Journal of Information Hiding and Multimedia Signal Processing ©2020 ISSN 2073-4212
Ubiquitous International Volume **11**, Number **4**, December **2020**

Improve the Accuracy of Link Predictions on Sparse Networks based on Similarity Measures and Multiple Ensemble Learning

Dzung Pham Thi Kim

Electric Power University of the Vietnam Ministry of Industry and Trade
235 Hoang Quoc Viet, Co Nhue, Tu Liem, Hanoi, Vietnam
zungptk@epu.edu.vn

Received June 2020; revised September 2020

Abstract. *In recent years, link prediction is the research direction that has attracted a lot of attention in the field of social network analysis. The real networks such as Facebook, Deezer, and DBLP have different properties and structures that affect the accuracy of link predictions. In this study, we aim to improve the accuracy of link predictions on these networks relating to their sparsity property. To do this, we analyzed the properties of the social networks, thereby building a similarity measure with node-based approach and proposing the application of a multiple ensemble learning algorithm on topology-based similarity features. The multiple ensemble learning is built from single classifiers or ensemble learning models and uses voting mechanisms to summarize the final predictive result. The experiments conducted on the social networks show that multiple ensemble learning models provide higher predictive efficiency than the existed ensemble learning models and basic classification algorithms such as Support Vector Machine, Logistic Regression, and Artificial Neural Networks.*

Keywords: Link prediction, Topology-based similarity feature, Multiple ensemble learning.

1. **Introduction.** A social network can be visualized as a graph in which each vertex is a node and each edge connecting the two vertices represents a certain form of the link between the two nodes [19]. These links can be formed based on common interests, friendships or relationships. Social networks are dynamic because their structures and characteristics change over time. There are many research problems related to learning and exploring social networks such as community detection, link prediction and network structure development [22]. In addition, narrower issues are also mentioned such as feature extraction [14], data visualization [8] , user mapping [24], etc.

In recent years, link prediction is the research direction that has attracted a lot of attention in the field of social network analysis and plays an important role in many areas such as terrorist prediction, collaborative filtering, disease transmission modeling, fashion and product promotion, virus distribution, etc. [22]. The link prediction problem is generally defined as predicting the ability between two nodes of a graph to be linked to each other in the future, while knowing that there is no link between them at the present time [18]. According to [19], the link prediction problem is summarized and classified into two main approaches: (1) similarity-based and (2) learning-based. From a network graph, using similarity measures the features of the edges are extracted and can be used for both approaches. For the approach (1), all of the similarity measure values of the edges after calculating are ranked in descending order, where the greater the value is, the higher the likelihood of the link between the two corresponding vertices. In this case, the top-k values will be selected as the predicted values. With the approach (2) similarity features and other additional features are chosen as the dataset for prediction. The binary classification problem is defined for this approach by classifying potential links (positive labels) and non-existent links (negative labels) in the dataset.

Following the approach based on learning models (2), Hasan and el. in [9] analyzed the efficiency of different popular classification algorithms on a co-authored network dataset, using basic similarity measures. There are many classification algorithms for supervised learning such as Support Vector Machine (SVM), Logistic Regression, Decision Trees, K-nearest neighbors (k-NN), Multi-layer Perceptron, Naive Bayes and Artificial Neural Networks (ANN) which are applied to solve link prediction problem. However, the accuracy of prediction depends heavily on the properties or the data domain of each social network, e.g., network density, average clustering index, average degree of nodes or specific attributes of each node in different types of network graph. The authors in [9] also indicated that the SVM algorithm is more efficient than other classification methods in terms of link prediction performance when was experimented with DBLP and BIOBASE datasets. Similarly, [3] adopted a supervised learning approach to Twitter network. The authors have compared the basic classification algorithms to the already developed ensemble learning models such as Bagging, RotationForest, AdaBoost, Bagging, RandomForest, etc. However, the authors have only used existing ensemble learning models and applied them on a single network, but have not analyzed why these models bring better results for link prediction and how they are affected by different datasets. In addition to the field of link prediction, these basic classification algorithms and combined learning models are also widely used in solving various problems such as: hyperspectral image recognition using SVM [11], driving behavior analysis based on AdaBoost algorithms [5], etc.

