

Chapter 5

A New Cost Based Feature for Link Prediction in Dynamic Networks (CFLP)

This chapter studies a feature-based solution that considers both individual snapshots and the overall network throughout the full-time span to solve the link prediction problem ¹. It is a novel feature called Cost-based Feature for Link Prediction (CFLP) for estimating edge behavior throughout the entire network, which uses a reward and penalty structure to summarize node activity across the entire network. It uses similarity indices, classified into four major categories: local similarity, global similarity, quasi-local similarity, and clustering coefficient-based similarity, to measure edge activity in individual snapshots. We have also selected fourteen different snapshot-based features to find the most excellent combination of minimum features for link prediction. We used regression and mutual information-based scoring for feature selection to correctly quantify the relative effect of features among themselves and the overall link prediction problem.

¹Published in Journal of Computational Science, CFLP : A New Cost Based Feature for Link Prediction in Dynamic Networks

5.1 Introduction

Local, Quasi-local and Global similarity-based indices are the three popular types of similarity-based indices. The local similarity indexes are constructed utilizing data from the vertices' immediate surroundings. To build global similarity-based indices, the complete topological knowledge of a network is employed. These approaches are more computationally intensive than local similarity-based approaches, but these approaches give a more comprehensive view of the network structure. To balance the qualities of local and global similarity-based measures, quasi-local similarity-based measures have been developed. We have also used clustering-based approaches to make a rich feature set. Clustering-based methods used here are Clustering Coefficient based Link Prediction (CCLP) and Node and Link clustering coefficient (NLC). These clustering methods provide more information about the nodes and edges, which helps for better accuracy.

The characteristic of vertices and connections emerging and disappearing with time characterizes dynamic networks [129, 130]. Link prediction challenge can be resolved by supervised machine learning (ML) methods for classification [1, 35, 50]. However, while several studies have shown that this strategy produces promising results, selecting the set of features (variables) to train the classifiers remains a significant challenge. According to literature [1, 129, 130], more local, global, quasi-local, and clustering information of the topology increase link prediction accuracy. We extracted local, global, quasi-local, and clustering information. Also, we have used feature selection methods from the Scikit-Learn library named *f_regression* and *mutual_info_regression* to optimize the number of features.

Integrating different types of topological information from different snapshots is the driving force behind our proposed method. All snapshots' various types of topological information (Local, Global, Quasi-local, Clustering-based, and CFLP based) are considered. For feature creation, we employed six local similarity indices: Adamic/Adar Index (AA), Common Neighbors (CN), Preferential Attachment (PA), Jaccard

Coefficient (JC), Salton Cosine Similarity Index (SI), and CAR-based common neighbor index (CARCN). The shortest path (SP), Cosine based (COSP), Average Commute Time (ACT), and Matrix Forest Index (MFI) were also used as global similarity indicators. We have used two quasi-local similarity indices: the Path of Length3 (L3) and the local path Index (LP). Two clustering-based similarity indices were also used: clustering coefficient-based link prediction (CCLP) and node and link clustering coefficient (NLC). We have also designed a new Cost-Based Feature for the Link Prediction (CFLP) method. We have used feature selection method named *f_regression* and *mutual_info_regression*, the feature score obtained from different similarity indices are shown in Table 5.1. We have selected the highest scoring similarity indices and used them for better accuracy in link prediction. More details regarding feature selection and optimized features are explained in Section 5.2.3. Seven different dynamic network datasets are employed to train and evaluate classification models based on these attributes. We have used five machine learning models named Neural Network (NN), Logistic Regression (LR), XGBoost (XGB), Random Forest Classifier (RFC), and Linear Discriminant Analysis (LDA). The experimental results show that our proposed method is better than state-of-the-art dynamic network link prediction methods.

The main contribution of this paper are as follows.

- In this paper, we introduced a novel Cost-based Feature for Link Prediction (CFLP) to solve the link prediction challenge in dynamic networks, which uses all the information from previous snapshots instead of the individual snapshot.
- In this study, we provide a link prediction framework that uses diverse topological and clustering data to improve link prediction. These include local similarity indices (CN, JC, AA, PA, Salton), global similarity indices (SP, Cosp, ACT, MFI), quasi-local similarity (L3, LocalPath), and clustering based similarity indices (CCLP, NLC, CARCN).
- We used a feature selection strategy to optimize the features by picking high-scoring similarity indices. The optimized features outperform all other features combined

and state-of-the-art link prediction algorithms. To correctly quantify the relative effect of features among themselves and the overall link prediction problem, we used regression and mutual information-based scoring for feature selection.

- The training data is processed using machine learning models like Neural Network, Logistic Regression, XGBoost, Random Forest Classifier, and Linear Discriminant Analysis. These models are then used to predict the probability of existing edges on test data.

5.2 Proposed Method

Most current research extracts feature sets from network topology (i.e., topological network information). These characteristics are generic and domain-independent, and they can be used in any network [131, 150, 151]. Other research focuses on identifying node and edge information critical for improving link prediction performance. Typical, neighborhood, and path-based features are examples of such features [152, 153]. Some related literature also shows that the clustering coefficient is related to the link prediction problem [1, 35, 154]. We perceive the link prediction challenge as a binary categorization challenge. The class label indicates the existence or lack of connections. When the connection is present among two nodes, the label is indicated by 1; otherwise, the label is taken as 0. Figure 5.1 shows the bird-view of our model.

5.2.1 A New Cost Based Feature for Link Prediction(CFLP) algorithm

CFLP method uses the concept of reward and penalty. While traversing through snapshots, a reward is added when the edge is present, and a penalty is subtracted if the edge is absent. The multiplication of i in addition/subtraction takes the closeness or

recentness of the snapshot into consideration, i.e., i will be higher for the recent snapshot and lower for the earlier snapshot. The initial value of v_o is taken such that addition/subtraction does not result in a negative value. The detailed algorithm is shown in the Algorithm 3.

$$\sum_{i=1}^5 v_o = v_o + (F_i(G_i(u, v)) \times i \times r) - (1 - (F_i(G_i(u, v)) \times i \times p)) \quad (5.1)$$

where $F_i(G_i(u, v))$ represents function of graph at i^{th} iteration and u & v are nodes between which edge is considered.

$$F_i(G_i(u, v)) = \begin{cases} 1 & \text{if edge is present in } i^{th} \text{ snapshot,} \\ 0 & \text{Otherwise.} \end{cases} \quad (5.2)$$

Algorithm 3: New Cost Based Feature for Link Prediction (CFLP)

Input: $G(V, E)$: Dynamic social graph, v : no. of nodes, e : edge for which the value is to be calculated,

Output: Return feature value

```

1  $v_o \leftarrow 30$  ▷ Initialization Phase
2  $p \leftarrow 2$ 
3  $r \leftarrow 5$ 
4 for  $i \leftarrow 1$  to 5 do
5   if edge present in  $i$  th snapshot then
6      $v_o = v_o + i * r$  ▷ Reward calculation
7   if edge is not present in  $i$  th snapshot then
8      $v_o = v_o - i * r$  ▷ Penalty calculation
9 return  $v_o$ ;

```

5.2.2 Algorithm for training and testing of CFLP model

Algorithm 4 shows the training and testing process. In line 1, we initialize an empty list l . In lines 2-4, we append sorted edge lists from each snapshot to l . After that, we generate

