JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

jiit

विद्या तत्व ज्योतिसम:

MINOR PROJECT REPORT

Link Prediction In Social Network

Group Members:

Sakshi gupta (17104039)
Manvi Chawla(17104041)
Adhar Agrawal(17104068)

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our mentor Dr. Parul Agrawal for teaching us, helping us with the project as well as guiding us with the relevant resources, and ultimately providing us with an opportunity to make and present this report.
We are highly indebted to Dr Parul Agrawal for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.
We would like to express our gratitude towards my parents & member of Jaypee Institute Of Information Technology for their kind co-operation and encouragement which help me in completion of this project.

DECLARATION

We hereby declare that the project report is based on our own work carried out during the course of our study under the supervision of our mentor Ms. Parul Agrawal.

I. We assert the statements made and conclusions drawn are an outcome of our research work.

II. I further certify that the work contained in the report is original .The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Sakshi Gupta(17104039)
Manvi Chawla(17104041)
Adhar Agrawal(17104068)

Table of contents

I. Introduction

We have picked Link Prediction in social networks as our project. Currently, with the rapid development, the online social network has been a part of people's life. A lot of sociology, biology, and information systems can use the network to describe, in which nodes represent individual and edges represent the relationships between individuals or the interaction between individuals.

Therefore, the study of complex networks has been an important branch of many scientific fields. Link prediction is an important task in link mining. Link prediction is to predict whether there will be links between two nodes based on the attribute information and the observed existing link information. Link prediction not only can be used in the field of social networks but can also be applied in other fields.

Predicting missing links and links that may occur in the future in social networks is an attention grabbing topic amid the social network analysts. Owing to the relationship between human-based system and social sciences in this field, some machine learning algorithms mentioned ahead in the report can help us to model the systems more effectively.

Link prediction not only has a wide range of practical value but also has important theoretical significance. For example, it is helpful to understand the mechanism of the evolution of a complex network. Since the statistical magnitude to describe characteristics of the network structure is very large, it is difficult to compare the advantages and disadvantages of different mechanisms. Link prediction can provide a simple and unified platform for a fair comparison of network evolution mechanisms, to promote the theoretical research on a complex network evolution model.

In this work, we propose a framework of composite and mutual link prediction on the network. The problem of the link prediction addresses the sign of links among the composite entities in the network. We address this problem and it is resolved by machine learning and to construct the structural feature for the machine learning.

II. PROBLEM STATEMENT

Social networks are social structures including some actors and relationships amid them. These networks are presented by employing some nodes and ties. The ties show some type of relationships among the nodes including kinships, friendships, collaborations, and any other interactions between the people in the network. Link prediction is an important research field in data mining. It has a wide range of scenarios. Many data mining tasks involve the relationship between the objects. Link prediction can be used for recommendation systems, social networks, information retrieval, and many other fields.

Given a graph G={V, E} of the social network at a moment of the node and the other node, link prediction is to predict the probability of the link between the node and the other node.

The two **challenges** we will be facing are:

1) Predict that the new link will appear in future time.
2) Forecast hidden unknown link in the space.

In our project, we have combined all these aspects mentioned above in a single problem statement using different algorithms SVM(Support Vector Machine), ANN, Logistic regression, Fuzzy model, PSO, ACO, hybrid ACO & PSO. In the end, we are comparing the accuracy of all these algorithms on different datasets of different social media platforms.

II.I Mathematical description of problem statement

The link prediction problem is usually described as:
Given a set of data instances V = v in i=1 ,
which is organized in the form of a social network

G = (V,E)
where E is the set of observed links

Then the task to predict how likely an unobserved link $e_{ij} \notin E$ exists between an arbitrary pair of nodes v_i , v_j in the data network.

Given a snapshot of a social network at time t (or network evolution between (t_1 and t_2), seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t′.

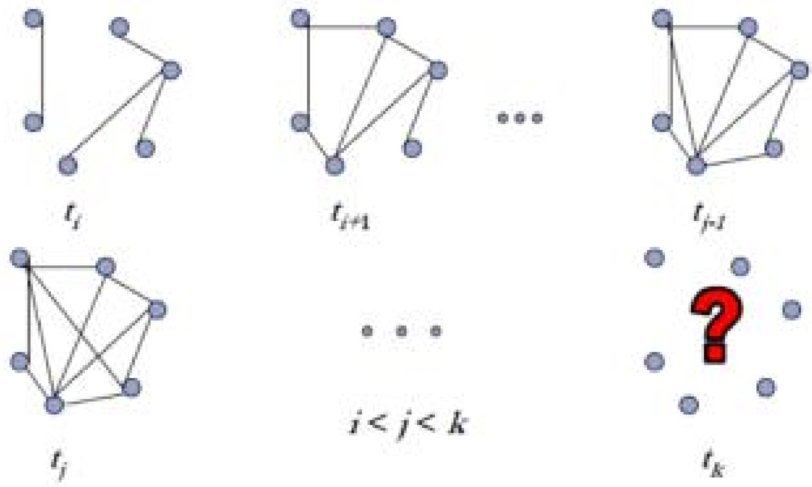


Fig 1. Visual Representation of Link Prediction Task

The easiest framework of link prediction algorithm is based on the similarity of the algorithm. Any pair of node and node,we have assigned to this node is a function,Similarly this function is defined as the similarity function between nodes and . Then sorting the nodes pair in accordance with the function values from the largest to smallest, the greater the value of the similarity function, the greater the probability of the link in the nodes.

III. Literature Review

Year	Title	Publication	Algorithm used	Data set	Advantages	Limitation
2013	Fuzzy Models for Link Prediction in Social Networks [1]	HKU scholars hub	Fuzzy networks -FCC and FCO	Data of facebook in form of graph	The accuracy of the predictions using fuzzy link prediction models is higher as compared to the results of the considered crisp models	The results also indicate that ANFIS is weak in predicting the strength of the links. It also seems that Mamdani inference engine cannot predict the strength of nodes due to the complexity of the social networks.
2016	A dynamic logistic regression for network link prediction [2]	Research gate	Logistic regression	Graph with random links	First, it is an interesting and novel time series model. Most classical time series literature focus on univariate time series. Recently Second that it is the simplest.	For rapidly changed network structure (e.g., a telecommunication) whether CLI can still perform competitively is not clear at this moment. Lastly, we assume here that β is fixed across different time points

2017	Time-Series Link Prediction Using Support Vector Machines [3]	Phillipine Journal of Science	Support vector Machine algorithm	Graph with random links	The VAR and SVM models achieved their highest AUC values of 84.96% and 86.32% respectively using five lags. Results indicate that the performance of both VAR and SVM are improved with more data from the lags	Furthermore, techniques to handle imbalanced datasets for SVM, which is the case for our co-authorship network, failed to improve link prediction.
2015	Hybrid Swarm Based Method for Link Prediction in Social Networks [4]	IEEE	Particle Swarm optimisation	Facebook, Arxiv, Contact	Similar to human behaviour	Sometimes add unncessary links
2014	A link prediction algorithm based on ant colony optimization [5]	Springer	Ant colony optimisation	Facebook data set	One is that it uses both the pheromone and heuristic information reflecting both local and global structure of the network. Another reason is that ACO_LP considers both attribute and structure	When the datasets have large number of attributes, since our algorithm ACO_LP creates lots of additional nodes in the augmented graph, it consumes more computation time.