Ensemble learning is considered as a solution for machine learning problems to improve the predictive performance of a single learning model by training many learning models simultaneously and combining predictive results from them [15]. The advantage of ensemble learning has been proven to be able to improve accuracy in some domains, such as speech recognition, health prediction, ect. [16, 17]. In recent years, many studies on ensemble learning methods, such as XGboost, Random forest, Adaboost and Bagging have been published widely and become popular. Sagi et al. [15] also indicated that some challenges to machine learning algorithms, such as class imbalance, drifting of content over time, or increasing the number of features can be solved by an ensemble learning approach. In fact, the social networks we analyze in this paper are sparse, such as the Deezer network and DBLP co-authoring network, which leads to a greater number of negative data points than positive ones. This also causes problems for link prediction when most of the elements of the corresponding adjacency matrix of a social network are zero [10, 20, 23]. Therefore, ensemble learning is seen as an effective solution for link prediction problems.

To gain a higher predictive accuracy, we first selected empirically the best basic classification algorithms and ensemble learning models, from which we build a set of multiple ensemble learning and evaluate final results by voting algorithms. The voting mechanisms have also been shown to be useful to increase the accuracy of ensemble learning models [17].

In this paper, we focus on improving the accuracy of link prediction in the two approaches (1) and (2) mentioned above. Our research has the following contributions:

- Analyze the different properties of three social networks Facebook, Deezer and DBLP.
- Develop a similarity measure, called Common Attribute Coefficient (CAC), that represents the binding between the common attributes of two nodes in a social network, thereby extracting the corresponding values as the topology-based similarity features of the edges.
- Experiment on Deezer network, compare and evaluate the predicted effectiveness based on different features extracted for edges (including CAC). Use two prediction methods that are based on rankings and machine learning algorithms.
- Propose the application of a multiple ensemble learning algorithm to improve the accuracy of link prediction.
- Experiment, compare and evaluate results among the single classifiers, ensemble learning and multiple ensemble learning models.

The next section of this paper is structured as follows. Section 2 presents related studies. Section 3 describes CAC similarity measure and the multiple ensemble learning algorithm. Experiments and evaluations on different social networks are presented in Section 4. Finally, conclusion and future research directions are summarized in Section 5.

2. **Related Work.** Let a social network be represented by the graph $G =< V, E >$, where each edge $e =< u, v >$ of E represents a relationship between u and v over time t. The link prediction problem determines whether two nodes without edges at time t are likely to join at time t'. In our experiment, each edge is defined as a data point, i.e., a co-authoring relationship in DBLP dataset or a friend relationship in the datasets of Facebook and Deezer. Corresponding to G, the adjacency matrix $A = \big((a_{ij})\big)_{nxn}$ is defined as follows: a_{ii} equals 0 with $\forall i = 1, 2..., n$ and a_{ij} denotes whether there is a relationship between node i and j, $a_{ij} \in \{0, 1\}$ with $i, j = 1, 2..., n$. Currently, there is no exact definition of whether a network is sparse or not. However, a graph is considered sparse with the implication that its corresponding adjacency matrix is sparse, namely that most of the elements of the matrix A equal zero [6]. Therefore, in this paper we briefly analyze the properties of the three networks Facebook, Deezer and DBLP and compare their sparsity properties.

2.1. **Similarity measures.** To prepare data sets for link prediction problems, using similarity measures the values between node pairs are calculated with two kinds of metrics: node-based and topology-based. For node-based metrics, a similarity measure of each unlinked pair (u, v), whose value is score (u, v), represents the correlation between two nodes u and v. The higher the score, the higher the probability that u and v will be linked in the future, and if the score is low, the likelihood of u and v will have no link.

In fact, each node can be a user whose attributes are personal information according to an online social network or an author whose articles have been published in a co-authoring network. Because most of the node attribute values are textual, many measures have been created to represent text-based similarity as mentioned in [7, 13] or based on the classification tree model to calculate the distance between keywords in the text [4]. The node-based measures reflect personal or social relationships between the pairs of nodes, so it's useful when many attributes and activities of users can be collected. The weakness of this type of measure is that it depends on the type of different attributes and activities of the nodes in each social network. This causes an issue that the formulas for calculating the similarity metrics will have to be changed accordingly.

