International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I5P207

Volume 71 Issue 5, 70-83, May 2023


*Original Article*

# Accurate Link Prediction Metric Based on Node Centrality and LSBC

Bara Samir[1], Jibouni Ayoub[2], Hammouch Ahmed[3]

[1]*Sultan Moulay Slimane High School of Technology, Khenifra, Morocco.*
[2]*Rabat IT Center, LRIT, Faculty of Sciences, Mohammed V University, Rabat, Morocco.*
[3]*ENSAM Rabat, Mohammed V University, Rabat, Morocco.*

[1]*Corresponding Author : samirbara00@gmail.com*

Received: 14 March 2023 Revised: 12 May 2023 Accepted: 18 May 2023 Published: 25 May 2023

***Abstract*** *- During the last two decades, there has been a lot of interest in social network analysis. These networks are dynamic, with new links appearing and disappearing all the time. The challenge of suggesting future links from the current state of the network is recognized as link prediction. We calculate user similarity using information from nodes and edges. The more similar two users are, the more likely they will connect. In the domain of link prediction, similarity measures are quite essential. Many authors have suggested and analyzed numerous metrics, such as Jaccard, AA, and Katz, because of their simplicity and flexibility. In this work, we extend a new parameterized approach [21] to enhance the AUC value of link prediction metrics by combining them with eigenvector centrality. This work proposes to enhance local similarity metrics due to their interpretability and low complexity. Experiments reveal that the suggested technique outperforms the state-of-the-art metrics in terms of AUC, and it also outperforms the LSBC metric in terms of time. In addition to that, we have used machine learning algorithms to solve the link prediction problem as a classification task.*

***Keywords*** *- Social network analysis, Link prediction, Similarity measures, Machine learning, Eigenvector centrality.*

## 1. Introduction

Connecting people is the primary cause for using social networks (SN) around the world; we study those networks in social network analysis (SNA), where scientist tries to discover the communities [22, 26] and influential nodes [23,24] on the networks. In social network analysis, the networks are described by graphs G (V, E), where V are nodes in the graphs; they may represent actors, proteins, or universities. E is the set of edges or connections between the nodes.

One of the most crucial aspects of a social network is its expansion. This augmentation can be observed from two unique perspectives: The first is the number of new nodes joining the network, and the second is the development of links or edges between users. This latter is explored in the link prediction. Link prediction has attracted substantial interest in the recent five years because of its application in numerous fields, such as spam E-mail detection [28], disease prediction [29], and system recommendations [27]. To completely understand and describe a social network where individuals are treated as nodes and interactions between them are represented by edges, SNA leverages methods from graph theory.

Link prediction is known to be one of the most important tasks of SNA, in which the researchers try to infer the outcoming links from the actual state of the network. We can define link prediction as follows: from the actual state of network G (V, E) at time t, could we estimate the forthcoming associations that will show up in time t+1? Where G (V, E2) is the network at time t+1, notice that the set of nodes remains the same, but the set of edges E becomes E2 where new edges are added because of that |E|<|E2|.

Previous works [30,31,32] have classified the link prediction solution into three main categories: maximum likelihood methods, probabilistic methods, and similarity-based metrics. Maximum likelihood methods are statistical methods that estimate the parameters of a model that best fit the observed data. In link prediction, a common maximum likelihood method is to estimate the probability of a link between two nodes in a network based on the observed links and use that probability to predict future links.

On the other hand, probabilistic methods are a class of methods that use probability distributions to model the data. A popular probabilistic method used in link prediction is the stochastic block model, which models the network as a probability distribution over blocks of nodes and links

between blocks. Unfortunately, those two methods cannot deal with large graphs. For that reason, we use local similarity-based metrics.

The developers of [33] investigate link prediction as a supervised learning task. Along the way, they discover a set of characteristics that are critical to performance in the supervised learning scenario. The identified qualities are simple to compute while also being quite successful in tackling the link prediction issue. They also explain why the qualities in their class density distribution are useful.

Then, using a 5-fold cross-validation, they compare numerous classes of supervised learning algorithms in terms of prediction performance using various performance metrics such as accuracy, precision-recall, F-values, squared error, and so on.

In [34], the authors have proposed two similarity metrics. The first one is based on combining the Preferential Attachment index and the Adamic-Adar index using a weighted combination. The second contribution uses the Infomap algorithm to detect the communities; that information has been used to enhance the AUC.

The authors of [35] presented \textbf{Common Neighbor and Centrality based Parameterized Algorithm], which is based on two key characteristics of nodes, namely the number of shared neighbors and their centrality. The common neighbors between two nodes are referred to as shared neighbors. A node's importance in a network is referred to as its centrality.

The major drawback of the previously introduced metrics is the high complexity and the time needed to classify the links; for those reasons, metrics still need to enable state-of-the-art users to enhance their results in a fair amount of time. To this end, we have extended the LSBC metrics [21] (we have focused on the best three metrics, JA, AA, and RA-based metrics); since the betweenness is very time-consuming, we have used the eigenvector centrality of the source and destination nodes.

From the above, we can conclude that most methods developed in state of the art are not parametrized (like CN, Jaccard …); as a result that they can not suit all the complex networks. Our metric rely on the use of an eigenvector that has low complexity (we have used the Maximum number of iterations in power method 50); this has reduced the execution time to about half and provides the same correctness in term of AUC.

This paper is organized as follows: in section 2, we shed light on the link prediction metrics from state of the art, and then, we propose the LSEV variations of AA, JA, and RA. In section 3, we introduce the evaluation criterion AUC and the datasets used to examine the validity of the proposed enhancement. Then, we study the impact of the parameter \alpha on the results and compare the performance of the state-of-the-art metrics against the proposed LSEV variations. Finally, we have compared LSBC, LSEV, and local metrics regarding time.

## 2. Methods

### *2.1. Overview of Link Prediction*

In this section, we introduce the most used state-of-the-art link prediction algorithms. Those algorithms are also used for comparison against the performance of our proposed metric. Through this paper, we use the following definitions: G (V, E) is a graph where V is a finite set of vertices/nodes. Those nodes could represent persons, football teams etc. E is a finite set of edges or links. We use e(x,y) to refer to the link between nodes x and y, \Gamma(x) is the set of neighbors of x. |\Gamma(x)| is the degree of node x (how many neighbors the node x has). Graph G can be directed where every link has a specific direction or undirected where edges have no direction; also, it can be weighted where each edge has weight or unweighted where all edges have no weights. In this work, we consider only simple graphs (graphs that are not directed or weighted; also, self-loops are not considered).

Link prediction metrics are categorized into three main categories:

Local measures rely exclusively on neighbor information, such as the common neighbor and the Jaccard coefficient. Semi-local measures that use local paths, such as the Local path metric. The final category is global metrics, which, like Katz, take into account all pathways. One of the biggest disadvantages of using semi-local or global techniques is the constantly increasing computational time. As a result, we will concentrate on local metrics in the following section.

