Chapter 2

Background

This chapter presents an overview of the state-of-the-art in link prediction and influence maximization in social networks. We discuss the literature review of link prediction and influence maximization techniques in sections 2.1 and 2.2, respectively. Section 2.3 discusses the key concepts, basic definitions, and preliminaries used in this thesis. Then, we discuss the principal evaluation metrics, public datasets used in this thesis, and baseline methodologies used to compare our proposed models in sections 2.4, 2.5, and 2.6, respectively. Section 2.7 mention the hardware and software used for experimental work performed in this thesis.

2.1 Literature Review for Link Prediction

The problem of Link prediction was initially introduced by Liben-Nowell and J. Kleinberg in [27] as a fundamental problem in social network analysis [28] and knowledge graph completion [29]. In literature, 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, 88, 24].

A graph data of social networks generally have significant evolution information, such as the pattern of change in the structure of the graph [89, 90]. Some probability-based models of link prediction [91, 92, 93, 94] consider the dynamic behaviour of social networks; however, they suffer from issues related to the model capacity and computation.

Link prediction techniques in [95, 96, 97] incorporated the information related to user behaviour such as similar hobbies, culture, language, geographical location, or interaction frequency to predict the links between users. A relational topic model proposed in [98] predicts the link among the text using the analysis of topic distribution in text data. Authors in [99] proposed a label propagation-based algorithm for similarity-based link prediction in social networks. However, the above methods consider the impact of interests derived directly from labels or keywords; they do not consider the user behaviour pattern, which may also be influenced by other factors such as structural information along with the interests of the users and the combined behaviour of the individual nodes and their neighbours in the networks.

Authors in [100, 101, 102] consider the location check-in information of users for link prediction, and paper [103] proposed a link prediction model based on users' popularity; however, they do not consider other important factors such as users interests, structural patterns, and behaviour of the nodes. Studies in [104, 105, 106] show that user behaviour learning is also responsible for predicting links in dynamic social networks. User behaviour-based techniques are applied for web-link prediction [107], mobile web systems [108], and recommendation systems [95, 109]; however, most of them consider only the single activity

and do not consider the others collectively. Authors in [110] use the analysis of multiple activities by users to measure the importance of their role in link formation in Facebook. Authors in [111] proposed an algorithm by weighting activities of users for collaborative filtering and recommendations to target users. In paper [112, 113], the analysis of user behaviour is applied for the link prediction task. However, these techniques can not be used directly in the considered scenario for link prediction in social networks because they consider specific factors based on above mentioned specific scenarios and applications. User relationship-based methods [114, 115] for link prediction uses users attribute similarities. Authors in [116] proposed an algorithm by combining the structural and attribute similarity for link prediction. Authors in [117] use network clustering coefficient and degree of nodes for the link prediction task. The graph neural network-based method in [49] used the subgraph structure information for each pair of nodes, which makes this model difficult to implement for large graphs. In the graph embedding-based method proposed in [118], network embeddings alone are not able to capture the most useful link prediction information located in the local structures.

In this thesis, we present methods to overcome the above-discussed limitations of existing link prediction techniques by considering the dynamic nature of the graph, users' behaviour pattern, the topic of interest of the users, nodes' popularity, and location-based information of the nodes. We present two models for link prediction in dynamic social networks. The first model uses conditional temporal Restricted Boltzmann Machine for predicting the links that may appear in the network by considering the evolutionary networks' temporal and structural patterns. The second model presents a modified Latent Dirichlet Allocation and Hidden Naive Bayesian-based link prediction technique named *Popularity, interests, the location used hidden Naive Bayesian-based model* for link prediction in dynamic social networks by considering behavioural controlling elements like relationship network structure, nodes' attributes, location-based information of nodes, nodes' popularity, users' interests, and learning the evolution pattern of these factors in the networks.

2.2 Literature Review for Influence Maximization

The study of influential nodes in viral marketing was first proposed in [63] by Domingos and Richardson. Further, David Kempe *et al.* in [59] formulated the influence maximization problem as a combinatorial optimization problem and suggested a greedy algorithm applied to IC and LT models (more specifically on graphs to represent diffusion using IC and LT models) with an approximation guarantee of $(1 - 1/e)$. However, for large networks, the proposed solution does not scale due to its requirement of a large number of Monte-Carlo simulations to estimate influence spread. This is due to the working of the greedy algorithm, which tests every node as a seed node for influence maximization in each iteration. Several techniques [119, 120, 121, 122, 123, 124, 125] have been proposed to handle this issue for influence maximization in static networks.