2014	Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET [6]	Springer	Particle Swarm optimisation	Graph based list of nodes	Even for a weak node, the performance of a route recovery mechanism is made in such a way that corresponding routes are diverted to the strong nodes. With the aid of the simulated results, the minimization of data loss and communication overhead is using PSO prediction	Large overhead
2017	Link Prediction across Aligned Networks with Sparse and Low Rank Matrix Estimation [7]	IEEE	SLAMPRED Algorithm	Foursquare And Twitter	Works well in accommodating the information distributions between the source and target domains.	Increasing the Weight of the intimacy term about the aligned source network will slightly degrade the performance of SLAMPRED
2019	A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges [8]	Elseiver	Naïve Bayes classifier, Logistic regression, ANN	Facebook Data from stanford site	Small computation cost	Explore the method to optimize the complexity and to make the method scalable

2019	Graph kernel based link prediction for signed social networks [9]	Elseiver	Graph kernel based link prediction method	Epinions	The node's properties, its neighbors' properties, triad information are considered	Does not work well on sparse data
2017	Structural Similarity Based Link Prediction in Social Networks Using Firefly Algorithm [10]	IEEE	Firefly algorithm	USAir,PB, Power	Outperforms the Jaccard and LHN1 similarity indices in terms of precision	Node attribute based link prediction not addressed
2017	Link Prediction Based on Whale Optimization Algorithm [11]	International Conference on New Trends in Computing Sciences	Whale Optimization Algorithm	Power, Political blogs	Competitive results and compatible to solve link prediction problem.	Algorithm efficiency could be improved by implementing a distributed version of the algorithm
2016	Link Prediction in Social Networks: A Similarity score based Neural Network Approach [12]	Research gate	Nueral network	Facebook, Arxiv	Introduced a technique based on similarity score and supervised learning for link prediction in co-authorship network	In future we will use similarity score for all pairs of nodes and evaluate the domain metrics for training data set and implement the supervised learning
2019	Attacking Similarity-Based Link Prediction in Social Networks [13]	ICTS	ATTACK MODEL	Twitter	This paper greatly advance the algorithmic understanding of attacking similarity-based link prediction.	The problem of minimizing Katz and that of maximizing ACT are NP-Hard

2018	3-HBP: A Three-Level Hidden Bayesian Link Prediction Model in Social Networks [14]	SPRINGER	A latent Dirichlet allocation (LDA)	Twitter	LDA is improved by Gaussian weighting which can reduce the negative impact of the interest distribution to the high-frequency users and the expression ability of the interests can be enhanced.	Link prediction can be used to infer users missing attributes, which may be used for targeted advertisements and recommendationand this algorithm is not that much efficient
2017	A systemic analysis of link prediction in social network [15]	Cross mark	Comparative study	Facebook	This model presented an analytical framework for link prediction in social networks and illustrated that there are different challenges and techniques	No limitation as it is a comparative study

IV. Features used for Local similarity index

IV.I Common Neighbors

Assume that the node v E V; that is, is the set of all the neighbors of node . The common neighbors of node and node refer to the jointly owned neighbors by node and node. For the undirected graph, the common neighbors can use the following definition:

$$\boxed{\text{Similarity } (u, v) = \left|\text{ɼ}(u) \cap \text{ɼ}(v)\right|}$$

IV.II Preferential Attachment

Preferential attachment mechanism can be used to generate scale-free network evolution model. The probability of generating a new link of node is directly proportional to the degree of the node. This is the same as the truth “the rich are getting richer” in economics. Therefore, the probability of the link between node and node is directly proportional to. Inspired by this mechanism, the PA similarity index can be defined as follows:

$$\boxed{\text{Similarity}(u, v) = d_u \times d_v}$$

IV.III Adamic-Adar

This similarity index assigns a higher similarity function value to a small degree node. Adamic-Adar algorithm believes that an affair owned by less objects, compared to owned by more objects, has greater effect on link prediction. Its definition is as follows:

$$\boxed{\text{Similarity}(u, v) = \boldsymbol{\Sigma}_{\mathbf{z} \in |\text{ɼ}(u) \cap \text{ɼ}(v)|} 1/ \log d_z}$$

IV.IV Resource Allocation

This similarity index is inspired by the ideas of complex network resources dynamically allocated. In pair of nodes u,v that have no direct link, node u can allocate some resources to the node v through their common neighbor. Their common neighbors assume the role of passers. In the simplest case, we assume that each passer has a unit of resources; it assigns these resources to its neighbors evenly. Therefore, the similarity of node u and node v can be defined as the number of resources that node u get from node v.

$$\text{Similarity}(u, v) = \boldsymbol{\Sigma}_{\mathbf{z} \in |\mathbf{r(u)} \cap \mathbf{r(v)}|} 1/ d_z$$

IV.V Jaccard’ s coefficient

The Jaccard similarity index (sometimes called the Jaccard similarity *coefficient*) compares members for two sets to see which members are shared and which are distinct. It’ s a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two populations.

$$\mathbf{J(X,Y) = |X \cap Y| \,/\, |X \cup Y|}$$

V.I Support Vector Machine Algorithm

Support Vector Machines (SVM) is a Machine Learning Algorithm which can be used for many different tasks.

The main objective in SVM is to find the optimal hyperplane to correctly classify between data points of different classes.The hyperplane dimensionality is equal to the number of input features minus one (eg. when working with three feature the hyperplane will be a two-dimensional plane)

Data points on one side of the hyperplane will be classified to a certain class while data points on the other side of the hyperplane will be classified to a different class (as in Figure below). The distance between the hyperplane and the first point (for all the different classes) on either side of the hyperplane is a measure of sure the algorithm is about its classification decision. The bigger the distance and the more confident we can be SVM is making the right decision.