Previous research has only concentrated on developing new metrics for datasets and attempting to achieve high AUC values; however, few researchers have addressed the topic of improving link prediction metrics. Several ways have been proposed to address the link prediction problem. The authors of [1] developed a preferential attachment in which the score is determined by the degree of both nodes. The higher the degree of both nodes, the higher the score. The following is how PA is defined:

$$\mathrm{PA}(x,y) = |\Gamma(x)| * |\Gamma(y)|$$

The authors in [2] have introduced common neighbor as follows:

$$CN(x,y) = |\Gamma(x) \cup \Gamma(y)|$$

The score represents the number of shared neighbors; if both nodes share many neighbors, the probability that they are acquaintances is high use.

The authors in [3] have proposed the ed Jaccard index, a normalized version of the common neighbors' index. The

Jaccard coefficient is defined as the probability of selection of a common neighbor considering all paths of the neighbors of either vertex.

$$Jaccard(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

Other variations of common neighbor are:

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

This metric was proposed to calculate the similarity between two web pages, but after has been used by Liben Nowell in the link prediction field.

Hub Promoted Index (HPI) metric was proposed for metabolic networks where the authors in [5]. HPI is defined as follows:

$$\text{Hub } promoted(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min(|\Gamma(x)|, |\Gamma(y)|)}$$

Note that HPI promotes the formation of links between hub nodes (nodes with a number of links that greatly exceed the average).

Hub Depressed Index [5] follows the same principle as HPI but has the opposite aim because it promotes link formation between hubs and low-degree nodes. It is defined as follows:

$$Hub\ depressed(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max(|\Gamma(x)|, |\Gamma(y)|)}$$

In the Leicht-Holme-Newman index (LHN) [6], the authors assume that two nodes are similar if their corresponding neighbors are similar.

$$LHN(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|.|\Gamma(y)|}$$

Sorensen index [8] is defined as the number of vertices adjacent to both nodes normalized by the sum of the degrees of both nodes:

$$Sorensen(x,y) = \frac{2.|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|}$$

Salton index [7] is like the Sorensen index described above; it is a normalization of common neighbors. They define Salton as:

$$Salton(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)|.|\Gamma(y)|}}$$

In the Resource Allocation index (RA) [9], the authors assume that nodes with a higher degree are meaningless compared to low-degree nodes. The basic idea behind RA is that a node shares equally a resource unit between all its neighbors; because of that, the high-degree nodes share less resources than low-degree nodes.

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}$$

### *2.2. The Proposed Metric*

In this article, we enhance a novel metric [21] to calculate the connectedness or similarity of two nodes. Consider x and y as two nodes on graph G; S (x, y) computes the similarity between x and y. The likelihood that x and y will be connected in the future is determined by the amount of power each of these nodes has over the network at time t.

The suggested measure is based on local similarity measures, which are very simple to calculate and may be applied to large-scale networks; however, they do not guarantee the same precision and accuracy for each network.

Indeed, several studies have demonstrated that using extra attributes such as community information and node information improves their effectiveness. As a consequence, we can develop a measure that can deal with large-scale networks and produce more accurate findings in a fair amount of time.

For these factors, many works have proposed to use the centrality of nodes, such as [21], where the authors added the Mean Received Resources in the first work. The Betweenness centrality of the source and destination nodes in the second work, both of which have improved the accuracy of the results but are time-consuming.

Those works rely on the following idea: If both nodes have several pieces of information, they are more likely to be linked to increase their amount of data, in other words, to have more control over the network. In our work, we have used the eigenvector centrality of the source and destination node combined with the local similarity measures to enhance the accuracy because the MRR and Betweenness centrality has a big complexity. Furthermore, we have focused on RA, AA, and Jaccard metrics that provide the best accuracy in almost all networks.

$$LSEV_{AA}(x,y) = \alpha \times AA(x,y)^{1-\alpha} + sigmoid\big(e(x) + e(y)\big)$$

$$LSEV_{JA}(x,y) = \alpha \times JA(x,y)^{1-\alpha} + sigmoid\big(e(x) + e(y)\big)$$

$$LSEV_{RA}(x,y) = \alpha \times RA(x,y)^{1-\alpha} + sigmoid\big(e(x) + e(y)\big)$$

Note that in order to speed up the time execution, we have used the Numpy library [36].

### *2.3. Results*

In this section, we will introduce the evaluation criterion AUC, and then we will introduce the datasets used to test the validity of our proposed metric. Then we studied the effect of

alpha on the results by tuning different values of $\alpha \in [0,1]$, after that, we compared the original metric and the enhanced metrics of AA, RA, and Jaccard. We have also compared the performance of all metrics on all the datasets. Finally, we have studied the time complexity.

### *2.4. Evaluation Metric AUC*

Let G(V, E) be a simple network ( Note that loops and multi-edges are not authorized); we divide our graph into $G_{train}(V, E_{train})$ and $\mathrm{G_{test}(V, E_{test})}$ Note that the train set comprises 90% of the interactions, and the test set includes 10% of the vertices.

$$E_{test} \cup E_{train} = E$$
$$E_{test} \cap E_{train} = \emptyset$$

We utilized AUC to measure the performance of all metrics; AUC is defined as the likelihood that a link picked at random from $E_{test}$ has a higher score than a link randomly chosen from $\bar{E}$:

$$AUC = \frac{N' + 0.5 \times N''}{N}$$

- $N'$ is the number of times an edge from $E_{test}$ and an edge from $\bar{E}$ have the same score.
- $N$" is the number of times the edges from $E_{test}$ had a higher score than the edge from $\bar{E}$.
- $N$ is the number of independent comparisons.

### *2.5. Data Sets*

a) Les misirables [10]: an undirected network comprises co-occurances of actors in Victor Hugo's novel 'Les Misérables'. A node represents a character, and an edge between two nodes shows that these two characters took place in the same chapter of the book. The weight of each link reflects how often such a co-appearance occurs.
b) USAir [13] is an airline network of the US where a node depicts an airport, and an edge describes the connectivity between two airports.
c) This is the popular and widely utilized Zachary karate club network [11,14]. Wayne Zachary collected the data from members of a university karate club in 1977. Each node represents a club member, and each edge represents a tie between two club members.
d) NetScience [16,17]: A graph of coauthorships among scientists who publish on the topic of network science.
e) C.elegans' neural network [14], in which an edge connects two neurons, whether they are connected by a synapse or a gap junction.
f) A bottlenose dolphin social network [15]. The collection is made up of a series of linkages, each reflecting a common association between dolphins.
g) This is the network of jazz musicians [12] who collaborate. Each node represents a Jazz performer, and an edge indicates that two musicians have collaborated in a band.

