

# LPCNN: convolutional neural network for link prediction based on network structured features

Asia Mahdi Naser Alzubaidi<sup>1</sup>, Elham Mohammed Thabit A. Alsaadi<sup>2</sup>

<sup>1</sup>Department of Computer Science, Faculty of Computer Science and Information Technology, University of Kerbala, Kerbala, Iraq

<sup>2</sup>Department of Information Technology, Faculty of Computer Science and Information Technology, University of Kerbala, Kerbala, Iraq

---

## Article Info

### Article history:

Received Jan 17, 2022

Revised Aug 24, 2022

Accepted Sep 03, 2022

---

### Keywords:

Convolutional neural network

Deep learning

Heuristic scores

Social network analysis

Structural information

---

## ABSTRACT

In a social network (SN), link prediction (LP) is the process of estimating whether a link will exist in the future. In prior LP papers, heuristics score techniques were used. Recent state-of-the-art studies, like Wesfeiler-Lehman neural machine (WLNLM) and learning from subgraphs, embeddings, and attributes for link prediction (SEAL), have demonstrated that heuristics scores may increase LP model accuracy by employing deep learning and sub-graphing techniques. WLNLM and SEAL, on the other hand, have some limitations and perform poorly in some kinds of SNs. The goal of this research is to present a new framework for enhancing the effectiveness of LP models throughout various types of social networks while overcoming the constraints of earlier techniques. We present the link prediction based convolutional neural network (LPCNN) framework, which uses deep learning techniques to examine common neighbors and predict relations. Adapts the LP task into an image classification issue and classifies the links using a convolutional neural network. On 10 various types of real-work networks, tested the suggested LP model and compared its performance to heuristics and state-of-the-art approaches. Results revealed that our model outperforms the other LP benchmark approaches with an average area under curved (AUC) above 99%.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

## Corresponding Author:

Asia Mahdi Naser Alzubaidi

Department of Computer Science, Faculty of Computer Science and Information Technology

University of Kerbala, Kerbala, Iraq

Email: asia.m@uokerbala.edu.iq

---

## 1. INTRODUCTION

Social networks are a common way of simulating user interactions in each community. It may be shown as a social graph, with each node representing a network member and each edge signifying the sort of interaction between involved individuals [1]. Link prediction (LP) is a subfield of social network analysis that determines if two nodes in a network are more likely to link soon. LP can be applied in various domains like knowledge graph completion, information retrieval to analyze the hyperlink structure of the web and recommendation frameworks to propose modern companions or common interests, bioinformatics within the think about of the protein-protein interaction network, connected examination, link analysis, and mining for recognizing hidden criminals in terrorist networks and e-commerce to facilitate purchasing the value of the customer by recommending products to consumers via over-targeting on past basis Purchase history and general customer data [2].

Various heuristics methods were proposed in early research to handle the link prediction problem from different areas, which finds proximity between potential nodes and predicts link presence based on the metrics. In contrast, heuristic techniques performed well in other social networks, such as protein-protein interaction

networks, where two proteins with many common neighbors have a low chance of interacting [3]. Recent research proposes latent approaches for improving link prediction accuracy. Those techniques, on the other hand, may be able to advance accuracy in certain sorts of social networks, but they fared worse in others than the simple heuristics approach [4]. Generally, the probable LP task is mainly considered from two views: structure-based prediction and features of nodes-based and edges-based prediction. Where network structure indicates how the network's nodes are organized fundamentally according to the popular notion that the more similar a node pair is, the more likely they are to connect [5].

The LP also has been considered from the view of the learning-based that referred to features of nodes in the graph followed machine learning approaches included decision tree, support vector machine (SVM), Naïve Bayes, deep learning, convolutional neural network (CNN), graph neural network, and random forest [6]. Despite improvements in prediction accuracy in similarity-based methods, balancing performance and computational complexity for attributes-based metrics is difficult. A social network is fed into the framework, which then analyses the data and makes predictions. Each step aims to address the limits of existing methodologies and to solve them in a novel way [7], [8].

Earlier studies concentrated on utilizing topological features to generate similarity scores and predict connections. Heuristics are the most basic and straightforward but effective techniques for the link prediction task. It computes specific heuristic node similarity scores as the probability of influences [9]. Existing heuristics, for example, maybe considered based on the number of neighbors that are necessary to compute the score. First-order heuristics like common neighbors and preferred attachment only contain one-hop neighbors of two chosen nodes. The common neighbor soundarajan hopcroft and cosine index are all first-order heuristics. Adamic-Adar and resource allocation, on the other hand, are second-order heuristics because they're based on the target nodes up to two-hop proximity. In addition, high-order heuristic methods frequently outperform low-order heuristic approaches, although they have a higher computational cost. Because numerous heuristic techniques have been developed to handle various graphs, finding a suitable heuristic approach becomes a difficult task [10]. Also, extended weighted common neighbors (EWCN), extended weighted Adamic-Adar (EWAA), extended weighted Jaccard coefficient (EWJC), and extended weighted preferential attachment (EWPA) are used in [11]. Furthermore, experiments employing well-known classification methods such as the J48 decision tree, weighted SVM, Gboost, Naïve Bayes, random forest, logistic regression, and extreme gradient boosting (XGBoost) showed that the expanded metrics considerably improved the output of all supervised approaches in every validation dataset. Furthermore, the underlying difficulty with heuristic techniques is that they do not function consistently across networks since they rely on derived characteristics from network topology, which varies from one social network to the next.