Broadly, the solutions proposed for influence maximization [126] can be divided into two categories: the algorithms of the first category aim to improve the performance of the greedy algorithm and give an approximation guarantee [127, 72]. Alternatively, the second category of algorithms put on several heuristics but lack verifiable approximation guarantee [67, 74, 128, 86]. However, all these approaches consider only static networks.

Some of the IM methods in the online social network consider the snapshots of the graph and then apply static IM algorithms. However, these approaches do not handle the real dynamics of the network. Some methods consider the graph streams, but they do not provide a theoretical guarantee of their seed quality and may return arbitrarily bad solutions. For instance, Aggarwal et al. [129], Zhuang et al. [130], and Song et al. [131] focuses on $t + \delta$ given the changing aspects of the progress of the network throughout the interval $[t, t + \delta]$, where δ denotes the small change in time. They apply diffusion maximization independently in each static graph G_t and use S_{t-1} as seed node set for influence spread. However, the previous seed set might become inefficient at a later stage because of the graph's dynamic nature. So, these solutions are not effective for dynamic social networks.

Recently, there have been many studies about IM in online social networks. A few significant contributions are discussed in this section. Wang et al. in [132] proposed a "Pairwise Factor Graph (PFG) model" to formalize the problem of IM in social networks using a probabilistic model and further extended it by incorporating the time information, which results in the "Dynamic Factor Graph (DFG) model" for IM in the dynamic social network. Aggarwal et al. in [129] use the communications of given social network entities, which can frequently be predicted based on past behaviour of the evolving network, and these represented future interactions, which were used to model the spread of information. Rodriguez et al. in [133] developed a method INFLUMAX for IM that considers time-based dynamics underlying the diffusion processes. This method allows for variable transmission (influence) rates between nodes of a network.

Zhuang et al. in [130] proposed an algorithm to determine a subset of seed nodes in the network so that the particular information diffusion method in the network can be best projected with the probing nodes. That is, it decreases the likely error between the evaluated network and the real network. Gayraud et al. in [134] introduced a persistent and transient variation of IC and LT model for justifying network evolution. Li et al. in [135] proposed a novel conformity-aware greedy algorithm called CINEMA for a conformity-aware cascade model that integrates the interplay among conformity and influence. Han et al. in [136] proposed a dynamic probing context that accepts the community structure as a unit and updates network topology to investigate the genuine changes of network and employs community-based influence maximization. Wang et al. in [137] proposed an IM query named Stream Influence Maximization (SIM) on social streams; it uses the sliding window model and keeps up a set of k seeds with the most significant influence value over the latest social activities. Tong et al. in [138] demonstrated the dynamic IC model and presented the idea of an adaptive seeding technique with a provable performance guarantee. However, all these approaches require high computation costs, and there is still a lot of scopes to do better for dynamic social networks.

In this thesis, our goal is to have maximum information spread in a minimum time span for online social networks. For achieving this goal, we predict the upcoming snapshot of the

graph as G_{t+1} by considering the temporal and structural behaviour of the evolving graph, and then we detect the seed set that maximizes the information spread in the upcoming snapshot of the graph.

Further, the influence maximization approaches can be generally categorized into different groups depending upon the algorithms' working technique. We have grouped the influence maximization methods into the following four categories:

2.2.1 Centrality based Approaches for IM

Chen et al. in [67] presented a *degree discount centrality* algorithm in which the node with the highest degree is selected and added to the seed set in each of the k iterations. In this approach, if node v_i is selected as a seed node, the edges between v_i and the other nodes are ignored in the computation of the spreading capability of the nodes. In [68], a *degree distance centrality* algorithm is proposed, which ensures a minimum distance between each selected seed node. Wang et al. in [69] presented an approach for IM named *degree punishment* with repetitive punishment process. Here, when a node is selected as a seed node, its first and second-level neighbours are punished by reducing their influentiability level. A *distance-based coloring* method is presented in [70]. In this approach, the nodes are firstly colored such that the distance between the same color node is higher than a threshold. Then, the color-based grouping is done, and within each group, the nodes are ranked on the basis of their degree. Finally, the top-k nodes in the groups based on the maximum degree are selected as the most influential nodes. Although most of the centrality-based IM approaches are efficient and scalable, accuracy is still an issue with these algorithms.