In the table 1, N is the number of nodes in the graph, E is the number of edges, GCC is the global clustering coefficient [19,20], Diameter is the maximum eccentricity, and Global efficiency [18] of the graph.

### *2.6. Results*

In this section, we will present the results of our proposed metric in the previous section and compare it with the previously introduced state-of-art metrics in section 2.

Note that we use only simple graphs G (V, E) where no loops or multiple links are also allowed. The graphs are undirected and unweighted. The train set contains 90% of the links, and the remaining 10% represents the test set.

Table 2 shows the performance of the enhanced versions of algorithms AA, RA and JA using different values of α in the range [0,1]. Note that when $\alpha = 0$ and $\alpha = 1$, the score is obtained using eigenvector centrality.

**Table 1. Datasets features**

| | N | E | GCC | Average degree | Diameter | Global efficiency |
|---|---|---|---|---|---|---|
| **Les miserables** | 77 | 253 | 0,559 | 3,285 | 5 | 0,434 |
| **Network Science** | 1461 | 2742 | 0,694 | 1,877 | Not connected | 0,016 |
| **USAir97** | 332 | 2126 | 0,625 | 6,404 | 6 | 0,406 |
| **Macaque** | 242 | 3054 | 0,45 | 12,62 | 4 | 0,5 |
| **Karate** | 34 | 78 | 0,571 | 2,294 | 5 | 0,492 |
| **C-elegans** | 131 | 687 | 0,245 | 5,244 | 6 | 0,449 |
| **dolphins** | 62 | 159 | 0,259 | 2,565 | 8 | 0,379 |
| **jazz** | 198 | 2742 | 0,617 | 13,848 | 6 | 0,513 |

**Table 2. Performance of algorithms using different values of $\alpha$**

We can notice from Table 2 that when the scores are received using eigenvector centrality ($\alpha = 0$ or $\alpha = 1$), the AUC is superior to 0.5 the uses of the importance of the nodes have an acceptable performance. For values between ]0,1[ we can notice that the shape of the curve increases to a maximal value, for instance, the $best_\alpha$ for the improved version of Adamic/Adar in the dataset Jazz, 0.6; for the dataset Dolphins is 0.7; for the rest of the datasets, the $best_\alpha$ is between [0.4,0.8]. For the improved version of Jaccard, the $best_\alpha$ of Jazz is 0.9, for the dataset Dolphins, it is 0.7, and for the rest of the datasets, the $best_\alpha$ is between [0.3,0.9], the same conclusion is applied to the improved version of RA. To sum up, to get the $best_\alpha$ user or system can use only values between [0.4,0.9] since the precision decrease for $\alpha = 0$ $and$ $\alpha = 1$ and also values of $\alpha \in [0.1,0.4[$ have a very small AUC compared to the maximum value of AUC.

From Table 4, we can conclude that the best three performing algorithms are PSI AA, PSI RA, and PSI Jaccard in all the datasets. These results offer a piece of powerful evidence that the proposed combination enhances the accuracy of metrics in general and the three metrics used in this study.

In order to compare the general performance of every algorithm in all datasets, we calculate the meaning of AUC. From 3, we can conclude that the performance of the improved version of RA, AA, and JA offers the best performance in all datasets, with AUC around 90%. The original version has an AUC of around 83%; because of that, the proposed metrics have improved the accuracy by 7%. The worst-performing algorithms are PA and Katz, respectively, with 70% and 60% performance.

**Table 3. Results of original algorithms and the improved version of AA, RA and Jaccard**

| | Jazz | Dolphins | C-elegans | Karate | Macaque | USAir97 | Netscience | Les miserables |
|---|---|---|---|---|---|---|---|---|
| AA | 0,962 | 0,78 | 0,782 | 0,643 | 0,897 | 0,931 | 0,933 | 0,928 |
| JC | 0,956 | 0,79 | 0,775 | 0,5 | 0,883 | 0,892 | 0,932 | 0,858 |
| PA | 0,77 | 0,613 | 0,615 | 0,729 | 0,772 | 0,862 | 0,618 | 0,774 |
| HD | 0,947 | 0,79 | 0,779 | 0,5 | 0,869 | 0,882 | 0,932 | 0,866 |
| HP | 0,948 | 0,79 | 0,758 | 0,621 | 0,855 | 0,867 | 0,933 | 0,832 |
| LHN | 0,904 | 0,8 | 0,76 | 0,471 | 0,797 | 0,776 | 0,932 | 0,8 |
| PD | 0,956 | 0,787 | 0,78 | 0,557 | 0,893 | 0,923 | 0,933 | 0,898 |
| RA | 0,97 | 0,787 | 0,779 | 0,657 | 0,899 | 0,939 | 0,933 | 0,928 |
| CN | 0,954 | 0,783 | 0,777 | 0,629 | 0,891 | 0,922 | 0,933 | 0,904 |
| SA | 0,963 | 0,783 | 0,77 | 0,514 | 0,887 | 0,9 | 0,932 | 0,87 |
| SO | 0,956 | 0,79 | 0,775 | 0,5 | 0,883 | 0,892 | 0,932 | 0,858 |
| Katz | 0,51 | 0,46 | 0,523 | 0,714 | 0,628 | 0,571 | 0,671 | 0,822 |
| PSI AA | 0,968 | 0,867 | 0,819 | 0,871 | 0,909 | 0,959 | 0,947 | 0,936 |
| PSI RA | 0,973 | 0,867 | 0,822 | 0,857 | 0,915 | 0,964 | 0,945 | 0,928 |
| PSI Jaccard | 0,968 | 0,853 | 0,812 | 0,829 | 0,906 | 0,939 | 0,941 | 0,908 |

