Machine Learning with Applications 6 (2021) 100086

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Supervised-learning link prediction in single layer and multiplex networks

Deepanshu Malhotra, Rinkaj Goyal [*]

University School of Information, Communication and Technology, Guru Gobind Singh (GGS) Indraprastha University, New Delhi 110078, India

ARTICLE INFO

Keywords:
Supervised learning
Link prediction
Complex networks
Multiplex networks
Social networks
Node similarity

ABSTRACT

The emergence of complex real-world networks has put forth a plethora of information about different domains. Link-prediction is one of the emerging research problems that utilizes the information from the networks to find future relationships between the nodes. The structure of real-world networks varies from having homogeneous relationships to having multiple associations. The homogeneous relationships are modeled by single-layer networks, while the multiplex networks represent the multiple associations. This study proposes a solution for finding future links in single-layer and multiplex networks by using supervised machine learning techniques. This study considers a set of topological features of the network for training the machine learning classifiers. The training and testing data set construction framework devised in this work helps in evaluating the proposed method on different networks. This study also contributes towards identifying four community-based features for the proposed mechanism.

1. Introduction

A complex network is a general model of the connections and relations among the components in a system. These networks are represented using graphs, where a node denotes a component or a user, and an edge signifies the connection between different users. Such representations have been used to examine complicated problems from diverse fields like biology (Jeong, Tombor, Albert, Oltvai, & Barabási, 2000), chemistry (Doye, 2002), social sciences (Travers & Milgram, 1969), web (Albert, Jeong, & Barabási, 1999), and inter-banking systems (Boss, Elsinger, Summer, & 4, 2004).

Prediction of future or hidden links in a complex network (Fire, Tenenboim, Lesser, Puzis, Rokach, & Elovici, 2011; Liben-Nowell & Kleinberg, 2007; Lü & Zhou, 2011) is one such problem that has attracted the keen interest of researchers for a long time. The discovery of these links has been utilized in many applications like fake profiles detection in social networks (Kagan, Elovichi, & Fire, 2018), community detection using similarity-based approaches (Cheng, Ning, Yin, Yan, Liu, & Zhang, 2018), improving product recommendations (Li, Zhang, Meng, & Li, 2014), discovering the latest and popular events (Zhang & Lv, 2018), etc. Daud, Ab Hamid, Saadoon, Sahran, and Anuar (2020) present a detailed review of the various link prediction techniques and their applications.

The ubiquitous presence of link prediction problems in different domains has encouraged the development of several techniques and algorithms. The classical approaches to solving this problem include extracting similarity-based topological features between the nodes like Common neighbors (Liben-Nowell & Kleinberg, 2007), Jaccard Index (Jaccard, 1901), Preferential Attachment (Newman, 2001), Resource Allocation Index (Zhou, Lü, & Zhang, 2009), Adamic–Adar Index (Adamic & Adar, 2003), etc. Although these methods are simple and computationally efficient, they are less successful on large real-world networks. The recent growth of artificial intelligence for various applications (Altan & Karasu, 2019, 2020; Altan, Karasu, & Bekiros, 2019; Altan, Karasu, & Zio, 2021; Altan & Parlak, 2020; Karasu, Altan, Bekiros, & Ahmad, 2020), has inspired the adoption of supervised learning methods for predicting links. These techniques have helped in attaining superior performances by using an effective feature-set. Al Hasan, Chaoji, Salem, and Zaki (2006) use a number of topological and non-topological features (like the overlap of interest among people) under the supervised learning setup for finding future links. They also rank and compare the different features based on their prediction ability. A similar setup was used by Fire et al. (2011) where they utilize simple, easy-to-compute structural features for machine learning algorithms. Furthermore, they present a new structural feature, that is friends measure (FM), to enhance their link prediction framework. Despite the results of above-mentioned studies, their methods do not consider the content based features. Ahmed, ElKorany, and Bahgat (2016) build upon the previous works by utilizing the additional information of share, mention, or reply action by users in the twitter network. They use different snapshots of the twitter data set to construct training and testing sets. Although they obtain robust performance, the experiments were conducted on a single data set. Recently, Kumari, Behera, Sahoo, Nayyar, Kumar Luhach, and

[*] Corresponding author.
E-mail addresses: deepanshumalhotra@gmail.com (D. Malhotra), rinkajgoyal@gmail.com, rinkaj@ipu.ac.in (R. Goyal).

https://doi.org/10.1016/j.mlwa.2021.100086
Received 27 April 2021; Received in revised form 6 June 2021; Accepted 21 June 2021
Available online 27 June 2021
2666-8270/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Prakash Sahoo proposed a unique test data set construction framework by computing a mean similarity index of topological features for link prediction. They also conduct different experiments on real-world and synthetic networks.

Although most of the existing techniques focus on finding links in single-layer networks, many real systems have multilayer structures (Boccaletti, Bianconi, Criado, del Genio, Gómez-Gardeñes, Romance, na Nadal, Wang, & Zanin, 2014; Kivelä, Arenas, Barthelemy, Gleeson, Moreno, & Porter, 2014). These systems are modeled by multiplex networks that consist of the same type of nodes in distinct layers, and each layer represents a different set of interactions between the nodes (Kinsley, Rossi, Silk, & VanderWaal, 2020). Since multiplex networks provide a different view of relations in each layer, the proper use of information from other layers could help achieve better efficiency. Sharma and Singh (2015) use the weighted combination of information from all the layers to predict the link in the target layer. A rank aggregation-based supervised learning approach for the node features from different layers is used in Manisha Pujari (2015) to predict the likelihood of links. Yao, Zhang, Yang, Yuan, Sun, Qiu, and Hu (2017) devise a node similarity-based method that uses the interlayer and intralayer information in the multiplex networks. Mandal, Mirchev, Gramatikov, and Mishkovski (2018) focus on finding the suitable network features from all layers to be utilized for machine learning models to increase the accuracy of predicting connections. Although these techniques work on multiplex networks, there are some deficiencies. The work of Sharma and Singh (2015) does not consider the link information in the target layer. Manisha Pujari (2015) test their method only on a single DBLP network and Yao et al. (2017) use a complex tunable parameter. Recently, Shan, Li, Zhang, Bai, and Chen (2020) presented a supervised learning-based link prediction technique by integrating the topological information from all the layers. They also develop two new friendship-based structural features for improving the accuracy of their technique on multiplex networks.

In light of the context above, a solution is presented for the link prediction problem in single-layer and multiplex networks by computing a set of topological features. The motivations behind the proposed method are:

1. To create a unified framework that could easily be used for single layer and multiplex networks.
2. To build the training and testing data set for networks that work with unknown ground truth link structure.
3. To effectively utilize the topological information of the network for improving the link prediction accuracy.