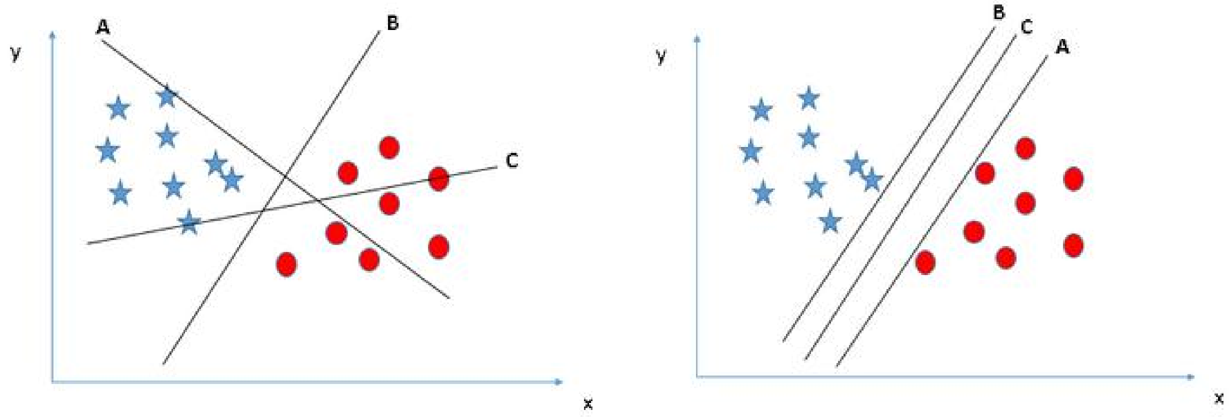


Fig 2. SVM Classifcation process

There are two main types of classification SVM algorithms Hard Margin and Soft Margin:

- Hard Margin: aims to find the best hyperplane without tolerating any form of misclassification.
- Soft Margin: we add a degree of tolerance in SVM.

In this way we allow the model to voluntarily misclassify a few data points if that can lead to identifying a hyperplane able to generalize better to unseen data.

We are implementing a hard margin SVM model in our project.

We are using the sklearn library to implement SVM. We could also use similar classification algorithms such as decision tree, k-NN, Logistic Regression, Naive Bayes Classifier, Random forest, and Artificial Neural Network.

V.II **Logistic Regression**

To solve the problem, we propose here a dynamic logistic regression method for link prediction. It combines various similarity measures under a unified model framework. Specifically, assume we have a network with size n and its network structure is observed at a sequence of time points indexed by $\{t : 1 \leq \quad t \quad \leq T \}$. For any two arbitrary nodes i and j, define $a^{t}_{ij} = 1$ if i follows j and 0 otherwise at time point t. Then the relationships among the nodes over different time points are represented by a sequence of n × n adjacency matrices $A_t = (a^{t}_{ijj})$. We then consider how to make accurate prediction for future link a t ij by carefully studying the historical network structure information $F_{t-1} = \sigma\{As : s < t\}$. This leads to a novel method of dynamic logistic regression. The new method can flexibly take various network structure information into consideration. It is remarkable that the new model allows the network structure to be extremely sparse. Furthermore, by introducing a novel binary random effect, a number of stylized network characteristics (e.g., reciprocation, transitivity) can be well accommodated . However, as a side effect, the resulting likelihood function can be too complicated. This makes the standard maximum likelihood estimation (MLE) computationally forbidden. To alleviate the computational cost, we propose a novel conditional likelihood estimation (CMLE) method, which is computationally feasible for large-scale networks. We demonstrate the efficiency of our model with both simulation studies and a real data example. The proposed new model makes two contributions to the existing literature. First, it is an interesting and novel time series model. Most classical time series literature focus on univariate time series . Recently, there is an increasing interest in multivariate time series, with particular focus on high-dimensional data. Second, in network analysis literature, much efforts have been made to understand the mechanism for network formation.

We consider a network with n nodes, which are denoted as {1, . . . , n}

The relationships among the nodes over different time points are represented by a sequence of n×n adjacency matrices $A_t = (a^{t}_{ij})$ in which $a^{t}_{ij} = 1$ $(i \neq j)$ if there is a link from i to j at time t, and 0 otherwise.

To this end, define $z_{ij} = z_{ji} \in \{0, 1\}$ to be a binary indicator, which could be treated as a binary random effect.

Define Z = (zij) ∈ R (n×n), which is a symmetric random effect matrix. We then assume a^{t} $_{ij}$ = zij a˜ ^{t} $_{ij}$, where a˜ ^{t} $_{ij}$ ∈ {0, 1} is another independent binary indicator such that:

$$P(\tilde{a}_{ij}^{t} = 1 \mid \mathcal{F}_{t-1}) = \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})},$$

Lastly we define A$_{t}$ = (a˜ ^{t} $_{ij}$) ϵ R$^{n×n}$,
As a consequence,

$$P(a_{ij}^{t} = 1 \mid \mathcal{F}_{t-1}) = P(z_{ij} = 1) P(\tilde{a}_{ij}^{t} = 1 \mid \mathcal{F}_{t-1}) = \alpha_{ij} \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}.$$

V.III Artificial Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired the brain. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning largely involves adjustments to the synaptic connections that exist between the neurons.

The training process consists of the following steps:

1. Forward Propagation:

   Take the inputs, multiply by the weights (just use random numbers as weights)

   Let $Y = W_i I_i = W_1 I_1 + W_2 I_2 + W_3 I_3$

   Pass the result through a sigmoid formula to calculate the neuron's output. The Sigmoid function is used to normalise the result between 0 and 1:

   $$1/(1 + e^{-y})$$

2. Back Propagation

   Calculate the error i.e the difference between the actual output and the expected output. Depending on the error, adjust the weights by multiplying the error with the input and again with the gradient of the Sigmoid curve:

   Weight += Error Input × Output × (1-Output)

Here, Output × (1-Output) is derivative of sigmoid curve.

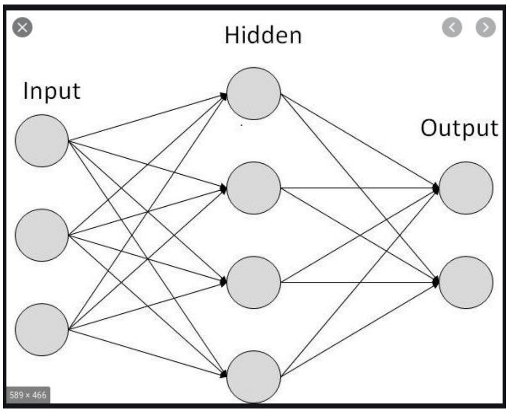


Fig 4. ANN Implementation

V.IV Fuzzy Model Implementation for link prediction

Fuzzy logic makes it possible to connect human thinking to mathematical concepts. In modeling the networks or predicting the links using fuzzy logic, like the crisp modeling of networks, there are common concepts and models with graph theory. In the following section, some basic concepts of graph theory are presented that are used in modeling the system using fuzzy logic.

Fuzzy logic deals with linguistic terms that are known as linguistic variables. The problem related to linguistic terms is that these words have different meanings for different people. As an example, concepts such as close, weak, and strong have different meanings for different people.
A graph is complete if there is a path with a length of 1 between each pair of nodes. In other words, in a complete graph, every pair of vertices is connected by a link. Clique is a complete subgraph of the graph.
Another important concept in undirected networks is the degree of a node. The degree of a node is the number of adjacent nodes with the considered node, that is, the number of nodes that are connected with the path by the length of 1 to that considered node.

Recently, with the advancement of Social Network Analysis (SNA), many studies based on the crisp logic were conducted in the domain of link prediction. According to a structural view toward the networks, efforts were made to find the most similar people based on the homophily theory. By considering the above-mentioned theory, people try to form stable relationships with others using their similar attitudes. In other words, people who have more friends in common have more chances to form a relationship with each other. Kleinberg and Liben–Nowell have studied many unsupervised methods that employed a proximity measure between nodes for link prediction [16]. Some studies are based on the random walks in the network, which try to find ways with the maximum probability and generate a connection based on it.