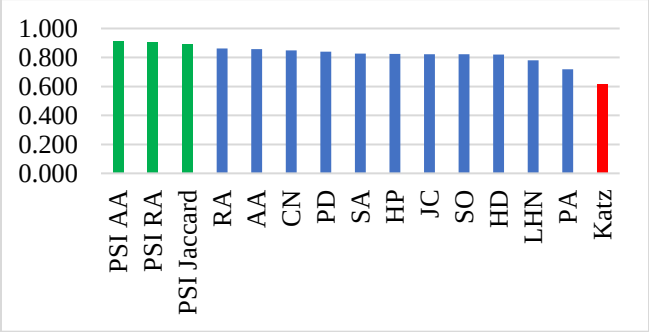
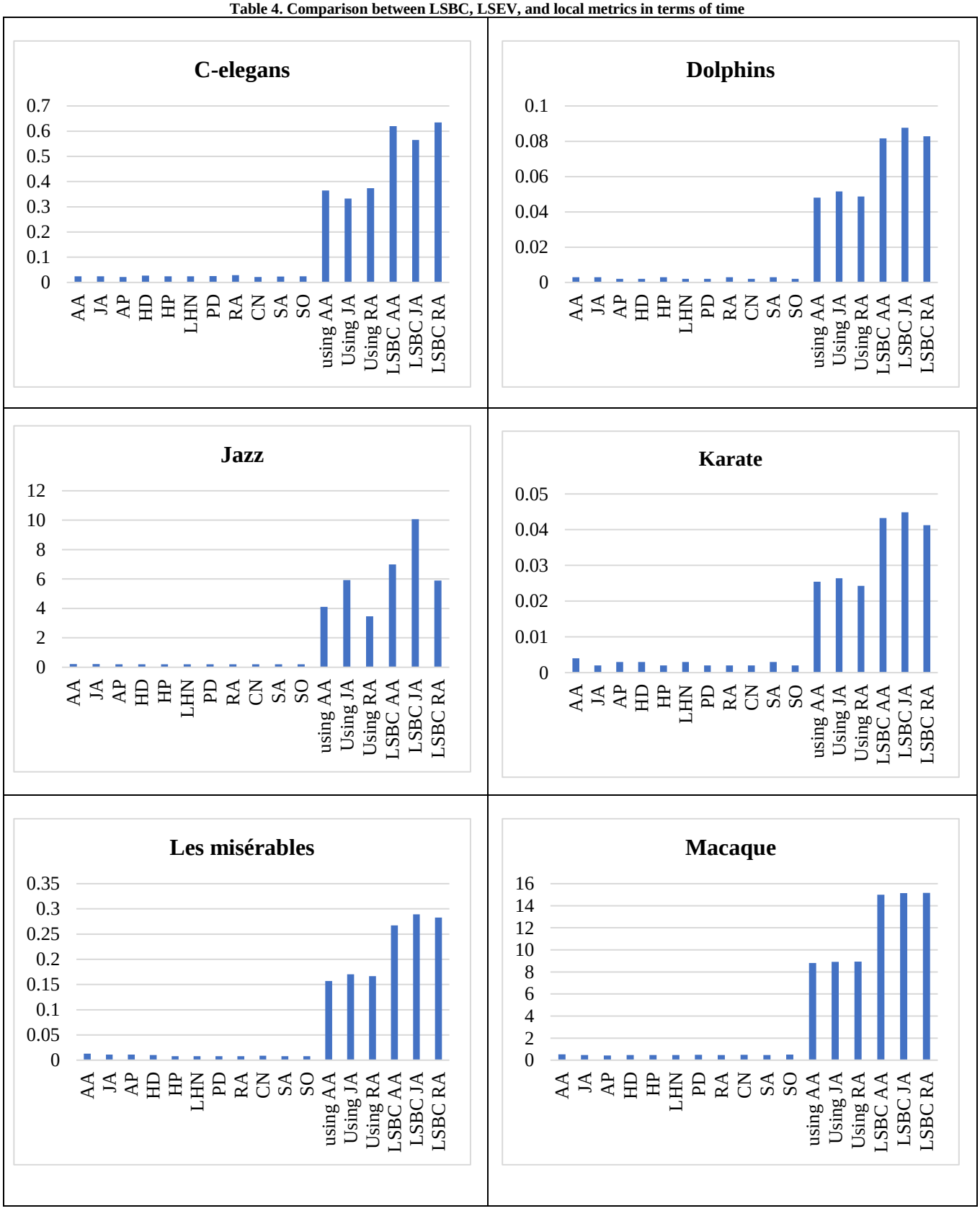
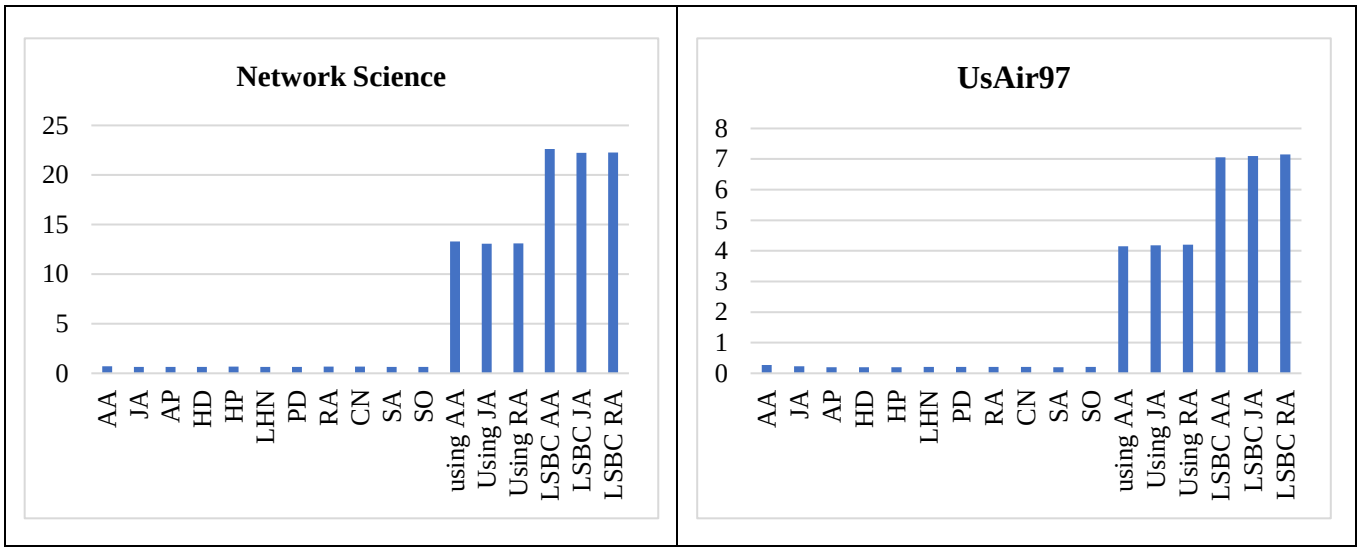



**Fig. 1 The performance of every algorithm on all datasets**

Next, we have calculated the time needed to experiment with different datasets. Results of Table 4 shows that the datasets network science, macaque, and jazz are the most time-consuming due to their size. In addition to that, for the dataset C-elegans, we can notice that the time needed in order to conduct the experiment is very high using the original LSBC; it takes about the double of time needed using our metric, also the time of local similarity metrics is very slow, but their performance is very low. To sum up, the proposed metric came as a trade-off between LSBC and local metrics; it enhances the performance of local metrics and is not time-consuming as the betweenness centrality.

**Table 4. Comparison between LSBC, LSEV, and local metrics in terms of time**






## 3. Link Prediction using Machine Learning

In this section, we will solve the link prediction problem as a classification task using machine learning algorithms.