In the proposed method, structural similarity features are computed and utilized for training the various machine learning classifiers. The fundamental technique of using the information from various layers in multiplex networks (Shan et al., 2020) is incorporated in our methodology. Four community-based features are also designed that are utilized in the proposed feature model. We expound on the basic framework presented in Ahmed et al. (2016) and combine it with appropriate topological features (Soundarajan & Hopcroft, 2012; Valverde-Rebaza & de Andrade Lopes, 2012). Finally, we modify the training and testing data set construction technique presented in Lü and Zhou (2011) for calculating the efficiency of the supervised learning methods on various complex networks. The main contributions of this paper are as follows:

1. A supervised learning-based framework is proposed that utilizes the topological information of the networks to predict connections in single layer and multiplex networks. The proposed method incorporates the information from different layers in a multiplex network.
2. An effective training and testing data set construction technique is presented for experimentation with supervised learning algorithms.
3. We propose four community-based features to enhance the information content, thereby improving the accuracy score.
4. Extensive experimentation is conducted on single and multilayer real-world network data set(s).

Table 1
The list of notations used in this paper.

Notation	Description
V	Set of nodes in a network.
E	Set of edges in a network.
$\|V\|$	Number of nodes in the network.
$\|E\|$	Number of edges in the network.
G_s	A single layer network.
G_m	A multiplex network.
$G_{mi} = (V_i, E_i)$	A graph for the ith layer in a multiplex network G_m.
G_α	The layer on which link is predicted, also known as the target layer.
G_β	Other layers, apart from target layer, are called auxiliary layers.
$\Gamma(x)$	The set of neighbors of a given node x.
k_x	Degree of a given node x.
$(A^2)_{xy}$	Paths of length two between the nodes x, y.
$(A^3)_{xy}$	Paths of length three between the nodes x, y.
z_{comm}	The number of common neighbors having same community as the nodes x, y.
t_{comm}	The total number of neighbors having same community as the nodes x, y.
E_T	The set of edges initially present in the network.
E_{Tb}	A random sample of 10% of the edges in E_T.
E_{Tc}	A set of all the pairs of disconnected nodes that are 2-hop away from each other.

2. Methodology

2.1. Network description

This study utilizes both single-layer and multiplex networks, which are modeled as graphs. A simple undirected network is represented by $G_s(V, E)$ where G_s signifies the network, V represents the nodes, and E is used to denote the set of links in the graph. $|V|, |E|$ are used to denote the number of vertices and edges in the graph, respectively. In the case of a multiplex network with N nodes and p layers, it is denoted by $G_m = (G_{m1}, G_{m2}, G_{m3}, \dots, G_{mp})$. Here $G_{mi} = (V_i, E_i)$ denotes the network for the ith layer. G_α represents the layer on which link is predicted, also known as the target layer. Other layers are symbolized by G_β, called auxiliary layers, that provide additional information about the existence of relations in the target layer. A multiplex network is shown in Fig. 1. Table 1 presents the list of notations used in this work.

2.2. Feature extraction

We use appropriate network information to calculate similarity-based features for the proposed supervised learning model. All the features detailed in this section are used for all types of network, that is, simple undirected network, multiplex network (both target and auxiliary layer) except for the Friendship in Auxiliary layer (FAL) (Shan et al., 2020) feature. FAL is only used for the auxiliary layers in a multiplex network. The various features are described below.

2.2.1. Features based on connectivity

1. Common neighbors (CN): The common neighbors (Liben-Nowell & Kleinberg, 2007) between a pair of vertices is the length of the set of vertices that have a connection to both the given vertices. The formal definition is given below where $\Gamma(x)$ and $\Gamma(y)$, represent the neighbors of nodes x and y respectively.

$$CN = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

2. Jaccard Index (JC): The jaccard index (Jaccard, 1901) is the ratio of the number of common neighbors to the total neighbors of the gives nodes. It is calculated using the equation below.

$$JC = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2}$$

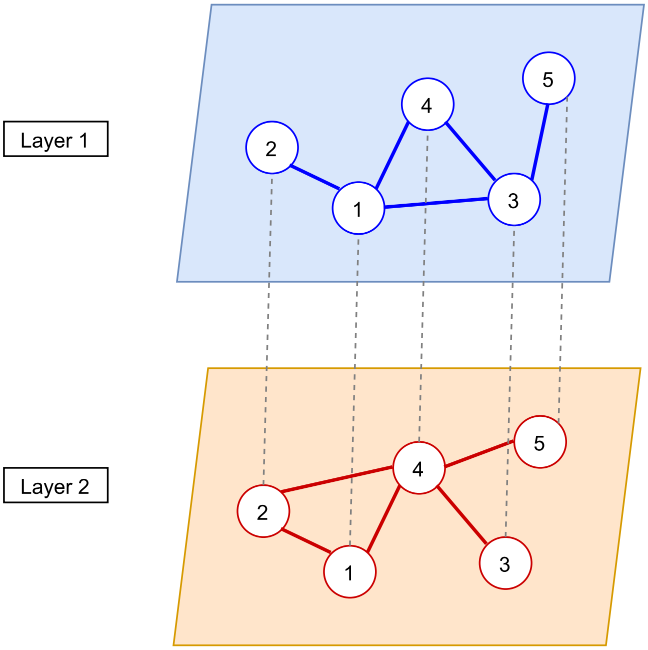

Fig. 1. A multiplex network with 2 layers.

3. Preferential Attachment (PA): The preferential attachment (Newman, 2001) index calculates the score for the existence of a link between two nodes by utilizing their degree information. It is computed as below.

$$PA = |\Gamma(x)| * |\Gamma(y)| \tag{3}$$

4. Adamic–Adar Index (AA): Adamic–Adar (Adamic & Adar, 2003) uses the information of the degree of the common neighbors between the given node. Its formal definition is given below.

$$AA = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\log|\Gamma(z)|} \tag{4}$$

5. Resource Allocation Index (RAI): Resource Allocation Index (Zhou et al., 2009) also uses the common neighbor degree information, except it directly utilizes the degree magnitude in place of its logarithmic value. The equation is given below.

$$RAI = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{5}$$

6. Hub Promoted Index (HPI): HPI (Ravasz, Somera, Mongru, Oltvai, & Barabási, 2002) uses the information of the number of common neighbors and degrees of the given nodes to assign a score. It priorities the nodes adjacent to a hub, that is a high degree vertex.

$$HPI = \frac{|\Gamma(x)\cap\Gamma(y)|}{min(k_x, k_y)} \tag{6}$$

7. Hub Depressed Index (HDI): HDI (Ravasz et al., 2002) also utilizes the information of the number of common neighbors and degrees of the given nodes to assign a score. It punishes the nodes adjacent to a hub.