V.IV.I Clustering Coefficient

An important concept that shows how much the neighbors of a node are related to each other is the clustering coefficient (CC). This measure calculates the number of triangles over the number of possible triangles related to the node. CC is based on the clique concept. Recently, Yager proposed softer definitions for C

If S shows a clique in the graph then the following criteria can define the clique:
C1 : “Most of the elements in S are closely connected.”
C2 : “None of the elements in S are too far from the others.”
C3 : “No element not on the clique is better connected to the members of clique than any element in the clique.”

In the above-mentioned criteria, there are some concepts that should be defined as fuzzy terms. The first one is the concept of close, which means how much two nodes are closely connected. The close concept can be defined as a path with minimum length that connects two nodes to each other. If the length of a path is k, then Q (k) is a function to show the closeness. Yager proposes some prototypes in [17] for the close function, such as ramp function, but according to the small world phenomenon in social networks, every two nodes on average meet each other with the length of 6.Therefore, it means that the closeness in social networks decreases exponentially. Thus, the new close function for undirected and unweighted social networks is proposed as follows:

$$\boxed{\text{close}(xi, xj) = \frac{q_{ij}}{2*10\text{^}(q_{ij}-2)}}$$

q_{ij} is the length of a path that relates x_i to $x_{j.}$

If the considered path is the shortest path that relates these two nodes. Therefore, close (xi, xj) = Q (qij).

Yager’s function for weighted fuzzy social networks can also be generalized. In a weighted fuzzy social network, a new close function is defined as follows:

$$close\,(i, j) = \begin{cases} 1 & q_{ij} < 2 \\ \dfrac{(w\,(i, z) + w(z, j))}{2 * 10^{q_{ij}-2}} & q_{ij} = 2 \\ \dfrac{(w\,(i, z) + w\,(z, e) + w\,(e, j))}{2 * 10^{q_{ij}-2}} & q_{ij} = 3 \\ 0 & q_{ij} > 3 \end{cases}$$

Another important concept that is used in the clustering definition is the Most concept. Most can also be defined as a fuzzy function like M(p), which indicates that the proportion p satisfies the Most. In other words, the considered node is closely connected to how many nodes in the cluster. This function computes how many nodes are closely connected to each other. M(p) should satisfy the following criteria:

$$\begin{cases} M(0) = 0 \\ M(p_2) \geq M(p_1) \quad p_2 \geq p_1 \\ M(1) = 1 \end{cases}$$

C1 criteria can be calculated by using the amount of M(pxi).

In the second criterion of the cluster, there are Far and Not Far concepts. Far is function like the close and Not indicates the negation of this fuzzy number. For any pairs of nodes, Not Far is calculated by :

$$\text{Not Far}(x, y) = \text{Max}_{k=1ton}[R^k(x, y) \wedge F(k)]$$

After defining the Not Far function, the second criterion can be calculated by the following equation, where u is a set that its members are pairs of nodes in the cluster of node x:

$$C_2(x) = \text{Min}_{u \varepsilon Ux}[\text{Not Far}(u)]$$

The third criterion shows that every node out of the cluster of considered nodes should not be close to most of the nodes in the cluster. This criterion shall be defined as follows:

$$M(y/S) = \text{Most}\left(\frac{\sum_{j=1}^{ns} \text{close}(y, xj)}{ns}\right)$$

$$M(x_i/S) = \text{Most}\left(\frac{\Sigma\, j=1tons\ \text{close}(xi, xj)}{ns-1}\right)$$

V.IV.II Fuzzy Link Prediction Based on Local Clustering Coefficient (FCC)

A number of networks tend to form a link between the neighbor nodes. The CC index quantifies how neighbors of nodes tend to be a clique. In social networks, nodes with higher CC tend to generate links with each other. In this paper, fuzzy quasilocal CC model is used for link prediction. The score of the node in the paper is calculated similar to the previous studies. The score of a link is calculated by the sum of CC of its nodes.

For instance, assume that there is an $x_i x_j$ link, as a result, its score is calculated by $C(x_i) + C(x_j)$. The CC for every node is the minimum satisfaction of criteria. It can be interpreted that for every node like x_i the $C(x_i)$ is calculated by the following equation:

$$C\ (x_i) = \mathrm{Min}_j\ [C_j\ (x_i)]$$

As it was mentioned, in social networks the degree of closeness will be decreased exponentially due to small world. To decrease the computational complexity, it can be assumed that the closeness between the nodes that are connected through paths with the length of 4 or above is zero.

V.IV.III Fuzzy Link Prediction Based on Cluster Overlapping (FCO)

In the previous models based on CC or clusters of the node, the clusters' overlapping is not taken into account. It is possible to have two nodes with highdensity clusters around them, but in two different parts of the network. To realize the cluster overlaps in social networks, some models were proposed by different scholars. For instance, Goldberg et al. proposed a model in [18] to find the cluster overlaps based on counting the nodes that are similar in both clusters over all of the nodes.

In social networks, the overlap between clusters of two nodes can be considered by the path, which crosses the nodes that are common in both clusters. To compute the cluster overlapping for two nodes x_i and x_j , the following new index is proposed:

$$S(xi, xj) = \frac{\sum_{z=1}^{n} \mathrm{close}(xi, xj)}{\left|\sum_{u,z \in S_{xi}} W_{uz}\right| + \left|\sum_{u,z \in S_{xj}} W_{uz}\right|}$$

The above Equation calculates the sum of closeness of two nodes (through all the paths that connect the nodes to each other) over the sum of weights of all edges within the clusters. The model is also compared with weighted clustering coefficient (WCC) model.

V.V Particle Swarm Optimization Algorithm In Link Prediction

Particle Swarm Optimization is an algorithm, proposed by Eberhart and Kennedy, to simulate the movement of flocks of birds in getting a food source. In the real world, a flock of birds may begin in random directions searching for food. Then every bird moves along its path, it can discover food in different places. Therefore, this bird memorizes its 'personal best position where food is found in plenty. At the same time, its movement from a position to another takes into consideration the 'global best position found in the whole. Via this process, a flock of birds will head towards the best location that has the largest amount of food.

The PSO approach consists of a set of particles, which are equivalent to the birds. Each particle has a set of attributes: current velocity, current position, the best position discovered by the particle so far and, the best position discovered by its neighbors so far. At each iteration, each particle is moved according to the given equations. Once the move of the particles is affected, the new positions are evaluated. The algorithm iterates until a stopping condition is verified.

$$\boxed{v_{id}^{t+1}= wv_{id}^{t}+c_1r_1(p_{id}-x^{t}_{id})+ c_2r_2(p_{gd}-x^{t}_{id})}$$

And

$$\boxed{X_{id}^{t+1} = v_{id}^{t+1} + X_{id}^{t}}$$

Where d is the dimension, i is the number of particles, w is the inertia weight, c_1 and c_2 are random numbers called learning factor or acceleration factor, r_1 and r_2 are random values in the range of (0,1), P_g is the best position found so far by all particles and p; is the best position discovered so far by the corresponding particle.