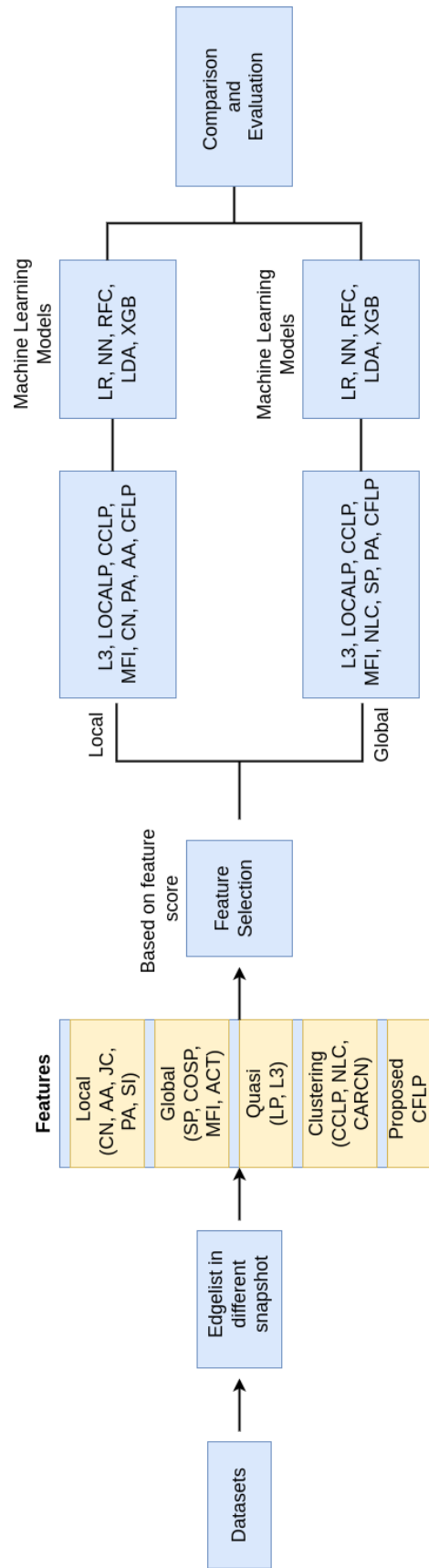
Algorithm 4: Algorithm for producing training and test set**Input:** dynamic network D , number of snapshots m **Output:** Training set and testing set

```

1  $l \leftarrow \{\}$ 
2 for each snapshot  $\in D$  do
3    $edgelist \leftarrow sorted(edgesin snapshot)$ 
4    $l \leftarrow append\ edgelist$ 
5  $all\_graphs \leftarrow generate\_graph(l)$   $\triangleright$  Generate graph for each snapshot
6  $TrueEdges \leftarrow l[m-1]$   $\triangleright$  True edges of last snapshot
7  $RandomNonEdges \leftarrow$  Random edgelist of non-edges of size equal to TrueEdges from graph[m-1]
8  $AllEdges \leftarrow TrueEdges + RandomNonEdges$ 
9  $FeatureSet \leftarrow \{\}$ 
10 for each edge  $\in AllEdges$  do
11    $SingleEdgeFeature[14 * m + 3] \leftarrow 0$ 
12   for  $t \leftarrow 0$  to  $m - 2$  do
13      $G \leftarrow all\_graphs[t]$ 
14      $i \leftarrow edge[0], j \leftarrow edge[1]$ 
15      $SingleEdgeFeature[14 * t] \leftarrow CommonNeighbour(G, i, j)$ 
16      $SingleEdgeFeature[14 * t + 1] \leftarrow JaccardCoefficient(G, i, j)$ 
17      $SingleEdgeFeature[14 * t + 2] \leftarrow AdamicAdar(G, i, j)$ 
18      $SingleEdgeFeature[14 * t + 3] \leftarrow PreferentialAttachment(G, i, j)$ 
19      $SingleEdgeFeature[14 * t + 4] \leftarrow ShortestPathP(G, i, j)$ 
20      $SingleEdgeFeature[14 * t + 5] \leftarrow CosineSimilarity(G, i, j)$ 
21      $SingleEdgeFeature[14 * t + 6] \leftarrow AverageCommuteTime(G, i, j)$ 
22      $SingleEdgeFeature[14 * t + 7] \leftarrow MatrixForestIndex(G, i, j)$ 
23      $SingleEdgeFeature[14 * t + 8] \leftarrow ClusteringCoefficientBasedLinkPrediction(G, i, j)$ 
24      $SingleEdgeFeature[14 * t + 9] \leftarrow SALTON(G, i, j)$ 
25      $SingleEdgeFeature[14 * t + 10] \leftarrow PathofLength3(G, i, j)$ 
26      $SingleEdgeFeature[14 * t + 11] \leftarrow NodeAndClusteringCoefficient(G, i, j)$ 
27      $SingleEdgeFeature[14 * t + 12] \leftarrow LocalPath(G, i, j)$ 
28      $SingleEdgeFeature[14 * t + 13] \leftarrow CAR - basedCommonNeighbourIndex(G, i, j)$ 
29    $SingleEdgeFeature[14 * m] \leftarrow CFLP(G, i, j)$ 
30    $SingleEdgeFeature[14 * m + 1] \leftarrow edge[0]$   $\triangleright$  Node1 of edge
31    $SingleEdgeFeature[14 * m + 2] \leftarrow edge[1]$   $\triangleright$  Node2 of edge
32    $FeatureSet \leftarrow append(SingleEdgeFeature)$ 
33  $FeatureSet \leftarrow shuffle(FeatureSet)$ 
34  $FeaturesTrainSet \leftarrow \{\}, LabelsTrainSet \leftarrow \{\}$ 
35  $FeaturesTestSet \leftarrow \{\}, LabelsTestSet \leftarrow \{\}$ 
36  $FeaturesTrainSet \leftarrow 70\%$  of  $FeatureSet$ 
37  $FeaturesTestSet \leftarrow$  remaining 30% of  $FeatureSet$ 
38 for each tuple  $\in FeaturesTrainSet$  do
39   if (tuple[14 * m + 1], tuple[14 * m + 2])  $\in all\_graphs[m - 1]$  then
40      $LabelsTrainSet \leftarrow append(1)$ 
41   else
42      $LabelsTrainSet \leftarrow append(0)$ 
43  $FeaturesTrainSet \leftarrow FeaturesTrainSet[0 : 14 * m]$   $\triangleright$  eliminate columns from feature set representing edge identifiers
44 Same process for  $FeaturesTestSet, LabelsTestSet$   $\triangleright$  lines 38-43
45 return  $FeaturesTrainSet, LabelsTrainSet, FeaturesTestSet, LabelsTestSet$ 

```

FIGURE 5.1: Proposed *CFLP* Model Overview



all the graphs from l . In lines 6-7, we store true edges and random non-edges. Then we combine them to get *AllEdges*. After that, we initialize another empty list for the feature set in line 9. In lines 10-32, we iterate through *AllEdges* for each snapshot, and we keep appending similarity scores generated from different similarity methods. The local similarity scores are calculated between the lines 15-18, global similarity scores are computed between the lines 19-22, and quasi-local similarity scores are computed between the lines 23-27. We calculate the similarity score from our method *CFLP* in line 28. In line 33, we shuffle the feature set. We initialize four lists for storing the training and testing feature set and the corresponding labels in lines 34-35. In lines 36-37, we transfer 70% of features to the training set and the remaining 30% to the test set. After that, in lines 38-42, we iterate through features in the training set and create labels for the training set. If the connection exists in the previous snapshot, we label it as 1; otherwise, it is 0. In line 44, we have described that we use the same process for the testing set. Finally, in line 45, we return the training and testing sets and corresponding labels.

Computational Complexity

The feature set generation algorithm which consists of not only the proposed feature *CFLP* but also fourteen other base-lined network metrics, we devise the complexity as follows: Let N : size of the dataset and V : the number of nodes in a snapshot and V' : the number of nodes in the overall graph, lines 1-9 are initialization steps with constant or linear complexity, next from lines 10-32 the number of classification edges i.e. *AllEdges* is of size $2 * (N/m)$ (having true edges of last snapshot and equal size randomized false edges), then we iterate over these edges for m snapshots with $O(m * 2(N/m)) \approx O(N)$ complexity and in each iteration, for each classification edge we extract network fourteen features, these features have a maximum complexity of $O(V^2)$ for a particular edge. After this, from lines 29-32, we extract our proposed *CLFP* feature where we iterate over m snapshots each of size N/m . Hence it has linear complexity $O(N)$ for checking m lists of cumulative size N . The subsequent calculations are trivial in terms of complexity. This results in overall complexity is $O(N * V^2)$. Note that $V \ll V'$.

TABLE 5.1: Feature Score of all similarity measures using Mutual Information Classifier (MIC) and F-Regression method (FREG). In this table R-EMAIL, COLLEGE, LKML and MATH are shortforms for RADOSLAW-EMAIL, COLLEGEMSG, LKML-REPLY and MATHOVERFLOW respectively.)

Method	DATASET	MIT	R-EMAIL	EU-CORE	FB-FORUM	COLLEGE	LKML	MATH
MIC	AA	0.8463	0.692	0.7612	0.0546	0.209	0.97	0.8906
	JC	0.7392	0.6656	0.7241	0.0669	0.2488	0.931	0.8629
	PA	1	0.5711	0.2712	0.2812	0.5128	0.9902	0.9751
	CN	0.791	0.6633	0.7031	0.0403	0.1895	0.9452	0.879
	SI	0.7317	0.6594	0.7687	0.0758	0.2146	0.9332	0.8642
	SP	0.7502	0.8449	1	1	0.9843	0.9283	0.8809
	COSP	0.751	0.9822	0.9907	0.9205	0.8601	0.9714	0.8988
	ACT	0.9942	0.6303	0.2892	0.4055	0.5844	0.9385	0.8835
	MFI	0.9004	1	0.9092	0.8372	0.7989	0.9017	0.8349
	L3	0.9799	0.6714	0.8174	0.8911	1	1	1
	LOCALP	0.8773	0.641	0.7049	0.5624	0.6663	0.981	0.9634
	CCLP	0.8278	0.6875	0.7799	0.0562	0.2105	0.963	0.8634
	NLC	0.8903	0.6761	0.7888	0.0804	0.2785	0.9591	0.8671
	CARCN	0.8363	0.6553	0.6876	0.0409	0.1845	0.8768	0.7112
FREG	AA	0.6463	0.8567	0.8525	0.0136	0.0317	0.3651	0.5129
	JC	0.2557	0.5324	0.7072	0.0015	0.0029	0.6773	0.671
	PA	0.8141	0.6147	0.212	0.1704	0.198	0.2116	0.3643
	CN	0.6803	0.9117	0.8018	0.0172	0.0343	0.427	0.5908
	SI	0.2992	0.6826	1	0.0018	0.0024	0.9794	1
	SP	0.4859	0.5588	0.6059	0.1418	0.0401	0.2413	0.2759
	COSP	0.0116	0.0484	0.3807	0.4306	0.517	1	0.1547
	ACT	0.1741	0.2545	0.0459	0.0357	0.0238	0.0237	0.0281
	MFI	0.0056	0.005	0.0948	0.151	0.1895	0.0112	0.0133
	L3	1	1	0.9533	1	1	0.3729	0.5454
	LOCALP	0.8436	0.9752	0.7208	0.2931	0.3428	0.4477	0.5894
	CCLP	0.517	0.56	0.5655	0.0153	0.0553	0.1194	0.1357
	NLC	0.5464	0.3937	0.3868	0.0099	0.0468	0.0792	0.0803
	CARCN	0.5114	0.1445	0.1494	0.0034	0.0144	0.0497	0.0632

5.2.3 Feature score used in CFLP

This section will discuss the two feature selection techniques, F-Regression (FREG) and Mutual Information Classifier (MIC), used in this paper for feature selection. Table 5.1 shows the feature score of different similarity indices. Initially, we used fourteen methods for feature creation. The methods used are Adamic Adar (AA), Common Neighbor (CN), Jaccard Coefficient (JC), Preferential Attachment (PA), Salton Cosine Similarity (SI), Shortest Path (SP), Cosine Similarity (COS+), Average Commute Time (ACT), Matrix Forest Index (MFI), Path of Length 3 (L3), Local Path (LOCALP), Clustering Coefficient Based Link Prediction (CCLP), Node and Link Clustering

Coefficient (NLC), CAR-based Common Neighbor Index (CARCN). As we add more methods to the prediction process, the overall complexity of the link prediction algorithm increases significantly. Therefore, we have applied feature selection techniques to optimize the features based on feature scores. We have used the highest similarity score of various similarity indices for feature creation. The experiment observed that L3, LOCALP, CCLP, MFI, and PA produce similar scores on both feature selection methods. The feature score of L3 is the highest of all datasets. The LOCALP, CCLP, and MFI methods give the second-highest score on all datasets. From the table, we can observe that in F-Regression (FREG) method, the best local scoring metrics are CN, PA, and AA. We have combined these three methods with the above four high-scoring methods and named this method CFLP-Local (CFLP-L). So based on the highest score, the methods used in CFLP-Local after optimization are L3, LOCALP, CCLP, MFI, CN, PA, and AA. Similarly, for Mutual Information Classifier (MIC) based feature selection, the best scoring global methods are NLC, SP, and PA. We have combined these three methods with the above four high-scoring methods and named them CFLP-Global (CFLP-G). Based on the highest scores, the methods used for feature creation in *CFLP_Global* are L3, LOCALP, CCLP, MFI, NLC, SP, and PA.