### *3.1. Methodology*

To evaluate the effectiveness of the suggested measure when modeling the link prediction as a classification problem, we utilize the classification accuracy:

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Consider an undirected and unweighted graph G (V, E). To transform the link prediction issue into a binary classification task, we construct two sub-sets:

The first one contains the edges of E; the second one has randomly chosen edges from [E]. Then, we assign a zero to the edges from [E] and 1 to those of E. We divided them into a test set as well as a training set in which the test size=10%. Then, we train our classification algorithms on the training set and predict the test set.

### *3.2. Decision Tree and Neural Network Results*

From Table 5, we can conclude that the best-performing algorithm for the Jazz dataset is the neural network using our metrics as additional parameters. These results are still valid for dolphins, c-elegans, and the rest of the datasets**.**

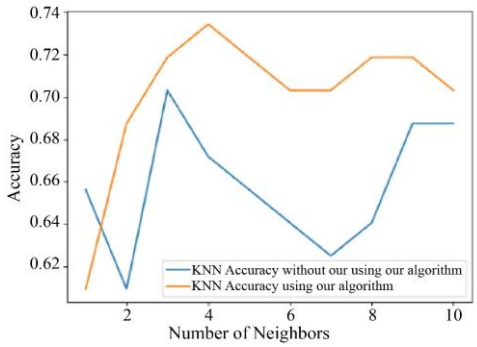
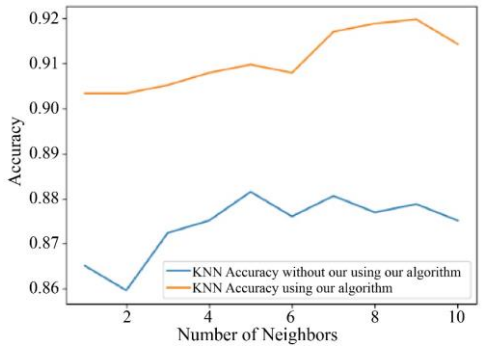
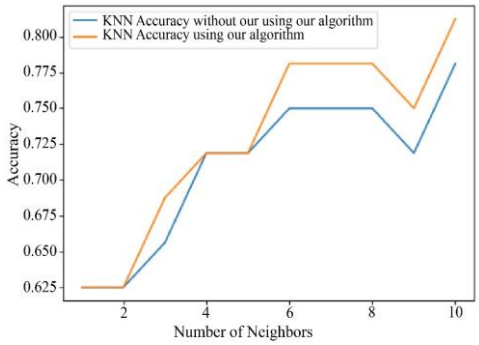
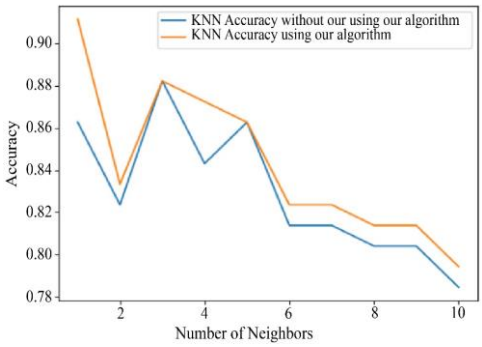
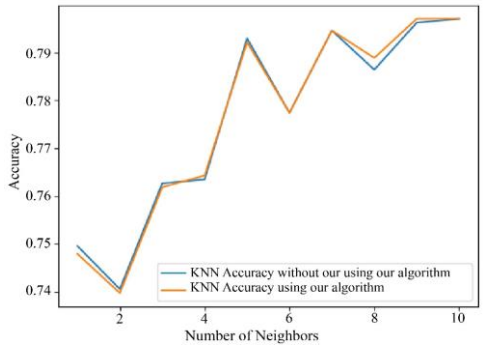
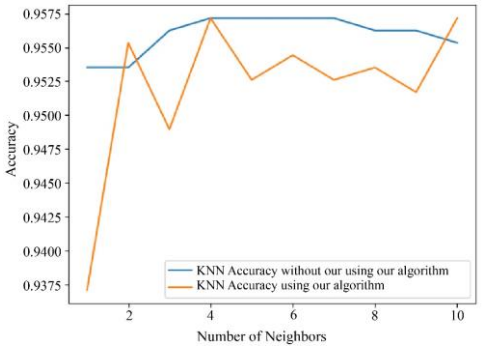
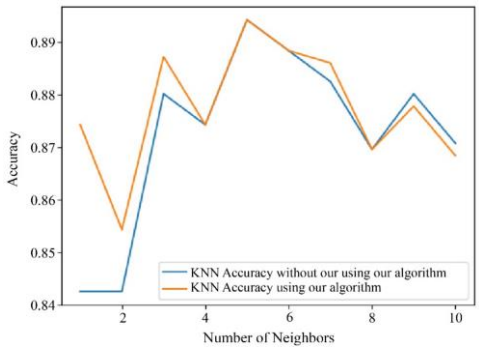
### *3.3. KNN Results*

Table 6 shows that using our algorithm as an additional parameter enhances the results' precision and makes higher accuracy. We can conclude that the results of KNN for the Dolphins, the dataset is higher when considering 4 neighbors; for the jazz dataset, the best performance is using 8 neighbors. Overall, the performance of the KNN using our metrics is higher in most datasets.

**Table 5. Decision tree and neural network results**

| | Jazz | Dolphins | C-elegans | Karate | Macaque | Usair | Network science | Les Miserables |
|---|---|---|---|---|---|---|---|---|
| **decision tree using our algorithms** | 0,893 | 0,703 | 0,735 | 0,781 | 0,794 | 0,898 | 0,96 | 0,941 |
| **decision tree without our algorithms** | 0,899 | 0,688 | 0,716 | 0,656 | 0,795 | 0,901 | 0,949 | 0,902 |
| **neural network using our algorithms** | 0,92 | 0,844 | 0,753 | 0,875 | 0,841 | 0,915 | 0,963 | 0,873 |
| **neural network without using our algorithms** | 0,921 | 0,874 | 0,749 | 0,875 | 0,84 | 0,908 | 0,963 | 0,863 |

**Table 6. The results of the KNN algorithm**

| | | | |
|---|---|---|---|
| Dolphins |  | Jazz |  |
| Karate |  | Les miserables |  |
| Macaque |  | Network Science |  |
| USAir |  | | |

### *3.4. Feature of Importance*

From Table 7, we can conclude that the improved version of Adamic/Adar Resource allocation and Jaccard has the heaviest weight in the decision-making according to the Random Forest algorithm. Besides, we have applied the random forest algorithm to classify the links; from Table 8, we can conclude that the improved version of Adamic/Adar Resource allocation and Jaccard have a positive impact on the accuracy of the algorithm Random Forest; whenever we add our metrics as additional parameters, the accuracy increases.

