Chapter 4

Multifeatures Analysis Based Link Prediction in Dynamic Social Networks

This chapter focus on the second objective of this thesis, i.e. Multifeatures Analysis Based Link Prediction in Dynamic Social Networks. We give an introduction of the considered problem in section 4.1. Section 4.2 gives the preliminaries and problem statement. Section 4.3 explains the proposed framework as the solution to the defined problem. Experimental details are given in section 4.4 and their outcomes are discussed in section 4.5. Section 4.6 concludes the overall outcome of the chapter.

4.1 Introduction

Link prediction [27] is a fundamental problem in social network analysis [28] and knowledge graph completion [29]. Real social networks/knowledge graphs are dynamic in nature that evolve over time, either by adding/deleting nodes or links between nodes. Data mining and machine learning algorithms have been used to predict the future or missing links in the network with the knowledge of existing links and nodes [43, 32, 44]. This has many applications, such as friend recommendation in social networks [31], click-through prediction for target marketing [32], academic recommender systems [33, 34], electric grid network [35], finding new connections in protein-protein interaction networks [36], metabolic network reconstruction [37] and missing link completion in the knowledge graph [38]. Most of the previous attempts [44] to solve the link prediction problem consider the static network. However, almost all real-world networks are dynamic in nature, and they evolve with time either in terms of change in structure (addition or deletion of nodes or edges) or in terms of change in attributes of nodes or edges.

A large category of link prediction methods is based on some heuristics such as Common Neighbours, Jaccard coefficient, Adamic-Adar [31], Preferential Attachment [39], Katz coefficient [40], PageRank [41], SimAttri [42] and their numerous variants. However, a major limitation of these heuristics is that they can not deal with high non-linearity in networks. To tackle this, many advanced models like probabilistic matrix factorization [45, 46], network embedding based models [47, 48], graph neural network (GNN) models [49], and stochastic block models [50] have been developed. These methods are powerful but still lack the ability to analyze the evolution of networks. The typical reason behind this may be the ignorance of nodes' individual behaviour, which may be predicted by considering various factors. Recent studies indicate that the network structure evolution highly depends on the dynamics of the structure as well as the attributes of the nodes [51, 52, 53, 24]. Incorporating the node attributes for link prediction proves to be helpful in achieving better performance in link prediction, especially for sparse graphs. In evolving networks, as the structure of the network changes with time, the respective attributes of the nodes also change with time [54, 55, 56]; few common examples include modification of posts/comments/reviews, updating educational qualification, job organization, political party, relationship status, and age [57, 58]. Including attributes information with structural information improves the accuracy of link prediction in dynamic networks; however, there is still a lot of scopes to improve it further. Many other factors, like location-based information of nodes and popularity of nodes can also play a major role in tracing the evolution pattern of dynamic attributed networks. Studies [172, 173] shows that the change in geographical location (mobility factor) affects the evolution of social networks. Some paper [100, 101, 102] use location-based analysis to find the social communities in the networks; we use it here as one factor for link prediction. Another important factor we consider for link prediction is the popularity of nodes [103]; as we see in our society, most people want to connect with the popular faces of society.

In this chapter of the thesis, we added mobility, popularity, and similar interests of the nodes as additional factors along with the structure and attributes to predict the network evolution pattern and the upcoming links in the evolving social networks. Here, we use an improved LDA topic model [174, 175] and Hidden Naive Bayesian algorithm [176] to propose *Popularity, interests, location used hidden Naive Bayesian-based model (PILHNB) model* for link prediction in dynamic social networks. See Appendix A for our research paper supporting this work.

4.2 Problem Description

4.2.1 Data Model

We consider n_t number of users as a set of vertices denoted as $V^t = \{v_1, \dots, v_{n_t}\}$ at timestamp t and the set of edges among these users as $E^t = e_{ij}$, where each edge e_{ij} indicates a link (e.g., friendship) between v_i and v_j at timestamp t. Each node v_i has a d-dimensional set of attributes $a^i \in \mathbb{R}^d$ at each timestamp. The node's geographical location information is given by $L^t = [l_1, l_2, \dots, l_{n_t}]^t$, where $l_i \in \mathbb{R}^M$, denotes the checked-in information of i^{th} user at M different locations. The common interest (interest similarity) vector for user v_i is given by $I^t_{v_i} = [I_1, I_2, \dots, I_j, \dots, I_{n_t}]$, where I_j denotes the number of common interests among user v_i and v_j. The interaction frequency vector for user v_i is given by $A^t_{v_i} = [A_1, A_2, \dots, A_j, \dots, A_{n_t}]$, where A_j gives the frequency of interaction between i^{th} and j^{th} user. The popularity vector is given by $\mathscr{P}^t = [\mathscr{P}_{v_1}, \mathscr{P}_{v_2}, \dots, \mathscr{P}_{v_j}, \dots, \mathscr{P}_{v_{n_t}}]$, where $\mathscr{P}_{v_j}$ gives the popularity of j^{th} node at timestamp t. The attribute similarity vector is given by $\mathscr{S}^t_{v_i} = [\mathscr{S}_{v_1}, \mathscr{S}_{v_2}, \dots, \mathscr{S}_{v_{n_t}}]$, where $\mathscr{S}^t_{v_i}$ represents the similarity vector of i^{th} node with all other nodes in the network.

For evaluating the popularity of a user, we consider the assumption that, if a user became popular in social networks, then in the recent past, many users have been added as a friend/follower to that user. To compute the popularity of a user, we can divide the friends/followers added to him/her into the fresh set and the old set. If the degree of node v_i at timestamp t is denoted as $d_{v_i}(t)$. The number of new edges added with node v_i in the next t' time span is given by equation 4.1.

$$\Delta d_{v_i}(t, t') = d_{v_i}(t + t') - d_{v_i}(t). \tag{4.1}$$

For a dataset spans starting from timestamp t_x to t_z, we divide its edges into the fresh set and the old set according to a boundary $t_y \in [t_x, t_z]$. If an edge was constructed in $[t_x, t_y)$ it belongs to the old set; otherwise, the fresh set.

Popularity: The popularity of a node v_i is defined as the fraction of freshly added edges to the overall edges connected to it. Mathematically it can be represented by the following equation:

$$\mathscr{P}_{v_i} = \frac{\Delta d_{v_i}(t_y, t_z - t_y)}{\Delta d_{v_i}(t_x, t_z - t_x)} = \frac{d_{v_i,fresher}}{d_{v_i,all}}, \tag{4.2}$$

where $d_{v_i,all}$ and $d_{v_i,fresher}$ denotes the overall degree and fresher degree of the node v_i, respectively. The value of popularity $\mathscr{P}_{v_i}$ lies in $[0, 1]$, and a higher value of $\mathscr{P}_{v_i}$ means higher popularity of node v_i. The popularity vector $\mathscr{P}^t = [\mathscr{P}_{v_1}, \mathscr{P}_{v_2}, \ldots, \mathscr{P}_{v_{n_t}}]$ gives the popularity of nodes at timestamp t. The total number of users at time t is denoted by n_t.

The user behaviour pattern distribution for link prediction is denoted as $\Omega^* = [\theta_1^*, \theta_2^*, \ldots, \theta_{n_t}^*]$ and it will be mined through our proposed method, where θ_x is the behaviour pattern distribution of user v_x. The links which will be predicted using our method are represented as E^*, where $E^* \subseteq (V^t \times V^t) \backslash E_t$. The problem of link prediction on dynamic social networks can be formally defined as follows:

Problem Definition: *Given,* $G = \{G^1, G^2, G^3, \ldots, G^t\}$ *as a series of snapshots of* *location-aware dynamic attributed network with evolving edges, change in geographical location and node's attributes during timestamps $T = \{1, 2, 3, \ldots, t\}$. The link prediction problem's objective is to use the key factors to capture the evolution pattern of the network and predict the future links that may appear in G^{t+1} as new links E^*. We can make our model learn users link behaviour pattern distribution Ω^* till the present snapshot and predict the new links E^* in the upcoming snapshot G^{t+1}. Formally, the problem definition can be represented as $G^{t+1} \Rightarrow f(G, I, A, L, \mathscr{P}, \mathscr{S}) \rightarrow \Omega^*, E^*$.*

4.3 Proposed Framework

To solve the problem of dynamic link prediction defined in the previous section, we propose a Hidden Naive Bayesian-based link prediction model employing users' relationships and behaviour patterns derived from attributes, geographical location, popularity, and interests of the users. The proposed model has three submodules: controlling elements quantification, user behaviour pattern modelling, and link prediction module, as shown in Figure 5.4. In the first module, methods to represent and quantify the various controlling elements are proposed. In the second module, the user behaviour pattern learning model for link prediction using topic modelling with modified LDA and HNB is constructed. In the third module, the trained model is used to determine the user's link behaviour pattern distribution and perform a link prediction task.