The feature scores of all these common methods are highlighted in Table 5.1. Therefore, we have considered two optimized version of proposed CFLP method CFLP-Local (CFLP-L) and CFLP-Global (CFLP-G). We have evaluated the performance of these two variations of CFLP on different machine learning models. The experimental result shows that the optimized methods are more accurate and less complex. The experimental analysis of two CFLP variations CFLP-Local and CFLP-Global are discussed in section 5.3.6.

- *f_regression* .

The best features are chosen using univariate statistical tests in univariate feature selection. In order to uncover potentially essential features for a downstream

classifier, *f_regression* [156] is considered a feature selection criterion, regardless of the correlation of the connection with the target variable.

- *mutual_info_regression*.

Mutual information (MI) [156] among two random variables is a non-negative number that indicates how dependent the variables are on each other. It is zero if two random variables are autonomous, while larger values indicate more dependence. For a continuous target variable, *mutual_info_regression* calculates mutual information correlation between feature score and predicted value.

5.3 Result Analysis

This section examines the experimental results received from the experiments. To begin, we employ five performance evaluation metrics to compare the CFLP result to individual features in different machine learning algorithms such as Neural Network (NN), Logistic Regression (LR), XGBoost (XGB), Random Forest Classifier (RFC), and Linear Discriminant Analysis (LDA) on seven well-known dynamic datasets. The experiments showed that machine learning variations of CFLP variation yield better accuracy on seven well-known dynamic networks with five evaluation metrics. We also compared and examined the final result with optimized local and global features with state-of-the-art algorithms.

5.3.1 Performance comparison of CFLP model with different similarity methods on Neural Network (NN)

This part will look at the experimental aspects and effectiveness of the CFLP approach while utilizing neural network for training and prediction. We employed two hidden layers with 1024 neurons to train the neural network model. The learning rate is 0.001 in

this case. We trained the model for 5 epochs with a batch size of 32. Table 5.2 compares the CFLP model's performance with various similarity indices using neural network. In terms of AUC, CFLP produces good results on CollegeMsg dataset while producing average results on other datasets. For AUPR, CFLP produces good results on Mit and CollegeMsg datasets while performing moderately well on other datasets. For balance accuracy score (BAL ACC), CFLP produces excellent results on Lkml-reply and Mathoverflow datasets and average results on other datasets. For F1 SCORE, CFLP produces moderately good results across all datasets. For average precision (AVG PREC), our model produces excellent results across all datasets. On Fb-forum dataset, it outperformed all models except SP and L3. On Mathoverflow dataset, it outperformed all except L3, and on Lkml-reply dataset, it outperformed all except L3 and COSP.

5.3.2 Performance comparison of CFLP model with different similarity methods on Logistic Regression (LR)

This part will look at the experimental details and effectiveness of the CFLP approach while employing Logistic Regression (LR) for training and prediction. The comparison of the effectiveness of the CFLP model and different similarity indices with Logistic Regression (LR) is displayed in Table 5.3. In terms of AUC, CFLP produces good results on Radoslaw-email and CollegeMsg datasets while generating average results on other datasets. For AUPR, CFLP produces good results on Fb-forum dataset and moderate results on other datasets. For BAL ACC, the CFLP method produces better results than the global feature method on the Mit dataset but local and quasi-local methods outperform it. CFLP produces average results on other datasets. For F1 SCORE, CFLP produces good results on CollegeMsg dataset but performs rather poorly on other datasets. For AVG PREC, CFLP produces good results in CollegeMsg dataset and average results in other datasets.

TABLE 5.2: Performance comparison and analysis of CFLP model with various individual similarity indices with Neural Network (NN)

Metric	Dataset	CN	AA	JC	PA	SP	CCLP	SI	L3	NLC	COSP	ACT	MFI	CARC	LOCALP	CFLP
AUC	Mit	0.71589	0.67528	0.76706	0.64257	0.84672	0.82773	0.75598	0.71028	0.85372	0.88594	0.64166	0.85232	0.61435	0.69663	0.67473
	Radoslaw-email	0.77209	0.83502	0.87213	0.55737	0.91299	0.86276	0.85782	0.79003	0.86666	0.92735	0.55627	0.92233	0.53275	0.70617	0.71587
	Eu-core	0.8827	0.92462	0.92413	0.51606	0.97139	0.93733	0.93347	0.93757	0.93784	0.97353	0.51582	0.94949	0.71612	0.81329	0.7733
	Fb-forum	0.53698	0.56339	0.61514	0.55235	0.90913	0.57425	0.61835	0.917	0.60116	0.9185	0.50574	0.89369	0.49892	0.80554	0.64437
	CollegeMsg	0.50907	0.51339	0.54974	0.54851	0.65529	0.51554	0.55037	0.67368	0.53449	0.67006	0.4927	0.64024	0.48715	0.61789	0.5447
	Lkml-reply	0.79572	0.7961	0.79237	0.72339	0.81011	0.78835	0.79291	0.81183	0.76794	0.81177	0.7605	0.80451	0.70687	0.81069	0.76356
Mathoverflow	0.70256	0.70357	0.70187	0.65191	0.73045	0.69012	0.70273	0.73451	0.67214	0.71779	0.68361	0.72107	0.62367	0.72324	0.68339	
AUPR	Mit	0.5069	0.43079	0.52295	0.53096	0.57598	0.61282	0.48561	0.53012	0.62776	0.67731	0.51119	0.59817	0.48481	0.54743	0.56843
	Radoslaw-email	0.48227	0.52815	0.62895	0.3615	0.77521	0.5855	0.58054	0.51215	0.60504	0.80591	0.36231	0.75076	0.37977	0.44446	0.50811
	Eu-core	0.64707	0.73014	0.69053	0.27753	0.84318	0.77023	0.71973	0.76023	0.7739	0.87008	0.34359	0.77172	0.58651	0.59686	0.61573
	Fb-forum	0.23124	0.23213	0.30111	0.33939	0.75887	0.27396	0.30847	0.81018	0.40464	0.83354	0.25179	0.75841	0.33922	0.4887	0.40534
	CollegeMsg	0.18607	0.19392	0.22026	0.2717	0.36163	0.2109	0.22816	0.45842	0.25291	0.47096	0.23765	0.41954	0.21237	0.32504	0.27393
	Lkml-reply	0.72804	0.73065	0.69956	0.56306	0.69764	0.72777	0.7039	0.72166	0.72997	0.71413	0.51994	0.66873	0.65525	0.71555	0.53094
Mathoverflow	0.62747	0.62888	0.60584	0.48827	0.58774	0.61657	0.61503	0.61489	0.61478	0.61253	0.44458	0.54106	0.55779	0.59541	0.44846	
BAL ACC	Mit	0.6714	0.6371	0.7056	0.6419	0.80994	0.74142	0.69681	0.68055	0.77952	0.86301	0.62574	0.77513	0.61357	0.68006	0.67469
	Radoslaw-email	0.71201	0.77041	0.79767	0.55445	0.83053	0.78714	0.78815	0.72252	0.79195	0.8459	0.55991	0.84324	0.56506	0.67279	0.69055
	Eu-core	0.80429	0.85649	0.85894	0.52115	0.93578	0.87677	0.86953	0.87039	0.87595	0.93599	0.51481	0.87041	0.76141	0.76659	0.74961
	Fb-forum	0.53347	0.55029	0.5836	0.55717	0.84425	0.55874	0.58796	0.86428	0.57098	0.90612	0.50701	0.86271	0.50022	0.75032	0.62816
	CollegeMsg	0.50945	0.51077	0.53283	0.53395	0.6008	0.51782	0.52985	0.64796	0.53556	0.66291	0.49691	0.62794	0.49331	0.60863	0.52993
	Lkml-reply	0.78209	0.78219	0.77984	0.74322	0.7804	0.75991	0.78165	0.78962	0.74222	0.7884	0.76052	0.77525	0.73283	0.78688	0.76223
Mathoverflow	0.69598	0.69677	0.69439	0.67493	0.7074	0.66893	0.69648	0.71943	0.64059	0.69896	0.68392	0.69711	0.63805	0.70972	0.68341	
F1 SCORE	Mit	0.46063	0.41592	0.524	0.3949	0.66006	0.5881	0.49999	0.45955	0.60968	0.72574	0.34915	0.57722	0.35716	0.46294	0.48211
	Radoslaw-email	0.48916	0.56327	0.6046	0.26757	0.68639	0.58011	0.586	0.51033	0.59282	0.70526	0.20822	0.67202	0.3017	0.44939	0.47222
	Eu-core	0.57295	0.66019	0.65649	0.19434	0.84425	0.69556	0.6913	0.69512	0.68186	0.81018	0.13188	0.72629	0.51061	0.53934	0.57163
	Fb-forum	0.26282	0.27591	0.3153	0.27236	0.65193	0.26152	0.324	0.73188	0.2704	0.81684	0.13324	0.75418	0.23	0.49996	0.36213
	CollegeMsg	0.21253	0.22119	0.22569	0.26131	0.32945	0.13737	0.22401	0.40952	0.16058	0.45884	0.2077	0.39148	0.16371	0.3443	0.25328
	Lkml-reply	0.6472	0.64903	0.64913	0.58895	0.6235	0.64704	0.64814	0.65187	0.63152	0.64422	0.54255	0.60874	0.61042	0.63802	0.54438
Mathoverflow	0.52672	0.52591	0.47764	0.51073	0.49115	0.52919	0.55451	0.63319	0.46319	0.54003	0.44328	0.50584	0.42676	0.52903	0.44144	
AVG PREC	Mit	0.49262	0.43699	0.52765	0.34551	0.60404	0.61912	0.49118	0.51284	0.63385	0.68144	0.42688	0.60432	0.31424	0.49024	0.37858
	Radoslaw-email	0.4847	0.53079	0.63024	0.27062	0.76718	0.58774	0.58211	0.51433	0.6074	0.80656	0.34859	0.75229	0.30048	0.4446	0.44349
	Eu-core	0.64771	0.73159	0.69302	0.19109	0.87317	0.72253	0.72253	0.76128	0.77508	0.87083	0.25115	0.77349	0.43303	0.59397	0.57348
	Fb-forum	0.22091	0.22902	0.29201	0.25477	0.7627	0.26604	0.2978	0.80953	0.29349	0.8316	0.19499	0.76073	0.18814	0.49013	0.35507
	CollegeMsg	0.1846	0.19092	0.20984	0.22516	0.35141	0.20136	0.21665	0.44767	0.23086	0.45531	0.17219	0.42195	0.16556	0.32495	0.20477
	Lkml-reply	0.63566	0.64031	0.60722	0.49198	0.64182	0.6362	0.61608	0.67096	0.6407	0.66504	0.36024	0.62509	0.50061	0.66384	0.37
Mathoverflow	0.49004	0.49524	0.47814	0.39236	0.51955	0.48191	0.48111	0.55149	0.48068	0.52161	0.27953	0.49855	0.36447	0.5307	0.2783	