Based on node embeddings, the similarity between pairs of nodes may also be calculated using node embeddings. As a result, embedding algorithms that can learn node features from network topology have been employed to resolve the LP issue; notable approaches in this line include matrix factorization and stochastic block modeling (SBM) [12]. SBM, on the other hand, is computationally costly and only works on specific types of social networks. Social representation of a graph's nodes, via modeling a stream of short random walks (DeepWalk) [13], large-scale information network embedding (LINE) [14], and node to vector representation (node2vec) [15] have been proposed as approaches for learning node embedding utilize the skip-gram approach that inspired from word embedding method used in natural language processing. The variational graph auto-encoder (VGAE) is an unsupervised learning model that employed the variational auto-encoder to analyze the pattern of graph structure data [16]. Latent techniques may learn useful features from the graph and hence perform well in the LP challenge. However, if the graph gets exceedingly sparse, the efficiency of the node link prediction based on embedding approaches may suffer.

Deep learning (DL), a novel way in machine learning, has been recently depicted in the literature. DL was used for learning the distribution of associates from a graph and developed to overcome the limitations of heuristic methods. One issue with traditional deep learning models is that the input is distributed independently and equally, which makes it unable to represent relational data. To address this issue, a bayesian deep learning framework that successfully learns relational data are suggested [17]. Predicting links by analyzing common neighbors (PLACN) a methodology based on convolutional neural networks is introduced and compared their technique to the state-of-the-art method, achieving 96% area under curve (AUC) in the benchmark [18]. Because of its accuracy, a subgraph technique known as Weisfeiler-Lehman neural machine (WLNLM) was recently designated as a state-of-the-art link prediction method [19]. To attain significant accuracy, the WLNLM utilized high-order algorithms like the Katz index and PageRank. This, on the other hand, necessitates many hops from the enclosing subgraph to the complete network, as well as additional calculation of time and memory. To address this issue, learning from subgraphs, embeddings, and attributes for link prediction (SEAL) presented a way to learn general graph structure features from local enclosing subgraphs using graph neural networks [20]. They computed first-order, second order, and high-order heuristic scores to create a vector of the feature. The authors used the double-radius technique to

organize nodes. Lastly, to categorize links, graph neural networks, the adjacency matrix, and the latent vector are employed. The SEAL model yields state-of-the-art performance for the link prediction issue due to the exceptional learning of graph neural networks. Determining an appropriate hop number for a given network, on the other hand, is a trial-and-error process, and putting all neighbor nodes in the subgraph raises the issue that hub nodes have many neighbors even at low hop numbers. Another issue with SEAL is that pooling layers miss topological information, and graph convolution layers fail to learn edge embeddings.

People communicate with one another through social media apps. The graph theory concept can be used to solve the problem of link prediction, which is a recent research approach. A social network is represented as a graph  $G(V, E)$  at any given time, where  $V$  and  $E$  are sets of nodes and links respectively. The goal of link prediction is to predict the missing or undetected links in the existing network, as well as future or removed connections between nodes for a period  $t'$  in the future. A potential link prediction task can be described by a simple social network to show the evolution of links as depicted in Figure 1, where solid links represent previously existing connections and dotted links indicate links that have recently arisen due to link classification algorithms.

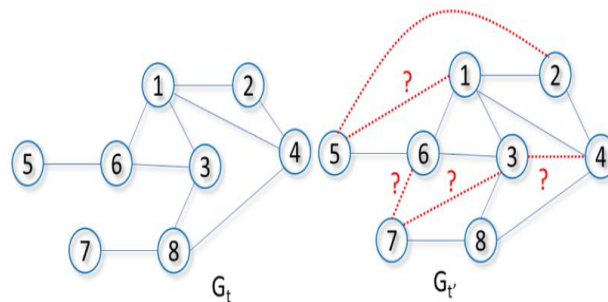


Figure 1. An example of link prediction

Social networks are extremely dynamic objects in reality; they expand and evolve over time as a result of the addition of new edges, which represent the emergence of new interactions in the underlying social structure. For example, in the case of Facebook, with their “friend discoverer” merit, they can also recommend the user you might be interested in and the relationship may lead to a real-life friendship that will enhance both parties’ commitment to the Facebook service. In the research community, the issue of link prediction has gained a lot of interest. However, researchers mostly focused on making predictions about how a social network may expand by including new ties. In other words, the majority of earlier studies on link prediction either restricted their research to the prediction of links that will be added to the network during the period from the present time to a specific future time or implicitly devoted the link prediction to specific domains like co-authorship. This manuscript makes the following contributions:

- 1) To find a novel LP approach that can learn how to optimally combine heuristic scores to improve the performance of the LP model in any network and not only in a size and type of network.
- 2) Create feature adjacency matrices as layers for a specific social network to obtain evidence about targeted nodes and common neighbors. We use a total of eight heuristic features, such as the Jaccard index, the Adar index, resource allocation, preferential attachment, and others.
- 3) The LP problem may be transformed into an image classification problem by using CNN to train and classify the positive and negative links.

The rest of the paper is organized as: section 2 specifies the link prediction framework. the experimental data and analyses are then presented in section 3. Finally, in section 4, we offer some concluding remarks and suggestions for further works.

## 2. METHOD

In network analysis, link prediction is a major study area. In recent years, new strategies based on graph features have emerged as effective methods for determining heuristic scores. The heuristic’s score is directly utilized to rank node pairings. They can also be used for supervised prediction by combining them with a classifier and using some or all of them as features. We combine all heuristics with a convolution neural network in this paper. This method of supervision has been proved to produce the best results. In this part, we want to get a better understanding of the mechanics underlying various link prediction heuristics metrics, therefore encouraging the notion of learning heuristics from graphs.