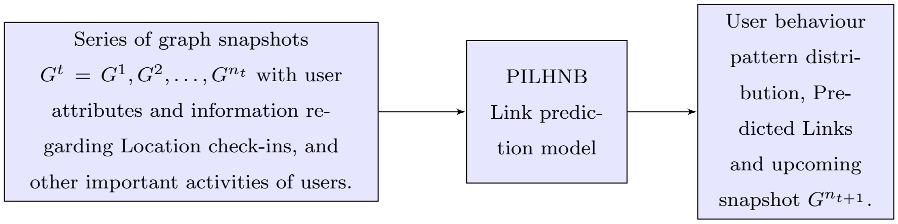

Series of graph snapshots $G^t = G^1, G^2, \ldots, G^{n_t}$ with user attributes and information regarding Location check-ins, and other important activities of users.
PILHNB Link prediction model
User behaviour pattern distribution, Predicted Links and upcoming snapshot $G^{n_{t+1}}$.

FIGURE 4.1: PILHNB Model

4.3.1 Controlling Elements Quantification

In the first module, for link prediction in dynamic social networks, we identify the dependency of link formation on network evolution pattern, which is governed by some controlling factors. The controlling factors can be categorized into behavioural elements and structural elements.

1. ***Behavioural Elements***: To predict the upcoming links, we extract the users' attributes and activities performed by them to analyze the user behaviour and their evolution pattern. The considered behavioural elements are defined and represented as follows:

   (a) ***Common Interests***: Link formation between a pair of users is also affected by their common interests [177]. If two users have many common interests, then they have fair chances to become friends on social networks. The interest may be in education, politics, sports, film, media, research, fashion, and technology. This factor is calculated by analyzing the messages posted or liked by the user in the recent past. The common interest (interest similarity) vector for user v_i is defined as follows:

   $$I^t_{v_i} = [I_1, I_2, \ldots, I_j, \ldots, I_{n_t}], \tag{4.3}$$

   where n_t gives the total number of users at time t and I_j denotes the number of common interests (similarity in interest) among user v_i and v_j.

   (b) ***Interaction Frequency behaviour***: Link prediction is affected by the users' activeness on social networks. Here, we measure the user activeness by analyzing the frequency of interactions [178] with other users. The interaction may be in the form of message posting, comments, like/dislike. The interaction frequency vector for user v_i is defined as follows:

   $$A^t_{v_i} = [A_1, A_2, \ldots, A_j, \ldots, A_{n_t}], \tag{4.4}$$

where, A_j gives the frequency of interaction between user v_i and user v_j. n_t gives the total number of users at time t.

(c) ***Location Check-ins Behaviour***: We consider if two people share a common geographical location repeatedly; then, they may become friends on the social network platform. The geographical location can be a gym, institute, club, workplace, seminar, workshop, conference, tourist place. The node's geographical location information at time t is given by location vector:

$$L^t = [l_1, l_2, \ldots, l_j, \ldots, l_{n_t}], \tag{4.5}$$

where $l_j \in \mathbb{R}^M$, denotes the checked-in information of user v_j at M different locations and n_t gives the total number of users at time t.

(d) ***Popularity***: The popularity of a node v_j is defined as the fraction of freshly added edges to the overall edges connected to it. Mathematically it can be represented by equation 4.2. The popularity vector is given as:

$$\mathscr{P}^t = [\mathscr{P}_{v_1}, \mathscr{P}_{v_2}, \ldots, \mathscr{P}_{v_j}, \ldots, \mathscr{P}_{v_{n_t}}], \tag{4.6}$$

where $\mathscr{P}_{v_j}$ gives the popularity of node v_j at timestamp t and n_t gives the total number of users at time t. From the real-life scenario, we consider a hypothesis which says that mostly the people want to make friendship with the popular faces of their society, so the user with a high value of popularity (considering recent snapshots for evaluation) have great chance to add friends in the social networks.

(e) ***Attribute similarity***: The extent of similarity of attributes between two users increases the chance of being friends on social networks if they are a few hops away in the networks. The attributes of users may include age, education, workplace, school, current city, and common groups. The attribute similarity vector is defined as follows:

$$\mathscr{S}_{v_i}^t = [\mathscr{S}_{v_1}, \mathscr{S}_{v_2}, \ldots, \mathscr{S}_{v_{n_t}}], \tag{4.7}$$

where $\mathscr{S}_{v_i}^t$ represents the similarity vector of node v_i with all other nodes in the network, and n_t, gives the total number of nodes in the network.

2. ***Structural Elements***:

   (a) ***Common neighbours***: In a real-life scenario, friendship/link formation between unknown persons in social networks also depends on the common neighbours/friends. Here, we consider common neighbours as one of the important controlling elements used for link prediction. The common neighbour information of each user is stored in a vector represented as:

   $$C_{v_i,v_j} = c_{ij} = [c_1, c_2, \ldots, c_{N_{ij}}], \tag{4.8}$$

   where $c_{ij} \in V^t$ represents the common neighbours of user v_i and v_j. N_{ij} denotes the number of common neighbours for the pair of the user.

   (b) ***Individual Dependency***: In social networks, common neighbours are not totally independent; there exists certain dependence among them. Individual dependence can be defined as the individual dependence of each common neighbour of a pair of nodes. To represent individual dependence, we can use conditional mutual information defined as follows:

   $$\alpha_{xy} = \frac{J_T(v_x, v_y|k)}{\sum_{y=1, y \neq x}^{N_{ij}} J_T(v_x, v_y|k)}, \tag{4.9}$$

   where, α_{xy} is the value of individual dependence between common neighbour v_x and v_y. $J_T(v_x, v_y|k)$ gives the conditional mutual information among them in case of link existence.

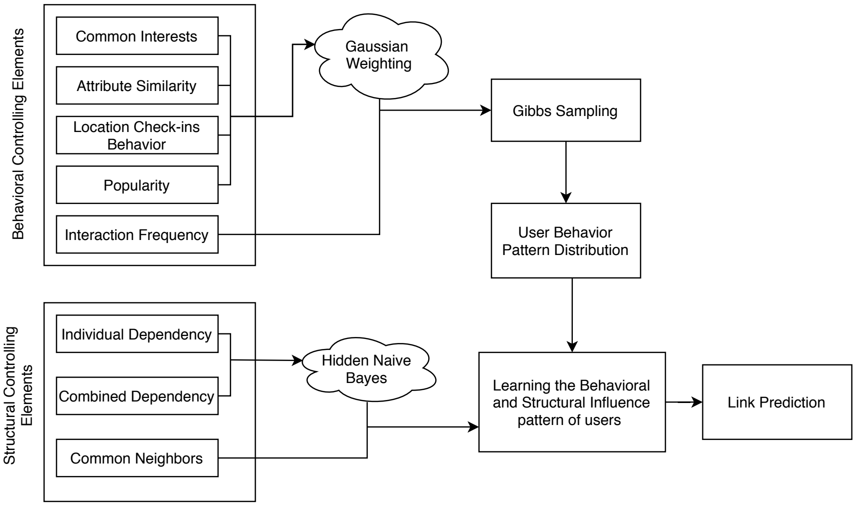

Behavioral Controlling Elements
Common Interests
Attribute Similarity
Location Check-ins Behavior
Popularity
Interaction Frequency
Gaussian Weighting
Gibbs Sampling
User Behavior Pattern Distribution
Structural Controlling Elements
Individual Dependency
Combined Dependency
Common Neighbors
Hidden Naive Bayes
Learning the Behavioral and Structural Influence pattern of users
Link Prediction

FIGURE 4.2: PILHNB Model Details.

(c) ***Combined Dependency***: The collective influence of all common neighbours of a pair of nodes is defined as a combined dependency. It can be represented as conditional mutual information weighted summation given as follows:

$$\beta_{xyz} = \frac{J_T(v_x,[v_y,v_z]|k)}{\sum_{y=1,y\neq x}^{N_{ij}}\sum_{z=1,z\neq x}^{N_{ij}} J_T(v_x,[v_y,v_z]|k)}, \tag{4.10}$$

where β_{xyz} is the value of combined dependence between common neighbour v_x and pair of common neighbour $[v_y, v_z]$. In the case of link existence, the conditional mutual information among them is given by $J_T(v_x,[v_y,v_z]|k)$.

4.3.2 Learning User Behaviour Pattern Distribution

We consider two basic controlling elements: behavioural and structural elements, which influence the link formation pattern in dynamic social networks. In this module, we describe the learning of user behaviour patterns and provide the detail of the PILHNB model proposed to solve the given link prediction problem. Figure 5.5 shows the block diagram for our proposed framework.

To track user behaviour patterns, we extract the behaviour controlling elements like common interests, interaction frequency, location check-ins, popularity, and attribute similarity from the given users' information. Then, we use the popularity vector, common location sharing pattern of users, and frequency of interaction vector, according to the definitions given in section 4.3.1. To extract the relevant information regarding the users' common interests and the common attributes of the users, we apply the technique of text mining to get user behaviour based on them.

Common interest and attribute similarity between a pair of users can be obtained by applying an improved topic modelling technique. Here, we use LDA topic modelling improved with the Gaussian weighting method for text mining and use it as a tool for user behaviour modelling. In this module, the user is represented as a document, and user behaviour is represented as vocabulary. By assuming interest as a topic, we can mine the user behaviour pattern distribution.