$$HDI = \frac{|\Gamma(x)\cap\Gamma(y)|}{max(k_x, k_y)} \tag{7}$$

8. Local Path Index (LP): LP (Lü, Jin, & Zhou, 2009) utilizes the number of paths of lengths two and three between a given node pair, to calculate the score for the presence of a link. Here, $(A^2)_{xy}, (A^3)_{xy}$ denote the number of paths of length two and three, respectively.

$$LP = (A^2)_{xy} + (A^3)_{xy} \tag{8}$$

9. Friends Measure (FM): Fire et al. (2011) developed the friends measure. FM uses the information of the number of links between the neighbors of a given node pair. Here $\delta(x, y)$ is 1, if x and y are connected, else it is 0.

$$FM = \sum_{x\in\Gamma(x)} \sum_{y\in\Gamma(y)} \delta(x, y) \tag{9}$$

10. Friendship in Auxiliary layer (FAL): Shan et al. (2020) develop the FAL method that utilizes the information of the link present in auxiliary layers. This builds up a confidence score for the link that exists between a pair of nodes in the target layer. This feature is only calculated for auxiliary layers in multiplex networks. The equation is described below. Here x, y signify a pair of nodes, i denotes the auxiliary layer being used to calculate FAL.

$$FAL(x, y, i) = \begin{cases} 1, & \text{if } (x, y) \in E_i \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

2.2.2. Features based on community

Community structure information helps in obtaining a unique perspective that brings forth the regions of dense link composition in the graph. Ahmed et al. (2016) use this perspective and present a model that increases the confidence score for link presence if both the given vertices belong to the same community. We build upon this approach by utilizing different topological features to get an accurate estimation for link presence. This also helps in removing the bias of grouping dissimilar nodes together by the community detection algorithm. Ten runs of community detection were performed using the label propagation algorithm (Raghavan, Albert, & Kumara, 2007). For each run, a similarity measure between a node pair was calculated. The final score is the average sum of all values for ten runs. The equation below represents the above framework where $f(x, y)$ is the topological feature between the nodes x, y. $f(x, y)$ is 0 if the communities of nodes x, y are different.

$$Community\_score(x, y) = \sum_{i=1}^{10} (0.1).f(x, y) \tag{11}$$

1. Ratio of same community common neighbors and total common neighbors (RACC): Valverde-Rebaza and de Andrade Lopes (2012) use the ratio of the number of common neighbors in the same community to the total number of common neighbors for efficiently predicting the links. This measure is utilized as a function between the nodes to the above mentioned framework. In the equation below z_{comm} is equal to the number of common neighbors having the same community as the node pair x, y. z is the number of total common neighbors between the node pair x, y.

$$RACC(x, y) = \sum_{i=1}^{10} (0.1).(\frac{z_{comm}}{z}) \tag{12}$$

2. Ratio of same community neighbors and total neighbors (RACN): We use the number of the neighbors of nodes x, y such that they all belong to the same community (represented by t_{comm}). This number is divided by the total number of neighbors of the given node pair (symbolized by t). The equation is shown below.

$$RACN(x, y) = \sum_{i=1}^{10} (0.1).(\frac{t_{comm}}{t}) \tag{13}$$

3. Extended CN Algorithm (ECN):
Soundarajan and Hopcroft (2012) propose an extended CN method that uses the information of the community structures to enhance link prediction. This method is used as a feature function in Eq. (11). In the equation below z_{comm} is equal to the number of common neighbors having the same community as the node pair x, y. z is the number of total common neighbors between the node pair x, y.

$$ECN = \sum_{i=1}^{10}(0.1).(z + z_{comm}) \tag{14}$$

4. Extended RAI Algorithm (ERAI):
Soundarajan and Hopcroft (2012) also present an extended RAI algorithm to increase the accuracy of the traditional RAI method. We utilize this in Eq. (11) to find the future links more accurately. In the equation below z_{comm} is equal to the number of common neighbors having the same community as the node pair x, y and k_i signifies the degree of a node i.

$$ERAI = \sum_{i=1}^{10}(0.1).(\sum_{i \in z_{comm}} \frac{1}{k_i}) \tag{15}$$

2.3. Construction of training and testing data set

To generate appropriate training and testing data set is an important requirement for supervised learning methods. We modify the data set construction technique (Lü & Zhou, 2011) for unsupervised learning algorithms to use it in the proposed supervised framework. Let E_T denote the set of edges initially present in the network. The proposed algorithm randomly divides the network's existing links into two sets having 90% (signified by E_{Ta}) and 10% (signified by E_{Tb}) of the links, respectively. Next, the algorithm removes all the edges in set E_{Tb} from the network and utilize them as ground truth labels. After that, it generates a set of all the pairs of disconnected nodes that are 2-hop away from each other. This set is denoted by E_{Tc}. This is based on the factor that 90% new links are formed between 2-hop away users (Valverde-Rebaza & de Andrade Lopes, 2012). From E_{Tc}, the algorithm randomly under samples the number of non-existent links to match the length of E_{Tb}, in order to balance the positive and negative edge data sets. Next, it appends the non-existent edges in E_{Tc} to E_{Tb} to create a data set with binary labels. Finally, the edges in E_{Tb} are split up into two sets having 80% (training data set) and 20% (testing data set) by using a 5-fold cross-validation approach. The machine learning models are trained using a training set and are subsequently evaluated by using different metrics on the test data set. Algorithm 1 shows the training and test data set construction. Furthermore, the training and test data set split is only performed for the target layer in the case of a multiplex network.

2.4. Proposed method

The first step in the proposed algorithm is to generate training and test sets as shown in Section 2.3. Next, the algorithm removes the positive edges present in the training, test set from the graph. In the case of multiplex networks, the above operations are only performed on the target layer, that is G_α. This is done to ensure that the topology-based similarity measures do not use the information of the link present in the above sets. Following this, the algorithm generate a set of features described in Section 2.2 for the node pairs in training and test data sets. The set of features are computed for both the target as well as auxiliary layers in multiplex networks. We normalize the feature-set before using them in the supervised learning classifiers. Subsequently, the proposed method train the classifiers using the binary labeled set of training data set. The method uses support vector machine (SVM), K Neighbors Classifier (KNN), C4.5 Decision Tree (DT), Artificial Neural Network (ANN), Ada Boost (ADA), Bagging Classifier (BAG), and Random Forest Classifier (RF) using scikit-learn (Buitinck, Louppe, Blondel, Pedregosa, Mueller, Grisel, Niculae, Prettenhofer, Gramfort, Grobler, Layton, VanderPlas, Joly, Holt, & Varoquaux, 2013). Finally, the labels predicted from the classifier were compared with the ground truth labels of the test data set using different performance metrics like accuracy, precision, recall, F-score, and area under the curve (AUC). Algorithms 2 and 3 delineate the complete steps for single-layered and multiplex networks, respectively.