TABLE 5.3: Performance comparison and analysis of CFLP model with various individual similarity indices with Logistic Regression (LR)

Metric	Dataset	CN	AA	JC	PA	SP	CCLP	SI	L3	NLC	COSP	ACT	MFI	CARCN	LOCALP	CFLP
AUC	Mit	0.7745	0.83328	0.77279	0.68626	0.8401	0.82496	0.75969	0.8494	0.82277	0.87906	0.81286	0.35364	0.51772	0.7535	0.68154
	Radoslaw-email	0.86043	0.85549	0.857	0.34486	0.86973	0.85991	0.85703	0.86554	0.85835	0.78161	0.80625	0.90671	0.50934	0.80444	0.77194
	Eu-core	0.923	0.93206	0.92637	0.2647	0.92637	0.92637	0.93864	0.93299	0.94306	0.9702	0.7648	0.94935	0.88046	0.9188	0.81017
	Fb-forum	0.56149	0.57048	0.54516	0.47573	0.80674	0.58042	0.55988	0.90721	0.61279	0.89925	0.68196	0.8728	0.49815	0.82185	0.70942
	CollegeMsg	0.54698	0.53599	0.49654	0.51641	0.54153	0.52325	0.50853	0.67869	0.53644	0.6416	0.55146	0.62432	0.48593	0.64518	0.56318
	Lkml-reply	0.79007	0.79085	0.7906	0.79975	0.76066	0.79525	0.79331	0.80589	0.79425	0.79978	0.27328	0.80263	0.76473	0.7969	0.5314
	Mathoverflow	0.70339	0.70443	0.69992	0.72442	0.6806	0.69831	0.70172	0.73151	0.69826	0.72039	0.34642	0.72244	0.65837	0.73186	0.50701
	Mit	0.55911	0.56509	0.43012	0.44655	0.46534	0.53863	0.41693	0.63918	0.55243	0.67097	0.53298	0.20847	0.36668	0.53491	0.44676
	Radoslaw-email	0.5755	0.56429	0.55139	0.18023	0.4795	0.5878	0.56446	0.60704	0.59212	0.5234	0.42242	0.6375	0.37746	0.55465	0.45454
	Eu-core	0.72108	0.74243	0.70832	0.11545	0.81108	0.77746	0.71814	0.75847	0.78857	0.82459	0.4378	0.75332	0.6873	0.70218	0.59082
	Fb-forum	0.24245	0.23234	0.19398	0.25922	0.35115	0.25489	0.20265	0.81951	0.31054	0.77041	0.25786	0.72193	0.37295	0.52235	0.36297
	CollegeMsg	0.22255	0.21266	0.26794	0.22215	0.22129	0.2181	0.20832	0.48242	0.26507	0.42224	0.31161	0.4031	0.27095	0.35612	0.25641
Lkml-reply	0.72834	0.73096	0.6797	0.6987	0.39435	0.72949	0.6986	0.71934	0.72877	0.70184	0.24614	0.65926	0.71737	0.71002	0.52044	
Mathoverflow	0.62874	0.63058	0.57765	0.58503	0.30758	0.61028	0.59111	0.60988	0.61355	0.61054	0.31772	0.53988	0.60314	0.60576	0.47437	
BAL ACC	Mit	0.73624	0.75948	0.71652	0.64561	0.84606	0.75629	0.70418	0.79936	0.76012	0.82959	0.73108	0.41118	0.46878	0.68953	0.65674
	Radoslaw-email	0.78104	0.77295	0.78553	0.37459	0.78825	0.76888	0.78559	0.7751	0.7581	0.78991	0.72851	0.78785	0.44474	0.75986	0.65974
	Eu-core	0.84364	0.85621	0.8538	0.32417	0.93594	0.84223	0.8662	0.86567	0.82433	0.90288	0.68175	0.8426	0.75578	0.85365	0.71246
	Fb-forum	0.5501	0.55651	0.52152	0.44842	0.79305	0.55435	0.56396	0.87128	0.56832	0.85953	0.64561	0.82014	0.49834	0.73088	0.65758
	CollegeMsg	0.53262	0.5317	0.49665	0.48449	0.53434	0.52221	0.50083	0.67182	0.52811	0.63329	0.55084	0.62133	0.48346	0.61885	0.55612
	Lkml-reply	0.75333	0.75674	0.74334	0.75039	0.74547	0.69888	0.77254	0.75487	0.66988	0.77845	0.38673	0.7758	0.63326	0.76448	0.41263
	Mathoverflow	0.69021	0.69055	0.67936	0.69753	0.67181	0.62151	0.69502	0.69945	0.59253	0.67565	0.41447	0.70613	0.56524	0.70369	0.44626
	Mit	0.57088	0.59809	0.52833	0.45121	0.70514	0.59581	0.51893	0.66035	0.60002	0.67349	0.52418	0.26677	0.29607	0.50003	0.46337
	Radoslaw-email	0.5951	0.5881	0.59115	0.21753	0.61541	0.59362	0.59204	0.59458	0.5947	0.64487	0.49066	0.63038	0.28742	0.57457	0.42879
	Eu-core	0.68882	0.7119	0.70629	0.15906	0.83882	0.7289	0.71211	0.72708	0.72646	0.80776	0.39854	0.72168	0.62691	0.67691	0.4407
	Fb-forum	0.26576	0.27741	0.28603	0.22399	0.55931	0.25506	0.29944	0.77978	0.258	0.7846	0.37963	0.73712	0.22835	0.51966	0.39549
	CollegeMsg	0.21694	0.21624	0.25933	0.26157	0.27711	0.14691	0.26004	0.47035	0.14284	0.40076	0.28842	0.38191	0.27515	0.36703	0.29286
Lkml-reply	0.64433	0.64951	0.60595	0.63806	0.52839	0.56143	0.64764	0.65058	0.5037	0.64693	0.21177	0.59588	0.41775	0.66544	0.22738	
Mathoverflow	0.52967	0.5307	0.49866	0.54194	0.42904	0.38899	0.52701	0.55003	0.31264	0.50762	0.22467	0.50622	0.23243	0.55107	0.24439	
AVG PREC	Mit	0.56559	0.57359	0.43757	0.45295	0.53517	0.54679	0.42385	0.64433	0.56121	0.67539	0.53924	0.21451	0.37389	0.54003	0.45573
	Radoslaw-email	0.57799	0.56672	0.55349	0.18228	0.63819	0.59028	0.56655	0.60914	0.59448	0.52475	0.4237	0.63968	0.37904	0.55685	0.45721
	Eu-core	0.72243	0.74386	0.7112	0.11687	0.85021	0.77847	0.72082	0.75975	0.78938	0.82696	0.44128	0.75601	0.68531	0.70367	0.59258
	Fb-forum	0.23524	0.24843	0.18658	0.26166	0.35558	0.247	0.19496	0.82007	0.29905	0.77043	0.25014	0.72573	0.18881	0.52518	0.36247
	CollegeMsg	0.20519	0.20194	0.16526	0.21969	0.17848	0.21226	0.19496	0.82007	0.29905	0.77043	0.25014	0.72573	0.18881	0.52518	0.36247
	Lkml-reply	0.63082	0.63573	0.65397	0.65374	0.63281	0.60651	0.66652	0.66572	0.65472	0.6514	0.11081	0.61573	0.58474	0.65591	0.39833
	Mathoverflow	0.49302	0.49821	0.44489	0.54055	0.26688	0.47282	0.45876	0.55344	0.4747	0.51631	0.12316	0.49698	0.4053	0.54846	0.30703

5.3.3 Performance comparison of CFLP model with different similarity methods with Extreme Gradient boosting XGBoost (XGB)

In this part, we examine the performance of the CFLP model on feature sets using XGBoost (XGB) as a training and testing model. In this model, we utilized a learning rate of 0.01 and several estimators of 50. The number of boosting rounds is indicated by *n_estimators*. A single tree's greatest depth is regarded to be six. XGBoost initializes tree pruning from the backward direction and utilizes the *max_depth* option. This increases the XGBoost framework's computational performance and speed. Table 5.4 compares the CFLP model's performance with various other similarity indices using XGBoost (XGB). If we observe AUC, CFLP is the best performing method on all datasets except on Radoslaw-email and Lkml-reply datasets, on which it is the second-best method slightly behind COSP in Radoslaw-email dataset and JC in Lkml-reply dataset. For AUPR, CFLP is the best performing method on all datasets except Fb-forum, CollegeMsg, and Radoslaw-email. It is the second-best performing method on Fb-forum and CollegeMsg datasets, lagging only behind SP, and the third-best performing method on Radoslaw-email dataset behind SP and COSP. For BAL ACC, CFLP is the best performing method on all the datasets except on Radoslaw-email and Lkml-reply datasets. On Radoslaw-email dataset, it is the second-best performing method only behind COSP, and on Lkml-reply dataset, CFLP is the fourth-best method. For F1 SCORE, CFLP is the highest scoring method across all datasets barring the Eu-core dataset, where it very closely follows the SP method. For AVG PREC, CFLP produces the best results on all the datasets except CollegeMsg and Radoslaw-email datasets. On CollegeMsg dataset, it is the second-best method only behind SP, while on Radoslaw-email dataset, it is the third-best method behind SP and COSP.