Assume that the set of users is $V^t = \{v_1, v_2, \ldots, v_{n_t}\}$. Each user's behaviour can be inferred as the component of its behavioural controlling element vectors, which can also be represented as the superposition of its component vectors. The superposition is defined as follows:

$$V_x = B_{v_x} = I_{v_x} \oplus A_{v_x} \oplus L_{v_x} \oplus \mathscr{P}_{v_x} \oplus \mathscr{S}_{v_x}, \tag{4.11}$$

where "$\oplus$" represents the superposition operator. Each user v_x is referred to as a behavioural user v_{x,n_t}. Each behavioural user obeys a multinomial distribution of interest $z_{x.n_t}$, and each interest z_{x,n_t} follows a multinomial distribution of user v_{x,n_t}.

Due to the power-law characteristics of user behaviour, the user behaviour pattern distribution will be tending toward high-frequency users. To remove the noise, the basic LDA is improved by the Gaussian weighting method, which provides weight to each user behaviour. Giving the parameters k_1, k_2, k_3, k_4 and k_5 as the number of interests, number of interactions, number of shared locations, number of popular nodes within 2 hops distance, and the number of common attributes, respectively. The joint distribution of all the observed and hidden variables can be computed as follows:

$$\begin{aligned} R(\Omega|I, A, L, \mathscr{P}, S) = \prod_{x=1}^{k_1} R(\theta^1|I) \cdot \prod_{x=1}^{k_2} R(\theta^2|A) \cdot \prod_{x=1}^{k_3} R(\theta^3|L) \\ \cdot \prod_{x=1}^{k_4} R(\theta^4|\mathscr{P}) \cdot \prod_{x=1}^{k_4} R(\theta^5|\mathscr{S}), \end{aligned} \tag{4.12}$$

here, the goal of user behaviour modelling is to get the behaviour distributions θ^1, θ^2, θ^3, θ^4, and θ^5 for each user. Owing to the coupling of these distributions, we cannot compute them directly, so the Gibbs sampling will be applied to extract the Ω and when the sampling converges, the convergent distribution Ω^* can be obtained.

4.3.3 Link Prediction

Let $K = \{k, \bar{k}\}$ be the set of classified edges, where k represent the existence of links, and $\bar{k}$ represents the absence of links. In this chapter, we have taken two types of dependence: individual dependency and combined dependency. Here, we use the controlling element α to represent the summation of individual dependency and β to represent the summation of combined dependency. For the variables α and β, the joint probability distribution is defined as follows:

$$\begin{aligned} P(c_{ij}, k) = P(k) \prod_{y=1}^{N_{ij}} P(c_y|\alpha_y, k) P(c_y|\beta_y, k), \\ P(c_{ij}, \bar{k}) = P(\bar{k}) \prod_{y=1}^{N_{ij}} P(c_y|\alpha_y, \bar{k}) P(c_y|\beta_y, \bar{k}). \end{aligned} \tag{4.13}$$

Proceeding with the common neighbour as an important condition, the probability of link establishment can be evaluated as follows:

$$\begin{aligned} P(k|c_{ij}) = \frac{P(k)}{P(c_{ij})} \prod_{y=1}^{N_{ij}} P(c_y|\alpha_y, k) P(c_y|\beta_y, k), \\ P(\bar{k}|c_{ij}) = \frac{P(\bar{k})}{P(c_{ij})} \prod_{y=1}^{N_{ij}} P(c_y|\alpha_y, \bar{k}) P(c_y|\beta_y, \bar{k}). \end{aligned} \tag{4.14}$$

We take the probability of link formation as the ratio of conditional probabilities from equation 4.13 and 4.14, and can be represented as follows:

$$\begin{aligned} P_L &= \log_2 \frac{P(k|c_{ij})}{P(\bar{k}|c_{ij})}, \\ &= \log_2 \frac{P(k)}{P(\bar{k})} \prod_{y=1}^{N_{ij}} \frac{P(c_y|\alpha_y, k)P(c_y|\beta_y, k)}{P(c_y|\alpha_y, \bar{k})P(c_y|\beta_y, \bar{k})}. \end{aligned} \tag{4.15}$$

Here, $P(k)$ and $P(\bar{k})$ denotes the probability of link existence and link absence, and it can be evaluated as follows:

$$\begin{aligned} P(k) &= \frac{2\mathscr{L}^t}{n_t(n_t-1)}, \\ P(\bar{k}) &= 1 - \frac{2\mathscr{L}^t}{n_t(n_t-1)}, \end{aligned} \tag{4.16}$$

where $\mathscr{L}^t$ represents the total number of links present in the networks, $P(c_y|\alpha_y, k)$, $P(c_y|\alpha_y, \bar{k})$, $P(c_y|\beta_y, k)$ and $P(c_y|\beta_y, \bar{k})$ denotes the corresponding dependencies in case of link presence and link absence.

The probability $P(c_y|\alpha_y, k)$ and $P(c_y|\beta_y, k)$ can be evaluated as follows:

$$\begin{aligned} P(c_y|\alpha_y, k) &= \sum_{m=1, m\neq y}^{N_{ij}} \alpha_{ym} \times P(c_y|c_m, k), \\ P(c_y|\beta_y, k) &= \sum_{m=1, m\neq y}^{N_{ij}} \sum_{n=1, n\neq m, n\neq y}^{N_{ij}} \beta_{ymn} \times P(c_y|[c_m, c_n], k). \end{aligned} \tag{4.17}$$

In equation 4.17, $P(c_y|c_m, k)$ and $P(c_y|[c_m, c_n], k)$ denotes the additional factor added by user c_m or pair of user $[c_m, c_n]$, and they can be defined as the reciprocal of node degree as:

$$\begin{aligned} P(c_y|c_m, k) &= \frac{1}{d_{c_m}}, \\ P(c_y|[c_m, c_n], k) &= \frac{1}{d_{c_m} d_{c_n}}, \end{aligned} \tag{4.18}$$

where d_{c_m} and d_{c_n} denotes the degree of common neighbour nodes c_m and c_n.

The calculation of conditional mutual information for individual dependence is as follows:

$$\begin{aligned} J_T(c_y, c_m|k) &= P(c_y, c_m|k) \log_2 \frac{P(c_y, c_m|k)}{P(c_y|k)P(c_m|k)}, \\ &= \frac{P(c_y, c_m, k)}{P(k)} \log_2 \frac{P(c_y, c_m|k)}{P(c_y|k)P(c_m|k)P(k)}. \end{aligned} \tag{4.19}$$

Here, the conditional probability of common neighbour c_y and c_m is given by $P(c_y|k)$ and $P(c_m|k)$, respectively.

The conditional probability $P(c_y|k)$ and $P(c_m|k)$ can be computed as:

$$\begin{aligned} P(c_y|k) &= \frac{2d_{c_y}}{n^t(n^t-1)}, \\ P(c_m|k) &= \frac{2d_{c_m}}{n^t(n^t-1)}. \end{aligned} \tag{4.20}$$

The similarity of common neighbour c_y and c_m is given by $P(c_y, c_m, k)$ and it can be calculated with cosine similarity, which depends on user behaviour pattern distribution as given by the following equation:

$$P(c_y, c_m, k) = cos(c_y, c_m) = \frac{\sum_{r=1}^{W} \theta_{yr} \times \theta_{mr}}{\sqrt{(\sum_{r=1}^{W} \theta_{yr}^2)(\sum_{r=1}^{W} \theta_{mr}^2)}}. \tag{4.21}$$

In case of cmbined dependence, the conditional mutual information can be calculated as follows:

$$\begin{aligned} J_T(c_y, [c_m, c_n]|k) &= P(c_y, [c_m, c_n]|k) \log_2 \frac{P(c_y, [c_m, c_n]|k)}{P(c_y|k)P([c_m, c_n]|k)}, \\ &= \frac{P(c_y, [c_m, c_n], k)}{P(k)} \log_2 \frac{P(c_y, [c_m, c_n]|k)}{P(c_y|k)P([c_m, c_n]|k)P(k)}. \end{aligned} \tag{4.22}$$

Here, the conditional probability for the pair of common neighbour $[c_m, c_n]$ is given by $P([c_m, c_n]|k)$ and it can be calculated as:

$$P([c_m, c_n]|k) = \frac{2(d_{c_m} + d_{c_n} - \Delta_{mn})}{n^t(n^t-1)}, \tag{4.23}$$