Algorithm 1: Training and Testing data set generation

```
Input: A network $G$ or a target layer graph $G_\alpha$ in case of a multiplex network
$E_T$ ← List of all edges present in the network
$V$ ← List of nodes in the network
$E_{Tb}$ ← Randomly sample 10% of the edges in $E_T$
$E_{Ta}$ ← $E_T$ - $E_{Tb}$
$E_{Tc}$ ← A set of all the pairs of disconnected nodes that are 2-hop away from each other
$E_{Tc}$ ← Randomly under sample $E_{Tc}$ such that length($E_{Tc}$) = length($E_{Tb}$)
$E_{Tb}$ ← $E_{Tc} \cup E_{Tb}$
\ * Combine $E_{Tc}$ and $E_{Tb}$ to create a balanced data set with equal number of binary labels.* \
Construct a graph $G'$ using $V$ and $E_{Ta}$
\ *Modified graph on which feature extraction will be performed. This ensures that the features do not use the information from $E_{Tb}$.* \
$TrainSet, TestSet$ ← Split up $E_{Tb}$ into two sets having 80% (Train set) and 20% (Test set) by using 5-fold cross-validation.
return $G', TrainSet, TestSet$
```

Algorithm 2: Proposed Method for Single Layer Networks

```
Input: A network $G_s$
$E_T$ ← List of all edges present in the network
$V$ ← List of nodes in the network
$G'_s, Train\_Set, Test\_Set$ ← Generate Training and Testing Data Set using $G_s, E_T, V$
\ * $G'_s$ is the modified graph on which feature extraction will be performed. This ensures that the features do not use the information from training and testing data set.* \
$Features$ ← Compute features for node pairs in training and testing sets
$Features$ ← Normalize the features
Train various classifiers using $Train\_Set$
$Predicted\_Values$ ← Compute the classifier scores for data in $Test\_Set$
$Scores$ ← Calculate the scores of different classifiers using evaluation metrics, $Predicted\_Values$ and ground-truth labels
return $Scores$
```

3. Results

This section presents the details of the data sets and the performance of our proposed work by utilizing various metrics.

3.1. Experimental settings

As described in Section 2.4, we use different machine learning classifiers like SVM, KNN, DT, ANN, ADA, BAG, and RF. Most of the parameters utilized in these algorithms are set to their default values except for a few. The number of neighbors required for each sample in KNN is set to 20, and the minimum number of samples at the leaf

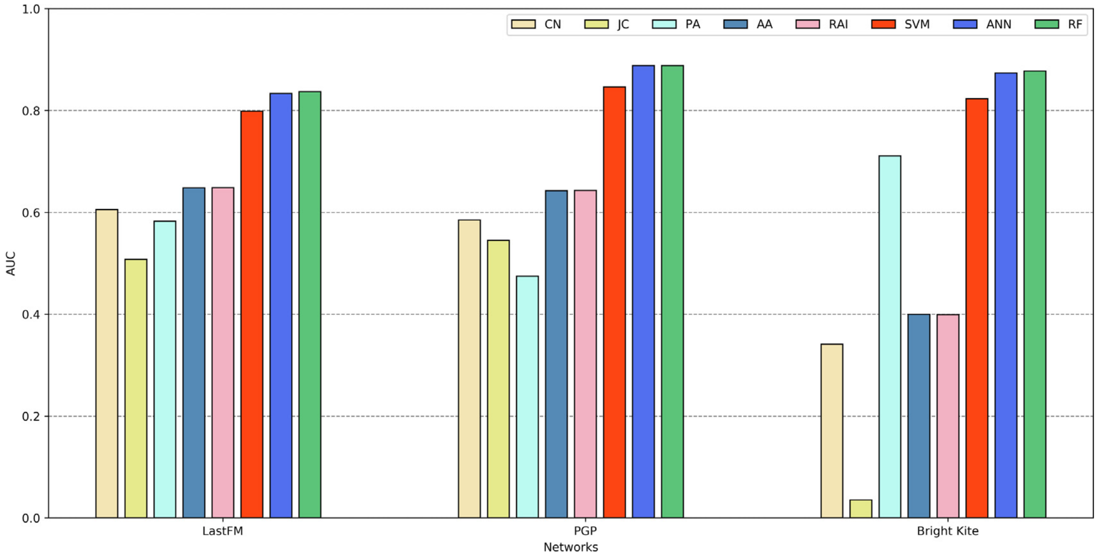

Fig. 2. Performance comparison between unsupervised and supervised learning methods.

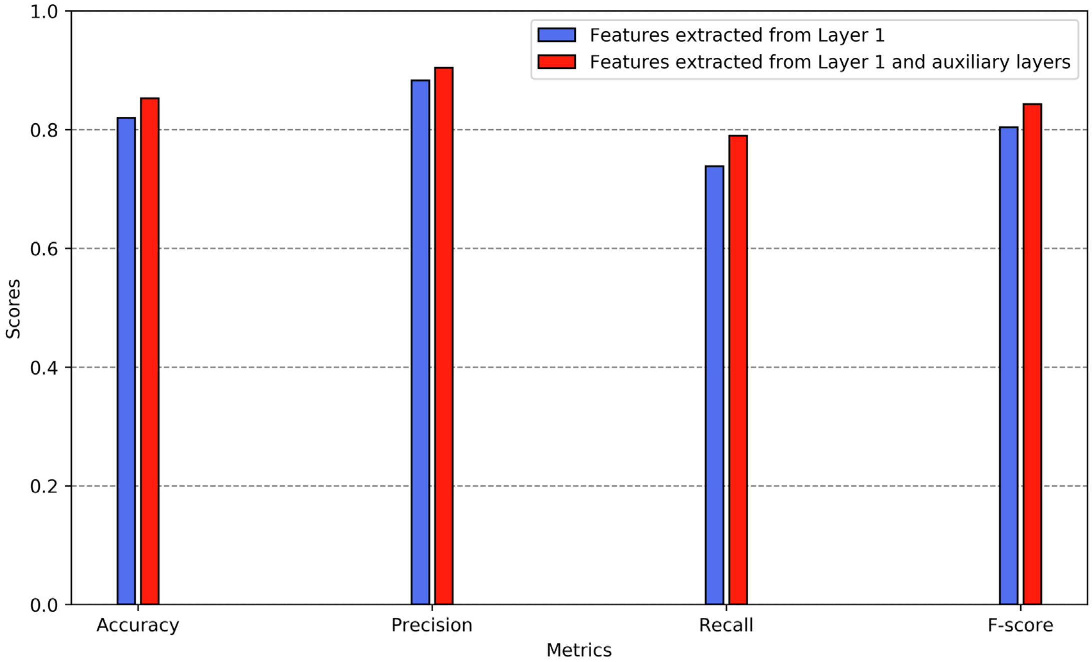

Fig. 3. Results of the proposed method with features from Layer 1 or all the layers.

node in DT, RF is set to 10. For ADA, DT is used as the base classifier with the parameter mentioned above, while for BAG, we utilized the SVM classifier with default parameters. Apart from the classification task, we also use a 5-fold cross-validation method and evaluate each algorithm for a total of five runs on every data set. The average of all the five values thus obtained is taken to determine a final score. The evaluation metrics used in this study for obtaining these scores are accuracy, precision, recall, F-score, and area under the curve (AUC).