The topology-based similarity takes a different approach compared to the node-based because it relies on the network graph structure to calculate features for each pair of nodes [12] and more widely used. To represent the popular topology-based measures, we use some common notations. Let $\Gamma(u)$ be the neighbors of node u and $|\Gamma(u)|$ be the number of neighbors or the degree of node u.

Common Neighbor (CN): The similarity measure CN is defined as the number of nodes that both u and v have the direct link, called mutual friends. A large number of mutual friends indicate whether u and v will be able to connect in the future.

$$CN(u,v) = |\Gamma(u) \cap \Gamma(v)| \tag{1}$$

Jaccard Coefficient (JC): The coefficient JC is constructed by standardizing the number of mutual friends between the nodes u and v, and is calculated using the following formula.

$$JC(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{2}$$

Resource Allocation (RA): The similarity measure RA is defined to represent resource allocation in network structure [25] and is calculated by the following formula.

$$RA(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{|\Gamma(z)|} \tag{3}$$

Adamic-Adar Coefficient (AA): The coefficient AA is a widely used similarity measure and is calculated regarding the number of neighbors of each mutual friend between the two nodes u and v [1]. This coefficient indicates that if the mutual friends between u and v have few friends, the link weight between the two nodes will be large, and vice versa if these mutual friends have many friends, the link weight will be low. The formula for coefficient AA is written as follows:

$$AA(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log|\Gamma(z)|} \tag{4}$$

Preferential Attachment (PA): The PA similarity measure is defined that new links are likely to form for high degree nodes rather than low degree ones.

$$PA(u,v) = |\Gamma(u) \cdot \Gamma(v)| \tag{5}$$

FriendTNS: The coefficient FriendTNS [18] is a combination of improved Jaccard coefficient and the result of multiplying similarity values between the nodes on the shortest path of two non-directly-linked nodes u and u.

$$FriendTNS(u,v) = \begin{cases} \frac{1}{|\Gamma(u)|+\Gamma(v)-1} \text{ where } (u,v) \in E \\ \\ \prod_{h=1}^{k} FriendTNS(u_h, u_{h+1}) \text{ where } (u,v) \notin E \end{cases} \tag{6}$$

In addition to the basic measures above, the authors in [2] have also developed new topology-based similarity measures, such as those based on triplet motifs. From the represented formulas, it can be seen the reason why basic topology-based measures are commonly used. Because these formulas are independent, do not depend on the attributes of nodes but only on the structure of network graph, they can be applied to many different types of networks. The basic ranking on a feature dataset which is extracted following similarity measures indicates experimentally that subsets of features always play an important role in link prediction [3].

2.2. **Single Classifier.** There are many classification algorithms applied to supervised learning for link prediction problems in social networks. However, the performance of each algorithm may vary due to the specific dataset and network domain. In this paper we have conducted the experiments with single classifiers such as SVM, Logistic Regression, ANN and Decision Tree to find the best classifiers for link prediction.

2.3. **Ensemble Learning.** Ensemble learning method uses the predictions from different classifiers to improve predictive performance, which helps to avoid overfitting by making the result of a model less dependent on the dataset [15]. By combining different learning models, the search space can be expanded and thus a more suitable data space can be achieved. One of the problems with machine learning is the class imbalance where one class has significantly more samples than the other classes. This problem also happens in social networks when the number of positive (linked) edges is much greater than the number of negative (unlinked) edges, and can be solved by combining random sampling techniques and ensemble learning techniques such as Bagging, AdaBoost, XGBoost, RandomForest and GBM, thereby increasing predictive efficiency. For a dataset of n samples and m features $D = (\overrightarrow{x_i}, \overline{y_i})$ (where $|D| = n, \overrightarrow{x_i}, \in R^m, \overline{y_i} \in R$), an ensemble learning model uses an aggregate function AG to synthesize g single classifiers $\{c_1, c_2, ..., c_g\}$, aiming at predicting the following output:

$$\widehat{y_i} = \varphi(\overrightarrow{x_i}) = AG(c_1, c_2, ..., c_g) \tag{7}$$

,where $\widehat{y_i} \in R$ is for the regression problem and $\widehat{y_i} \in Z$ for the classification problem. The ensemble learning problem in general is to build a combination model that involves selecting a methodology for training the participating models and selecting an appropriate procedure for combining the outputs of single classifiers.