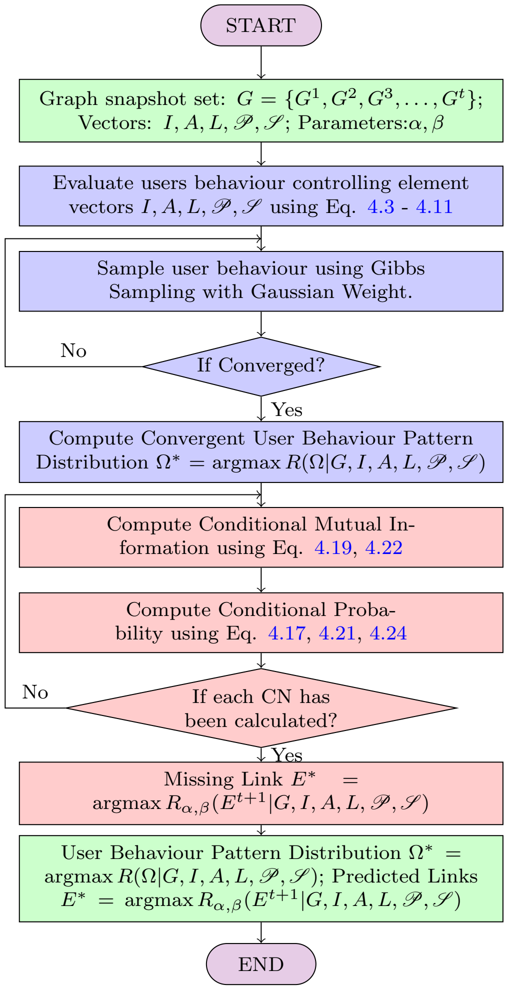

START
Graph snapshot set: $G = \{G^1, G^2, G^3, \ldots, G^t\}$;
Vectors: $I, A, L, \mathscr{P}, \mathscr{S}$; Parameters:$\alpha, \beta$
Evaluate users behaviour controlling element
vectors $I, A, L, \mathscr{P}, \mathscr{S}$ using Eq. 4.3 - 4.11
Sample user behaviour using Gibbs
Sampling with Gaussian Weight.
No
If Converged?
Yes
Compute Convergent User Behaviour Pattern
Distribution $\Omega^* = \text{argmax}\, R(\Omega|G, I, A, L, \mathscr{P}, \mathscr{S})$
Compute Conditional Mutual In-
formation using Eq. 4.19, 4.22
Compute Conditional Proba-
bility using Eq. 4.17, 4.21, 4.24
No
If each CN has
been calculated?
Yes
Missing Link $E^* =$
$\text{argmax}\, R_{\alpha,\beta}(E^{t+1}|G, I, A, L, \mathscr{P}, \mathscr{S})$
User Behaviour Pattern Distribution $\Omega^* =$
$\text{argmax}\, R(\Omega|G, I, A, L, \mathscr{P}, \mathscr{S})$; Predicted Links
$E^* = \text{argmax}\, R_{\alpha,\beta}(E^{t+1}|G, I, A, L, \mathscr{P}, \mathscr{S})$
END

FIGURE 4.3: Flowchart Showing Steps of PILHNB Model

where, Δ_{mn} is the presence ($\Delta_{mn} = 1$) and absence ($\Delta_{mn} = 0$) of links between pair of common neighbour $[c_m, c_n]$.

The probability $P(c_y, [c_m, c_n], k)$ can be represented as the similarity of common neighbour c_y and the pair of users $[c_m, c_n]$. Based on user behaviour pattern distribution and cosine similarity, the probability $P(c_y, [c_m, c_n], k)$ is defined as:

$$\begin{aligned} P(c_y, [c_m, c_n], k) &= cos(c_y, [c_m, c_n]), \\ &= \frac{\sum_{r=1}^{W} \theta_{yr} \times (\theta_{mr} + \theta_{nr})}{\sqrt{(\sum_{r=1}^{W} \theta_{yr}^2)(\sum_{r=1}^{W} (\theta_{mr} + \theta_{nr})^2)}}. \end{aligned} \tag{4.24}$$

The probabilities $P(c_y|\alpha_y, \bar{k})$ and $P(c_y|\beta_y, \bar{k})$ can also be calculated using a similar method, as mentioned above.

The computational complexity of conditional mutual information can be reduced by using the selection rule given as follows:

$$J_T(c_y, [c_m, c_n]|k) > \max\{J_T(c_y, c_m|k), J_T(c_y, c_n|k)\}. \tag{4.25}$$

If the effect of the influence factor β is larger than the effect of influence factor α, then we use the joint influence of α and β. Otherwise, we use the influence of factor α as:

$$Q_L = \begin{cases} \log_2 \frac{P(k)}{P(\bar{k})} \prod_{y=1}^{N_{ij}} \frac{P(c_y|\alpha_y,k)P(c_y|\beta_y,k)}{P(c_y|\alpha_y,\bar{k})P(c_y|\beta_y,\bar{k})}, & \text{if } J_{ymn} > \max\{J_{ym}, J_{yn}\}, \\ \log_2 \frac{P(k)}{P(\bar{k})} \prod_{y=1}^{N_{ij}} \frac{P(c_y|\alpha_y,k)}{P(c_y|\alpha_y,\bar{k})}, & \text{Otherwise.} \end{cases} \tag{4.26}$$

Here, Q_L gives the probability of new link formation between pair of nodes. Now, the link prediction task can be performed using this probability. Here, we use a specific threshold value ρ for link prediction. If the value of Q_L is greater than the threshold value, the link will form; otherwise, the link will not form. For each missing link e^*, we can define the rule of link prediction as follows:

$$e^* = \begin{cases} 1, & \text{if } Q_L \geq \rho, \\ 0, & \text{Otherwise.} \end{cases} \tag{4.27}$$

4.3.4 PILHNB Learning Algorithm

We mine user behaviour pattern distribution by utilizing user behaviour controlling elements and using it for link prediction. Here, the steps used for mining user behaviour pattern distribution come under the training process, and the task of link prediction comes under the testing process. Figure 4.11 shows the flowchart for the overall model, where blue blocks represent the training process steps, red blocks represent the steps involved in the testing process, and green blocks represent the input and output of the proposed

Algorithm 5 The PILHNB Algorithm

```
Input: Graph snapshot set: $G = \{G^1, G^2, G^3, \ldots, G^t\}$; Vectors: $I, A, L, \mathscr{P}, \mathscr{S}$;
    Parameters:$\alpha, \beta$; No. of nodes in $G^t$: $n_t$
Output: User behaviour Pattern Distribution $\Omega^* = \operatorname{argmax} R(\Omega|G, I, A, L, \mathscr{P}, \mathscr{S})$; Missing
    Link $E^* = \operatorname{argmax} R_{\alpha,\beta}(E^{t+1}|G, I, A, L, \mathscr{P}, \mathscr{S})$; Predicted upcoming Graph snapshot $G^{t+1}$;
//initialization
Get Graph Snapshot Set $G = \{G^1, G^2, G^3, \ldots, G^t\}$;
Compute User Behaviour controlling element vectors $I, A, L, \mathscr{P}, \mathscr{S}$ for each snapshot
    by using Eq. 4.3 - 4.11;
//model training
do
    for User $k \leftarrow 1$ to $n_t$ do
        Sample User Interest $z_i$ using Gibbs Sampling;
    end for
while Converged;
Obtain Convergent User Behaviour Pattern distribution $\Omega^* =$
    $\operatorname{argmax} R(\Omega|G, I, A, L, \mathscr{P}, \mathscr{S})$;
//model testing(link prediction);
for each user pair $(v_i, v_j)$ of $V$ do
    for common neighbour $c_x \leftarrow 1$ to $N_{ij}$ do
        Evaluate conditional mutual information using-
          Eq. 4.19, 4.22;
        Evaluate conditional probability using-
          Eq. 4.17, 4.21, 4.24;
    end for
    Evaluate link formation probability $Q_i$ by Eq. 4.26;
end for
Predicted links $E^* = \operatorname{argmax} R_{\alpha,\beta}(E^{t+1}|G, I, A, L, \mathscr{P}, \mathscr{S})$;
```

model. Algorithm 5 gives the steps involved in the proposed PILHNB model for link prediction.

In the proposed algorithm, the user behaviour controlling vectors $I, A, L, \mathscr{P}$, and $\mathscr{S}$ are used to extract the influencing factors responsible for the prediction of links. Further, the Gaussian weighting improved LDA is used to extract the user behaviour pattern distribution. So, the convergent criterion here is the user behaviour pattern distribution i.e., Ω^*. The Ω^* gets converged when no further improvement in the performance of the model is possible after the training and testing process. Combining the user behaviour pattern distribution and HNB-based common neighbour contribution algorithm, the link prediction task is performed by the PILHNB model.

4.4 Experiments

4.4.1 Datasets

We used six real-world network datasets for the performance evaluation of our proposed model. These datasets are from online social networks and coauthor networks. The considered datasets are: Facebook, Epinions, Brightkite, DBLP, Gowalla, and Twitter. The description of these network datasets are given in section 2.5.2.

4.4.2 Baseline Methods

We compare our proposed model with ten state-of-the-art methods using their published codes or our implementation. The considered baseline methods are introduced in section 2.6.2. Four of these methods used only network structure for link prediction, and the rest of other methods use structure and attribute both for predicting the interactions.

4.4.3 Evaluation Metrics

We have used four evaluation metrics to compare the performance of the link prediction of our proposed model with other methods. The four considered metrics are precision, recall, F1-Measure [148] and the Area Under Receiver Operating Characteristics Curve (AUROC) [149]. The formal definitions of these metrics are given in section 2.4.1. Better prediction results have greater precision, recall, F1-measure, and AUROC values.