3.2. Data sets

In order to evaluate the performance of our proposed methods, we use various real-world network data sets. The *Hamsterster friends* (Kunegis, 2013) network contains friendship links between the users of hamsterster.com. *LastFM* (Rozemberczki & Sarkar, 2020) is a social network that consists of users from Asian countries and depicts the relationship between them. The users and their interactions in an algorithm for information exchange constitute the *Pretty-Good-Privacy* (PGP) network (Boguñá, Pastor-Satorras, Díaz-Guilera, & Arenas, 2004; Kunegis, 2013). The co-authorship relations between the authors of the preprints on the Astrophysics E-Print Archive make up the *Astrophysics* network (Kunegis, 2013; Newman, 2011). The mutual connections between the various official Facebook pages is modeled by the *Facebook* network (Rozemberczki, Allen, & Sarkar, 2019). The *Caida* network (Kunegis, 2013; Leskovec, Kleinberg, & Faloutsos, 2007) is composed of the autonomous systems that are connected with each other over the internet. *Condensed matter* (con) is a co-authorship network (Kunegis, 2013; Newman, 2011) for the papers uploaded to the condensed matter section of arXiv. *Brightkite* is a location-based social network (Cho, Myers, & Leskovec, 2011) where users can exchange their locations by checking in. Apart from the above-mentioned single-layer real-world networks, we also use a multiplex network for our proposed algorithm. The *twitter and foursquare* (TF) network (Jalili,

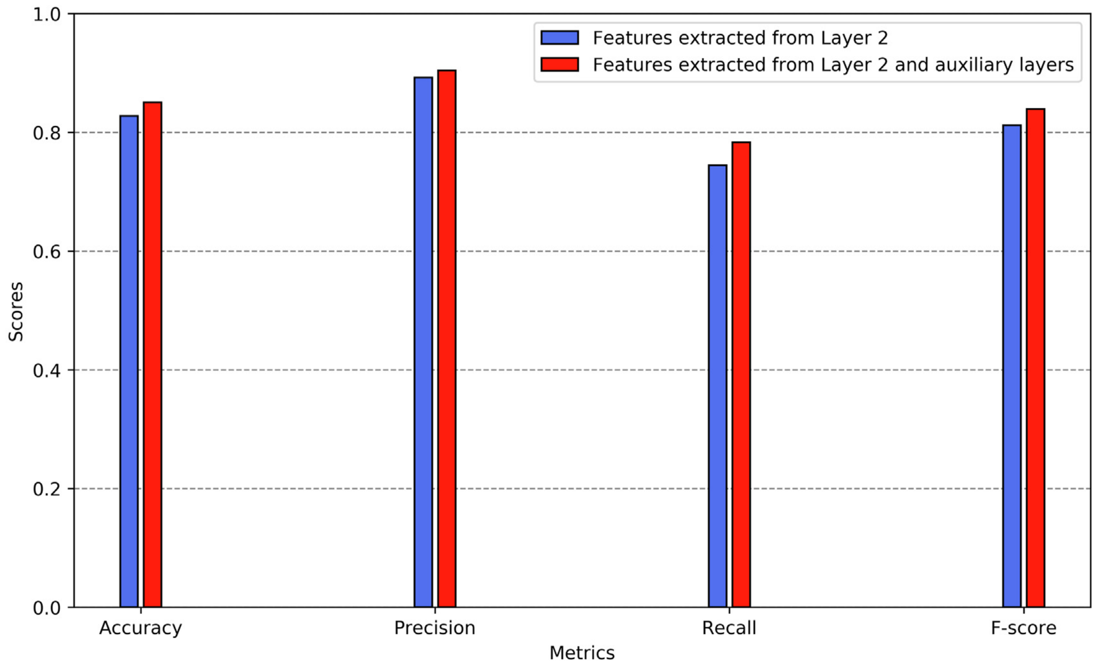

Fig. 4. Results of the proposed method with features from Layer 2 or all the layers.

Table 2
Description of the networks used in the experiments.

Name	Acronym	Nodes	Edges	Type	Avg. Degree	Avg. CC
Hamsterster friends	Ham	2426	16 631	Single Layer	13.71	0.53
LastFM	Lfm	7624	27 806	Single Layer	7.29	0.21
Pretty Good Privacy	Pgp	10 680	24 316	Single Layer	4.55	0.26
Astrophysics	Ast	16 046	121 251	Single Layer	15.11	0.66
Facebook	Fac	22 470	171 002	Single Layer	15.22	0.36
Caida	Cai	26 475	53 381	Single Layer	4.03	0.21
Condensed Matter	Con	39 577	175 691	Single Layer	8.88	0.65
Brightkite	Bri	58 228	214 078	Single Layer	7.35	0.17
TF Layer - 1	TF	1564	18 471	Multiplex	23.62	0.33
TF Layer - 2	TF	1564	14 090	Multiplex	18.02	0.13

Algorithm 3: Proposed Method for Multiplex Network

```
Input: A network $G_m$ with target layer $G_\alpha$ and auxiliary layers $G_\beta$
$E_T$ ← List of all edges present in $G_\alpha$
$V$ ← List of nodes in a single layer of $G_m$
$G_\alpha', Train\_Set, Test\_Set$ ← Generate Training and Testing Data Set using $G_\alpha, E_T, V$
\ * $G_\alpha'$ is the modified target layer. This ensures that the features do not use the information from training and testing data set. * \
$Features$ ← Compute features for node pairs in training and testing sets using $G_\alpha'$ and $G_\beta$
\ * FAL feature is only calculated using $G_\beta$ * \
$Features$ ← Normalize the features
Train various classifiers using $Train\_Set$
$Predicted\_Values$ ← Compute the classifier scores for data in $Test\_Set$
$Scores$ ← Calculate the scores of different classifiers using evaluation metrics, $Predicted\_Values$ and ground-truth labels
return $Scores$
```

Orouskhani, Asgari, Alipourfard, & Perc, 2017) comprises the same users that have different interactions on the given social networking platforms. Henceforth, the graph model of this system is best represented by a two-layer multiplex network. The details of the networks as mentioned above are presented in Table 2.