A notable comment is that ensemble learning methods can use many different classifiers, whereby the predictive accuracy also varies [17]. However, in the context of this paper, we do not evaluate the effectiveness of ensemble learning using different classifiers, but only focus on finding solutions for the multiple ensemble learning model from ensemble learning and the classifiers.

3. Proposed Method.

In order to improve the prediction accuracy, in this section we present our proposed method for multiple ensemble learning as below.

3.1. **Common Attribute Coefficient - CAC.** For a node-based approach, we add a measure to determine the similarity of an edge based on the attributes of each node, in which each mutual friend between the two nodes of that edge is assigned a weight based on their number of intersecting attributes. We formulate the common constraints between the attributes of two nodes x and y, called CAC, that are calculated as follows:

$$CAC(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{|Att_{uz}| + |Att_{vz}|}{|Att_{uz} \cup Att_{vz}|} \tag{8}$$

, where Att_{uz} is the intersection of two attribute sets of two nodes u and z. This coefficient is high when u, v and z have a large number of common attributes, but their total number of attributes is small. This formula is generally defined for different networks to take advantage of the information gained from the attributes of each node and enrich the features of social networks.

Assuming that with the Deezer network, each node has attributes that are the number of favorite music types. The favorite music types of node u are $\{Pop, Rock, Jazz\}$ and node v are $\{Pop, Jazz, Dance, Electro\}$, in which between u and v there is a common friend z_1 whose music hobby is $\{Pop, Film, Jazz, Dance\}$. Then, we can calculate

$Att_{uz} = \{Pop, Jazz\}$, $Att_{vz} = \{Pop, Jazz, Dance\}$, $Att_{uz} \cup Att_{vz} = \{Pop, Jazz, Dance\}$ and $CAC(u, v) = (2 + 3)/3 = 1.667$.

Formula 8 is more complex than the basic similarity formulas (see Section 2.1) because the amount of common attributes must be calculated between nodes. However, this complexity is acceptable in social network analysis. We use our proposed coefficient CAC along with other similarity measures such as CN, RA, Jaccard, AA and TNS to extract features, build empirical datasets from the structure of network graph and node attributes.

3.2. **Multiple Ensemble Learning model.** To prepare multiple ensemble learning, we divided the classifiers into three categories: single classifier, ensemble learning model and multiple ensemble learning model, in which multiple ensemble learning is built from single classifiers or the sets ensemble learning models based on predictive performance criteria and then uses voting mechanisms to summarize the final result. The pseudo-code describing how to find multiple ensemble learning models is presented in Algorithm 1.

Algorithm 1: MEL - an algorithm of multiple ensemble learning for spare social networks.

```
Input:
$G$ - a social network graph,
$s$ - number of steps between two nodes ,
$C$ - a set of learning models, where each element is a single classifier or an ensemble
learning model,
$\theta$ - sampling coefficient,
$\delta$ - threshold,
$S$ - a prediction function using $f$-fold cross-evaluation,
$par_v$ - voting parameter ('hard'/'soft').
Output:
$ES$ - a set of estimators,
$ME$ - multiple ensemble learning model,
$pMEL$ - prediction accuracy of $ME$.
begin
    Step 1: Extract a dataset from $G$.
        Take positive samples that are hidden edges: $H = \textbf{\textit{Hidden}}(G, \theta)$.
        Take negative samples that are no-linked edges: $F = \textbf{\textit{NoLinked}}(G, \theta, s)$.
        Establish a dataset: $D = H \cup F$.
        Normalize data: $D_{nor} = \textbf{Normalize}(D)$ .
    Step 2: Find the best classifies and ensemble learning models.
        Get a accuracy $acc_i$ for each $c_i \in C$ using Formula 9: $AC = \{acc_i\}$.
        Calculate an average accuracy: $m = \frac{\sum_{i=1}^{|C|} acc_i}{|C|}$ .
        Select estimators: $ES = \{c_i | acc_i > m + \delta; i = 1, 2, ...|C|\}$.
    Step 3: Arrange multiple ensemble learning.
        Set a multiple ensemble learning model: $ME = \textbf{\textit{VoteClassifier}}(ES, par_v)$.
        Calculate $pMEL = S(D_{nor}, ME, f)$.
    return $ES, ME,$ and (optional) $pMEL$.
end
```