2.2.2 Sub-modularity based Approaches for IM

In the paper [71], Svirdenko et al. modified the influence maximization problem proposed in [59] by adding the constraint of node price. The authors proved that the objective function of the proposed problem is sub-modular under IC and LT models. Therefore,

the *Greedy Algorithm* proposed in [59] can be applied with the constraint of node price. This approach gives a performance guarantee but is suitable for small networks due to the time-consuming *Monte Carlo (MC)* simulations. Leskovec et al. in [72] presented an algorithm named *cost-effective lazy forward (CELF)*. This approach uses the property of sub-modular function for cascade influence and reduces the computations of marginal gain, which makes it 700 times faster than the normal greedy algorithm. Goyal et al. in [73] proposed an improved version of CELF named *CELF++*. This algorithm computes two marginal gain values simultaneously, which makes it 30% to 50% faster than CELF experimentally. The above-discussed sub-modularity-based algorithms are much faster than the greedy algorithm and also give a performance guarantee. However, they still use time-consuming MC simulations and hence not suitable for large-scale networks.

2.2.3 Path based Approaches for IM

In paper [74], Kimura et al. presented an IM approach based on the shortest path named as *shortest path 1 model (SP1M)*. This method considers that only the shortest and second shortest paths are important in influence spread. It does not use MC simulations. The *Maximum influence arborescence (MIA)* method is proposed by Chen et al. in [75]. To estimate the influence propagation from node v to other nodes, MIA uses local structures' arborescence. The arborescence of a node v is computed as the set of nodes that are located in paths starts from v and includes edges with propagation probability greater than the threshold. Kim et al. in [76] presented an IM method named *independent path algorithm (IPA)*. This approach considers that the influence path from u to v are independent of each other, and all paths having edges' propagation probability above a threshold are considered for influence spread. Rossi et al. in [77] proposed an algorithm named *matrix influence algorithm (MATI)* for influence maximization. This method considers all possible paths for information spread and uses the *pruning threshold* technique to reduce the computation of influence paths. Influence path-based IM techniques [139] maintain a tradeoff between accuracy and efficiency compared with centrality and sub-modularity-based techniques. However, they cannot provide a theoretical guarantee for the optimal solution. They also need a large amount of memory to maintain the information regarding the large set of influential paths.

2.2.4 Context-aware Approaches for IM

Literature shows that in social networks, users' social-behavioural information plays a vital role in determining their influentiability [78, 79, 80, 81, 82, 83, 84]. Therefore, we need to combine this information with the structural information to find the effective influence maximization. Mochalova and Nanopoulos in [78] presented a technique for selecting seed set by considering the marketing potential of interested users. However, no specific criterion is used to compute the value of interest. Z. Zhu in [79] introduced a seed selection technique based on users' interest in a particular product and also the sincerity and trust between the users. Y. Li et al. in [80] presented an IM algorithm based on users' interest in various topics. Here the authors used topic-based query processing to improve the effectiveness of the seed set. S. Li et al. in [81] proposed an approach to identify influential users based on each users' interest in various topics computed based on users' interactions and activities in earlier times. Zareie et al. in [82] presented an Influential Marketer User Detection (IMUD) algorithm to select a seed set with k members so that the selected seeds are located close to interested users covering as many such users as possible. In [83], Chen et al. presented a *Topic-aware Influence Maximum (TIM)* algorithm based on the topic-aware query. In paper [84], Zareie et al. presented an IM approach called *Multi-criteria influence maximization (MCIM)* for finding a set of influential nodes selected as the initial core in the spreading process with the *Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)* [140] method so that the seed set featured maximal influence spread and minimal overlap.

Most of the above-discussed techniques tried to consider the topic-related information along with the structural information of the network to find the suitable seed set for influence maximization. However, there exist many possibilities to improve the technique of topic distribution among nodes, topic-aware diffusion process as well as to consider other important factors responsible for efficient and effective seed selection.

Compared with the traditional IM approaches, topic-aware methods are more efficient and

effective. Still, these algorithms ignore many other important factors such as the popularity of nodes, location information, the behaviour of the nodes etc., which may also be responsible for influence maximization in dynamic social networks. Thus, accuracy in seed selection and effectiveness in information spread is still an issue.

In this thesis, we propose methods to overcome the limitations of existing influence maximization techniques by considering the dynamic nature of the graph, users' behaviour patterns, the topic of interest of the users, nodes' popularity, and location-based information of the nodes. In the upcoming chapters, we present the study of the Influence Maximization problem in a social network that evolves with time and propose two new frameworks: *Link Prediction based Influential Node Tracking*, and *Multifeature based Influential Nodes Tracking*.

2.3 Preliminaries

In this segment, we briefly describe some of the theoretical concepts, which have been used in further chapters.

2.3.1 Link Prediction

Link Prediction: Link prediction aims to predict the edges that are expected to be added to the network at a future time $t+1$ given the snapshot of the network at time t [27].