**Table 7. The importance of features using the Random Forest algorithm**

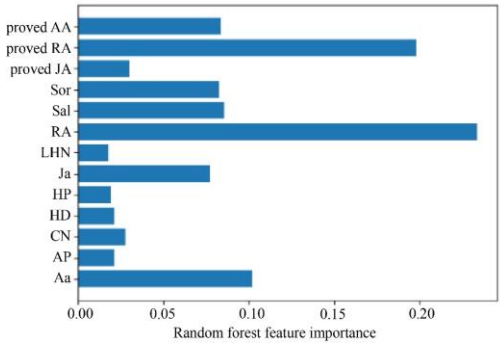
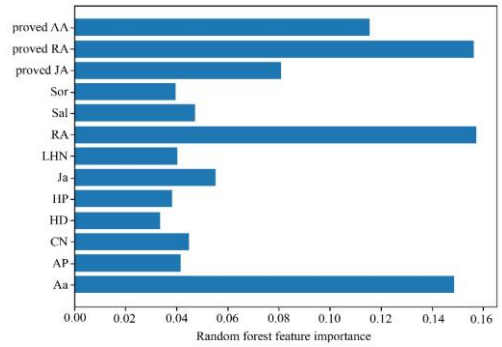
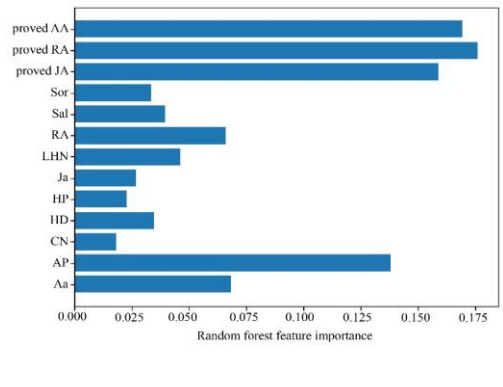
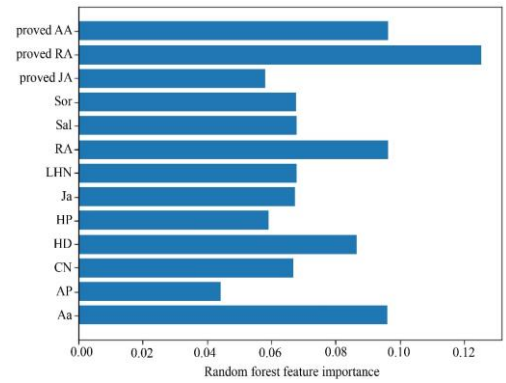
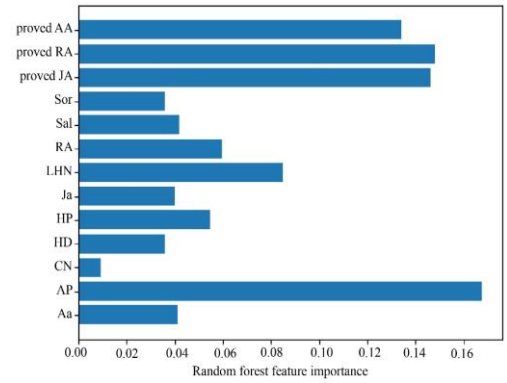
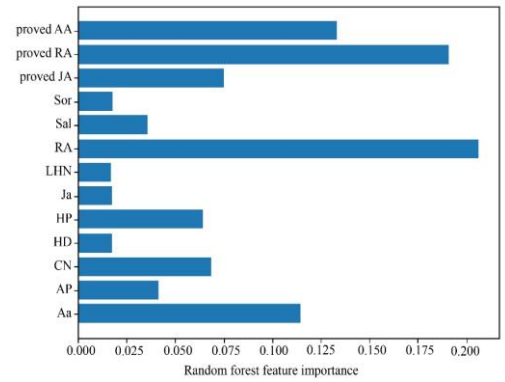
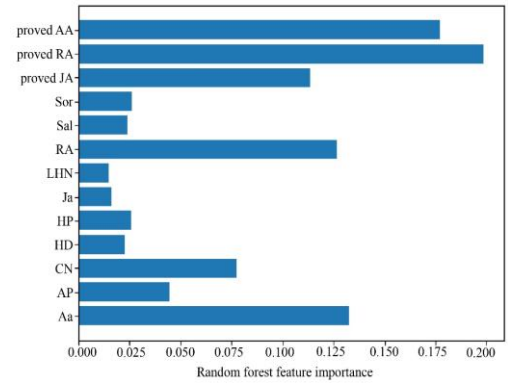
| | |
|---|---|
| Jazz |  |
| Dolphins |  |
| Karate |  |
| Les Miserables |  |
| Macaque |  |
| Network Science |  |
| USAir |  |

