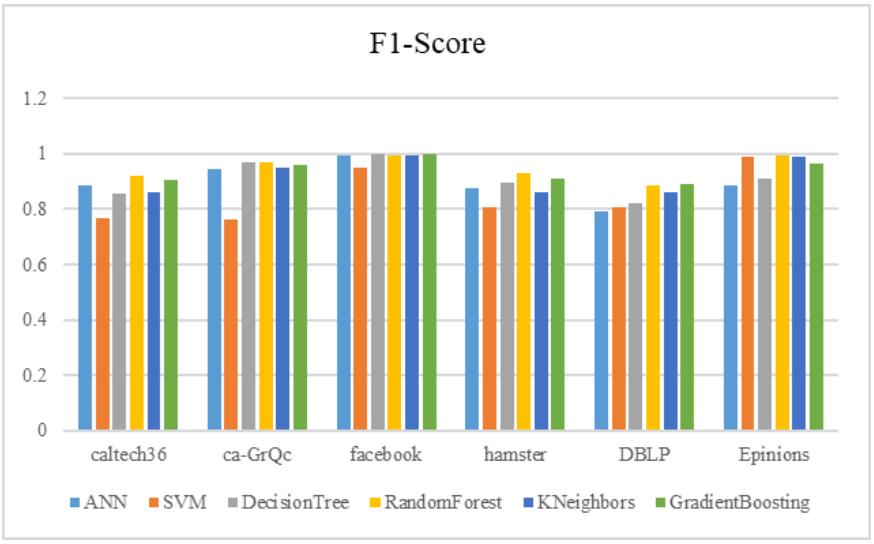

Figure 6. The F1 score comparative analysis

Comparing the link prediction models with the F1-score metric. Figure (6) shows the comparative analysis of the performance of these models using F1-score. It can be seen that Random Forest algorithms perform better on Caltech36, Ca-GrQc, Epinions, and Hamsterster networks, while Gradient Boosting outperforms all the other algorithms on DBLP and Facebook networks. These results indicate balanced and improved performance across multiple and diverse networks.

Table (2) displays the AUC for a variety of algorithms. For all of the considered Caltech, DBLP, Epinions, and Facebook datasets, Gradient Boosting provided a higher AUC value. Meanwhile, for Ca-GrQc and hamster networks, in comparison to the values obtained by the other algorithms, the AUC for the random forest model appeared to have a high value. The various performance measures obtained for all datasets showed that the SVM model had the lowest performance value.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

Table 2: Analyzing link prediction methods comparisons in terms of AUC value

	Caltech 36	**Ca-GrQc**	**Facebook**	**Hamster**	**DBLP**	**Epinions**
ANN	0.883431	0.939349	0.993007	0.872608	0.812	0.869
SVM	0.765861	0.751422	0.948427	0.80718	0.829	0.861
DT	0.85499	0.964657	0.996503	0.896597	0.79	0.942
RF	0.9017	**0.972639**	0.993007	**0.928344**	0.801	0.862
KN	0.861246	0.948959	0.993007	0.858532	0.856	0.8601
G B	**0.966346**	0.957876	**0.996503**	0.908208	**0.969**	**0.982**

Figures 3 to 6 show the performance evaluation of the proposed method for link prediction based on evaluation metrics such as AUC, precision, recall, and F1 score. We can see from these results that the model maintains its consistency and performance by changing the size and type of networks. It can be seen that for a small network such as the Caltech social network, our model achieved an AUC value of 90.6%, while for a larger network such as the Facebook social network; the proposed algorithm achieved an AUC value of 98.4%. In addition, for a relatively large Epinions citation network with 26588 nodes and 100120 edges, our model achieved an AUC value of 98.2%. Thus, we can conclude that with increasing changes in the type and size of the network, the performance of the proposed model is uniform and does not deteriorate.