The remarkable interactions that occur to an individual are the interactions with his friends, or friends of friends. We can say that, in most social networks, significant interactions between individuals are based on their local network environment. But we cannot ignore the possibility of recognizing stranger people. So for solving the link prediction problem and modeling the dynamic interaction, we tried to apply the principle of the particle swarm movement paradigm which presents a compromise between local and global exploration.

The proposed method models the individuals within a social network as particles in a swarm. So, we have considered each node i ϵ V in the graph G = (V, E) as a particle in the swarm. Then, we have assigned to each node its position X_i which presents all of its connected neighbors N_i. A particle is allowed to interact with its neighbors and with the global best particle in the whole swarm Gbest. The Gbest presents the most influential node in the give social network. In our case, we have eliminated the notion of velocity and subsequently the velocity update equation. During the simulation process, nodes interact and try to improve their status by establishing new relationships. The process of interacting and updating is based on the following equation. In this equation the subtraction operator presents the difference between both sets of neighbors and the addition operator adds new neighbors.

$$\boxed{X_i^{t+1} = X_i^t + \alpha \sum (X_l^t - X_i^t) + \beta(X_{Gbest} - X_i^t)}$$

Where,

X_i^t : The current position;

$\alpha \sum (X_l^t - X_i^t)$: The local interaction step;

$\beta(X_{Gbest} - X_i^t)$: The global interaction step.

V.VI Ant Colony Optimization (ACO)

Ant colony (ACO) is a new population-based stochastic algorithm that has shown good search abilities on many optimization problems. However, the original ACO shows slow convergence speed during the search process. In order to enhance the performance of ACO, this paper proposes a new artificial bee colony algorithm, which modifies the search pattern of both employed and onlooker bees. A solution pool is constructed by storing some best solutions of the current swarm. New candidate solutions are generated by searching the neighborhood of solutions randomly chosen from the solution pool. ACO is a new swarm intelligence algorithm that is inspired by the behavior of honey bees. Since the development of ACO, it has been applied to solve different kinds of problems. Similar to other stochastic algorithms, ACO also faces up some challenging problems. For example, ACO shows slow convergence speed during the search process. Due to the special search pattern of bees, a new candidate solution is generated by updating a random dimension vector of its parent solution. Therefore, the offspring (new candidate solution) is similar to its parent, and the convergence speed becomes slow. Moreover, ACO easily falls into local minima when handling complex multi nodal problems. The search pattern of bees is good at exploration but poor at exploitation. However, a good optimization algorithm should balance exploration and exploitation during the search process.
In the algorithm, artificial ants are employed to travel on a logical graph. Each ant chooses its path according to the value of the pheromone and heuristic information on the edges. The paths the ants passing through are evaluated, and the pheromone information on each edge is updated according to the quality of the path it located. Finally, the pheromone on each edge is used as the score of the similarity between the nodes.

We consider a network represented by an undirected simple network *G*(*V,E*), where *V* is the set of nodes and *E* is the set of links. Multiple links and self-connections are not allowed in *G*. Let *N* = |*V* | be the number of nodes in *G*. We use *U* to denote the universal set containing all possible links. The task of link prediction is to find out missing links (or the links that will appear in the future) in the set of non-existing links *U-E*.
The purpose of our method is to assign a score, *Score*(*x,y*), to each pair of nodes (*x,y*) ∈*U*. This score reflects the similarity between the two nodes. For a nodes pair (*x,y*) in *U E* , the larger *Score*(*x,y*) is, the higher probability there will exist a link between nodes *x* and *y*.
We also extend the method to solve the link prediction problem in the networks with node attributes. The pheromones on the edges are used to predict links as well as infer node attributes.

The edges with pheromones are assigned scores relatively.The ones with the highest scores are selected.The pheromones evaporates with time on the edges. The intensity of pheromones is increased by the ant's iteration on the respective path. This pheromone information in turn influences other ants to pick up the next path.

We set the initial value of pheromone on the edge between the nodes

(v_i, v_j) as

$$\boxed{\tau_{ij}=\lambda\times(a_{ij}+\varepsilon)}$$

Here, λ and ε are positive constants, namely if $(v_i, v_j) \in \varepsilon$ the initial value of pheromone τ_{ij} is set as $\lambda\times(1+\varepsilon)$,otherwise it is set as $\lambda\times\varepsilon$.

It is obvious that the edges which have link connection will have higher initial pheromone value. Such pheromone information will guide the ants to walk through the existing links and their neighbors with higher probability. The ants are likely to choose the path with the highest pheromone and update it as the global path.The evaporation on the path takes place according to the variable Ʈ.

The pheromone updation for each path take place as follows:

$$\boxed{Ʈ(t+1) = ß \,.\, Ʈ(t) + \Delta Ʈ(t)}$$

$$\boxed{\Delta Ʈ(t) = \sum_{k=1}^{m} \Delta Ʈ(t)}$$

$$\boxed{\Delta Ʈ(t) = Q(S)}$$

The algorithm ceases the iterations according to a certain termination condition. We stop the iterations when the pheromone values on each edge obtained in adjacent iterations tend to stabilize. In addition, we also set up a threshold Nc, which is the maximum number of iterations. The iterations should be ended as well when the number of iterations goes beyond Nc.
Finally, the algorithm outputs the pheromone matrix as the score matrix, namely, the final score of nodes pair v_i and v_j is $\text{Score}(i,j) = \tau_{ij}$.
Finally, the pheromone on each edge is used as the final similarity score of the node pair. Empirical results show that our algorithm can achieve higher quality results of link prediction using less computation time than other algorithms.There are two reasons for ACO achieving high quality results. One is that it uses both the pheromone and heuristic information reflecting both local and global structure of the network. Another reason is that ACO considers both attribute and structure information.

V.VII Hybrid ACO and PSO algorithm (PACO)

The PACO Algorithm is the name given to a hybridized algorithm using Particle Swarm Optimization algorithm and the Ant Colony algorithm.In the proposed algorithm, each artificial ant, like a particle in PSO, is allowed to memorize the best solution ever found. After solution construction, only elite ants can update pheromone according to their own best-so-far solutions. Moreover, a pheromone disturbance method is embedded into the ACO framework to overcome the problem of pheromone stagnation. Two sets of benchmark problems were selected to test the performance of the proposed algorithm. The computational results show that the proposed algorithm performs well in comparison with existing swarm intelligence approaches. The PACO algorithm incorporates the merits of PSO into the ACO algorithm. One of the advantages of applying ACO to the Link prediction is that ACO can cluster all nodes. However, laying pheromone (long-term memory) on trails as ant communication medium is time consuming. The merit of PSO is that it can speed convergence through memorizing personal and global best solutions to guide the search direction. Inspired by the merit of PSO, the PACO algorithm allows artificial ants to memorize their own best solution so far and to share the information of swarm best solution.