4.4.4 Experimental Settings

We have formulated the link prediction problem as a supervised binary classification problem. If the pair of user links exist ($k = 1$), it forms a positive sample, and if it does not exist ($k = 0$), it forms a negative sample. Supervised learning methods can appropriately handle the class imbalance problem [179] of datasets (e.g., online social networks). We have divided each dataset into a series of snapshots $G = \{G^1, G^2, G^3, \ldots, G^t\}$. The experiments are performed on each snapshot dataset divided from the original dataset into the proportion of 90% training set and 10% testing set by using the method of hold-out [180]. The model is trained with a training set and is used to predict the links in the test set. Five sets of probe links (i.e., percentage of removed links=$10, 20, 30, 40, 50$) are used to evaluate each considered performance metric. For training purposes, we removed the probe links from a snapshot of the graph and used it to train the model and then predict the probe links in the testing phase. Similarly, we perform this for each fraction of the removed links on each dataset. Finally, the trained model up to time t is used to predict the links of the upcoming snapshot of the graph and tested with the actual snapshot graph at time $t + 1$. We use their standard parameter settings for all the baseline methods to implement them on our considered datasets.

To evaluate the common interest vector, we perform preprocessing of text data available as messages/comments of users. To improve the quality of the text, we processed the raw content by applying the following normalization steps: (a) removing non-Latin characters and stop words; (b) removing words with document frequency less than 10; (c) filtering out messages with length less than 3; (d) removing duplicate messages. For evaluating location information, we consider only M key locations based on the frequency of visits of the networks' users.

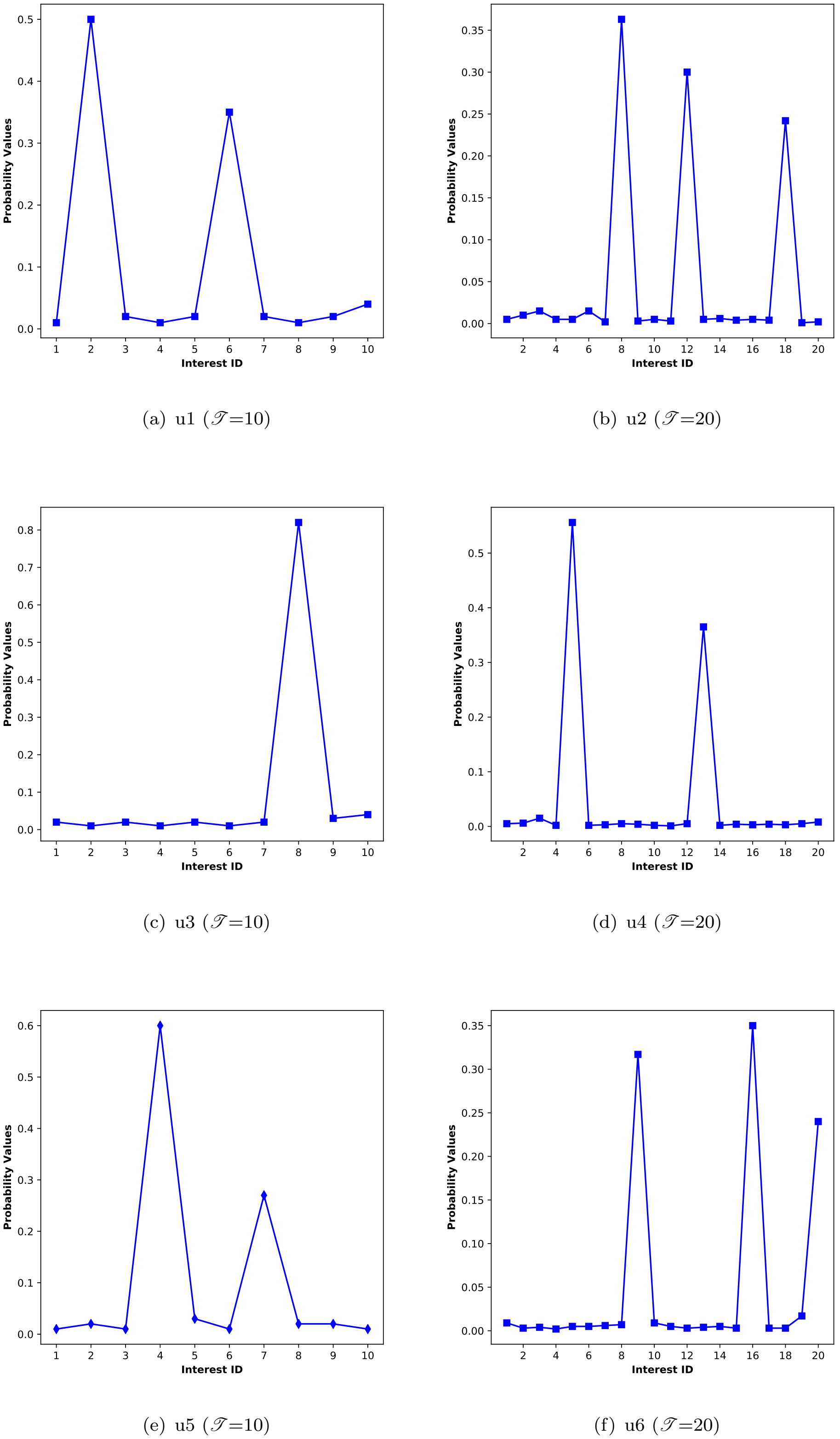
The latent interest distribution gives the interest vector of each user. This vector quantifies and evaluates the interest distribution of each user for $\mathscr{T}$ number of topics. Examples of topics can be sports, movies, politics, fashion, study, travelling, and so on. Each topic can be denoted with an interest ID such as $1, 2, 3, \ldots, \mathscr{T}$.

4.5 Results and Discussions

In this section, the results of the experiments performed are presented. Firstly, the result of user latent interest distribution is analyzed. We select the representative users from Facebook, DBLP, and Twitter datasets to show their latent interest distribution. In Fig. 4.4 (a)-(f), the graphs show the interest ID (latent interest number $\mathscr{T}$=10, and $\mathscr{T}$=20) on the x-axis and the probability of latent interest on the y-axis. As shown in Fig. 4.4 (a), (c), and (e), when $\mathscr{T}$=10, user u1 from the Facebook dataset has interest mainly concentrated in Interest ID=2, 6; for user u3 from DBLP has an interest in Interest ID=8, and user u5 from the Twitter dataset has an interest in Interest ID=4, 7. We can observe that users u1, u4, and u5 have some prominent interests, and user u3 has a concentrated interest. Here, the user with many interests has prominent interests and the user with a few interests has concentrated interests. Similarly, latent interest distribution for users u2 and u6 has a wide range of interests. We can find that each user's latent interest preferences are different and because of their differences in latent interests, the impact of this factor will affect the link prediction task.

The results are shown in Fig.4.5 verify that the latent interest has an effect on the link prediction task, and it acts as an important factor for link prediction. The x-axis shows the latent interest number, and the y-axis shows the values of Precision, Recall, F1-Measure and AUROC. It shows that the values of evaluation metrics first increases and then decreases for all the datasets. We observe that the peak of the evaluation metrics reaches when $\mathscr{T} = 15$ in all the considered datasets. So the value of $\mathscr{T}$ in this model should select the small value preferably, in the range of 10-15. A too-large value of $\mathscr{T}$ may make the model more sensitive to noise information. A too-small value of $\mathscr{T}$ may overestimate user interest and increase the estimation error.

Further, we experimented with checking the effect of popularity on link prediction tasks. The popularity of nodes is evaluated by considering the window of size ten snapshots and by varying the value of t_y as $\{1, 2, 3, 4\}$ for fresh links and links formed during the last ten snapshots as all links in equation 4.6 and keeping all other variables fixed. Figure 4.6



(a) u1 ($\mathscr{T}$=10)

(b) u2 ($\mathscr{T}$=20)

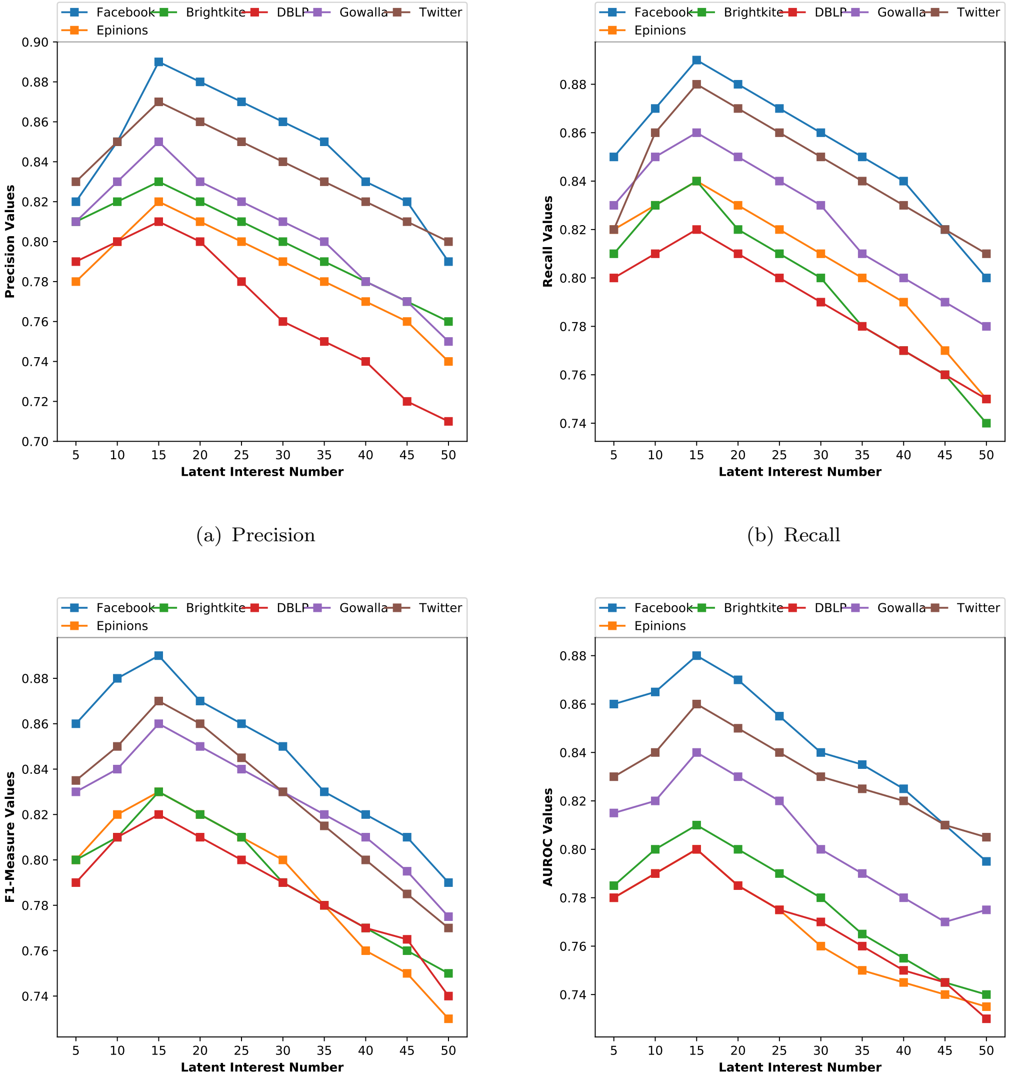
(c) u3 ($\mathscr{T}$=10)