## 2.1. Heuristic methods

Heuristic link prediction approaches are often based on topological structures, which can be node-based, neighbor-based and path-based, or random walk-based. Neighbor-based heuristic approaches include Jaccard coefficient, Adamic-Adar coefficient, preferential attachment, and resource allocation, while additional topological heuristic methods, also referred to as higher-order heuristic methods, include Katz index, PageRank, and simRank. Path-based metrics take into account more topological information than node-based and neighbor-based metrics, which solely employ local topological information. These metrics take into account paths between node pairs in addition to local neighbors and other significant global information. Path-based metrics have a larger time complexity than neighbor-based ones. However, longer links aren't necessarily better than shorter ones. Longer paths should only be taken into account in the theoretical basis of well-known link prediction heuristics metrics if shorter paths are insufficiently common. As a result, if networks contain enough shorter paths, path-based metrics can improve their performances by omitting excessively long paths. In our model, we solely examine the neighbor-based heuristic approaches as listed below.

### 2.1.1. Jaccard's coefficient (JC)

We shall utilize the Jaccard coefficient to compute the similarity of node pairs in unweighted networks in this paper. For an arbitrarily selection attribute that may be  $X$  or  $Y$ , this method calculates the probability that both  $X$  and  $Y$  have it. This metric solves the problem of two nodes having more different neighbors ( $\Gamma$ ) simply because they have a lot of neighbors and are closely connected [21].

$$S_{JC}(X, Y) = |\Gamma(X) \cap \Gamma(Y)| / |\Gamma(X) \cup \Gamma(Y)| \quad (1)$$

### 2.1.2. Cosine (Salton) index (CI)

The Salton similarity is strongly connected to the Jaccard index, but it generates a value that is approximately double that of the Jaccard index. This approach would perform the vector multiplication for each pair of nodes with common neighbors.

$$S_{CI}(X, Y) = \frac{r(x)|r(y)}{\|r(x)\| \times \|r(y)\|} \quad (2)$$

### 2.1.3. Adamic-Adar (AA)

Adamic Adar index used to rank features based on their logarithm and uses these features to measure the prediction scores. This metric can be understood in a real-world social network as: if a common neighbor of two nodes has more friends, they are less likely to expose the two nodes to each other than if they have just a few friends [22].

$$S_{AA}(X, Y) = \sum_{z \in \Gamma(X) \cap \Gamma(Y)} \frac{1}{\log|\Gamma(z)|} \quad (3)$$

Where  $Z$  represents the common neighbors between  $X$  and  $Y$ .

### 2.1.4. Resource allocation (RA)

This index is somewhat like Adamic-Adar, but it does not use the logarithm function, which decreases the influence of nodes to a large degree. Since these nodes are linked to so many other nodes in the graph, they have little insight for relation prediction[23].

$$S_{RA}(X, Y) = \sum_{z \in \Gamma(X) \cap \Gamma(Y)} \frac{1}{|\Gamma(z)|} \quad (4)$$

### 2.1.5. Preferential attachment (PA)

The PA was chosen as a prediction technique since a node with a high score is more likely to get new connections. This index is a fundamental prediction tool that is commonly used as a baseline to assess the efficiency of other prediction approaches. Rather than only the neighboring nodes, it computes a similarity score for each pair of nodes in the network [24].

$$S_{PA}(X, Y) = \Gamma(X) \times \Gamma(Y) \quad (5)$$

### 2.1.6. SimRank (SR)

The SR is based on the notion that two nodes are similar if their neighbors are similar as well. SimRank of two nodes ( $X, Y$ ) is recursively computed as:

$$S_{SR}(X, Y) = C \frac{\sum_{Z \in \Gamma(X)} \sum_{Z' \in \Gamma(Y)} S_{SimRink}(Z, Z')}{k_X \cdot k_Y} \quad (6)$$

Where  $Z$  is the set of neighbors of node  $X$  and  $Z'$  is the set of neighbors of node  $Y$ .  $C \in [0 \dots 1]$  is the decay factor [25].

### 2.1.7. PageRank (PR)

The stationary distribution of a random walker starts at  $X$  and iteratively moves to a random neighbor of its current position with probability  $\alpha$  or return to  $X$  with probability  $1 - \alpha$  calculated by the PageRank of node  $X$ .

$$S_{PR}(X) = \sum_{X \in Bv} \frac{PR(v)}{L(v)} \quad (7)$$

The PageRank value for a node  $v$  can be calculated by dividing the PageRank of each node  $X$  in the set  $Bv$  to the number  $L(v)$  of links from node  $v$  [26].

### 2.1.8. Katz index (KI)

Katz centrality calculates a node's relative influence in a network by counting its direct neighbors and all other nodes in the network that are connected to it through these common neighbors. Influences made with distant neighbors are, however, penalized by an attenuation factor  $\alpha$ . Each link between a pair of nodes is assigned a weight value determined by  $\alpha$  and the distance between nodes as  $\alpha^K$ .

$$S_{KT}(X) = \sum_{K=1} \sum_{Y=1} \alpha^K (A^K)_{YX} \quad (8)$$

Where the element at location  $(X, Y)$  of the adjacency matrix  $A$  raised to the power  $k$  (i.e.  $A^{\{K\}}$ ) reflects the total number of  $k$  degree connections between nodes  $X$  and  $Y$  [27].