V.VII.I Reasons why we have hybridized PSO and ACO Algorithm-

1. Many other algorithms (other than the used ones)perform the so-called single-starting-point search and thus their performance relies highly on a good initial solution.

However, PSO, and ACO are all population-based algorithms and can start the search from multiple points. Their initial solutions have little influence on their performance. Thus, we consider adopting population-based algorithms to solve Link prediction.

2. PSO and ACO have a memory that enables the algorithms to retain the knowledge of good solutions, In view of these two considerations, we select PSO and ACO as the solution for CVRP.

3. The PSO algorithm uses the least computational time and it could easily be combined with ACO algorithm in comparison to neural networks algorithm which uses a high computational power.

4. Both algorithms hybridized are already providing us with good results.

V.VII.II Steps involved in PACO algorithm

During the searching process, artificial ants construct solution routes, memorize the best solution ever found, and lay pheromone on the routes of swarm and personal best solutions. To prevent being trapped in local optima and to increase the probability of obtaining better solutions, PACO performs pheromone(long-term memory) disturbance and short-term memory resetting operations to adjust stagnated pheromone trails. Disturbed pheromone trails guide ants to find new Pbest and Gbest solutions. The merits of PSO adopted in PACO can speed convergence during a run, even after pheromone (long-term memory) disturbance operations. Computational results show that the performance of PACO is competitive in terms of solution quality when compared with existing ACO- and PSO-based approaches. We have also modified the existing PACO Algorithm with time windows and having multiple depots. The results that we have achieved will be of great use in future. These results will bring a change in Link prediction in the future.

STEPS :

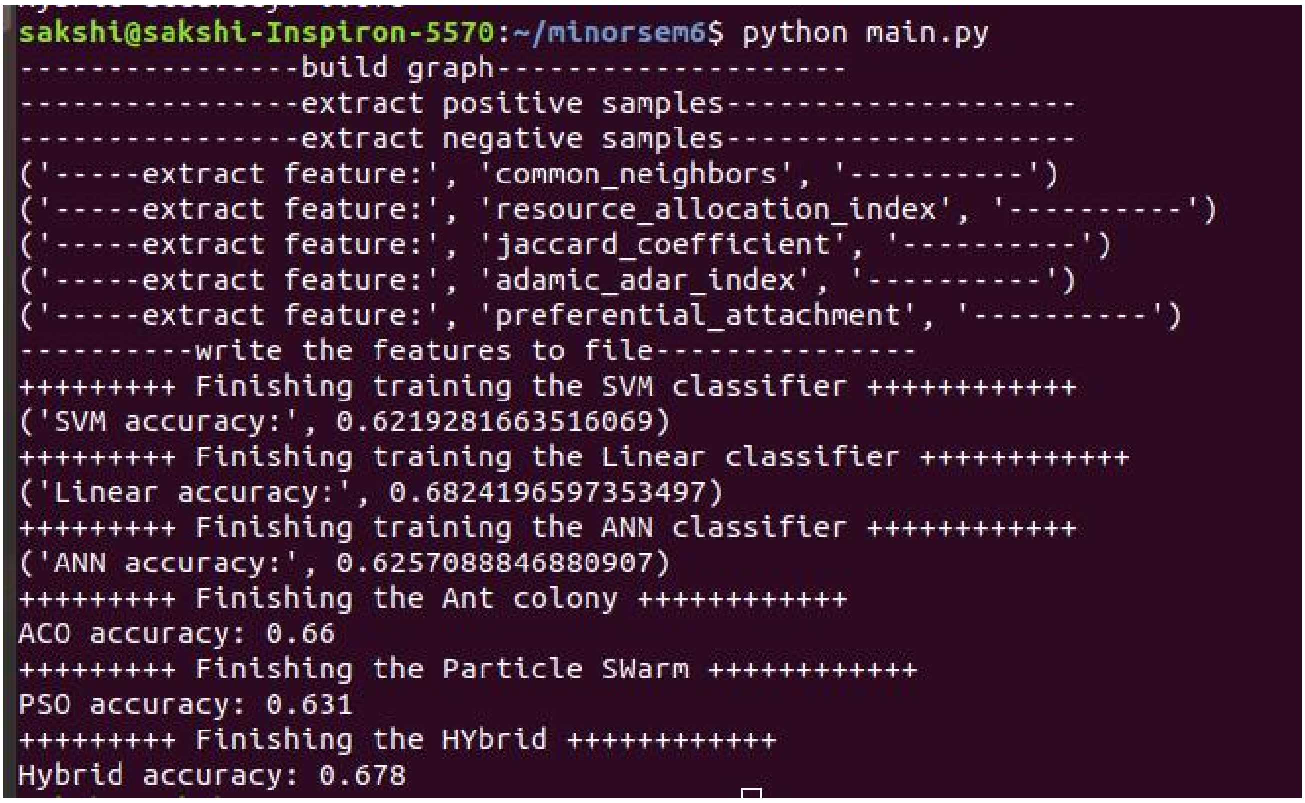
1) Initialization (Initialize all parameters).
2) Let m ants construct solution routes.
3) The top r best ants perform local search.
4) Update the Gbest and Pbestsolutions of ants and select the r elite ants.
5) If Gbest is not improved within successive iterations, go to Step 6; otherwise, go to Step 7.
6) Randomly disturb the pheromone matrix, reset the Pbestsolutions for some ants, and go to Step 8.
7) Update the pheromone matrix based on the Pbestsolutions of elite ants.
8) If iteration number reaches the maximum number of iterations (MaxIte), go to Step 9; otherwise, go to Step 2 for the next iteration.
9) Output Gbest, the best solution ever found. We have combined ant bee colony elitist mode and particle swarm to obtain good results. For comparison of results, refer to section 6.

VI. Results

The AUC value from the provided rank of all the non-observed links is the probability of a randomly chosen missing edge allotted with higher score than a randomly chosen nonexistent link. The implementation of calculation of the score of non-observed edge is easy compared to obtaining the ordered list of nonexistent edge as it would be more difficult. mathematically every time randomly a missing edge and nonexistent edge are picked and their scores are compared, if out of n independent comparisons there are n’ times when the missing edge have higher score and n” times they have the same score then the AUC value is given by

$$\boxed{AUC = \frac{n'+0.5n''}{n}}$$

If the generated scores are from independent identical distribution, the AUC would be around 0.5, therefore the degree to which the value exceeds would be 0.5 that indicates how better.



```
sakshi@sakshi-Inspiron-5570:~/minorsem6$ python main.py
----------------build graph--------------------
----------------extract positive samples--------------------
----------------extract negative samples--------------------
('-----extract feature:', 'common_neighbors', '----------')
('-----extract feature:', 'resource_allocation_index', '----------')
('-----extract feature:', 'jaccard_coefficient', '----------')
('-----extract feature:', 'adamic_adar_index', '----------')
('-----extract feature:', 'preferential_attachment', '----------')
----------write the features to file---------------
+++++++++ Finishing training the SVM classifier ++++++++++++
('SVM accuracy:', 0.6219281663516069)
+++++++++ Finishing training the Linear classifier ++++++++++++
('Linear accuracy:', 0.6824196597353497)
+++++++++ Finishing training the ANN classifier ++++++++++++
('ANN accuracy:', 0.6257088846880907)
+++++++++ Finishing the Ant colony ++++++++++++
ACO accuracy: 0.66
+++++++++ Finishing the Particle SWarm ++++++++++++
PSO accuracy: 0.631
+++++++++ Finishing the HYbrid ++++++++++++
Hybrid accuracy: 0.678
```