TABLE 5.4: Performance comparison and analysis of CFLP model with various individual similarity indices with XGBoost (XGB)

Metric	Dataset	CN	AA	JC	PA	SP	CCLP	SI	I3	NLC	COSP	ACT	MFI	CARCN	LOCALP	CHLP	
AUC	Mit	0.7201	0.70537	0.68384	0.75841	0.82573	0.73124	0.6734	0.76724	0.73509	0.82741	0.76179	0.801	0.73366	0.75422	0.82972	
	Radoslaw-email	0.68733	0.69024	0.70848	0.61422	0.79646	0.709	0.71487	0.67596	0.70409	0.80967	0.64458	0.79854	0.68861	0.66633	0.80636	
	Eu-core	0.78839	0.79836	0.79309	0.58579	0.82579	0.81752	0.81079	0.82304	0.81715	0.90082	0.59705	0.85635	0.78984	0.77917	0.92079	
	Fb-forum	0.51547	0.52021	0.5306	0.58274	0.85933	0.52924	0.52924	0.85692	0.54035	0.86826	0.61475	0.84618	0.51244	0.73325	0.88664	
	CollegeMsg	0.50374	0.50351	0.51192	0.57215	0.65147	0.50991	0.50895	0.63791	0.5216	0.63603	0.56265	0.62596	0.50302	0.59889	0.63986	
	Lkml-reply	0.75157	0.75265	0.77709	0.75085	0.76879	0.75001	0.77457	0.75963	0.75704	0.75196	0.72849	0.74324	0.74997	0.75632	0.7681	
	Mathoverflow	0.68422	0.67794	0.68313	0.68899	0.68108	0.67697	0.68106	0.69463	0.67586	0.68188	0.65861	0.68219	0.64798	0.68614	0.69896	
	Mit	0.62625	0.5984	0.58922	0.68087	0.74597	0.63603	0.58658	0.68837	0.64951	0.76213	0.68131	0.71961	0.63686	0.66398	0.76097	
	Radoslaw-email	0.59358	0.59676	0.6178	0.49785	0.77395	0.61116	0.62531	0.58737	0.60544	0.76579	0.54894	0.75489	0.59083	0.57371	0.75937	
	Eu-core	0.69628	0.71029	0.70206	0.45924	0.87151	0.73703	0.7299	0.73754	0.74635	0.8311	0.46734	0.76396	0.70035	0.68993	0.86962	
	Fb-forum	0.34143	0.35031	0.39454	0.44669	0.82946	0.36985	0.38737	0.82522	0.41904	0.82889	0.52051	0.79621	0.35738	0.63642	0.85218	
	CollegeMsg	0.29127	0.21793	0.32036	0.44565	0.62995	0.27928	0.26966	0.58564	0.34531	0.59764	0.45002	0.56692	0.27329	0.49475	0.57928	
Lkml-reply	0.71726	0.72167	0.70166	0.71247	0.69959	0.7208	0.69964	0.72815	0.72452	0.71539	0.68318	0.696	0.71735	0.7179	0.74163		
Mathoverflow	0.63681	0.64311	0.63881	0.64687	0.64303	0.63968	0.63712	0.66016	0.64812	0.65138	0.61731	0.62227	0.6405	0.65352	0.67245		
AUPR	Mit	0.7201	0.70537	0.68384	0.75841	0.82573	0.73124	0.6734	0.76724	0.73509	0.82741	0.76179	0.801	0.73366	0.75422	0.82972	
	Radoslaw-email	0.68733	0.69024	0.70848	0.61422	0.79646	0.709	0.71487	0.67596	0.70409	0.80967	0.64458	0.79854	0.68861	0.66633	0.80636	
	Eu-core	0.78839	0.79836	0.79309	0.58579	0.82579	0.81752	0.81079	0.82304	0.81715	0.90082	0.59705	0.85635	0.78984	0.77917	0.92079	
	Fb-forum	0.51547	0.52021	0.5306	0.58274	0.85933	0.52924	0.52924	0.85692	0.54035	0.86826	0.61475	0.84618	0.51244	0.73325	0.88664	
	CollegeMsg	0.50374	0.50351	0.51192	0.57215	0.65147	0.50991	0.50895	0.63791	0.5216	0.63603	0.56265	0.62596	0.50302	0.59889	0.63986	
	Lkml-reply	0.75157	0.75265	0.77709	0.75085	0.76879	0.75001	0.77457	0.75963	0.75704	0.75196	0.72849	0.74324	0.74997	0.75632	0.7681	
	Mathoverflow	0.68422	0.67794	0.68313	0.68899	0.68108	0.67697	0.68106	0.69463	0.67586	0.68188	0.65861	0.68219	0.64798	0.68614	0.69896	
	Mit	0.62625	0.5984	0.58922	0.68087	0.74597	0.63603	0.58658	0.68837	0.64951	0.76213	0.68131	0.71961	0.63686	0.66398	0.76097	
	Radoslaw-email	0.59358	0.59676	0.6178	0.49785	0.77395	0.61116	0.62531	0.58737	0.60544	0.76579	0.54894	0.75489	0.59083	0.57371	0.75937	
	Eu-core	0.69628	0.71029	0.70206	0.45924	0.87151	0.73703	0.7299	0.73754	0.74635	0.8311	0.46734	0.76396	0.70035	0.68993	0.86962	
	Fb-forum	0.34143	0.35031	0.39454	0.44669	0.82946	0.36985	0.38737	0.82522	0.41904	0.82889	0.52051	0.79621	0.35738	0.63642	0.85218	
	CollegeMsg	0.29127	0.21793	0.32036	0.44565	0.62995	0.27928	0.26966	0.58564	0.34531	0.59764	0.45002	0.56692	0.27329	0.49475	0.57928	
Lkml-reply	0.71726	0.72167	0.70166	0.71247	0.69959	0.7208	0.69964	0.72815	0.72452	0.71539	0.68318	0.696	0.71735	0.7179	0.74163		
Mathoverflow	0.63681	0.64311	0.63881	0.64687	0.64303	0.63968	0.63712	0.66016	0.64812	0.65138	0.61731	0.62227	0.6405	0.65352	0.67245		
BAL_ACC	Mit	0.7201	0.70537	0.68384	0.75841	0.82573	0.73124	0.6734	0.76724	0.73509	0.82741	0.76179	0.801	0.73366	0.75422	0.82972	
	Radoslaw-email	0.68733	0.69024	0.70848	0.61422	0.79646	0.709	0.71487	0.67596	0.70409	0.80967	0.64458	0.79854	0.68861	0.66633	0.80636	
	Eu-core	0.78839	0.79836	0.79309	0.58579	0.82579	0.81752	0.81079	0.82304	0.81715	0.90082	0.59705	0.85635	0.78984	0.77917	0.92079	
	Fb-forum	0.51547	0.52021	0.5306	0.58274	0.85933	0.52924	0.52924	0.85692	0.54035	0.86826	0.61475	0.84618	0.51244	0.73325	0.88664	
	CollegeMsg	0.50374	0.50351	0.51192	0.57215	0.65147	0.50991	0.50895	0.63791	0.5216	0.63603	0.56265	0.62596	0.50302	0.59889	0.63986	
	Lkml-reply	0.75157	0.75265	0.77709	0.75085	0.76879	0.75001	0.77457	0.75963	0.75704	0.75196	0.72849	0.74324	0.74997	0.75632	0.7681	
	Mathoverflow	0.68422	0.67794	0.68313	0.68899	0.68108	0.67697	0.68106	0.69463	0.67586	0.68188	0.65861	0.68219	0.64798	0.68614	0.69896	
	Mit	0.56838	0.54324	0.51847	0.63305	0.71601	0.58302	0.507	0.64368	0.59676	0.72991	0.63545	0.68334	0.58582	0.61812	0.72903	
	Radoslaw-email	0.51765	0.52235	0.55223	0.37559	0.71698	0.55065	0.56304	0.5002	0.54186	0.7241	0.43997	0.70861	0.51803	0.4818	0.71779	
	Eu-core	0.66253	0.67837	0.66963	0.29696	0.86012	0.70909	0.69949	0.71129	0.716	0.81683	0.32399	0.74311	0.66643	0.65348	0.85835	
	Fb-forum	0.07238	0.09341	0.12852	0.29119	0.80139	0.10671	0.1251	0.79708	0.16183	0.80503	0.36616	0.76881	0.05801	0.58424	0.83264	
	CollegeMsg	0.06685	0.03379	0.06042	0.26071	0.41101	0.05632	0.05338	0.42316	0.41936	0.22975	0.39494	0.06626	0.32778	0.4271		
Lkml-reply	0.64299	0.64608	0.66024	0.64011	0.65215	0.64524	0.65704	0.65716	0.65259	0.64267	0.60047	0.62407	0.64111	0.64886	0.67286		
Mathoverflow	0.52059	0.51092	0.51917	0.53083	0.51652	0.50858	0.51526	0.54541	0.50815	0.51956	0.47025	0.5138	0.45126	0.52756	0.55352		
F1 SCORE	Mit	0.43189	0.40477	0.3995	0.49417	0.57059	0.44177	0.39519	0.50272	0.45797	0.59555	0.49373	0.53689	0.44219	0.47167	0.59258	
	Radoslaw-email	0.39181	0.39434	0.41636	0.30492	0.60507	0.41114	0.42335	0.3841	0.40366	0.59845	0.34618	0.58274	0.38911	0.37034	0.58922	
	Eu-core	0.50357	0.52171	0.51037	0.2527	0.76245	0.55647	0.585817	0.58187	0.69434	0.25909	0.59179	0.50919	0.49577	0.7588		
	Fb-forum	0.1839	0.18615	0.19756	0.24879	0.28912	0.19339	0.1984	0.68444	0.20971	0.69173	0.29427	0.64157	0.17969	0.42956	0.73006	
	CollegeMsg	0.17282	0.17631	0.17794	0.23837	0.35786	0.18049	0.17297	0.34282	0.18798	0.3465	0.23385	0.32433	0.16913	0.27249	0.34198	
	Lkml-reply	0.51497	0.51994	0.50839	0.50985	0.50399	0.51764	0.50568	0.53019	0.52488	0.51354	0.47076	0.4902	0.51451	0.51818	0.54828	
	Mathoverflow	0.40686	0.40622	0.40729	0.41703	0.40872	0.40367	0.4048	0.43063	0.40813	0.41473	0.37493	0.39524	0.37892	0.41976	0.44277	
	AVG PREC	Mit	0.43189	0.40477	0.3995	0.49417	0.57059	0.44177	0.39519	0.50272	0.45797	0.59555	0.49373	0.53689	0.44219	0.47167	0.59258
		Radoslaw-email	0.39181	0.39434	0.41636	0.30492	0.60507	0.41114	0.42335	0.3841	0.40366	0.59845	0.34618	0.58274	0.38911	0.37034	0.58922
		Eu-core	0.50357	0.52171	0.51037	0.2527	0.76245	0.55647	0.585817	0.58187	0.69434	0.25909	0.59179	0.50919	0.49577	0.7588	
		Fb-forum	0.1839	0.18615	0.19756	0.24879	0.28912	0.19339	0.1984	0.68444	0.20971	0.69173	0.29427	0.64157	0.17969	0.42956	0.73006
		CollegeMsg	0.17282	0.17631	0.17794	0.23837	0.35786	0.18049	0.17297	0.34282	0.18798	0.3465	0.23385	0.32433	0.16913	0.27249	0.34198
Lkml-reply		0.51497	0.51994	0.50839	0.50985	0.50399	0.51764	0.50568	0.53019	0.52488	0.51354	0.47076	0.4902	0.51451	0.51818	0.54828	
Mathoverflow		0.40686	0.40622	0.40729	0.41703	0.40872	0.40367	0.4048	0.43063	0.40813	0.41473	0.37493	0.39524	0.37892	0.41976	0.44277	