(d) u4 ($\mathscr{T}$=20)

(e) u5 ($\mathscr{T}$=10)

(f) u6 ($\mathscr{T}$=20)

FIGURE 4.4: User Latent Interest Distribution over $\mathscr{T}$ Topics in Different Networks, (a)-(b) User u1 & u2 from Facebook Dataset, (c)-(d) User u3 & u4 from DBLP Dataset, (e)-(f) User u5 & u6 from Twitter Dataset.



(a) Precision

(b) Recall

(c) F1-Measure

(d) AUROC

Figure 4.5: Effect of Latent Interest Number on (a) Precision, (b) Recall, (c) F1-Measure, and (d) AUROC Values in Considered Datasets.

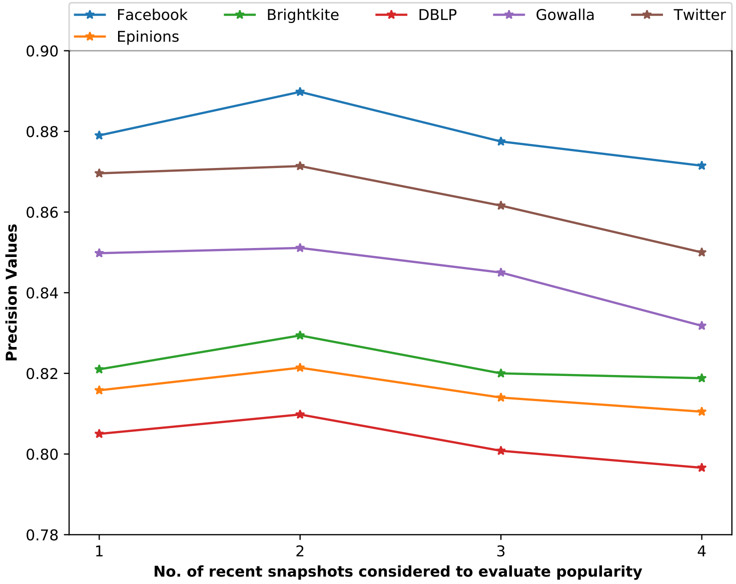
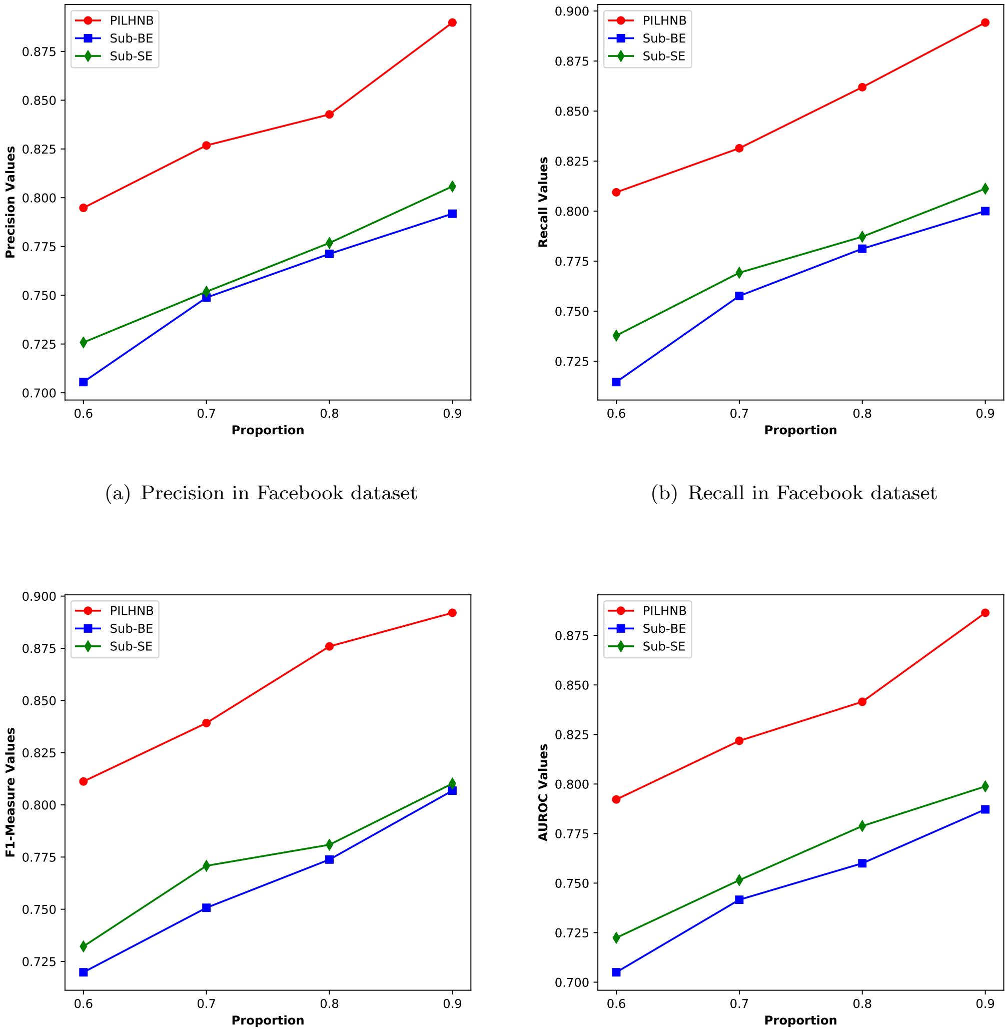


Figure 4.6: Precision Values of Link Prediction using PILHNB by Varying the Number of Recent Snapshots Considered to Evaluate the Popularity of Nodes.

shows that the precision value for link prediction is better when $t_y = 2$. So, for all the experiments, we take $t_y = 2$ for computation of popularity factor.

Next, we obtain two submodels: Sub-BE and Sub-SE, by extracting the behavioural elements (BE) as a driving factor for link prediction and the structural elements (SE) as a driving factor for link prediction separately. We compared the submodels with PILHNB and show the relations between the proportion of training sets and performance metrics of the proposed model in Figures 4.7 and 4.8. Here, the x-axis represents the proportion of the training sets, and the y-axis shows the values of considered metrics. The results clearly show that the combined effect of both the structural and behavioural elements improves the performance of link prediction significantly. Similar results are obtained for other datasets also.