7.3 Information Gain Result

Information gain is a measure of the relative improvement of an algorithm over other algorithms. The information gain achieved by our proposed method is compared to the best algorithm used in other research according to the data in Table (3). In [37], the authors proposed the Heterogeneous Degree Penalty (HDP) for the Caltech network. The AUC result was 0.9231, while the study [4] compared it with the AUC of the Ca-GrQc, DBLP, and Epinions networks. The study used several machine-learning classifiers, and the best results were obtained using the Light Gradient Boosted Machine (LGBM) classifiers, 0.933, 0.955, and 0.964, respectively. In the research [5], the AUC for the Facebook network is 0.891 in the best SVM algorithm, as the researchers used several models of machine learning algorithms to link the predictions, while the AUC on the Hamster dataset reached 0.923 in the SVM model.

Table 3. Gain in the area under the curve (AUC) of the suggested method compared to the optimal baseline

Datasets	Against best baseline (%)
Caltech	0.0468 (Gradient Boosting) [37]
Ca-GrQc	0.0424 (Random Forest) [4]
Facebook	0.1178 (Gradient Boosting) [5]
Hamster	0.0275 (Random Forest) [5]
DBLP	0.0146 (Gradient Boosting) [4]
Epinions	0.0186 (Gradient Boosting) [4]

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

As can be seen, the recommended way offered a significant knowledge gain. This illustrates that combining particular node centralities gave our method more crucial information that was required to achieve a high level of success for the link prediction challenge.

8. Conclusions

One of the most researched problems associated with complex network analysis is the prediction of links between nodes. Predicting the existence of links that have not yet been observed but will be observed in the future can be understood as the problem of link prediction. In this research, we proposed a method for establishing link predictions by developing a collection of features that take into account both node centralities and node similarity. Different binary machine-learning classifiers are then fed these node features. Some real, complex networks were used in the experiments, and a variety of performance metrics was determined. The obtained experimental results suggest that a classifier based on Gradient Boosting and Random Forest, with node centralities as additional features, is likely to perform the best. Experimental results show that when using traditional methods such as accuracy, precision, and recall to validate the work, it is possible to handle the correlation prediction problem by using one of the appropriate classification models. In the not-too-distant future, we may use content-based features in feature vector creation for reliable forecasting. To keep things simple, we just thought about the unweighted static network to foresee future connections. The dynamic network, in which new connections and nodes are constantly added, may be considered. In the near future, a promising approach to link prediction based on deep learning can be used to capture the complex transitional variability and local effects to find the link prediction in dynamic networks. One way to quantify the closeness of ties in a weighted network is through the link weights to take into account the role of links in networks In addition, link prediction algorithms could use that. This can be incorporated into the model that has been presented to foretell future connections.

References

[1] A. K. Singh and L. Kailasam, "Link prediction-based influence maximization in online social networks," *Neurocomputing*, vol. 453, pp. 151–163, 2021. DOI: https://doi.org/10.1016/j.neucom.2021.04.024.

[2] E. A. Abbas and H. N. Nawaf, "Improving Louvain algorithm by leveraging cliques for community detection," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, 2020, pp. 244–248. DOI: https://doi.org/10.1109/CSASE49402.2020.9142067.

[3] F. Aziz, L. T. Slater, L. Bravo-Merodio, A. Acharjee, and G. V. Gkoutos, "Link prediction in complex networks using information flow," *Scientific Reports*, vol. 13, no. 1, p. 14660, 2023. DOI: https://doi.org/10.1038/s41598-023-41476-9.

[4] S. Kumar, A. Mallik, and B. S. Panda, "Link prediction in complex networks using node centrality and light gradient boosting machine," *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022. DOI: https://doi.org/10.1007/s11280-021-00945-1.

[5] A. Kumari et al., "Supervised link prediction using structured-based feature extraction in social networks," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 13, p. e5839, 2022. DOI: https://doi.org/10.1002/cpe.5839.