The first step of the algorithm is to extract the dataset that contains values of the selected feature values corresponding to the similarity measures from graph G. The functions *Hidden* and *Nolinked* randomly take positive and negative samples that are hidden edges and no-linked edges of G. The sampling coefficient θ is used to adjust the number of hidden/no-linked edges taken from G, and s to change the number of steps between two nodes of no-linked edges. The samples are unionized, normalized to establish a input dataset D_{nor} that is used for all classification cases.

The selection of the best classifiers and ensemble learning models for building a multiple ensemble learning model is presented in Step 2. We independently run each element c_i of C on the dataset D_{nor} to calculate the corresponding accuracy acc_i by executing a prediction function S using f-fold cross-evaluation that is represented via Formula 9.

$$acc_i \overset{\text{def}}{=} S(D_{nor}, c_i, f), \text{where } c_i \in C \text{ and } i = 1, 2, ...|C| \tag{9}$$

The classification algorithms that are selected with the highest predictive performance are called *estimators* ES. The selection is based on the difference in predictability so that it must be large enough by comparison with the average m of the predicted results of all algorithms in C. The coefficient δ is a threshold used to adjust this selection.

In the third step, we set up a multiple ensemble learning model ME by conducting classification on dataset D_{nor} with the algorithms selected in ES and use a voting mechanism to synthesize the predictive results. The voting parameter par_v is set to one of two popular types, where 'hard' is for majority rule voting and 'soft' is on the argmax of the sums of the predicted probabilities. The final output of the algorithm MEL includes ES, ME and $pMEL$ (the prediction accuracy of ME).

4. **Experiment and Evaluation.** In this section, we describe how to prepare the datasets for our experiments and explain the results obtained from the empirical cases.

Datasets. In Table 1, we present the properties of the three social datasets Facebook, Deezer and DBLP taken from the website https://snap.stanford.edu/data/. The Facebook dataset is a list of friend connections collected from users on the Facebook application. Deezer is another dataset, collected from the music streaming service (November 2017) and representing a network of users from three European countries, Romania (HR), Croatia (HO) and Hungary (HU), along with the attributes that are a list of each user's favorite music genres. The DBLP data is a co-authoring network containing the information about various research publications in the field of computer science, where two authors are linked if they published at least one article together. The number of nodes and edges of DBLP is the biggest, followed by Deezer and Facebook.

The accuracy of link prediction depends on the properties and domain of each particular dataset. So, we started by calculating and analyzing some of the characteristics of the three social networks mentioned above. For example, the average clustering (AC) index of Facebook is 0.606 and this index is equivalent to DBLP by 0.632. However, Facebook's average node degree (AND) index is 7 times higher than that of DBLP and the network density (DEN) index is about 540 times higher. Therefore, DBLP is considered a sparse network compared to Facebook. The subnetworks RO, HR and HU of Deezer have small AC indices, which are about $\frac{1}{6}$ of Facebook and DBLP. Although compared to DBLP, the AND index of Deezer is not much different, but the DEN index is 5-15 times higher. Based on the results in Table 1, we can obtain two preliminary conclusions that the prediction accuracy for Facebook will be higher than the two remaining networks because Facebook has the highest network density, and DBLP is the sparsest network of the three networks we consider. The adjacency matrix extracted from the DBLP dataset also shows that most of its elements are *null*.