3.3. Performance evaluation on single layer networks

Table 3 presents the comparison of the seven supervised learning classifiers utilized in our work. Four performance metrics, namely, accuracy, precision, recall, and f-score, have been used for the evaluation. We take eight different real-world networks and train, test each of the classifiers using the 5-fold cross-validation technique. This method generates five scores which are then averaged to compute a final score for each evaluation metric on every network. Every column in Table 3 shows the performance of a single classifier, and each row represents the evaluation metric scores of different classifiers on a particular network. The bold values signify the highest score of a classifier in a given row. Random forest and Artificial Neural Network achieve the best accuracy, precision, recall, and f-score values in most cases. RF achieves highest scores for all four evaluation metrics on *Cai, Con, Bri* networks (0.9551, 0.9572, 0.9528, 0.9550 are the scores on *Cai*). ANN produces superior accuracy, precision values on *Ham, PGP, Facebook*

Table 3
Performance comparison on single layer networks.

Data Set	Algorithm	SVM	KNN	DT	ANN	ADA	BAG	RF
Hamsterster	Accuracy	0.8308	0.8272	0.8476	**0.8681**	0.8549	0.8296	0.8633
	Precision	0.8837	0.8633	0.8554	**0.8982**	0.8543	0.8849	0.8667
	Recall	0.7621	0.7778	0.8375	0.8308	0.8559	0.7580	**0.8591**
	F-score	0.8182	0.8182	0.8460	**0.8628**	0.8549	0.8164	0.8627
LastFM	Accuracy	0.7985	0.7911	0.8094	0.8333	0.8060	0.7990	**0.8367**
	Precision	0.8827	**0.8983**	0.8308	0.8729	0.8200	0.8850	0.8751
	Recall	0.6892	0.6570	0.7781	0.7807	0.7851	0.6880	**0.7864**
	F-score	0.7738	0.7587	0.8032	0.8241	0.8017	0.7739	**0.8281**
PGP	Accuracy	0.8466	0.8464	0.8703	**0.8884**	0.8682	0.8479	0.8882
	Precision	0.9226	0.9232	0.8838	**0.9305**	0.8739	0.9283	0.9131
	Recall	0.7574	0.7557	0.8533	0.8398	**0.8605**	0.7546	0.8584
	F-score	0.8317	0.8310	0.8680	0.8827	0.8671	0.8323	**0.8847**
Astrophy	Accuracy	0.9203	0.9231	0.9400	0.9447	0.9419	0.9207	**0.9479**
	Precision	0.9431	0.9382	0.9456	**0.9516**	0.9453	0.9433	0.9497
	Recall	0.8945	0.9059	0.9338	0.9371	0.9381	0.8953	**0.9458**
	F-score	0.9181	0.9217	0.9397	0.9443	0.9417	0.9187	**0.9478**
Facebook	Accuracy	0.8026	0.8296	0.8468	**0.8727**	0.8541	0.8033	0.8722
	Precision	0.8722	0.8666	0.8566	**0.8809**	0.8567	0.8741	0.8785
	Recall	0.7091	0.7793	0.8332	0.8622	0.8506	0.7087	**0.8640**
	F-score	0.7822	0.8206	0.8447	**0.8714**	0.8536	0.7828	0.8711
Caida	Accuracy	0.9320	0.9291	0.9490	0.9448	0.9481	0.9315	**0.9551**
	Precision	0.9245	0.9439	0.9551	0.9513	0.9475	0.9247	**0.9572**
	Recall	0.9409	0.9127	0.9421	0.9376	0.9486	0.9396	**0.9528**
	F-score	0.9326	0.9280	0.9485	0.9444	0.9480	0.9321	**0.9550**
Con	Accuracy	0.9066	0.8951	0.9050	0.9173	0.9082	0.9065	**0.9207**
	Precision	0.9211	0.9225	0.9149	0.9233	0.9140	0.9210	**0.9257**
	Recall	0.8893	0.8626	0.8931	0.9103	0.9012	0.8893	**0.9149**
	F-score	0.9049	0.8915	0.9039	0.9167	0.9075	0.9049	**0.9202**
Brightkite	Accuracy	0.8232	0.8362	0.8516	0.8737	0.8568	0.8235	**0.8775**
	Precision	0.8958	**0.9273**	0.8653	0.8967	0.8664	0.8959	0.8979
	Recall	0.7314	0.7295	0.8329	0.8450	0.8437	0.7321	**0.8520**
	F-score	0.8053	0.8166	0.8488	0.8700	0.8549	0.8058	**0.8743**

data sets (0.8681, 0.8982 are the accuracy, precision value respectively on *Ham*). Apart from them, KNN achieves the best precision values on *Lfm, Bri* networks (0.8983 and 0.9273, respectively). The other supervised learning methods produce comparative if not better scores on different networks (0.9419, 0.9453 are accuracy, precision value respectively by ADA on *Ast* network). Overall the proposed feature design produces better performance with various supervised learning algorithms on different network data sets. This shows the effectiveness of feature set in providing highly useful information.

We also compare the area under the curve (AUC) values of supervised and unsupervised methods on three networks, namely *LastFM, PGP* and *Brightkite* in Fig. 2. CN, JC, PA, AA, RAI are the five unsupervised learning algorithms, while SVM, ANN, and RF are the three supervised learning methods utilized for this experiment. The training and test data for unsupervised methods are generated as stated in Algorithm 1. However, here the edges in the training data set are added back to the network. This is done because unsupervised learning methods use only the network topology information and hence do utilize the training data set like the machine learning classifiers. In Fig. 2, SVM, ANN, and RF achieve far superior AUC scores on all three networks. The unsupervised algorithms achieve lower and unpredictable scores. For example, PA achieves the highest AUC value among the unsupervised methods on *Brightkite* network, while it has the lowest value on *PGP* network. This experiment demonstrates the limitation of such structural similarity methods as they use only a few features to compute values. On the other hand, supervised learning classifiers use a combination of topological features, thereby attaining a much larger perspective and subsequently higher scores. Finally, the better performance of our technique could be attributed to the contribution of community based features, which are missing in the unsupervised learning methods.

Table 4
Performance comparison on multiplex network.