**Table 8. The results of the Random Forest algorithm**

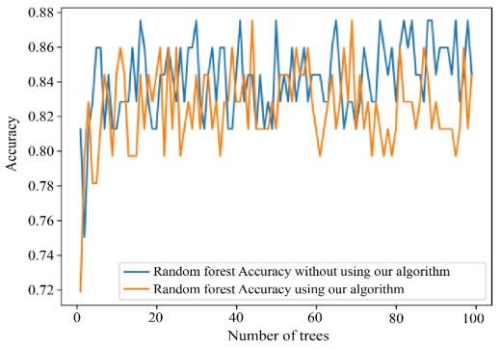
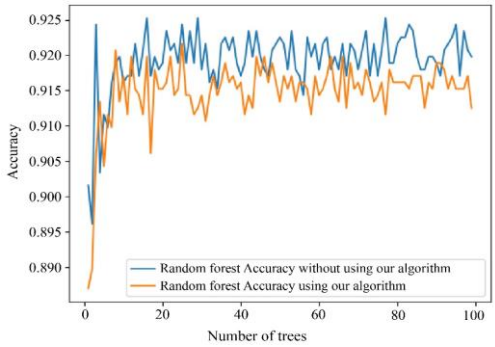
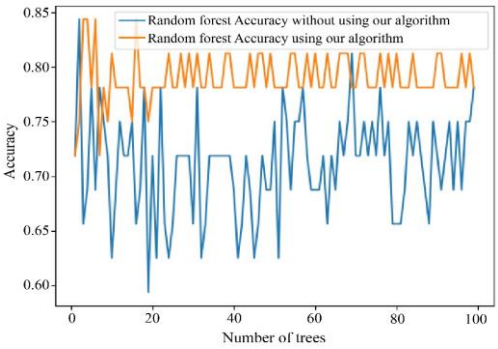
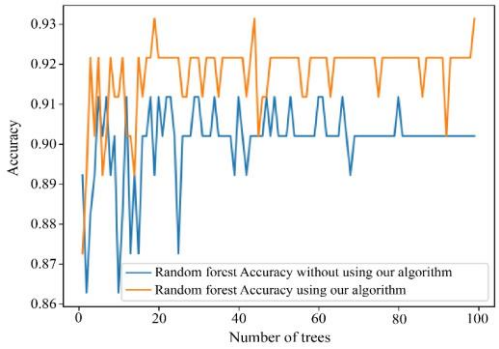
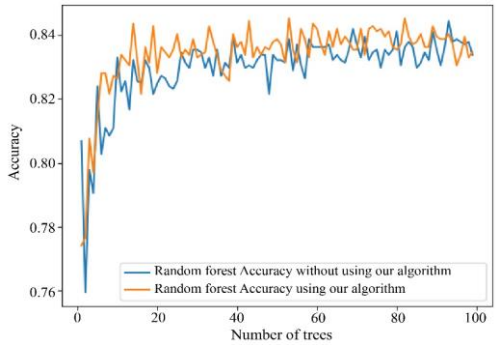
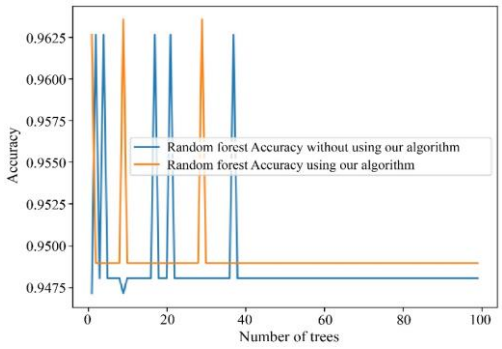
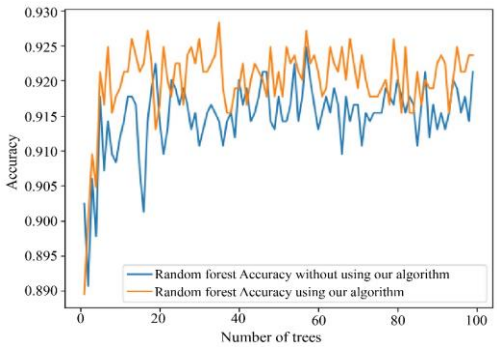
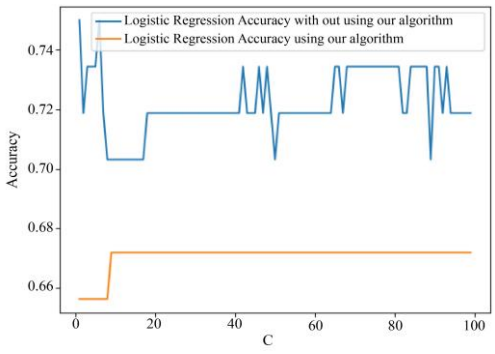
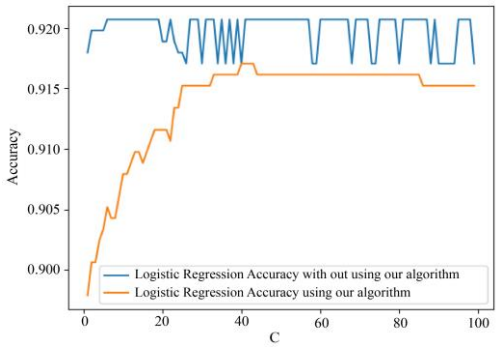
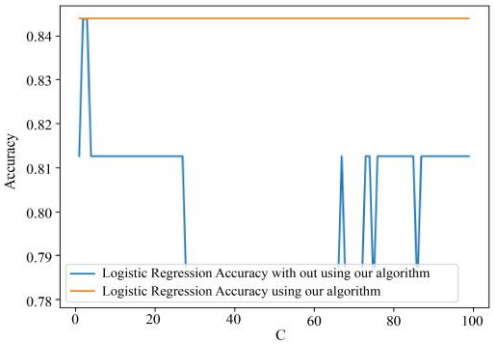
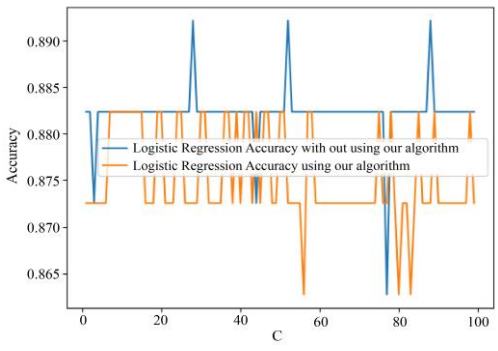
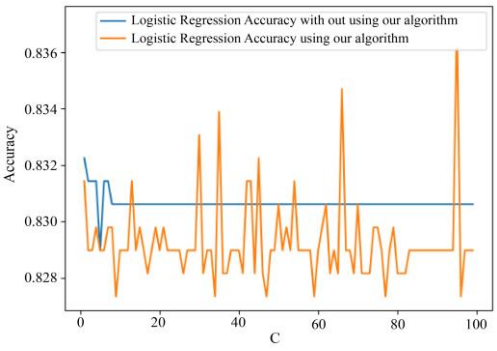
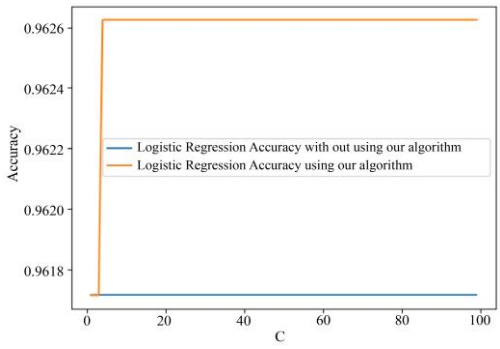
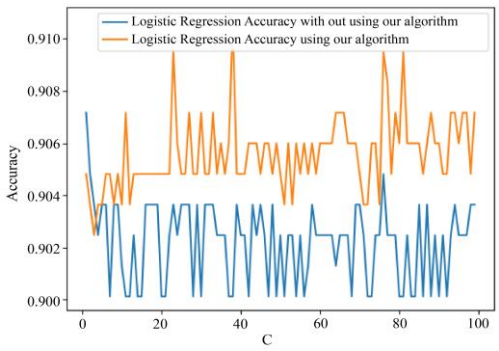
| | | | |
|---|---|---|---|
| Dolphins |  | Jazz |  |
| Karate |  | Les miserables |  |
| Macaque |  | Network Science |  |
| USAir |  | | |

**Table 9. The results of Logistic regression results**

| | | | |
|---|---|---|---|
| Dolphins |  | Jazz |  |
| Karate |  | Les miserables |  |
| Macaque |  | Network Science |  |
| USAir |  | | |

### *3.5. Logistic Regression Results*

From Table 9, we can conclude that using the improved versions of Adamic/Adar Ressource allocation and Jaccard has increased the accuracy of the Logistic regression results. According to the results, we can deduce that for the dolphins dataset, all the values of C provides high accuracy. The same conclusion could be drawn for Jazz, Karate, Network Science and USAir.