We represent each of the above social networks as a source graph. On this graph, we prepare a corresponding feature dataset extracted randomly with a ratio 0.02% of the total number of edges (This ratio is applied for all networks to ensure fairness). Each data point represents the link between the two nodes in the graph, so depending on the fact that a data point with or without a link in the source graph will have a positive label

Table 1. Properties of datasets

Dataset		Number of nodes	Number of edges	AC	DEN	AND
ego-Facebook		4,039	88,234	0.606	0.0108	43.691
gemsec-Deezer	RO	41,773	125,826	0.091	0.0001	6.024
	HR	54,573	498,202	0.136	0.0003	18.258
	HU	47,538	222,887	0.116	0.0002	9.377
com-DBLP		317,080	1,049,866	0.632	**0.00002**	6.622

(+) or negative labels (-). To balance the training feature dataset, the number of negative data points is taken at random and equal to the number of positive data points. In fact, 90% of new links are formed between 2-step node pairs [21], so we choose only negative data points that satisfy this condition (i.e. the parameter s equals 2 in Algorithm 1). We also select topology-based similarity measures that are ranked as the most effective ones in link prediction, namely CN, RA, Jaccard, AA, TNS, and CAC to extract the features of each respective data point. Finally, we divide the feature dataset to experiment and analyze the effect of different measures on prediction accuracy.

Link prediction can be stated as a binary classification problem, which can be solved by implementing the features extracted from graphs according to a supervised learning approach. Therefore, we have used the feature datasets as the classification input datasets for the machine learning models. The classification model of the link prediction problem needs to predict hidden, missing or non-existent links by distinguishing positive and negative classes from the input feature dataset.

Our experiments were performed on a 1.4 GHz Dual-Core Intel Core i5 processor, with 4 GB 1600 MHz DDR3 memory and all algorithms are implemented in Python. We conducted experiments based on the feature datasets extracted from the source graphs corresponding to Facebook, Deezer and DBLP in Table 1. The data points with positive and negative labels will be randomly mixed. Using the ranking method, we arrange the feature values in descending order and select top-k elements ($k = 0.02*$number of edges) as the predicted values. For all classification cases, we use a cross-evaluation method with a fold of 5 ($f = 5$). Each experiment will be repeated 10 times (with each loop, the training feature dataset is randomly mixed) and take the average value.

We study the effectiveness of link prediction on a source dataset using different similarity measures. Specifically, we test how performance will be affected, when (1) using separately similarity measures, including the one we proposed, by ranking method and by a single classifier (i.e. SVM) in Experiment 1; (2) using three types of learning models at the same time in Experiment 2.

Experiment 1. The effect of different similarity measures on predictability.

To evaluate the effectiveness of the specific measures including CAC proposed in this paper, we experimented on three sub-networks of Deezer. The two prediction methods used are ranking and SVM. The data in Table 2 represents the accuracy for each selected measure: $CN, RA, JC, AA, PA, TNS, CAC$, and ALL (for all features). The values in Table 2 indicate that the prediction method by ranking similarity measure values gives relatively low results, while the predictive results are quite high with SVM. Even with CAC, the accuracy is over 80%, not inferior to predictability compared to other features such as RA, JC, PA and TNS. The accuracy of using the whole features also significantly increased over 81%.

We conducted the same experiment with the other two networks Facebook and DBLP, but the prediction results while using ranking and SVM did not differ significantly as

Table 2. Results of nine similarity measures

gemsec-Deezer	CN	RA	JC	AA	PA	TNS	CAC	ALL	Method
RO	0.397	0.325	0.291	0.329	0.435	0.278	0.316	0.340	top-k
	0.838	0.781	0.805	0.828	0.552	0.736	**0.803**	**0.838**	SVM
HR	0.762	0.648	0.630	0.658	0.485	0.487	0.618	0.613	top-k
	0.807	0.722	0.799	0.815	0.526	0.509	**0.787**	**0.818**	SVM
HU	0.545	0.468	0.462	0.462	0.459	0.403	0.425	0.461	top-k
	0.838	0.805	0.788	0.831	0.550	0.646	**0.810**	**0.826**	SVM

with Deezer. This little difference proves that the properties of the Deezer network (with a low average clustering AC and a low density DEN) affect the predictive results. The single similarity measure, i.e. CAC, also contributes to the outcome of link prediction. In Experiment 1, the subset of features ALL giving quite high predictive results confirms the conclusion in [3] that the basic ranking algorithm on this feature subset plays an important role in link prediction.