## 2.2. Research method

The suggested framework termed link prediction based convolutional neural network (LPCNN) attempts to improve the efficiency of LP models in social networks by studying the common neighbors of targeted nodes. The LPCNN model also aims to create a methodology that can adjust to any size and sort of social network and automate the learning of the optimal combination of heuristic scores for that networks. LPCNN proposed a new model that combines heuristic features to generate adjacency matrices and then classifies the positive and negative links using CNN. The LPCNN model uses the feature of CNN, which is greatest identified for image classification, to map the link prediction task into an image classification issue. Another reason is that earlier studies only looked at targeted nodes for LP, but our technique looks at common neighbors between targeted nodes as well. Figure 2 depicts the LPCNN model's architecture as well as the link prediction process stages.

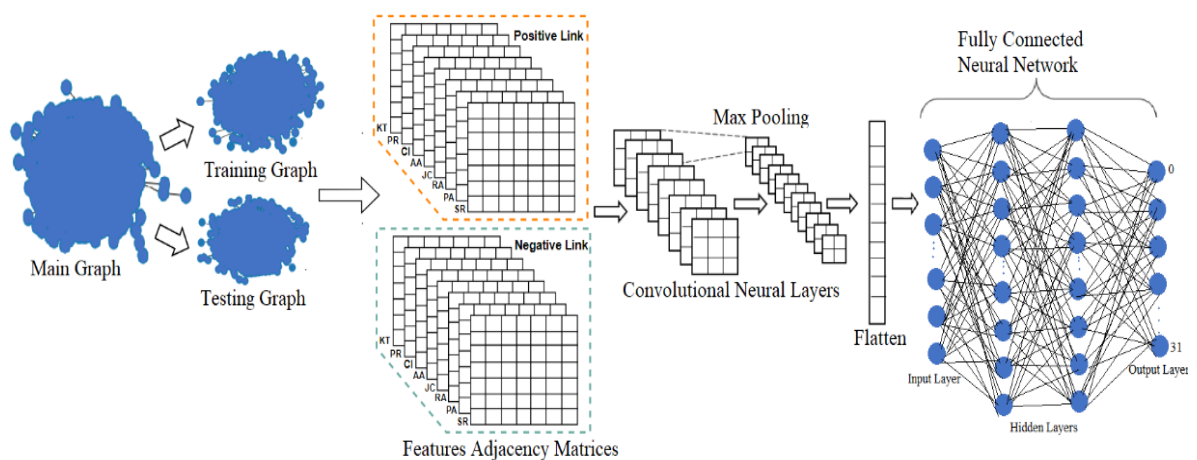


Figure 2. The Architecture of LPCNN for link prediction model

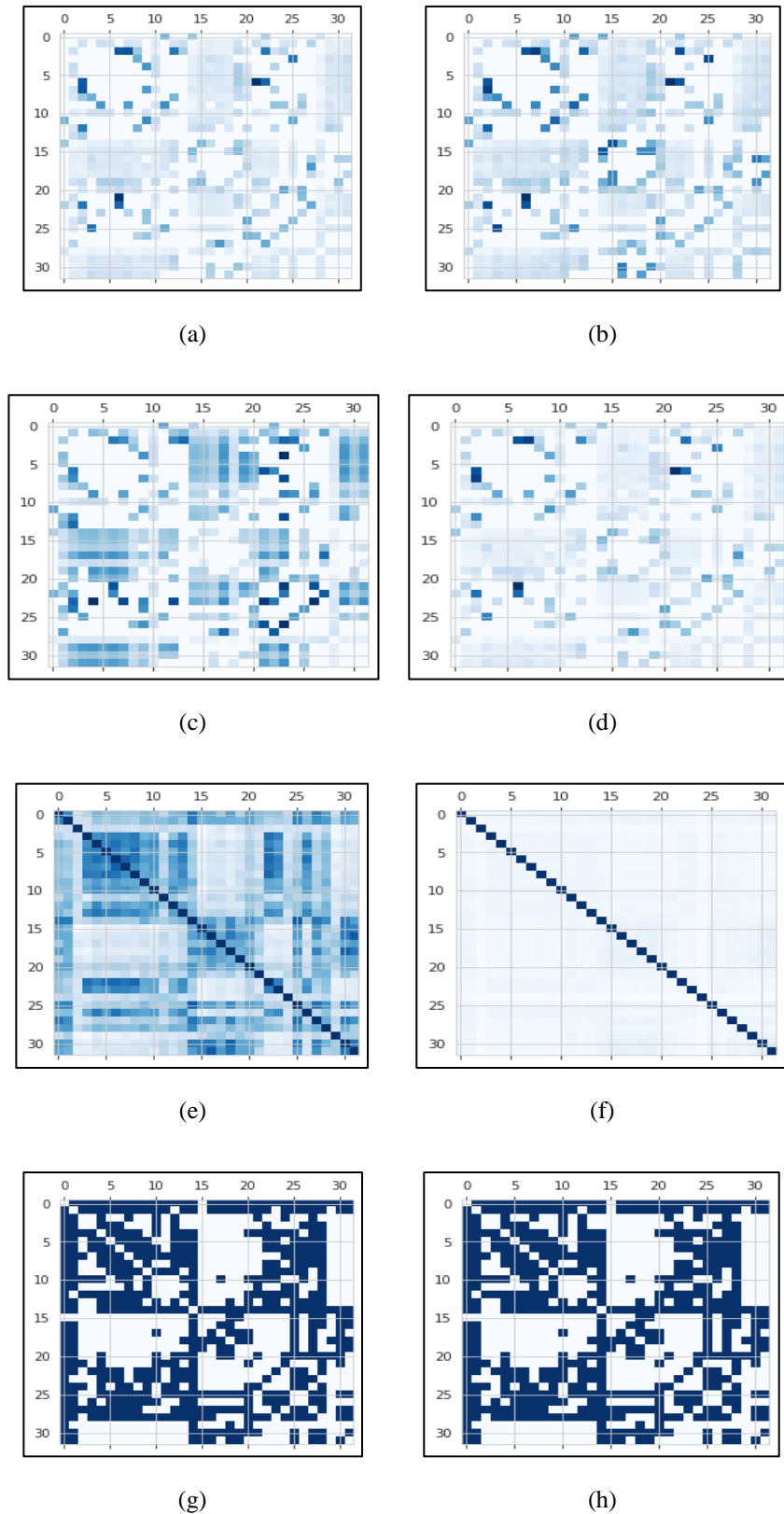


Figure 3. The eight selected heuristic scores as adjacency matrices where (a) Adamic-Adar, (b) Jaccard's coefficient, (c) preferential attachment, (d) resource allocation, (e) cosine index, (f) SimRank index, (g) Katz index, and (h) PageRank adjacency matrix

To begin, social network visualization as a graph representation for the original, training, and testing datasets are investigated by modeling the links between each pair of nodes. To construct the feature adjacency matrices, the proposed LPCNN framework takes a given network as input and creates dataset of start and goal nodes with the labels for positive link class (a link that will occur soon) and negative link class (a link that will not appear soon). Each feature adjacency matrix of the training and testing graphs can be represented as  $A_{i,j}^{KT}$ ,  $A_{i,j}^{PR}$ ,  $A_{i,j}^{CI}$ ,  $A_{i,j}^{AA}$ ,  $A_{i,j}^{JC}$ ,  $A_{i,j}^{PA}$ , and  $A_{i,j}^{SR}$  where  $(i,j)$  in range of  $(1:K)$  and ordered in the sequence of each of the heuristic methods. Our LPCNN framework tends to improve the accurateness of link prediction by analyzing common neighbors between targeted nodes.

To evaluate nodes and their relationships with the goal node, we must compute the features of common nodes. Heuristic scores are similarity ratings that quantify the degree to which two nodes are similar. In the past, researchers produced different heuristic scores between just specified nodes and attempted to estimate the links based on the results. Some heuristic scores outperformed others in certain types of social networks. A heuristic score combination surpassed a single heuristic score. As first-order techniques, the