## 4. Conclusion

This work proposes an upgrade of the proposed metric in LSBC metric. We built the proposed metric on the eigenvector centrality of the node and a state-of-the-art local likeness measure (LM). The proposed metric was used to enhance the performance of Jaccard, Adamic/Adar, and Resource allocation. We examined the performance of the proposed metric using the AUC value on real-world datasets from diverse fields and compared its results with the current metrics. The finding of this study reveals that the proposed metric has outstanding performance over the local similarity metrics in all networks. It captures the interactions between unrelated nodes, even if they do not have shared neighbours.

## References


[1] A.L Barabási et al., "Evolution of the Social Network of Scientific Collaborations," *Physical: Statistical Mechanics and its Applications*, vol. 311, no. 3-4, pp. 590-614, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[2] M. E. J. Newman, "Clustering and Preferential Attachment in Growing Networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[3] P. Jaccard, "Comparative study of the Floral Distribution in a Portion of the Alps and the Jura," *Bulletins – Vaud Society of Natural Sciences*, vol. 37, pp. 547-579, 1901. [Google Scholar]

[4] Lada A Adamic, and Eytan Adar, "Friends and Neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211-230, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[5] E. Ravasz et al., "Hierarchical Organization of Modularity in Metabolic Networks," *Science*, vol. 297, no. 5586, pp. 1551-1555, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[6] E. A. Leicht, Petter Holme, and M. E. J. Newman, "Vertex Similarity in Networks," *Physical Review E*, vol. 73, no. 2, p. 026120, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[7] G. Salton, and M. J. McGill, *Introduction to Modern Information Retrieval*, 1986.

[8] T. A. Sorensen, "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons," *Biol. Skar.*, vol. 5, pp. 1-34, 1948. [Google Scholar]

[9] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang, "Predicting Missing Links via Local Information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623-630, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ryan A. Rossi, and Nesreen K. Ahmed, "The Network Data Repository with Interactive Graph Analytics and Visualization," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jerome Kunegis, "KONECT – The Koblenz Network Collection," *Proceedings of the 22nd International Conference on World Wide Web*, pp. 1343–1350, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[12] M. Girvan, and M. E. J. Newman, "Network of American Football Games Between Division IA Colleges during Regular Season Fall 2000," *Proceedings of National Acadamic Science*, USA 99, pp. 7821-7826, 2002. [Publisher Link]

[13] Vladimir Batagelj, and Andrej Mrvar, Spider Datasets, 2006. [Online]. Available: http://vlado.fmf.uni−lj.si/pub/networks/data/

[14] Duncan J. Watts, and Steven H. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, vol. 393, no. 1, pp. 440–442, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[15] Robert E. Ulanowicz, Cristina Bondavalli, and Michael S. Egnotovich, *Network Analysis of Trophic Dynamics in South Florida Ecosystems–The Florida Bay Ecosystem*, Annual Report to the U.S. Geological Survey, Biological Resources Division, 1997.

[16] Shiwei Sun et al., "Topological Structure Analysis of the Protein-Protein Interaction Network in Budding Yeast," *Nucleic Acids Research*, vol. 31, no. 9, pp. 2443-2450, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[17] Software Package Protein Interaction Network

[18] Vito Latora, and Massimo Marchiori, "Efficient Behavior of Smallworld Networks," *Physical Review Letters*, vol. 87, no. 19, p. 198701, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jari Saramäki et al., "Generalizations of the Clustering Coefficient to Weighted Complex Networks," *Physical Review E*, vol. 75, no. 2, p. 027105, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[20] Marcus Kaiser, "Mean Clustering Coefficients: The Role of Isolated Nodes and Leafs on Clustering Measures for Small-World Networks," *New Journal of Physics*, vol. 10, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jibouni Ayoub, Dounia Lotfi, and Ahmed Hammouch, "Link Prediction Using Betweenness Centrality and Graph Neural Networks," *Social Network Analysis and Mining*, vol. 13, no. 1, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Manuel Guerrero et al., "Adaptive Community Detection in Complex Networks Using Genetic Algorithms," *Neurocomputing*, vol. 266, pp. 101-113, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Duanbing Chen et al., “Identifying Influential Nodes in Complex Networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 4, pp. 1777-1787, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[24] Fragkiskos D. Malliaros, Maria-Evgenia G. Rossi, and Michalis Vazirgiannis, “Locating Influential Nodes in Complex Networks,” *Scientific Reports*, vol. 6, no. 1, pp. 1-10, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] Sang-Uk Jung et al., "Predicting Neologisms for Marketing:A Text Mining Approach," *SSRG International Journal of Economics and Management Studies*, vol. 7, no. 7, pp. 5-9, 2020. [CrossRef] [Publisher Link]

[26] Fataneh Dabaghi Zarandi, and Marjan Kuchaki Rafsanjani, “Community Detection in Complex Networks Using Structural Similarities,” *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 882-891, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[27] Ilham Esslimani, Armelle Brun, and Anne Boyer, “Densifying a Behavioral Recommender System by Social Networks Link Prediction Methods,” *Social Network Analysis and Mining*, vol. 1, no. 3, pp. 159–172, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[28] Zan Huang, Xin Li, and Hsinchun Chen, “Link Prediction Approach to Collaborative Filtering,” *Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, pp. 141-142, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[29] Francesco Folino, and Clara Pizzuti, “Link Prediction Approaches for Disease Networks,” *International Conference on Information Technology in Bio-and Medical Informatics*, Springer, Berlin, Heidelberg, pp. 99-108, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[30] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero, “A Survey of Link Prediction in Complex Networks,” *Acm Computing Surveys*, vol. 49, no. 4, pp. 1-33, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[31] Linyuan Lü, and Tao Zhou, “Link Prediction in Complex Networks: A Survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150-1170, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[32] Mohammad Al Hasan, and Mohammed J. Zaki, “A Survey of Link Prediction in Social Networks,” *Social Network Data Analytics*, Springer, Boston, MA, pp. 243-275, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[33] Mohammad Al Hasan et al., “Link Prediction Using Supervised Learning,” SDM06: Workshop On Link Analysis, Counterterrorism, and Security, vol. 30, pp. 798-805, 2006. [Google Scholar] [Publisher Link]

[34] Tadej Matek, and Svit Timej Zebec, “Github Open-Source Project Recommendation System,” *arXiv preprint arXiv:1602.02594*, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[35] Iftikhar Ahmad et al., “Missing Link Prediction Using Common Neighbor asnd Centrality Based Parameterized Algorithm,” *Scientific Reports*, vol. 10, no. 1, pp. 1-9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Charles R. Harris et al., “Array Programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020. [CrossRef] [Google Scholar] [Publisher Link]