Experiment 2. Compare the accuracy of learning models.

In this experiment, we conduct experiments on different cases: baseline classification algorithm, ensemble learning and multiple ensemble learning. We choose a set C of learning models that includes *SVM, Logistic Regression, ANN, Decision Tree* (for baseline classification algorithm) and *Bagging, AdaBoost, XGBoost, RandomForest, GBM* (for ensemble learning). The parameters are set for Experiment 2 with: number of step $s = 2$, sampling coefficient $\theta = 0.02\%$, parameter of voting $= 'hard'$, cross-evaluation with f-fold $= 5$ and threshold δ is adjusted from $0.2 - 0.25$ to filter the limit of the number of estimators ES for the best results.

Table 3. Results of learning models

Type	Method name	gemsec-Deezer			com-DBLP
		RO	HR	HU	
Baseline classification algorithm	SVM	0.830	0.820	0.857	0.885
	Logistic	0.736	0.694	0.619	$0.888^{(*)}$
	ANN	$0.83^{(*)}$	$0.825^{(*)}$	$0.891^{(*)}$	$0.891^{(*)}$
	Decision Tree	0.745	0.749	0.787	0.857
Ensemble learning	Random Forest	0.829	$0.821^{(*)}$	0.820	$0.898^{(*)}$
	Bagging	$0.830^{(*)}$	$0.820^{(*)}$	$0.862^{(*)}$	0.885
	AdaBoost	0.827	0.818	0.814	$0.895^{(*)}$
	XGBoost	0.817	0.819	0.845	$0.901^{(*)}$
	GBM	0.816	0.819	0.840	$0.903^{(*)}$
Multiple ensemble learning	**MEL**	**0.833**	**0.826**	**0.910**	**0.907**

Note: $^{(*)}$ *marks the accuracy* acc_i *of the element* $c_i \in C$ *which is selected for establishing MEL.*

The results obtained from Experiment 2 are described in Table 3 of Deezer showing that ANN and Bagging bring the highest results. The combination of ANN and Bagging by voting mechanism created a multiple ensemble learning set that brings a predictive result increased from $0.3 - 1.9\%$, the highest for HU subnetwork. In a sparse network as

DBLP, classifiers such as Logistic and ANN are proved to be more efficient than SVM, reaching nearly 90%. Ensemble learning models such as XGBoost, GBM also achieve over 90%. Finally, the multiple ensemble learning model increases the final predictive result by 90.7%. This experiment proves that multiple ensemble learning models established by our proposed algorithm MEL are highly effective in predicting links.

The authors in [17] have demonstrated that multiple ensemble learning can improve accuracy in supporting diabetes decision-making using the Pima Indian diabetes UCI dataset with voting techniques. Unlike this study, our experiments do not focus on the types of voting but on the algorithm to select single classifiers or ensemble learning suitable for multiple ensemble learning to provide the highest link prediction efficiency.

5. **Conclusions.** The techniques used to improve the performance of link prediction in sparse social networks have been proposed in this paper. We described the properties of three popular social networks (Facebook, Deezer and DBLP) and compared their sparsity. Our proposed similarity measure CAC represents the common attributes between two nodes in a network graph and is proven to be an effective feature for link prediction. The multiple ensemble learning models are built from the baseline classifiers and ensemble learning models using our proposed algorithm. The experiments have been conducted on these models and the evaluation based on predictive performance show that the multiple ensemble learning provides the higher accuracy when applied to the datasets of the social networks DBLP and Deezer.

In the future, we expand this research direction to apply to different types of graphs such as directed, multidirectional or weighted. The community feature in social networks also influences the user's connection decision, which needs to be further exploited. Another potential research approach can also be explored to predict whether links will be hidden or will be canceled in the future. The multiple ensemble learning method can also be upgraded to use in more complex multi-layer networks.

REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Soc. Networks*, vol. 25, no.3, pp.211-230, 2003.

[2] F. Aghabozorgi and M. R. Khayyambashi. A new similarity measure for link prediction based on local structures in social networks. *Physica A: Statistical Mechanics and its Applications*, 501(C), pp. 12-23, 2018.