Fig 5. The accuracy of all the implemented algorithms for link prediciton

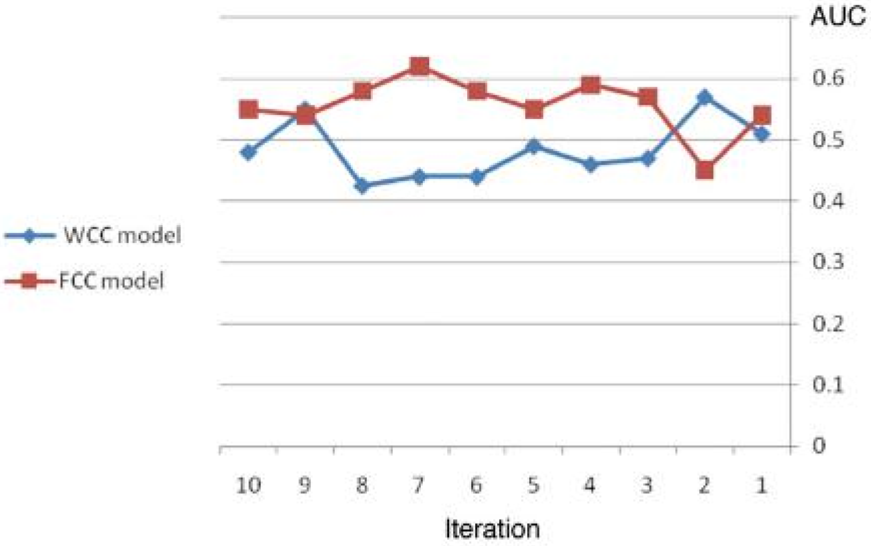


Fig 6. The comparison of the accuracy of the FCC model and WCC model

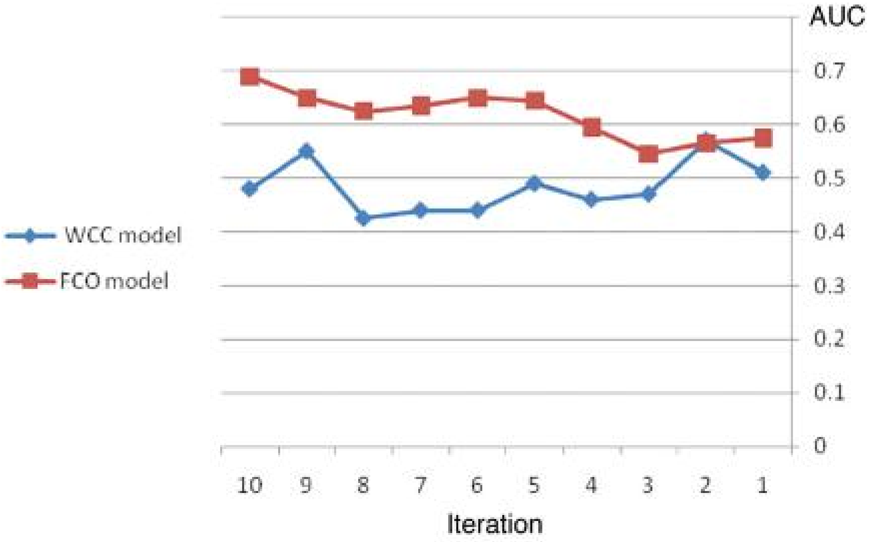
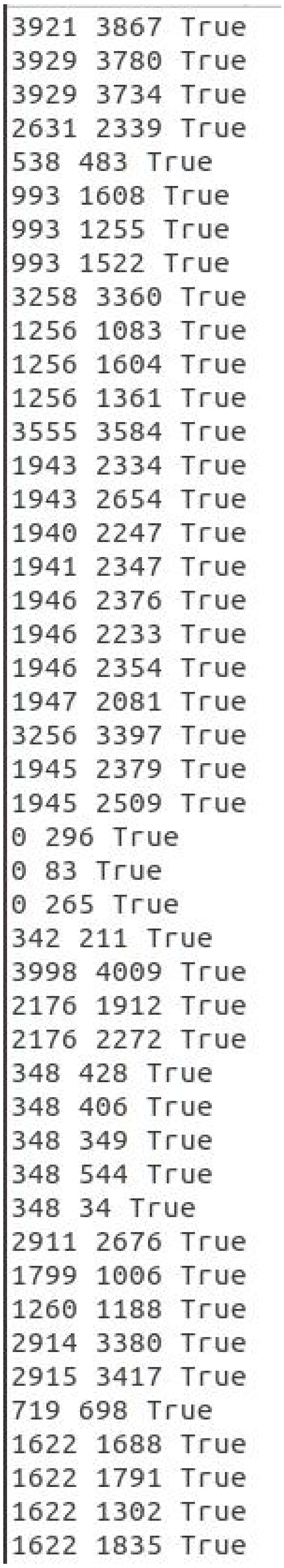
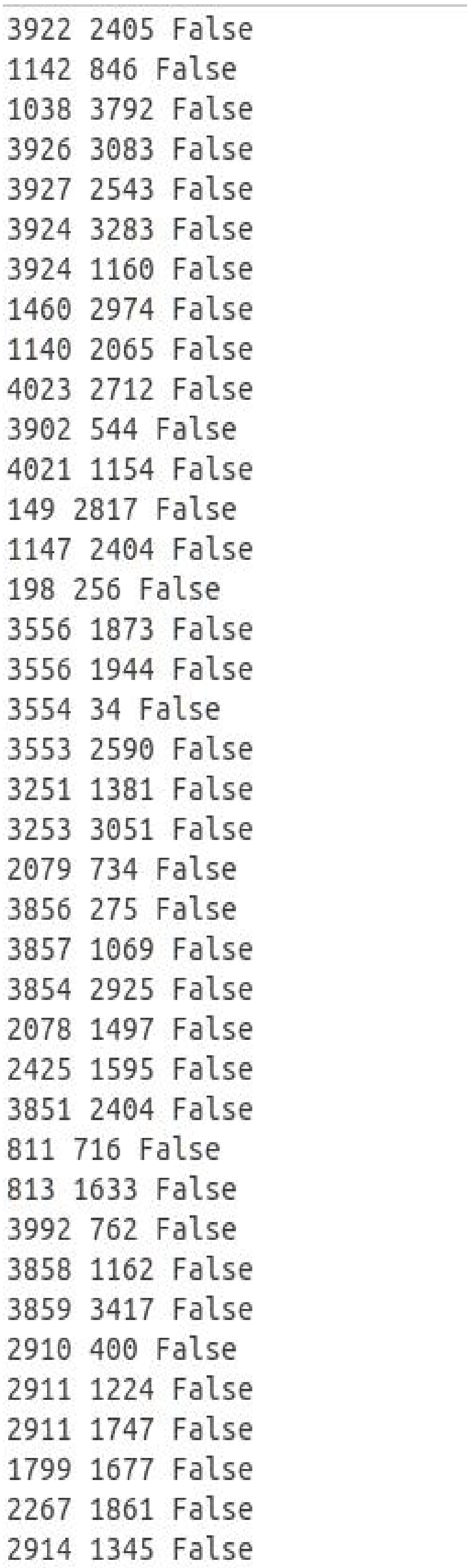