suggested LP model takes into account eight distinct heuristic scores such as Jaccard's coefficient, cosine score, and preferential attachment. While resource allocation, Adamic Adar index, and Katz index are second-order approaches, and PageRank, SimRank index, and SimRank are high-order methods. Heuristics scores indicated above are obtained for all nodes in training and testing graphs. Each heuristic method used to construct feature adjacency matrix then eight feature adjacency matrices will be stacked by using eight scores. The final feature matrix will be  $N \times K \times K \times 8$  in size for each score, where  $N$  is the size of training and testing datasets and  $K$  is the size of the image. They're also symmetric, and diagonal values are 0 because they represent nodes that link to one other.

CNN is well-known for its ability to classify images. To tackle the LP problem, we use CNN's features. By generating feature matrices with different heuristic scores, our LPCNN model changes the LP issue into an image classification task. Images have three channels, red, green, and blue (RGB) are handled as a three-dimensional matrix while the constructed feature adjacency matrices are  $N \times K \times K$  images with 8 channels and  $K = 32$ . Figure 3 showed eight selected heuristic scores as image from adjacency matrices where Figure 3(a) Adamic-Adar, Figure 3(b) Jaccard's coefficient, Figure 3(c) preferential attachment, Figure 3(d) resource allocation, Figure 3(e) cosine index, Figure 3(f) SimRank index, Figure 3(g) Katz index, and Figure 3(h) PageRank.

We train CNN to distinguish between two types of links: positive and negative. Backpropagation with loss function is used to optimize a neural network during training. The loss function is different depending on the problem. We choose binary cross-entropy as the loss function and area AUC as the CNN monitor because this is a binary classification task.

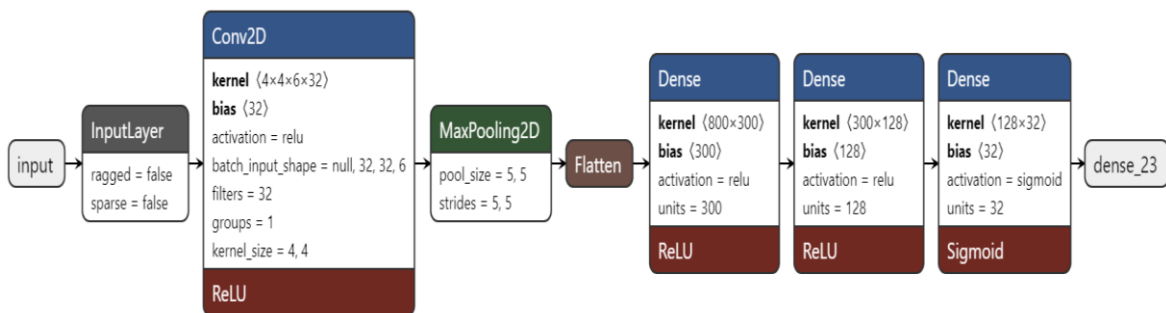


Figure 4. The convolution neural network configurations

To perform classification, CNN has multiple layers. The input layer is the primary layer, followed by one or more convolutional hidden layers and pooling layers. Finally, there are one or more dense layers and an output layer. The convolutional layer is distinct because it collects information from input matrices. The activation mapping describes how the neurons in this layer are organized in two-dimensional arrays. The kernel is a three-dimensional array that holds the weights. For different input sizes, the height, width, and depth can be modified. Even for the identical input matrix, several kernels of various sizes can exist. Kernel develops a series of activation maps by sliding through the input matrix. Figure 4 shows the CNN layers and their parameters in detail.