5.3.4 Performance comparison of CFLP model with different similarity methods with Random Forest Classifier (RFC)

In this part, we examine the performance of the CFLP model on feature sets using Random Forest Classifier (RFC) as a training and testing model. Table 5.5 compares the CFLP model's performance with various other similarity indices using the random forest classifier. If we observe AUC, CFLP produces excellent results surpassing all other methods on all tested datasets, sometimes by a large margin. If we consider AUPR, then also CFLP outperforms every other method on all tested datasets. Upon considering BAL ACC, CFLP once again is the best scoring method across all the tested datasets. For F1 SCORE, CFLP produces the best results on all datasets except Radoslaw-email, Eu-core, and Fb-forum datasets. It is the second-best method lagging only behind SP on these datasets. For AVG PREC, CFLP generates the best results among all other methods on every tested dataset.

5.3.5 Performance comparison of CFLP model with different similarity methods Linear Discriminant Analysis (LDA)

The comparison and analysis of the performance of the CFLP model with Linear Discriminant Analysis (LDA) are shown in Table 5.6. In terms of AUC, CFLP is the best method across all the datasets except Mathoverflow dataset, where it is the third-best method behind L3 and LOCALP. For AUPR, CFLP is the best performing method across all datasets except Lkml-reply and Mathoverflow datasets. On Lkml-reply dataset, it is the fourth-best method, while on the Mathoverflow dataset, it produces above average results. For BAL ACC, CFLP produces the best results across all the tested datasets.

For F1 SCORE, CFLP generates the best output on Radoslaw-email, CollegeMsg, Lkml-reply, and Mathoverflow datasets. It is the second-best performing method on Mit,

TABLE 5.5: Performance comparison and analysis of CFLP model with various individual similarity indices with Random Forest Classifier (RFC)

Metric	Dataset	CN	AA	JC	PA	SP	CCLP	SI	L3	NLC	COSP	ACT	MFI	CARCN	LOCALP	CFLP	
AUC	Mit	0.8239	0.83665	0.80935	0.87627	0.89485	0.84256	0.81649	0.88141	0.85917	0.89556	0.89519	0.90508	0.84149	0.86094	0.92359	
	Radoslaw-email	0.82338	0.85081	0.85139	0.80656	0.91189	0.84853	0.85647	0.85484	0.85151	0.91729	0.82845	0.91648	0.83904	0.84163	0.92326	
	Eu-core	0.88986	0.92031	0.90955	0.73316	0.97597	0.9222	0.94579	0.92172	0.92745	0.7588	0.97245	0.7588	0.96021	0.89413	0.90937	0.98184
	Fb-forum	0.56679	0.55809	0.56503	0.71631	0.93072	0.57034	0.57183	0.91647	0.56219	0.91211	0.78449	0.92463	0.54067	0.86116	0.94445	
	CollegeMsg	0.5315	0.53311	0.53833	0.63617	0.72636	0.52584	0.54171	0.70837	0.54028	0.66714	0.64998	0.69883	0.50764	0.66038	0.76694	
	Lkml-reply	0.79263	0.78688	0.7878	0.80068	0.81174	0.78591	0.78368	0.78647	0.78234	0.78468	0.77877	0.79931	0.76434	0.80387	0.81767	
	Mathoverflow	0.70251	0.69513	0.69605	0.73234	0.73484	0.68831	0.69033	0.71613	0.68775	0.69664	0.70046	0.721	0.65683	0.72883	0.75037	
	Mit	0.55936	0.5916	0.53592	0.67012	0.70546	0.56775	0.55547	0.66824	0.60391	0.71285	0.68237	0.71668	0.58651	0.65341	0.74276	
	Radoslaw-email	0.52742	0.59078	0.58602	0.5003	0.80918	0.59529	0.60087	0.58373	0.60115	0.79987	0.54783	0.79148	0.56907	0.56092	0.81503	
	Eu-core	0.65703	0.72692	0.70351	0.39398	0.86129	0.73735	0.72222	0.76218	0.75203	0.86251	0.43611	0.80594	0.68188	0.67968	0.90028	
AUPR	Fb-forum	0.24188	0.22507	0.25416	0.40396	0.84152	0.24954	0.25559	0.82422	0.26712	0.81633	0.49603	0.83065	0.28978	0.65634	0.87512	
	CollegeMsg	0.20956	0.20554	0.21106	0.29562	0.49765	0.20686	0.22821	0.48568	0.2353	0.46035	0.31815	0.45151	0.22715	0.36595	0.53615	
	Lkml-reply	0.72107	0.71577	0.69933	0.69146	0.70169	0.71515	0.69218	0.69519	0.72432	0.67772	0.65904	0.67497	0.71974	0.70306	0.73914	
	Mathoverflow	0.62268	0.61983	0.61349	0.59191	0.59564	0.61337	0.60438	0.59898	0.62308	0.59717	0.54289	0.55275	0.6214	0.60027	0.6372	
	Mit	0.76332	0.78212	0.73823	0.80679	0.85671	0.78432	0.74479	0.80367	0.79426	0.85318	0.82605	0.84933	0.78081	0.78789	0.86488	
	Radoslaw-email	0.76087	0.77557	0.77527	0.73327	0.83567	0.77181	0.78282	0.78579	0.7745	0.84342	0.74869	0.83692	0.77123	0.77204	0.8526	
	Eu-core	0.83703	0.85907	0.84222	0.67823	0.94347	0.86718	0.86064	0.88699	0.86294	0.9366	0.68971	0.91513	0.84557	0.83978	0.94472	
	Fb-forum	0.54835	0.5429	0.56522	0.66052	0.89896	0.57316	0.57253	0.86608	0.58	0.88562	0.72192	0.883	0.5437	0.80712	0.90565	
	CollegeMsg	0.52372	0.51931	0.52908	0.58744	0.64509	0.51515	0.53035	0.65567	0.52903	0.6361	0.59461	0.6343	0.51796	0.62059	0.67987	
	Lkml-reply	0.77974	0.77667	0.77933	0.78037	0.77836	0.77608	0.77607	0.77799	0.77115	0.77446	0.77668	0.77557	0.75823	0.78426	0.79386	
BAL ACC	Mathoverflow	0.69648	0.69289	0.69401	0.7129	0.71073	0.68645	0.69063	0.70921	0.68306	0.69656	0.69398	0.70433	0.65557	0.71364	0.72593	
	Mit	0.58805	0.61386	0.54797	0.64553	0.7327	0.60118	0.56241	0.65409	0.61931	0.72207	0.67485	0.71172	0.61404	0.6245	0.72742	
	Radoslaw-email	0.55221	0.56869	0.5684	0.52001	0.67829	0.5663	0.58419	0.58684	0.57157	0.68737	0.53934	0.67687	0.56423	0.56696	0.69281	
	Eu-core	0.63776	0.65738	0.64173	0.41488	0.84103	0.67849	0.66311	0.70037	0.67114	0.80931	0.42721	0.74663	0.65138	0.62985	0.82411	
	Fb-forum	0.27366	0.25717	0.28058	0.39608	0.80374	0.2879	0.2879	0.71723	0.29926	0.7644	0.48846	0.74497	0.11951	0.59296	0.79563	
	CollegeMsg	0.20767	0.19409	0.20915	0.3162	0.39318	0.18013	0.22014	0.41935	0.19172	0.39817	0.33461	0.38206	0.11637	0.36668	0.4408	
	Lkml-reply	0.6461	0.65174	0.65348	0.63485	0.65215	0.65169	0.64817	0.64146	0.65657	0.63164	0.61331	0.61257	0.64063	0.64239	0.65815	
	Mathoverflow	0.52992	0.52664	0.52833	0.53741	0.51474	0.5159	0.52334	0.5385	0.51489	0.52575	0.50237	0.51241	0.46328	0.54074	0.55937	
	Mit	0.56331	0.59347	0.58374	0.67025	0.68212	0.57102	0.55833	0.6704	0.60746	0.71229	0.66834	0.71999	0.56908	0.65399	0.74276	
	Radoslaw-email	0.52667	0.58914	0.5841	0.4989	0.77613	0.59298	0.59913	0.58217	0.59832	0.79572	0.54554	0.78688	0.56770	0.55894	0.81241	
AVG PREC	Eu-core	0.65644	0.77429	0.70089	0.39275	0.87395	0.7344	0.72073	0.78587	0.74865	0.85825	0.43438	0.80312	0.67695	0.67659	0.89893	
	Fb-forum	0.2338	0.21536	0.24276	0.40347	0.83448	0.22984	0.24048	0.81675	0.23834	0.80091	0.49453	0.82533	0.20973	0.65253	0.8711	
	CollegeMsg	0.19621	0.19471	0.19169	0.23909	0.19291	0.19291	0.20959	0.46063	0.20072	0.41648	0.31251	0.4328	0.18132	0.35212	0.51279	
	Lkml-reply	0.62362	0.61391	0.59975	0.63348	0.64124	0.61258	0.59113	0.62009	0.61284	0.60591	0.59275	0.61802	0.58421	0.64213	0.67885	
	Mathoverflow	0.48461	0.47182	0.46779	0.5163	0.52288	0.46053	0.45386	0.50197	0.46247	0.47411	0.46548	0.49382	0.41888	0.52605	0.56126	

TABLE 5.6: Performance comparison and analysis of CFLP model with various individual similarity indices with Linear Discriminant Analysis (LDA)