Fig 7. The comparison of the accuracy of the FCO model and WCC model



```
3921 3867 True
3929 3780 True
3929 3734 True
2631 2339 True
538 483 True
993 1608 True
993 1255 True
993 1522 True
3258 3360 True
1256 1083 True
1256 1604 True
1256 1361 True
3555 3584 True
1943 2334 True
1943 2654 True
1940 2247 True
1941 2347 True
1946 2376 True
1946 2233 True
1946 2354 True
1947 2081 True
3256 3397 True
1945 2379 True
1945 2509 True
0 296 True
0 83 True
0 265 True
342 211 True
3998 4009 True
2176 1912 True
2176 2272 True
348 428 True
348 406 True
348 349 True
348 544 True
348 34 True
2911 2676 True
1799 1006 True
1260 1188 True
2914 3380 True
2915 3417 True
719 698 True
1622 1688 True
1622 1791 True
1622 1302 True
1622 1835 True
```

Fig 8. Existing Links



```
3922 2405 False
1142 846 False
1038 3792 False
3926 3083 False
3927 2543 False
3924 3283 False
3924 1160 False
1460 2974 False
1140 2065 False
4023 2712 False
3902 544 False
4021 1154 False
149 2817 False
1147 2404 False
198 256 False
3556 1873 False
3556 1944 False
3554 34 False
3553 2590 False
3251 1381 False
3253 3051 False
2079 734 False
3856 275 False
3857 1069 False
3854 2925 False
2078 1497 False
2425 1595 False
3851 2404 False
811 716 False
813 1633 False
3992 762 False
3858 1162 False
3859 3417 False
2910 400 False
2911 1224 False
2911 1747 False
1799 1677 False
2267 1861 False
2914 1345 False
```

Fig 9. Non-Existing Links

VII. Dataset and libraries used

Dataset is taken from-

http://snap.stanford.edu/data/

Libraries Used-

1. **SKLEARN**- It is used for implementing SVM, Logistic Regression, and Artificial Neural Network (ANN) algorithm.
2. **Networkx**- It is used for implementing the undirected network graph.
3. **Numpy**- It is used for performing matrix calculations.
4. **Matplotlib**- It is used to plot the AUC curves.

The model was implemented and made to run on Ubuntu 18.04

VIII. Conclusions

We discussed that social networks are made of social actors and connections or relations between them and these networks are best represented by Graphs where nodes are social actors and edges are the connections between nodes. Social networks are often very large and grow at a fast rate.

These social network graphs consist of so much data that it is a challenging task to process and do the analysis. In the link prediction methods, we take a small snapshot of these large networks to analyze and predict future edges between the nodes.

As discussed, there are many potential areas of our social life where link prediction plays many important roles and still, there are a lot of fields where it has open opportunities.

We have also seen various methods to solve the link prediction problems which have their accuracies listed as below:

Algorithm Used	Accuracy Evaluated
Support Vector Machines (SVM)	0.6219
Logistic Regression	0.6824
Artificial Neural Networks (ANN)	0.6257
Fuzzy Link Prediction Based on Local Clustering Coefficient (FCC)	0.6200
Fuzzy Link Prediction Based on Cluster Overlapping (FCO)	**0.7000**
Particle Swarm Optimization (PSO)	0.6310
Ant colony Optimization (ACO)	0.6600
Hybrid PSO & ACO (PACO)	0.6780

IX. References

[1] Bastani, Susan, Ahmad Khalili Jafarabad, and Mohammad Hossein Fazel Zarandi. "Fuzzy models for link prediction in social networks." *International journal of intelligent systems* 28.8 (2013): 768-786.

[2] Zhou, Jing, DanYang Huang, and HanSheng Wang. "A dynamic logistic regression for network link prediction." *Science China Mathematics* 60.1 (2017): 165-176.

[3] Co, Jan Miles & Fernandez, Proceso. (2017). Time-Series Link Prediction Using Support Vector Machines. Philippine Journal of Science. 146. 105-116.

[4] Aouay, Saoussen, Salma Jamoussi, and Faiez Gargouri. "Hybrid Swarm Based Method for Link Prediction in Social Networks." *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2015.

[5] Chen, Bolun, and Ling Chen. "A link prediction algorithm based on ant colony optimization." *Applied Intelligence* 41.3 (2014): 694-708.

[6] Manickavelu, Devi, and Rhymend Uthariaraj Vaidyanathan. "Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET." *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014): 107.

[7] Zhang, Jiawei, et al. "Link prediction across aligned networks with sparse and low rank matrix estimation." *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017.

[8] Pandey, Babita, et al. "A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges." *Expert Systems with Applications* (2019).

[9] Yuan, Weiwei, et al. "Graph kernel based link prediction for signed social networks." *Information Fusion* 46 (2019): 1-10.

[10] Srilatha, P., and R. Manjula. "Structural similarity based link prediction in social networks using firefly algorithm." *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. IEEE, 2017.

[11] Barham, Reham, and Ibrahim Aljarah. "Link prediction based on whale optimization algorithm." *2017 International Conference on New Trends in Computing Sciences (ICTCS)*. IEEE, 2017.

[12] Sharma, Upasana, and Bhawna Minocha. "Link prediction in social networks: a similarity score based neural network approach." *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. 2016.

[13] Zhou, Kai, et al. "Attacking similarity-based link prediction in social networks." *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[14] Xiao, Yunpeng, et al. "3-HBP: A three-level hidden Bayesian link prediction model in social networks." *IEEE Transactions on Computational Social Systems* 5.2 (2018): 430-443.

[15] Haghani, Sogol, and Mohammad Reza Keyvanpour. "A systemic analysis of link prediction in social network." *Artificial Intelligence Review* 52.3 (2019): 1961-1995.

[16] Liben‐Nowell, David, and Jon Kleinberg. "The link‐prediction problem for social networks." *Journal of the American society for information science and technology* 58.7 (2007): 1019-1031

[17] Yager, Ronald R. "Intelligent social network analysis using granular computing." *International Journal of Intelligent Systems* 23.11 (2008): 1197-1219.

[18] Goldberg, Mark K., Mykola Hayvanovych, and Malik Magdon-Ismail. "Measuring similarity between sets of overlapping clusters." *2010 IEEE Second International Conference on Social Computing*. IEEE, 2010.