### 3. RESULTS AND DISCUSSION

The suggested LP model (LPCNN) aims to fit different networks while also increasing the link prediction model's efficiency. We utilized 80% of the data set to train the model and 20% to test it over 150 epochs. After training, we loaded the best model and reviewed the results. Running the LP model five times and selecting the average result of AUC as the best-given prediction model.

We conduct tests using real-world networks to evaluate the proposed model, and we use the area under curve-receiver operating characteristics (AUC-ROC) as the assessment metric. A ROC curve is a graph that shows the probability of true positive rate and false positive rate at various threshold settings. The AUC reflects a model's level of distinction. The AUC value varies between 0 and 1. When the AUC reaches one, the model can predict positive classes as positive and negative classes as negative. A larger AUC implies that the supplied model is not making random predictions and has been trained to find a pattern to classify the missing links.

In our LP model we not relied on calculating the time complexity of the proposed model, because the methods that were compared with did not calculated the time complexity, and most of used networks are not very large scale in size. By dividing links into two classes, the positive class, and the negative class, our approach handles link prediction as a binary classification problem. The positive class indicates that the given link will be available in the future, whereas the negative class indicates that the link will not be available soon.

#### 3.1. Real-world datasets

We compare the proposed missing link prediction model with state-of-the-art methodologies using ten diverse kinds of real-world network datasets from diverse locations and sizes. All the datasets are freely available on the internet. Table 1 lists the datasets that have been tested, with  $|V|$  and  $|E|$  representing the number of nodes and edges respectively,  $\langle k \rangle$  representing the average degree, and  $CC$  representing the clustering coefficient.

Table 1. The Basic topological features of the networks

#	Datasets	Size		$\langle K \rangle$	$\langle CC \rangle$	Type
		$ V $	$ E $			
1.	NSC	1461	2742	3.7536	0.694	Co-authorship
2.	Yeast	2375	11693	9.8467	0.306	Biology network
3.	Power	4941	6594	2.6691	0.080	Electrical grid network
4.	PB	1222	16714	27.3552	0.320	US political
5.	Router	5022	6258	2.4922	0.012	Internet routing
6.	E.coli	1805	14660	16.2438	0.516	Pairwise reaction
7.	Facebook	4039	88234	43.6910	0.606	Friendship network
8.	Wikipedia	4777	92517	38.6879	0.359	Online encyclopedia
9.	PPI	3890	38739	19.9172	0.146	Protein-protein interactions
10.	USAir	332	2126	12.8072	0.625	Transportation dataset

#### 3.2. Results and analysis

We compare our LP model (LPCNN) against the state-of-the-art methods: WLN and SEAL. To evaluate the LPCNN model, experimental and analysis the results are conducted. Our proposed framework provides efficient model for link prediction and delivers a best performance at variety of networks. Table 2 depicted the results of AUC. We compared LPCNN's performance to first-order heuristics methods like common neighbors, Jaccard coefficient, cosine index, and preferential attachment, second-order heuristics like Adamic Adar and resource allocation, high-order heuristics like Katz and PageRank, and finally, state-of-the-art methods like WLN and SEAL. Figure 5 is the chart plot of the average AUC score of all used methods on five test runs on each dataset.

Some global indices such as SimRank and Katz index are not performing well and displaying lower performance than most of the other heuristic approaches. Regardless of whether intra-dominant or inter-dominant, similarity indices better as the structural gap between intra-connection and inter-connection expands. However, in inter-dominant structures, several global indices such as Katz index perform poorly. The number of paths between node pairs is the basis for these indices. Due to the decay factor of these indices, connected pairs have a larger similarity than unconnected pairs. As a result, in the inter-dominant condition, there are more node pairs of different types with higher similarity, which results in subpar performances. However, heuristic scores such as PageRank and Adamic-Adar exhibit great performance when compared to other heuristics that show similar results. Moreover, most of heuristics techniques perform worse than the state-of-the-art subgraphing methods WLN and SEAL.



We can see that our LPCNN model outperforms all other approaches and has superior graph feature learning ability overall heuristics and subgraphing methods. This means that discovering and concatenating new heuristics scores for networks to catch more and more of their structural properties and utilizing them in the learning classifier where no existing heuristics work can improve model performance dramatically. Furthermore, due to the range of network sizes, the experimental results of similarity-based link prediction revealed that slight variances in overall AUC values, such as those found in the USAir dataset, do not always imply low predictability for that dataset. Because the AUC is based on the precision-recall curve, predicting a larger number of links increases the danger of false positives, as the number of new links generated by a network may not keep up with its growth. Furthermore, because they are built manually, existing LP algorithms based on similarity are unable to properly represent several non-linear modes that play a critical role in the LP network.