Algorithm	Layer	Accuracy	Precision	Recall	F-score
SVM	1	0.8531	0.9044	0.7897	0.8431
	2	0.8504	0.9047	0.7830	0.8394
	Average	0.8517	0.9045	0.7864	0.8412
KNN	1	0.8474	0.8820	0.8023	0.8402
	2	0.8486	0.8902	0.7951	0.8397
	Average	0.8480	0.8861	0.7987	0.8400
DT	1	0.8385	0.8552	0.8150	0.8342
	2	0.8560	0.8639	0.8446	0.8540
	Average	0.8473	0.8595	0.8298	0.8441
ANN	1	0.8634	0.8694	0.8553	0.8622
	2	0.8794	0.8999	0.8540	0.8761
	Average	**0.8714**	0.8847	0.8547	0.8692
ADA	1	0.8474	0.8580	0.8331	0.8450
	2	0.8521	0.8579	0.8440	0.8507
	Average	0.8498	0.8579	0.8386	0.8479
BAG	1	0.8536	0.9060	0.7891	0.8435
	2	0.8511	0.9067	0.7823	0.8398
	Average	0.8523	**0.9064**	0.7857	0.8416
RF	1	0.8642	0.8601	0.8696	0.8647
	2	0.8752	0.8755	0.8755	0.8753
	Average	0.8697	0.8678	**0.8726**	**0.8700**

3.4. Performance evaluation on multiplex network

Table 4 shows the scores obtained by supervised learning classifiers on the *Twitter Foursquare* (TF) multiplex network data set. Every row of the table presents the algorithm utilized for link prediction using a

Table 5
Information gain value of different features.

Feature	Data Set							
	Hamsterster	LastFM	PGP	Astrophy	Facebook	Caida	Con	Brightkite
CN	0.2346	0.2547	0.3231	0.3672	0.2723	0.3487	0.2823	0.3195
JC	0.3496	0.2953	0.3815	0.4605	0.3267	0.5535	0.3231	0.3232
PA	0.2808	0.1492	0.2107	0.1234	0.1251	0.4033	0.0754	0.1447
AA	0.6024	0.4909	0.5313	0.6304	0.5705	0.4521	0.5560	0.5031
RAI	0.5990	0.4877	0.5311	0.6297	0.5684	0.4518	0.5532	0.5024
HPI	0.2814	0.2509	0.3336	0.4359	0.2984	0.3572	0.3369	0.2923
HDI	0.3230	0.2793	0.3737	0.4389	0.3100	0.5422	0.2980	0.3124
FM	0.2232	0.1019	0.1273	0.2005	0.1374	0.1778	0.0980	0.0953
RACC	0.0374	0.0582	0.0930	0.2240	0.1083	0.2200	0.1106	0.1206
RACN	0.3010	0.3093	0.2442	0.4055	0.3683	0.3707	0.2797	0.3227
ECN	0.2558	0.2863	0.3686	0.4049	0.3037	0.3962	0.3124	0.3356
ERAI	0.5527	0.3815	0.3306	0.5462	0.4545	0.3339	0.5020	0.3849
LP	0.5789	0.4311	0.4627	0.5854	0.4670	0.5264	0.4856	0.4250

given target layer. The remaining layers were used as auxiliary layers for extracting additional information according to Algorithm 3. The 5-fold cross-validation technique was used for training and test data sets division, and the average of the values from all the runs was calculated. Finally, the mean of the layer 1 and layer 2 scores for each evaluation metric was computed to represent the overall score for each algorithm on different evaluation metrics. Columns in Table 4 are used to compare scores obtained by different classifiers using a particular evaluation metric. The bold values represent the highest overall average score achieved by a given classifier. RF algorithm obtains the highest recall and f-score values, i.e., 0.8726 and 0.8700, respectively. BAG and ANN classifiers achieve the best precision (0.9064) and accuracy scores (0.8714), respectively. KNN and DT methods achieve lower but comparative scores with respect to the above-mentioned classifiers (0.8473 accuracy score by DT and 0.7987 recall score by KNN). It is observed that all the classifiers achieve greater than 80% accuracy and precision scores. This shows that the features extracted from different layer provide a wide range of data for the machine learning classifier to achieve a superior performance.

We also conduct an experiment to evaluate the benefits of utilizing the auxiliary layer information, in addition to the information from the target layer. In this experiment, we calculate the overall score obtained by the proposed method with or without the use of auxiliary layers on four evaluation metrics. The scores are computed on a single classifier, i.e., SVM, to maintain consistency across the evaluation metrics. Fig. 3 shows a comparison between the scores calculated by using information only from layer 1 (target layer) and the combined information of all the layers. Similarly, Fig. 4 shows a similar comparison by taking layer 2 as the target layer. It is evident from the figures that the combined information from all the layers achieves higher performance in both cases on various evaluation metrics. Therefore, the topological features extracted from auxiliary layers play a vital role in improving the link prediction algorithm.

3.5. Information gain from feature set

Table 5 presents the information gain values for different features. This experiment was performed to indicate the usefulness of various features for the supervised learning classifiers. It is observed that AA, RAI, ERAI, and LP provide the maximum amount of information to the classifiers in majority of the networks (0.5527 by ERAI, 0.5789 by LP, 0.6024 by AA are the higher information gain scores obtained on *Ham* network). Apart from them, ECN, RACN, HDI, and JC also provide adequate data to the machine learning algorithms (0.4055, 0.4049 are the information gain scores obtained by RACN, ECN respectively on *Astrophy* network). Although RACC, PA attain low information gain scores, they are vital in some cases. It can be clearly deduced that three out of the four proposed community detection based features provide better information on various data sets. Therefore, the combination of topological features used in this study is highly suitable for efficiently finding future links in complex networks.

4. Discussion

The main motive of this study was to develop a computationally efficient set of features that can be utilized to find links in complex networks with high accuracy. This was accomplished by utilizing appropriate structural similarity techniques and proposing four community detection based topological methods to construct an effective feature set. Extensive experimentation was conducted by using different supervised learning techniques on a variety of network data sets. The results obtained show high accuracy; this is mainly because of the wide perspective of information gathered efficiently by the proposed algorithms. Furthermore, the training and testing framework utilized in this study helped in experimenting on networks with unknown ground truth link structure. Finally, this study also contributes towards a unified framework model is developed that can be executed on single-layer and multiplex networks with minor modifications.

5. Conclusion

In this study, a technique is presented to predict the future links in single-layer and multiplex networks. We construct an appropriate set of topological features and utilize them for the supervised learning approaches. The information from various layers is utilized in the multiplex network to improve the algorithm performance. Furthermore, we construct a framework for creating the training and testing data sets to evaluate the algorithm's performance. The results show the high accuracy (greater than 80% on the majority of the networks) and consistency of our method on various real-world network data sets. The future scope of the current exposition includes incorporating content-based and time-based features for finding links in dynamic networks, which could include edge and node attributes in the present framework to improve accuracy measures.

CRediT authorship contribution statement

Deepanshu Malhotra: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Rinkaj Goyal:** Conceptualization, Validation, Resources, Writing - review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, *25*(3), 211–230.