Metric	Dataset	CN	AA	JC	PA	SP	CCLP	SI	L3	NLC	COSP	ACT	MFI	CARC	LOCALP	CFLP
AUC	Mit	0.8268	0.82563	0.7508	0.86383	0.83425	0.82802	0.75707	0.88133	0.82166	0.86587	0.80906	0.63673	0.81008	0.85404	0.91941
	Radoslaw-email	0.8566	0.85943	0.85467	0.8238	0.86816	0.86305	0.85555	0.86227	0.85993	0.74247	0.77618	0.80672	0.85565	0.85529	0.92263
	Eu-core	0.92041	0.92629	0.92195	0.75647	0.97208	0.94184	0.93016	0.94488	0.93806	0.82787	0.76907	0.85924	0.88557	0.92553	0.97406
	Fb-forum	0.55702	0.57263	0.56117	0.73531	0.79898	0.58098	0.55417	0.91001	0.59876	0.82987	0.6893	0.82912	0.53911	0.81781	0.94413
	CollegeMsg	0.54155	0.54218	0.50968	0.6275	0.54296	0.53319	0.49162	0.67248	0.52797	0.65494	0.53291	0.62624	0.50044	0.64627	0.73842
	Lkml-reply	0.76008	0.76053	0.78945	0.7825	0.76296	0.78943	0.79109	0.7796	0.79255	0.79035	0.72285	0.78774	0.75058	0.77976	0.80593
	Mathoverflow	0.6969	0.70145	0.70104	0.71988	0.68202	0.69687	0.70228	0.72726	0.69262	0.70586	0.66913	0.69228	0.65847	0.72636	0.72574
	Mit	0.53799	0.5371	0.4158	0.62971	0.47188	0.55254	0.41953	0.6747	0.53299	0.65641	0.5187	0.31194	0.56349	0.60117	0.74122
	Radoslaw-email	0.57238	0.5677	0.55409	0.51833	0.48064	0.58691	0.54784	0.61289	0.59229	0.48561	0.36279	0.47914	0.55809	0.58258	0.79747
	Eu-core	0.69802	0.72893	0.69267	0.44424	0.80637	0.77852	0.72184	0.76965	0.77437	0.69356	0.43426	0.65798	0.67771	0.71609	0.85647
Fb-forum	0.23946	0.25541	0.20252	0.42755	0.34255	0.25549	0.20065	0.8041	0.31613	0.72993	0.26257	0.68498	0.27514	0.51841	0.84533	
CollegeMsg	0.22371	0.22471	0.18597	0.32005	0.19907	0.23308	0.16188	0.47724	0.24147	0.45868	0.28428	0.40066	0.27878	0.35068	0.53196	
Lkml-reply	0.71532	0.71801	0.6783	0.68944	0.39848	0.72862	0.69461	0.70486	0.731	0.69963	0.35767	0.61381	0.71683	0.70005	0.70948	
Mathoverflow	0.62735	0.62794	0.58091	0.58111	0.3123	0.60936	0.59179	0.6068	0.60691	0.59938	0.27707	0.46612	0.62079	0.60031	0.58044	
BAL ACC	Mit	0.76514	0.76202	0.69843	0.80204	0.84234	0.75838	0.70445	0.81029	0.74676	0.83697	0.72352	0.58585	0.75203	0.78194	0.86912
	Radoslaw-email	0.76788	0.76681	0.78123	0.73274	0.7836	0.76989	0.78033	0.75897	0.75116	0.7693	0.6846	0.73092	0.70357	0.76052	0.83649
	Eu-core	0.81257	0.81941	0.83384	0.67412	0.93053	0.81678	0.85177	0.84062	0.78425	0.82794	0.65849	0.79604	0.72236	0.81151	0.92202
	Fb-forum	0.54533	0.55968	0.56113	0.66877	0.78843	0.55307	0.55545	0.84451	0.57486	0.81631	0.65423	0.78179	0.53674	0.72541	0.87071
	CollegeMsg	0.53117	0.5302	0.49791	0.60804	0.53844	0.52201	0.48908	0.66275	0.52749	0.54315	0.61905	0.50969	0.61796	0.61796	0.67017
	Lkml-reply	0.72227	0.72087	0.73151	0.6927	0.69234	0.66424	0.7639	0.71091	0.63117	0.76986	0.60444	0.73831	0.59878	0.71625	0.77845
	Mathoverflow	0.68072	0.6763	0.67866	0.65805	0.67197	0.60266	0.69352	0.66765	0.57183	0.66799	0.5677	0.6538	0.57127	0.67414	0.70522
	Mit	0.61682	0.60343	0.5158	0.67197	0.70853	0.59368	0.51114	0.69188	0.57133	0.71921	0.5244	0.36725	0.59792	0.64324	0.75717
	Radoslaw-email	0.59416	0.59109	0.59477	0.54504	0.6124	0.60426	0.58013	0.58984	0.58832	0.60397	0.44715	0.52932	0.53681	0.5804	0.73073
	Eu-core	0.67228	0.69309	0.70004	0.44445	0.81615	0.71894	0.71981	0.72763	0.68286	0.73356	0.37672	0.66485	0.58039	0.67959	0.84131
Fb-forum	0.25379	0.28218	0.28825	0.43503	0.56224	0.24965	0.2909	0.77557	0.27839	0.73563	0.38556	0.68102	0.16996	0.51958	0.80769	
CollegeMsg	0.19571	0.19457	0.26586	0.3416	0.28697	0.14112	0.24201	0.4609	0.13232	0.423	0.29501	0.37294	0.09139	0.36441	0.48144	
Lkml-reply	0.60154	0.60014	0.59373	0.54882	0.46806	0.49192	0.64258	0.58465	0.41488	0.65282	0.34233	0.55885	0.32993	0.59223	0.6583	
Mathoverflow	0.51425	0.50784	0.50026	0.47322	0.42873	0.34051	0.52817	0.4949	0.25202	0.49188	0.27491	0.42103	0.25007	0.50594	0.51468	
AVG PREC	Mit	0.5474	0.54588	0.42291	0.63542	0.5506	0.5605	0.42688	0.67883	0.54182	0.66118	0.52484	0.3186	0.57129	0.60931	0.74651
	Radoslaw-email	0.57486	0.57013	0.55614	0.52088	0.63962	0.58939	0.54997	0.61455	0.59466	0.48667	0.36424	0.48146	0.56053	0.58506	0.79827
	Eu-core	0.69973	0.73049	0.69538	0.44646	0.85283	0.77961	0.72426	0.77092	0.77519	0.69599	0.43743	0.66041	0.67616	0.71772	0.85823
	Fb-forum	0.23264	0.25103	0.19537	0.43066	0.34788	0.24749	0.19428	0.8051	0.30558	0.72767	0.25517	0.68792	0.21679	0.52203	0.84614
	CollegeMsg	0.20829	0.20854	0.1735	0.32323	0.18333	0.21926	0.16379	0.47535	0.22601	0.43415	0.17851	0.39868	0.17927	0.34897	0.49926
	Lkml-reply	0.60754	0.61137	0.58481	0.64078	0.36184	0.63292	0.6009	0.64582	0.63654	0.64621	0.31182	0.5684	0.57877	0.64118	0.66063
	Mathoverflow	0.48557	0.49309	0.44801	0.53519	0.27174	0.47083	0.45877	0.54626	0.46751	0.49813	0.23527	0.41937	0.42203	0.53933	0.5249

Eu-core, and Fb-forum datasets behind SP. For AVG PREC, metric CFLP is the best performing method across all the tested datasets.

5.3.6 Comparison of CFLP with state-of-the-art methods after optimization

Table 5.7 compares the performance of CFLP and its variations with state-of-the-art methods on five performance metrics and seven publicly available dynamic datasets. The performance metrics used here are AUC, AUPR, BAL ACC (Balance Accuracy Score), F1 SCORE, and AVG PREC (Average Precision). Here we have compared two variations of our proposed CFLP method called CFLP-Local (CFLP-L) and CFLP-Global (CFLP-G) in different machine learning models. The similarity metrics used in CFLP-L and CFLP-G are discussed in section 5.2.3. The CFLP-L and CFLP-G variations are the optimized version of CFLP. We have used the highest similarity score of various similarity indices for feature selection. We have selected the features by two different feature selection methods, F-Regression and Mutual Information Classifier. The machine learning models used here are Neural Network (NN), XGBoost (XGB), Logistic Regression (LR), Random Forest Classifier (RFC), and Linear Discriminant Analysis (LDA).

In terms of AUC, our proposed method CFLP-L-RFC and CFLP-G-RFC generate the best results on the MIT dataset. CFLP-L-LDA and CFLP-G-LDA are second-best performing methods. On Radoslaw-email and Eu-core datasets, CFLP-L-RFC and CFLP-G-RFC are the best-performing methods. CFLP-G-LDA is the method with the second-highest performance. CFLP-L-LDA is comparable with CFLP-G-LDA. CFLP-G-LDA produces the best result on Fb-forum, followed closely by CFLP-L-RFC. CFLP-G-RFC and CFLP-L-LDA produce comparable and third best results. on CollegeMsg dataset, CFLP-L-RFC and CFLP-G-RFC give the best results, whereas CFLP-G-LDA is the second-best performing algorithm. On Lkml-reply and

TABLE 5.7: Comparison of CFLP machine learning variation with state-of-the-art methods after optimization