Finally, we evaluate our proposed method's performance by comparing it with other baseline methods. The value of $\mathscr{T}$ is taken as 15 for these experiments. The value of the



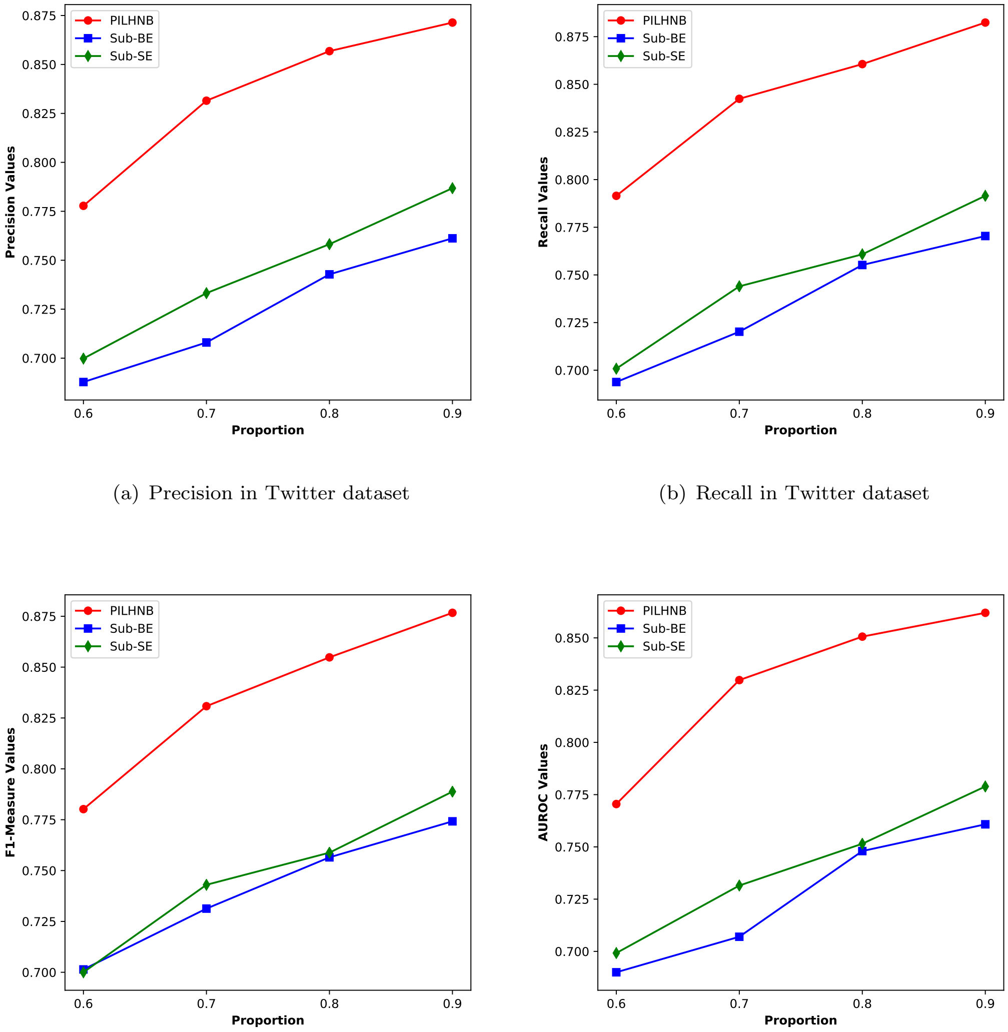
(a) Precision in Facebook dataset

(b) Recall in Facebook dataset

(c) F1-Measure in Facebook dataset

(d) AUROC in Facebook dataset

FIGURE 4.7: Comparison of Prediction Results Between Submodels and PILHNB, (a)-(d) Comparison of Prediction Results in Facebook Dataset.



(a) Precision in Twitter dataset

(b) Recall in Twitter dataset

(c) F1-Measure in Twitter dataset

(d) AUROC in Twitter dataset

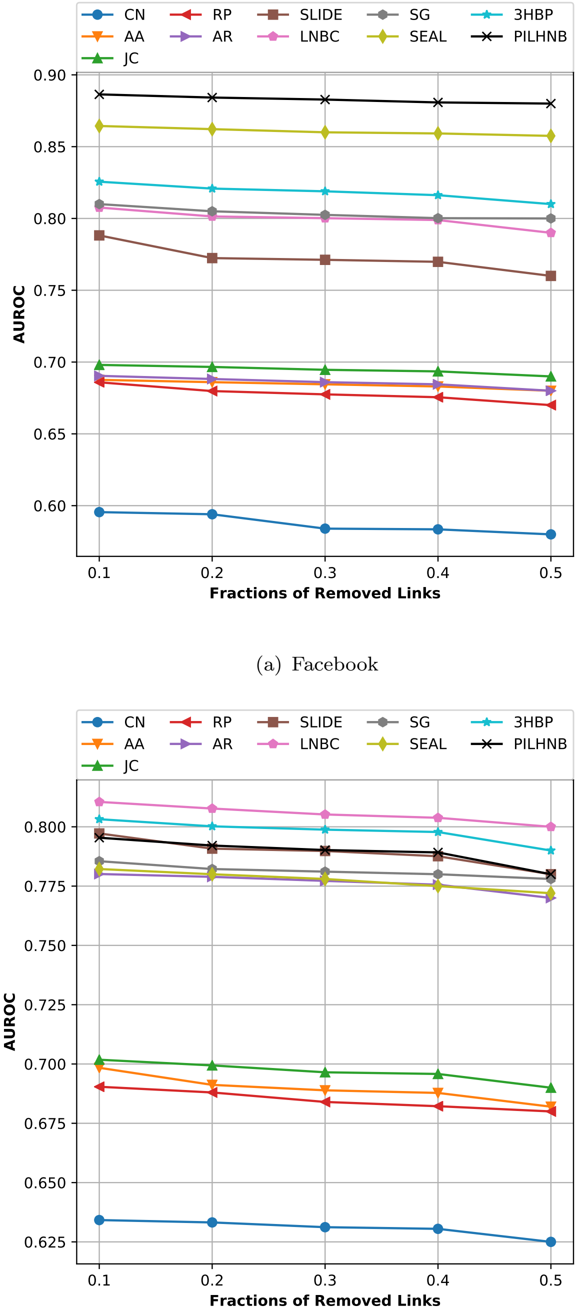
FIGURE 4.8: Comparison of Prediction Results Between Submodels and PILHNB, (a)-(d) Comparison of Prediction Results in Twitter Dataset.

threshold for interaction frequency behaviour to find the active users is taken as ten interactions. So, if a user has interacted more than or equal to 10 times after the previous snapshot, it will be considered as an active user. Similarly, we consider the location as a factor for link prediction when the users shared the same geographical location more than five times after the previous snapshot. The proposed method is abbreviated as "PILHNB", and other baseline methods are also abbreviated as with the abbreviation given in their introduction in section 4.4.2. Our method is tested on four well-known accuracy measures of link prediction, namely precision, recall, F1-Measure, and AUROC. For each method (proposed + baseline) on considered datasets, Tables 4.1, 4.2, 4.3, and 4.4 represents the average values of precision, recall, F1-Measure, and AUROC, respectively. The average is taken over the different fraction of removed links (percentage of link removed for testing, i.e., $10, 20, 30, 40, 50$). The result shows that the proposed method's evaluation metrics are better than the baseline methods for all the considered datasets except the Epinions dataset; however, it is comparable with the baseline methods for this dataset. Improvement by the proposed method in comparison with baseline methods lies up to 13.4% percent for AUROC value and between $8\% - 12.3\%$ for other considered metrics.

Table 4.1: The Comparison of Algorithms based on the Precision Value.

Algorithms	Facebook	Epinions	Brightkite	DBLP	Gowalla	Twitter
CN	.6050	.6422	.6685	.5998	.6558	.6715
AA	.6910	.7112	.6915	.6728	.7956	.7090
JC	.7024	.7236	.7448	.6956	.7248	.7319
RP	.6989	.7122	.7846	.6918	.7214	.7278
AR	.7004	.8110	.7392	.7210	.8317	.8144
SLIDE	.7972	.8122	.7097	.7227	.8211	.8006
LNBC	.8192	.8198	.7968	.7477	.7814	.7442
SG	.8080	.7850	.7868	.7461	.7685	.7849
SEAL	.8628	.7789	.8066	.7839	.8142	.8078
3HBP	.8304	.8110	.8156	.7787	.7526	.8315
PILHNB	**.8898**	**.8214**	**.8294**	**.8098**	**.8511**	**.8714**

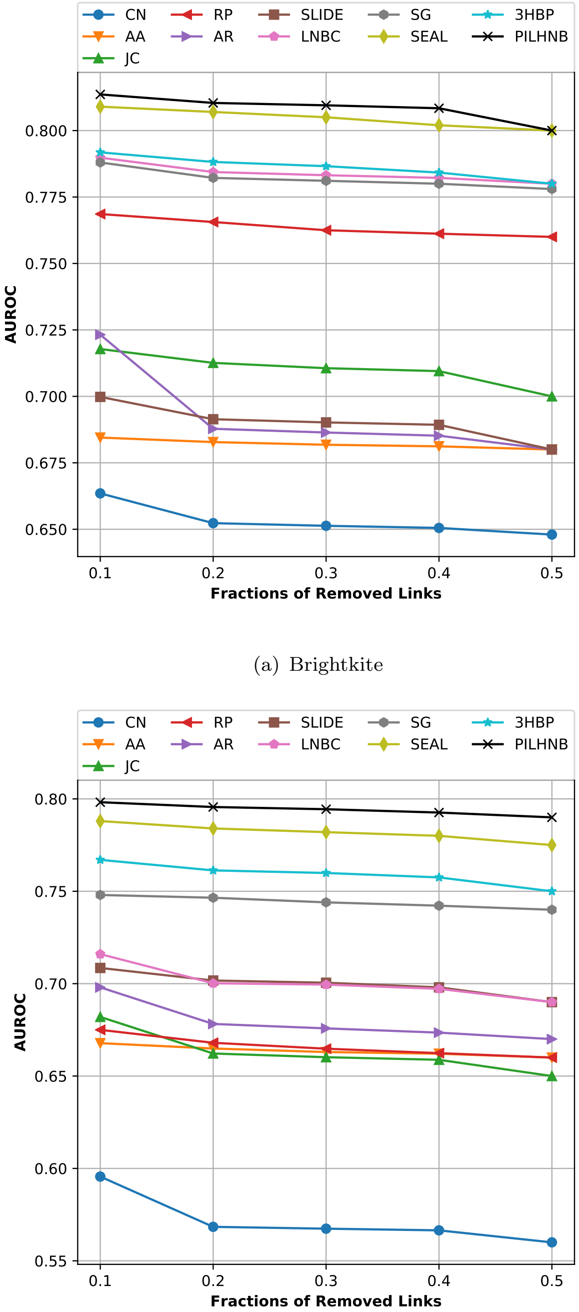
In Figures 4.9, 4.10 and 4.11 the fraction of removed links are represented on the x-axis, and the AUROC values are represented on the y-axis. The graph shows that the performance of the proposed method is optimal in comparison to other baseline methods in terms of link prediction. It also shows a slight decrease in the value of AUROC when the fraction of removed links increased. It may be due to an increase in the sparseness