Ahmed, C., ElKorany, A., & Bahgat, R. (2016). A supervised learning approach to link prediction in Twitter. *Social Network Analysis and Mining*, *6*(1), 24.

Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006). Link prediction using supervised learning. In *SDM06: Workshop on link analysis, counter-terrorism and security (vol. 30)* (pp. 798–805).

Albert, R., Jeong, H., & Barabási, A. L. (1999). Diameter of the world-wide web. *Nature*, *401*(6749), 130–131.

Altan, A., & Karasu, S. (2019). The effect of kernel values in support vector machine to forecasting performance of financial time series. *The Journal of Cognitive Systems*, *4*(1), 17–21.

Altan, A., & Karasu, S. (2020). Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique. *Chaos, Solitons & Fractals*, *140*, Article 110071.

Altan, A., Karasu, S., & Bekiros, S. (2019). Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. *Chaos, Solitons & Fractals*, *126*, 325–336.

Altan, A., Karasu, S., & Zio, E. (2021). A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Applied Soft Computing*, *100*, Article 106996.

Altan, A., & Parlak, A. (2020). Adaptive control of a 3D printer using whale optimization algorithm for bio-printing of artificial tissues and organs. In *2020 innovations in intelligent systems and applications conference* (pp. 1–5). IEEE.

Boccaletti, S., Bianconi, G., Criado, R., del Genio, C., Gómez-Gardeñes, J., Romance, M., et al. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, *544*(1), 1–122, The structure and dynamics of multilayer networks.

Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., & Arenas, A. (2004). Models of social networks based on social distance attachment. *Physical Review E*, *70*, Article 056122.

Boss, M., Elsinger, H., Summer, M., & 4, S. T. (2004). Network topology of the interbank market. *Quantitative Finance*, *4*(6), 677–684.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., et al. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD workshop: languages for data mining and machine learning* (pp. 108–122).

Cheng, H. M., Ning, Y. Z., Yin, Z., Yan, C., Liu, X., & Zhang, Z. Y. (2018). Community detection in complex networks using link prediction. *Modern Physics Letters B*, *32*(01), Article 1850004.

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1082–1090). New York, NY, USA: Association for Computing Machinery.

Daud, N. N., Ab Hamid, S. H., Saadoon, M., Sahran, F., & Anuar, N. B. (2020). Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, *166*, Article 102716.

Doye, J. P. K. (2002). Network topology of a potential energy landscape: A static scale-free network. *Physical Review Letters*, *88*, Article 238701.

Fire, M., Tenenboim, L., Lesser, O., Puzis, R., Rokach, L., & Elovici, Y. (2011). Link prediction in social networks using computationally efficient topological features. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing* (pp. 73–80).

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, *37*, 241–272.

Jalili, M., Orouskhani, Y., Asgari, M., Alipourfard, N., & Perc, M. (2017). Link prediction in multiplex online social networks. *Royal Society Open Science*, *4*(2), Article 160863.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., & Barabási, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, *407*(6804), 651–654.

Kagan, D., Elovichi, Y., & Fire, M. (2018). Generic anomalous vertices detection utilizing a link prediction algorithm. *Social Network Analysis and Mining*, *8*(1), 1–13.

Karasu, S., Altan, A., Bekiros, S., & Ahmad, W. (2020). A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. *Energy*, *212*, Article 118750.

Kinsley, A. C., Rossi, G., Silk, M. J., & VanderWaal, K. (2020). Multilayer and multiplex networks: An introduction to their use in veterinary epidemiology. *Frontiers in Veterinary Science*, *7*, 596.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, *2*(3), 203–271.

Kumari, A., Behera, R. K., Sahoo, K. S., Nayyar, A., Kumar Luhach, A., & Prakash Sahoo, S. Supervised link prediction using structured-based feature extraction in social network. Concurrency and Computation: Practice and Experience, e5839.

Kunegis, J. (2013). KONECT: The Koblenz network collection. In *Proceedings of the 22nd international conference on world wide web* (pp. 1343–1350). New York, NY, USA: Association for Computing Machinery.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, *1*(1), 2–es.

Li, J., Zhang, L., Meng, F., & Li, F. (2014). Recommendation algorithm based on link prediction and domain knowledge in retail transactions. *Procedia Computer Science*, *31*, 875–881, 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014.

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, *58*(7), 1019–1031.

Lü, L., Jin, C. H., & Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, *80*, Article 046122.

Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, *390*(6), 1150–1170.

Mandal, H., Mirchev, M., Gramatikov, S., & Mishkovski, I. (2018). Multilayer link prediction in online social networks. In *2018 26th telecommunications forum* (pp. 1–4).

Manisha Pujari, R. K. (2015). Link prediction in multiplex networks. *Networks & Heterogeneous Media*, *10*(1), 17–35.

Newman, M. E. J. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, *64*, Article 025102.

Newman, M. E. (2011). The structure of scientific collaboration networks. In *The structure and dynamics of networks* (pp. 221–226). Princeton University Press.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*, Article 036106.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, *297*(5586), 1551–1555.

Rozemberczki, B., Allen, C., & Sarkar, R. (2019). Multi-scale attributed node embedding. arXiv preprint arXiv:1909.13021.

Rozemberczki, B., & Sarkar, R. (2020). Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1325–1334). New York, NY, USA: Association for Computing Machinery.

Shan, N., Li, L., Zhang, Y., Bai, S., & Chen, X. (2020). Supervised link prediction in multiplex networks. *Knowledge-Based Systems*, *203*, Article 106168.

Sharma, S., & Singh, A. (2015). An efficient method for link prediction in complex multiplex networks. In *2015 11th international conference on signal-image technology internet-based systems* (pp. 453–459).

Soundarajan, S., & Hopcroft, J. (2012). Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st international conference on world wide web* (pp. 607–608). New York, NY, USA: Association for Computing Machinery.

Travers, J., & Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, *32*(4), 425–443.

Valverde-Rebaza, J. C., & de Andrade Lopes, A. (2012). Link prediction in complex networks based on cluster information. In *Brazilian symposium on artificial intelligence* (pp. 92–101). Springer.

Yao, Y., Zhang, R., Yang, F., Yuan, Y., Sun, Q., Qiu, Y., et al. (2017). Link prediction via layer relevance of multiplex networks. *International Journal of Modern Physics C*, *28*(08), Article 1750101.

Zhang, S., & Lv, Q. (2018). Hybrid EGU-based group event participation prediction in event-based social networks. *Knowledge-Based Systems*, *143*, 19–29.

Zhou, T., Lü, L., & Zhang, Y. C. (2009). Predicting missing links via local information. *The European Physical Journal B*, *71*(4), 623–630.