[6] A. Samad et al., "A comprehensive survey of link prediction techniques for social networks," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 7, no. 23, pp. 1–21, 2020. DOI: https://doi.org/10.4108/eai.10-11-2020.167963.

[7] X. Wang, H. Yang, and M. Zhang, "Neural common neighbor with completion for link prediction," *arXiv preprint arXiv:2302.00890*, pp. 1–17, 2023. DOI: https://doi.org/10.48550/arXiv.2302.00890.

[8] S. Li et al., "Friend recommendation for cross-marketing in online brand community based on intelligent attention allocation link prediction algorithm," *Expert Systems with Applications*, vol. 139, p. 112839, 2020. DOI: https://doi.org/10.1016/j.eswa.2019.112839.

[9] B. Kaya, "A hotel recommendation system based on customer location: A link prediction approach," *Multimedia Tools and Applications*, vol. 79, pp. 1745–1758, 2020. DOI: https://doi.org/10.1007/s11042-019-08344-y.

[10] T. K. T. Ho, Q. V. Bui, and M. Bui, "Co-author relationship prediction in bibliographic networks: A new approach using geographic factors and latent topic information," in *Proceedings of the 10th International Symposium on Information and Communication Technology*, 2019, pp. 69–77. DOI: https://doi.org/10.1145/3368926.3368937.

[11] A. Breit et al., "OpenBioLink: A benchmarking framework for large-scale biomedical link prediction," *Bioinformatics*, vol. 36, no. 13, pp. 4097–4098, 2020. DOI: https://doi.org/10.1093/bioinformatics/btaa598.

[12] E. C. Mutlu et al., "Review on graph feature learning and feature extraction techniques for link prediction," *arXiv preprint arXiv:1901.03425*, pp. 1–16, 2019. DOI: https://doi.org/10.48550/arXiv.1901.03425.

[13] K. Zhou et al., "Attacking similarity-based link prediction in social networks," *arXiv preprint arXiv:1809.08368*, pp. 1–9, 2018. DOI: https://doi.org/10.48550/arXiv.1809.08368.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