[3] C. Ahmed, A. ElKorany, and R. Bahgat. A supervised learning approach to link prediction in twitter. *Social Netw. Analys. Mining*, vol. 6, no. 1, pp. 24-:1-24:11, 2016.

[4] P. Bhattacharyya, A. Garg, and S. F. Wu. Analysis of user keyword similarity in online social networks. *Social Netw. Analys. Mining*, vol. 1, no. 3, pp. 143-158, 2011.

[5] S. Chen, J. Pan, and K. Lu. Driving behavior analysis based on vehicle obd information and adaboost algorithms. *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, pp. 18-20, 2015.

[6] S. Goswami, C. A. Murthy, and A. K. Das. Sparsity measure of a network graph: Gini index. *Inf. Sci.*, vol. 462, pp. 16-39, 2018.

[7] P. A. V. Hall and G. R. Dowling. Approximate string matching. *ACM Comput. Surv.*, vol. 12, no. 4, pp. 381-402, 1980.

[8] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52-74, 2017.

[9] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[10] Z. Huang, X. Li, and H. Chen. Link prediction approach to collaborative
ltering. In M. Marlino, T. Sumner, and F. M. S. III, editors, *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2005, Denver, CO, USA, June 7-11, 2005, Proceedings*, pp. 141-142. ACM, 2005.

[11] Y. Li, J. Li, and J.-S. Pan. Hyperspectral image recognition using svm combined deep learning. *Journal of Internet Technology*, vol. 20, no.3, pp. 851-859, May. 2019.

[12] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.*, vol. 58, no. 7, pp.1019-1031, 2007.

[13] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31-88, 2001.

[14] B. Rozemberczki, R. Davies, R. Sarkar, and C. A. Sutton. GEMSEC: graph embedding with self clustering. In F. Spezzano, W. Chen, and X. Xiao, editors, *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*, pp. 65-72. ACM, 2019.

[15] O. Sagi and L. Rokach. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, 2018.

[16] R. Z. Samira, S. Klaylat, L. Hamandi, and Z. Osman. Ensemble models for enhancement of an arabic speech emotion recognition system. *Advances in Information and Communication. FICC 2019.Lecture Notes in Networks and Systems*, 70, 2019.

[17] S. Sathurthi and K. Saruladha. An analysis of parallel ensemble diabetes decision support system based on voting classifier for classification problem. *Electron. Gov. an Int. J.*, 16(1/2), pp.25-38, 2020.

[18] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. In X. Amatriain, M. Torrens, P. Resnick, and M. Zanker, editors, *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pp. 183-190. ACM, 2010.

[19] P.Wang, B. Xu, Y.Wu, and X. Zhou. Link prediction in social networks: the state-of-the-art. *CoRR*, abs/1411.5118, 2014.

[20] L. Xu, X. Wei, J. Cao, and P. S. Yu. Multiple social role embedding. In *2017 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2017, Tokyo, Japan, October 19-21, 2017*, pp. 581-589. IEEE, 2017.

[21] D. Yin, L. Hong, X. Xiong, and B. D. Davison. Link formation analysis in microblogs. In W. Ma, J. Nie, R. Baeza-Yates, T. Chua, and W. B. Croft, editors, em Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011, pp. 1235-1236. ACM, 2011.

[22] J. Zhang, L. Tan, and X. Tao. On relational learning and discovery in social networks: a survey. *Int. J. Machine Learning & Cybernetics*, vol. 10, no. 8, pp. 2085-2102, 2019.

[23] J. Zhang, L. Tan, and X. Tao. On relational learning and discovery in social networks: a survey. *Int. J. Machine Learning & Cybernetics*, vol. 10, no. 8, pp. 2085-2102, 2019.

[24] W. Zhao, S. Tan, Z. Guan, B. Zhang, M. Gong, Z. Cao, and Q. Wang. Learning to map social network users by unified manifold alignment on hypergraph. *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 12, pp. 5834-5846, 2018.

[25] T. Zhou, L. Lü, and Y.-C. Zhang. Predicting missing links via local information. *The European Physical Journal B*, vol. 71, no. 4, pp. 623-630, Oct 2009.