Metric	Dataset	PROXM		N2V		WEAK		CFLP-NN		CFLP-XGB		CFLP-LR		CFLP-RFC		CFLP-LDA		
		Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	
AUC	Mit	0.89392	0.68906	0.65496	0.65503	0.65177	0.80378	0.81896	0.69567	0.69596	0.92378	0.92154	0.9067	0.91997	0.91811	0.9197	0.97003	0.94268
	Radoslaw-email	0.91898	0.73568	0.60541	0.59421	0.68277	0.80666	0.80757	0.37536	0.37944	0.93311	0.93398	0.96861	0.97003	0.93437	0.94268	0.74131	0.71238
	Eu-core	0.9552	0.94494	0.70495	0.75419	0.79433	0.90868	0.91839	0.44793	0.41665	0.97719	0.97989	0.6861	0.6861	0.93437	0.94268	0.74131	0.71238
	Fb-forum	0.79522	0.85596	0.64254	0.68104	0.72098	0.8771	0.87531	0.49323	0.49811	0.94024	0.93472	0.76058	0.76788	0.93437	0.94268	0.74131	0.71238
	CollegeMsg	0.63843	0.71109	0.57384	0.52315	0.56659	0.64332	0.63906	0.52141	0.52443	0.76058	0.76788	0.82008	0.82001	0.76058	0.76788	0.74131	0.71238
	Lkml-reply	0.92316	0.80217	0.79669	0.79908	0.80454	0.76533	0.76478	0.80462	0.80392	0.82008	0.82001	0.74637	0.74639	0.74637	0.74639	0.68744	0.71123
	Mathoverflow	0.78762	0.71539	0.73463	0.73049	0.73424	0.69724	0.69828	0.72414	0.72463	0.74637	0.74639	0.68744	0.68744	0.68744	0.68744	0.71123	0.71123
AUPR	Mit	0.37668	0.34857	0.55145	0.5301	0.55692	0.72948	0.74247	0.46591	0.46472	0.74961	0.75753	0.70924	0.74089	0.78847	0.79177	0.83356	0.83936
	Radoslaw-email	0.43212	0.41985	0.33067	0.37371	0.45008	0.76406	0.76372	0.19673	0.19642	0.82011	0.82112	0.78847	0.79177	0.83356	0.83936	0.83356	0.83936
	Eu-core	0.66827	0.7356	0.48863	0.57578	0.64035	0.84179	0.85851	0.23669	0.21637	0.87159	0.88828	0.85724	0.85699	0.85724	0.85699	0.85724	0.85699
	Fb-forum	0.58368	0.65555	0.38666	0.48611	0.51955	0.84412	0.83604	0.28527	0.28212	0.87059	0.87227	0.85724	0.85699	0.85724	0.85699	0.85724	0.85699
	CollegeMsg	0.25977	0.42017	0.29695	0.24937	0.28772	0.60177	0.58979	0.22276	0.22426	0.53946	0.53863	0.51705	0.51944	0.51705	0.51944	0.62722	0.62722
	Lkml-reply	0.72403	0.64506	0.61187	0.62054	0.63358	0.73684	0.73602	0.69786	0.69816	0.73534	0.73447	0.67933	0.62722	0.67933	0.62722	0.62722	0.62722
	Mathoverflow	0.56755	0.49082	0.54906	0.53719	0.5493	0.66547	0.66667	0.58641	0.58595	0.62976	0.63064	0.55642	0.51898	0.55642	0.51898	0.55642	0.51898
BAL ACC	Mit	0.78816	0.63489	0.65212	0.65261	0.64836	0.80378	0.81896	0.65337	0.64771	0.859	0.85705	0.84782	0.86371	0.83329	0.83747	0.89124	0.91567
	Radoslaw-email	0.82508	0.6507	0.59196	0.58755	0.65976	0.80666	0.80757	0.38728	0.3902	0.85433	0.85488	0.89124	0.91567	0.89124	0.91567	0.89124	0.91567
	Eu-core	0.93282	0.88257	0.68926	0.7408	0.77659	0.90868	0.91839	0.4258	0.40269	0.93702	0.94378	0.86199	0.87013	0.86199	0.87013	0.86199	0.87013
	Fb-forum	0.82043	0.78762	0.61863	0.65646	0.69126	0.8771	0.87531	0.45606	0.46206	0.90226	0.9022	0.66351	0.66339	0.66351	0.66339	0.66351	0.66339
	CollegeMsg	0.64007	0.63855	0.56032	0.5184	0.55262	0.64332	0.63906	0.48833	0.48844	0.67672	0.68562	0.73367	0.73857	0.73367	0.73857	0.6806	0.68686
	Lkml-reply	0.88832	0.77343	0.78094	0.78039	0.78422	0.76533	0.76478	0.7502	0.75035	0.79313	0.7929	0.6806	0.68686	0.6806	0.68686	0.6806	0.68686
	Mathoverflow	0.78381	0.69452	0.71551	0.71372	0.71399	0.69724	0.69828	0.69822	0.6978	0.72191	0.72221	0.6806	0.68686	0.6806	0.68686	0.6806	0.68686
F1 SCORE	Mit	0.23581	0.42454	0.41404	0.42263	0.39484	0.69406	0.71033	0.46681	0.44332	0.71736	0.7277	0.73096	0.75328	0.71893	0.72825	0.81118	0.83245
	Radoslaw-email	0.25144	0.42119	0.28952	0.28761	0.39128	0.72084	0.72138	0.22151	0.22318	0.69483	0.70153	0.81118	0.83245	0.81118	0.83245	0.81118	0.83245
	Eu-core	0.48031	0.69285	0.43604	0.51166	0.5767	0.82868	0.84668	0.22216	0.20686	0.79467	0.82161	0.80658	0.8142	0.80658	0.8142	0.47126	0.46755
	Fb-forum	0.47254	0.57337	0.35492	0.40466	0.45763	0.82178	0.81443	0.23089	0.2366	0.78445	0.79586	0.47126	0.46755	0.47126	0.46755	0.47126	0.46755
	CollegeMsg	0.24988	0.38145	0.29882	0.25258	0.2851	0.43644	0.42516	0.25991	0.25727	0.43536	0.44844	0.65634	0.65545	0.65634	0.65545	0.6131	0.56237
	Lkml-reply	0.56876	0.59426	0.63416	0.62633	0.64052	0.66757	0.66658	0.63758	0.6381	0.65634	0.65545	0.50659	0.4561	0.50659	0.4561	0.50659	0.4561
	Mathoverflow	0.49236	0.47727	0.54243	0.5331	0.53749	0.54898	0.55095	0.543	0.54225	0.55053	0.55143	0.50659	0.4561	0.50659	0.4561	0.50659	0.4561
AVG PREC	Mit	0.38432	0.3569	0.38221	0.37684	0.38129	0.55266	0.56762	0.4718	0.46976	0.74999	0.7578	0.71391	0.74529	0.71391	0.74529	0.78906	0.79234
	Radoslaw-email	0.43367	0.42233	0.2934	0.3084	0.36062	0.5956	0.59587	0.19875	0.19847	0.81778	0.8183	0.78906	0.79234	0.78906	0.79234	0.83538	0.8412
	Eu-core	0.63934	0.73792	0.37745	0.50817	0.52382	0.71155	0.73935	0.23905	0.21862	0.87	0.88627	0.83538	0.8412	0.83538	0.8412	0.85717	0.85717
	Fb-forum	0.53808	0.65866	0.34988	0.42924	0.44097	0.717	0.70363	0.28716	0.28347	0.86714	0.86805	0.85717	0.85717	0.85717	0.85717	0.85717	0.85717
	CollegeMsg	0.19046	0.41533	0.25821	0.22378	0.23799	0.35409	0.34432	0.22069	0.21976	0.51579	0.51497	0.5091	0.48022	0.5091	0.48022	0.5091	0.48022
	Lkml-reply	0.69856	0.59778	0.57838	0.57084	0.5976	0.54185	0.54055	0.65738	0.65714	0.67912	0.67839	0.62773	0.58766	0.62773	0.58766	0.62773	0.58766
	Mathoverflow	0.48114	0.44387	0.49389	0.45328	0.48814	0.43659	0.43807	0.5407	0.54129	0.5542	0.55494	0.4962	0.47645	0.4962	0.47645	0.4962	0.47645

Mathoverflow datasets, CFLP-G-RFC and CFLP-L-RFC produce the second-best results behind state-of-the-art method PROXM.

In terms of AUPR, our proposed method is CFLP-G-RFC, and CFLP-L-RFC produces the best results on the Mit dataset. On Radoslaw-email, Eu-core, and Fb-forum datasets, CFLP-G-RFC and CFLP-L-RFC produce the best results. On CollegeMsg, Lkml-reply, and Mathoverflow datasets, our proposed method CFLP-G-XGB and CFLP-L-XGB produce the best results while CFLP-G-RFC and CFLP-L-RFC are the second-best performing algorithms.

For BAL ACC, our proposed method CFLP-G-LDA generates the highest score on Mit dataset, and CFLP-L-RFC is the second-best performing algorithm. On Radoslaw-email, Eu-core, Fb-forum, and CollegeMsg datasets, our proposed methods CFLP-G-RFC and CFLP-L-RFC produce the best results. On Lkml-reply dataset, CFLP-G-RFC and CFLP-L-RFC produce second-best scores slightly behind state-of-the-art methods. On the Mathoverflow dataset, CFLP-G-RFC and CFLP-L-RFC produce second-best results behind the state-of-the-art.

For F1 SCORE, our proposed model CFLP-G-LDA and CFLP-L-LDA produce the best result on the Mit dataset. On the Radoslaw-email dataset, CFLP-G-LDA produces the best result while CFLP-G-XGB is the second-best performing algorithm. On the Eu-core dataset, CFLP-G-XGB produces the best result while CFLP-G-LDA is the second-best performing algorithm. On the Fb-forum dataset, CFLP-L-XGB generates the highest score. On the CollegeMsg dataset, CFLP-L-LDA produces the best result, while CFLP-G-LDA is the second-best performing algorithm. On the Lkml-reply dataset, CFLP-G-XGB and CFLP-L-XGB produce the best results. On the Mathoverflow dataset, CFLP-G-RFC produces the best result, while CFLP-G-XGB is the second-best performing algorithm.

For AVG PREC, on Mit, Radoslaw-email, Eu-core, Fb-forum and CollegeMsg datasets, our proposed methods CFLP-L-RFC and CFLP-G-RFC produce the best results.

CFLP-G-LDA and CFLP-L-LDA are the second best-performing algorithms on the above datasets. On the Lkml-reply dataset, our proposed method CFLP-L-RFC and CFLP-G-RFC produce the second-best results, slightly behind state-of-the-art methods. On the Mathoverflow dataset, CFLP-G-RFC and CFLP-L-RFC produce the best results, while CFLP-L-LR and CFLP-G-LR are the second-best performing algorithms.

5.4 Conclusion

This paper presents an enhanced feature set to address the link prediction problem in dynamic networks. We try to incorporate features that perform edge scoring individual snapshots and the whole dynamic graph through this feature set. Our proposed feature CFLP is used for estimating edge behavior throughout the entire dynamic network irrespective of the particular snapshot under consideration. We use four similarity indices for estimating edge behavior on individual snapshots: local similarity-based, global similarity-based, quasi-local similarity-based, and clustering coefficient-based similarity. We have also made feature selection on fourteen different snapshot-based features to find the best possible combination of minimum optimized features for link prediction. This combined feature set, which incorporates various categories of similarity, is tested with different machine learning models and compared with individual features and state-of-the-art algorithms to verify the improved performance of our approach. Experiments are conducted on seven real-world datasets, and our feature set provides the best performance on XGB and RFC models with five evaluation metrics.