Table 2. Comparison of AUC with state-of-art and heuristic methods

Dataset	JC	CI	AA	RA	PA	SR	PR	KT	WLMN	SEAL	LPCNN
NSC	94.43	95.55	94.45	94.45	68.65	94.79	94.89	94.85	98.61	98.85	99.03
Yeast	89.32	89.20	89.43	89.45	82.20	91.49	92.76	92.24	95.62	97.91	99.67
Power	58.79	60.37	58.79	58.79	44.33	76.15	66.00	65.39	84.76	87.61	99.56
PB	87.41	85.64	92.36	92.46	90.14	77.08	93.54	92.92	93.49	94.72	98.83
Router	56.40	56.4	56.43	56.43	47.58	37.40	38.76	38.62	94.41	96.38	99.31
E.coli	80.19	84.538	86.95	87.49	74.79	77.07	90.32	86.34	97.21	97.64	99.70
Facebook	98.56	98.62	98.88	8.929	83.09	96.31	96.31	98.93	99.24	99.40	99.65
Wikipedia	40.75	47.19	90.46	91.63	91.70	28.64	96.31	50.0	99.05	99.63	99.76
PPI	81.42	81.36	83.57	83.53	89.42	82.70	50.0	50.0	88.79	93.52	99.44
USAir	89.79	88.619	95.06	95.77	88.84	78.89	94.67	92.88	95.95	96.62	95.73
Average	77.706	78.749	84.638	75.893	76.074	74.052	81.356	76.217	94.713	96.23	99.068

Although the shallow neural network-based LP technique may make effective use of the network nodes' potential features, it cannot capture the deeply non-linear attributes such as link structural features. Because of their excellent attribute learning capacity, CNNs can capture deeply non-linear data and learn more valuable features, improving LP model output. As a result, we predict the missing link using a deep CNN. In general, the research results show that our model can successfully capture the most relevant missing LP information in many situations, implying that combining the power of eight different types of heuristic attributes with CNN learning can result in significantly better model performance than subgraphing methods.

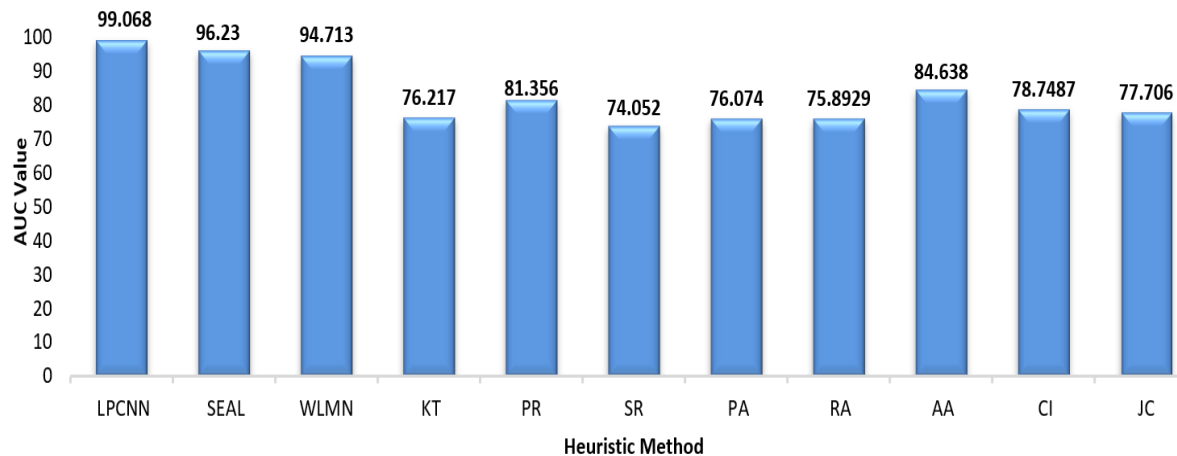


Figure 5. The average AUC score of all methods

#### 4. CONCLUSION

The goal of link prediction is to discover missing links in a network and predicate the future links. Link prediction attracted the interest from a variety of scientific disciplines as a key research issue in complex network analysis. Heuristics-based methods have gained the majority among diverse approaches due to their minimal complexity and excellent interpretability.

In this paper, we implemented extended feature extraction and generated eight different heuristics feature matrices. These features provide our framework autonomy in learning and adopting the topological patterns of diverse networks. LPCNN transforms the link prediction problem into an image classification problem, which CNN then classifies. Our model outperforms both state-of-the-art and the heuristic baseline approaches, according to AUC metric. However, we should agree that assessing associates using heuristics scores as features in the learning model is a reliable way to differentiate between the test and non-existent node edges. The goal of future research will be to improve node association predictions. By extracting and adding additional node features to the model instead of graph structure attributes may assist in improve the performance because it adds more information. Our methodology also opens new avenues for study into recommendation systems and knowledge graph completion.