[14] S. Haghani and M. R. Keyvanpour, "A systemic analysis of link prediction in social networks," *Artificial Intelligence Review*, vol. 52, pp. 1961–1995, 2019. DOI: https://doi.org/10.1007/s10462-018-9632-8.
[15] J. Wu et al., "General link prediction with influential node identification," *Physica A: Statistical Mechanics and Its Applications*, vol. 523, pp. 996–1007, 2019. DOI: https://doi.org/10.1016/j.physa.2019.02.007.
[16] A. Samad, M. Azam, and M. Qadir, "Structural importance-based link prediction techniques in social networks," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 7, no. 25, pp. 1–13, 2021. DOI: https://doi.org/10.4108/eai.22-1-2021.167785.
[17] X. Liu et al., "Link prediction approach combined graph neural network with capsule network," *Expert Systems with Applications*, vol. 212, p. 118737, 2023. DOI: https://doi.org/10.1016/j.eswa.2023.118737.
[18] I. Ahmad et al., "Missing link prediction using common neighbor and centrality-based parameterized algorithm," *Scientific Reports*, vol. 10, no. 1, pp. 1–9, 2020. DOI: https://doi.org/10.1038/s41598-020-62844-w.
[19] A. Sharma, S. Soni, and K. Rai, "Link Prediction in Social Network using Artificial Neural Network," *International Journal of Computer Applications*, vol. 174, pp. 26–30, 2021. DOI: https://doi.org/10.5120/ijca2021911861.
[20] L. Yin et al., "An evidential link prediction method and link predictability based on Shannon entropy," *Physica A: Statistical Mechanics and Its Applications*, vol. 482, pp. 699–712, 2017. DOI: https://doi.org/10.1016/j.physa.2017.04.001.
[21] P. Raut et al., "A comparative study of classification algorithms for link prediction," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, IEEE, 2020, pp. 479–483. DOI: https://doi.org/10.1109/ICIMIA48430.2020.9074957.
[22] S. Kumar, D. Lohia, D. Pratap, A. Krishna, and B. S. Panda, "MDER: Modified degree with exclusion ratio algorithm for influence maximisation in social networks," *Computing*, vol. 104, no. 2, pp. 359–382, 2022. DOI: https://doi.org/10.1007/s00607-021-00946-y.
[23] S. Behrouzi, Z. S. Sarmoor, K. Hajsadeghi, and K. Kavousi, "Predicting scientific research trends based on link prediction in keyword networks," *Journal of Informetrics*, vol. 14, no. 4, p. 101079, 2020. DOI: https://doi.org/10.1016/j.joi.2020.101079.
[24] E. A. Abbas and H. N. Nawaf, "Influence maximization based on a non-dominated sorting genetic algorithm," *Karbala International Journal of Modern Science*, vol. 7, no. 2, p. 5, 2021. DOI: https://doi.org/10.33640/2405-609X.2670.
[25] T. A. Diame et al., "Data management and decision-making process using machine learning approach for enterprises," *Full Length Article*, vol. 8, no. 1, p. 75, 2023. DOI: https://doi.org/10.1016/j.diba.2022.101076.
[26] D. Arrar, N. Kamel, and A. Lakhfif, "A comprehensive survey of link prediction methods," *Journal of Supercomputing*, vol. 80, no. 3, pp. 3902–3942, 2024. DOI: https://doi.org/10.1007/s11227-023-05060-1.
[27] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 1255–1260. DOI: https://doi.org/10.1109/ICCS45141.2019.9065762.
[28] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, Springer, 2019, pp. 758–763. DOI: https://doi.org/10.1007/978-3-030-03146-6_70.
[29] M. Badiy and F. Amounas, "Embedding-based method for the supervised link prediction in social networks," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 3, pp. 105–116, 2023. DOI: https://doi.org/10.47941/ijritcc.v11i3.7214.
[30] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 641–650. DOI: https://doi.org/10.1145/1772690.1772756.
[31] A. Ullah et al., "Identifying vital nodes from local and global perspectives in complex networks," *Expert Systems with Applications*, vol. 186, p. 115778, 2021. DOI: https://doi.org/10.1016/j.eswa.2021.115778.
[32] M. Azam et al., "Evaluations of similarity-based link prediction techniques in social networks," *Journal of Engineering Science and Technology*, vol. 18, no. 2, pp. 1055–1082, 2023. DOI: https://doi.org/10.5281/zenodo.7824236.
[33] M. Wang et al., "Graph ranking auditing: Problem definition and fast solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 10, pp. 3366–3380, 2020. DOI: https://doi.org/10.1109/TKDE.2019.2949253.
[34] X. Zhou and Z. Zhang, "Opinion maximization in social networks via leader selection," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 133–142. DOI: https://doi.org/10.1145/3543507.3583230.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024

[35] S. Behrouzi, Z. S. Sarmoor, K. Hajsadeghi, and K. Kavousi, "Predicting scientific research trends based on link prediction in keyword networks," *Journal of Informetrics*, vol. 14, no. 4, p. 101079, 2020. DOI: https://doi.org/10.1016/j.joi.2020.101079.

[36] H. Hairani, A. Anggrawan, and D. Priyanto, "Improvement performance of the random forest method on unbalanced diabetes data classification using Smote-Tomek Link," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 1, pp. 258–264, 2023. DOI: https://doi.org/10.30630/joiv.7.1.1305.

[37] R. Song et al., "Link prediction based on heterogeneous degree penalization with extending neighbors and clustering coefficient," *International Journal of Modern Physics C*, vol. 33, no. 3, p. 2250033, 2022. DOI: https://doi.org/10.1142/S0129183122500334.

DOI: https://doi.org/10.54216/FPA.180104
Received: June 20, 2024 Revised: September 19, 2024 Accepted: December 23, 2024