(a) Facebook

(b) Epinions

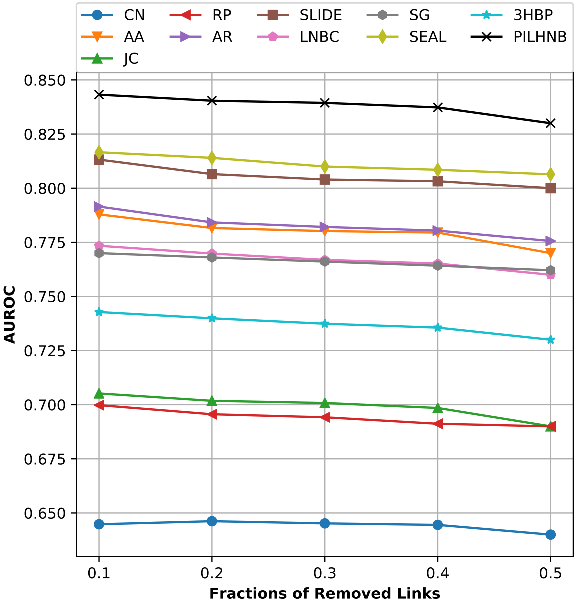
FIGURE 4.9: AUROC Values on Changing the Fraction of Removed Links for Facebook and Epinions Datasets.



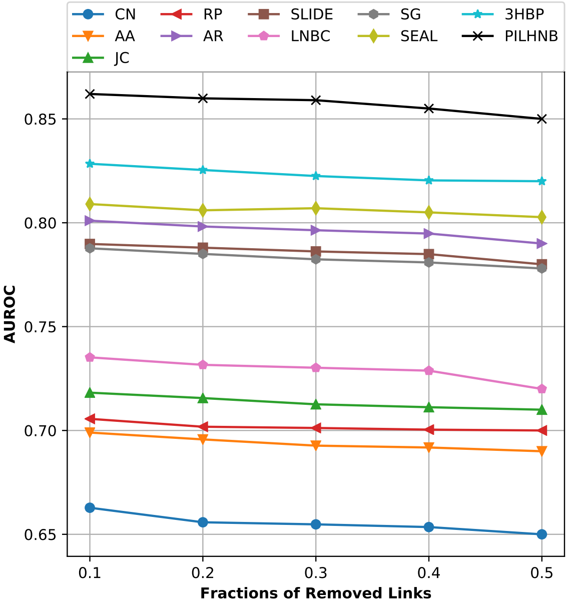
(a) Brightkite

(b) DBLP

Figure 4.10: AUROC Values on Changing the Fraction of Removed Links for Brightkite and DBLP Datasets.



(a) Gowalla



(b) Twitter

FIGURE 4.11: AUROC Values on Changing the Fraction of Removed Links for Gowalla and Twitter Datasets.

Table 4.2: The Comparison of Algorithms based on the Recall Value.

Algorithms	Facebook	Epinions	Brightkite	DBLP	Gowalla	Twitter
CN	.6148	.6512	.6711	.6094	.6612	.6744
AA	.6946	.7134	.6986	.6880	.7972	.7122
JC	.7443	.7178	.7346	.6998	.7190	.7278
RP	.6947	.7086	.7866	.6824	.7112	.7206
AR	.7096	.7990	.7459	.7130	.8006	.8155
SLIDE	.7945	.8114	.7198	.7233	.8302	.7986
LNBC	.8128	.8361	.8154	.7328	.7884	.7514
SG	.8120	.7898	.7909	.7502	.7708	.7892
SEAL	.8670	.7815	.8099	.7877	.8189	.8106
3HBP	.8354	.8146	.8165	.7810	.7517	.8405
PILHNB	**.8942**	**.8422**	**.8323**	**.8206**	**.8632**	**.8824**

Table 4.3: The Comparison of Algorithms based on the F1-Measure.

Algorithms	Facebook	Epinions	Brightkite	DBLP	Gowalla	Twitter
CN	.6099	.6467	.6698	.6046	.6585	.6729
AA	.6928	.7123	.6950	.6803	.7964	.7106
JC	.7227	.7207	.7396	.6976	.7219	.7298
RP	.6968	.7104	.7856	.6879	.7163	.7242
AR	.7050	.8049	.7390	.7169	.8158	.8149
SLIDE	.7958	.8118	.7147	.7230	.8256	.7995
LNBC	.8160	.8279	.8060	.7402	.7849	.7478
SG	.8108	.7882	.7898	.7488	.7695	.7867
SEAL	.8646	.7796	.8084	.7856	.8165	.8087
3HBP	.8329	.8128	.8160	.7798	.7522	.8360
PILHNB	**.8920**	**.8317**	**.8308**	**.8152**	**.8571**	**.8767**

Table 4.4: The Comparison of Algorithms based on the AUROC Curve.

Algorithms	Facebook	Epinions	Brightkite	DBLP	Gowalla	Twitter
CN	.5955	.6342	.6635	.5956	.6448	.6628
AA	.6876	.6984	.6845	.6678	.7879	.6990
JC	.6980	.7018	.7178	.6820	.7052	.7182
RP	.6859	.6904	.7686	.6750	.6998	.7056
AR	.6904	.7801	.7232	.6980	.7915	.8010
SLIDE	.7882	.7972	.6998	.7085	.8132	.7898
LNBC	.8076	**.8105**	.7898	.7160	.7734	.7352
SG	.8035	.7814	.7819	.7442	.7662	.7828
SEAL	.8607	.7774	.8063	.7818	.8112	.8059
3HBP	.8256	.8032	.7918	.7670	.7428	.8284
PILHNB	**.8864**	.7954	**.8136**	**.7982**	**.8432**	**.8620**

of the graph. We can also notice that the result of our proposed model PILHNB is much better in the networks having detailed information about users (user's attributes) and the content of interactions between them, such as Facebook, Twitter, and DBLP datasets. In

Gowalla and Brightkite, location information is in more detail; however, a few other pieces of information were missing. In the Epinions dataset, maybe the network's sparseness restricts our method to outperform the baseline. Overall the result of the PILHNB model is better than the baseline methods. Therefore, the experimental results show that the proposed model can effectively improve link prediction performance in dynamic social networks.

4.5.1 Insightful Discussion

Our proposed model learns the individual nodes' behaviour pattern with time, making the model more consistent, robust, and best suited for noisy networks because it considers each users' importance in the network. Considering the location and popularity feature makes the model more accurate. Using common interest and attribute similarity feature makes the model more effective than the considered baseline methods. However, in the baseline methods based on graph embedding and graph neural networks consider mainly the structural information of the nodes and their neighbours; they do not consider the content of communication messages or the other factors which we have considered. As the proposed model consider many factors together, it makes the model theoretically complicated and increases the preprocessing overheads for finding different feature vectors.

If the size of the network is n in terms of nodes to be processed, the time complexity of the algorithm is $T_{PILHNB} = T_{EXTRACT} + T_{PREDICT} = O(n)$. Here,$T_{EXTRACT}$ is the time to extract the latent interest of the users and $T_{PREDICT}$ is the time to predict the upcoming links in the network. The dynamic network may change during the meantime; however, we are neglecting this change, considering this as a very minor change in the network. It is also a limitation of our proposed model.

4.6 Conclusions

In this chapter, a multifeature analysis-based link prediction model PILHNB is proposed to predict links among users of the dynamic social networks by utilizing the user behaviour and the network structure change pattern of the evolving network. We used the LDA topic model for user behaviour pattern discovery and to infer the user interest distribution. To reduce the adverse impact of interest distribution, the LDA is improved by the Gaussian weighting technique. Then the HNB algorithm is used to analyze the overall effect of all the considered controlling elements responsible for the prediction of links in the networks. The performance of link prediction is improved in our proposed model by considering and combining both the behavioural and structural evolution pattern of the nodes.

We used six real-world datasets for our experiments. The experimental results validate that the proposed model PILHNB gives better performance in terms of precision, recall, F1-Measure, and AUROC on almost all the considered datasets compared with other considered baseline methods. By using our proposed model, we can effectively predict links among users of social networks. We can learn the user behaviour pattern, which changes over time, and also the pattern of structural changes in the networks. It can be applied to understand the evolution pattern of dynamic networks and can be useful in many applications of link prediction.