## REFERENCES

- [1] M. A. Hasan and M. J. Zaki, "A Survey of Link Prediction in Social Network," *Social Network Data Analytics*, pp. 243–275, 2011, doi: 10.1007/978-1-4419-8462-3\_9.
- [2] N. Barbieri, F. Bonchi, and G. Manco, "Who to follow and why: link prediction with explanations," in *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 1266–1275, doi: 10.1145/2623330.2623733.
- [3] L. Cai, J. Li, J. Wang, and S. Ji, "Line Graph Neural Networks for Link Prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5103–5113, 2022 doi: 10.1109/TPAMI.2021.3080635.
- [4] P. M. Chuan, C. N. Giap, L. H. Son, C. Bhatt, and T. D. Khang, "Enhance link prediction in online social networks using similarity metrics, sampling, and classification," *Information Systems Design and Intelligent Applications*, vol. 672, pp. 823–833, 2018, doi: 10.1007/978-981-10-7512-4\_81.
- [5] S. Fu, W. Liu, K. Zhang, Y. Zhou, and D. Tao, "Semi-supervised classification by graph p-Laplacian convolutional networks," *Information Sciences*, vol. 560, pp. 92–106, 2021, doi: 10.1016/j.ins.2021.01.075.
- [6] F. Gao, K. Musial, C. Cooper, and S. Tsoka, "Link prediction methods and their accuracy for different social networks and network metrics," *Scientific Programming*, vol. 2015, 2015, doi: 10.1155/2015/172879.
- [7] S. Gaucher, O. Klopp, and G. Robin, "Outlier detection in networks with missing links," *Computational Statistics and Data Analysis*, vol. 164, 2021, doi: 10.1016/j.csda.2021.107308.
- [8] A. Gupta and Y. Raghav, "Deep Learning Roles based Approach to Link Prediction in Networks," *Computer Science and Information Technology (CS & IT)*, pp. 203–222, 2020, doi: 10.5121/csit.2020.101416.
- [9] D. L. -Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007, doi: 10.1002/asi.20591.
- [10] L. Lu and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011, doi: 10.1016/j.physa.2010.11.027.
- [11] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *Proc. of the 2014 SIAM International Conference on Data Mining (SDM)*, 2014, pp. 289–297, doi: 10.1137/1.9781611973440.33.
- [12] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec," in *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 459–467, 2018, doi:10.1145/3159652.3159706.
- [13] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, Aug. 2014, doi: 10.1145/2623330.2623732.
- [14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale Information Network Embedding," in *Proc. of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.
- [15] K. Prajapati, H. Shah, and R. Mehta, "A survey of link prediction in social network using deep learning approach" *International Journal of Scientific and Technology Research*, vol. 9, no. 4, pp. 2540–2543, 2020. [Online]. Available: <http://www.ijstr.org/final-print/apr2020/A-Survey-Of-Link-Prediction-In-Social-Network-Using-Deep-Learning-Approach.pdf>
- [16] K. Narang, K. Lerman, and P. Kumaraguru, "Network flows and the link prediction problem," in *Proc. of the 7th Workshop on Social Network Mining and Analysis*, 2013, no. 3, pp. 1–8, doi: 10.1145/2501025.2501031.
- [17] H. Wang, X. Shi, and D. -Y. Yeung, "Relational deep learning: A deep latent variable model for link prediction," in *Proc. of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017, vol. 31, no. 1, pp. 2688–2694, doi: 10.1609/aaai.v31i1.10805.
- [18] K. Raguathan, K. Selvarajah, and Z. Kobti, "Link prediction by analyzing common neighbors based subgraphs using convolutional neural network," *Frontiers in Artificial Intelligence and Applications*, vol. 325, pp. 1906–1913, 2020, doi: 10.3233/FAIA200308.
- [19] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 575–583. [Online]. Available: [https://muhanzhang.github.io/papers/KDD\\_2017.pdf](https://muhanzhang.github.io/papers/KDD_2017.pdf)
- [20] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018, pp. 5165–5175. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/file/53f0d7c537d99b3824f0f99d62ea2428-Paper.pdf>
- [21] P. Srilatha and R. Manjula, "Similarity index based link prediction algorithms in social networks: A survey," *Journal of Telecommunications and Information Technology*, pp. 87–94, 2016. [Online]. Available: <https://core.ac.uk/download/pdf/235205592.pdf>
- [22] S. A. Fadaee and M. A. Haeri, "Classification using link prediction," *Neurocomputing*, vol. 359, pp. 395–407, 2019, doi: 10.1016/j.neucom.2019.06.026.
- [23] Y. Xiao, R. Li, X. Lu, and Y. Liu, "Link prediction based on feature representation and fusion," *Information Sciences*, vol. 548, pp. 1–17, 2021, doi: 10.1016/j.ins.2020.09.039.
- [24] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link Prediction in Social Networks: the State-of-the-art," *Science China Information Sciences*, vol. 58, pp. 1–38, 2015, doi: 10.1007/s11432-014-5237-y.

- [25] R. Guns, "Link Prediction," in *Measuring scholarly impact*, Springer, 2018, pp. 35-55, doi: 10.1007/978-3-319-10377-8\_2.
- [26] L. Yao, L. Wang, L. Pan, and K. Yao, "Link Prediction Based on Common-Neighbors for Dynamic Social Network," *Procedia Computer Science*, vol. 83, pp. 82–89, 2016, doi: 10.1016/j.procs.2016.04.102.
- [27] I. Ahmad, M. U. Akhtar, S. Noor, and A. Shahnaz, "Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm," *Scientific Reports*, vol. 10, no. 364, 2020, doi: 10.1038/s41598-019-57304-y.

## BIOGRAPHIES OF AUTHORS



**Asia Mahdi Naser Alzubaidi**    received the BSc degree in computer science from the University of Babylon, Hilla, Iraq, and the master's degree in computer science/ Artificial Intelligence. The Ph.D from College of Information Technology at Babylon University-Iraq. She is currently an Assist Professor in Department of computer science, College of Computer Science & Information Technology, University of Kerbala. Her current research interests include Data Mining, Computer Vision, Natural Language Processing, and Customer Churn Prediction. She can be contacted at email: asia.m@uokerbala.edu.iq.



**Elham Mohammed Thabit A. Alsaadi**    received the BSc. in Computer Science from Al-Mustansiriya University -Iraq, MSc. in Real Time Systems from UK. PhD. from College of Information Technology at Babylon University-Iraq. She is a lecturer in College of Computer Science & Information Technology, Karbala University. Her research interests include: Artificial Intelligence, Computer Vision, Image Processing, Database, System Analysis, and E- business. She can be contacted at email: elham.thabit@uokerbala.edu.iq.