Link predictor: Consider a graph $G(\mathbb{V}, E)$, where $\mathbb{V}$ is the set of nodes, and E is the set of connections. Multiple connections and self-association are not permitted. Denoted by U, the universal set containing all $\frac{|\mathbb{V}|.(|\mathbb{V}|-1)}{2}$ potential connections, where $|\mathbb{V}|$ signifies the number of elements in set $\mathbb{V}$. The set of nonexistent connections is $U - E$. We expect some missing connections (or the connections that will show up in the future) from the

set $U - E$, and to discover this connection is the task of link predictor.

Traditional Link Prediction method: All strategies give an association weight score $s(x, y)$ to pair of nodes (x, y) in the input graph G. And then produce a positioned list in decreasing order of the score $s(x, y)$. In this way, they can be seen as computing a vicinity or similarity between nodes x and y concerning the network topology.

Time series-based Link prediction: In time series-based link prediction, the essential idea is to make a time-based ordering of each non-connected pair of nodes of the network utilizing similarity scores are given by a topological metric [141]. The time series based link prediction problem is formally introduced as follows:

Given, $G = \{G^1, G^2, G^3, \ldots, G^t\}$ *as a series of snapshots of dynamic network with evolving edges during timestamps* $T = \{1, 2, 3, \ldots, t\}$. *The time series based 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}.

2.3.2 Influence Maximization

2.3.2.1 Diffusion Models

Independent Cascade (IC) model: It describes a straightforward and intuitive diffusion process. Beginning from a seed set S, which is active, the diffusion process happens in discrete-time steps. When a node u becomes active in step t, it attempts to activate all of its inactive neighbours in step $t + 1$. For each neighbour v, it succeeds with the known probability p_{uv}. If it succeeds, v ends up active; else, v stays inactive. When u has made all these attempts, it does not get the chance to make further activation attempts at later occasions.

Linear Threshold (LT) model: In this model, each node has an activation threshold Θ_u, and the node u become active or accepts a new idea if the influence from all its active neighbours has crossed the threshold, i.e., if $\sum_{v \in N(u)} b_{u,v} \geq \Theta_u$.

Here, each edge between u and v has an arbitrary weight $b_{u,v} \in [0, 1]$ such that the sum of the weights of the incoming edges of u is $\sum_{v \in N(u)} b_{u,v} \leq 1$; and each node $u \in V$ has a threshold $\Theta_u \in [0, 1]$ sampled uniformly at random and independently from the others. $N(u)$ denotes the set of neighbours of node u. Node v is an active neighbour of node u. In both the IC and LT diffusion models, each node is independent and autonomously and asynchronously perform the diffusion [142]. Several nodes can become active at the same time because there are various nodes in the seed set trying to activate their inactive neighbours. During this attempt, many nodes will become active at the same time.

Other terms related to the definition of diffusion models are defined below:

Seed Set: The set of active nodes before the start of the diffusion process is termed as a seed set. In this thesis the sed set is represented by S and the size of the seed set is given as k.

Live Edge: The edge between the two active nodes is termed as live edge.

Live Path: The path between two active nodes which contains only live edges are termed as a live path.

Active Nodes are the nodes that got influenced by the message being spread in the network.

Inactive Nodes are the nodes that are either not influenced by the message received from their active neighbours or did not get any message from their neighbours.

In the influence maximization problem, we aim to maximize the total number of active nodes in the network at the end of the diffusion process.

2.3.2.2 Influence Maximization

Given the seed set S, we describe the influence spread of S as the expected number of activated nodes when the diffusion procedure stops, represented by the influence function $\sigma(S)$.

Influence Maximization: The Influence Maximization process is to find a seed set $S \subseteq V$ of maximum size k to maximize the influence function $\sigma(S)$. Formally, the Influence Maximization method can be defined as the following optimization problem:

$$I^* = \arg\max_{|S| \leq k} \sigma(S). \tag{2.1}$$

It has been shown by David Kempe et al. in [59] that the IM problem under IC/LT model is NP-hard. It can be shown that the influence function $\sigma(S)$ under the IC model is monotone and submodular. A set function f is monotone if $f(S+e) \geq f(S)$ for an element e; and f is submodular if it has diminishing returns as $f(S+e) - f(S) \geq f(T+e) - f(T)$ for an element $e \in U/T$ for a finite set U (in this case, the set U is equal to the set of nodes in the graph) whenever $S \subseteq T$. These properties of the IC/LT model allow for an approximation algorithm with a guarantee.

In particular, there are elementary Greedy Algorithms (see Algorithm 2) proposed by Nemhauser et al. in [143] for maximizing monotone submodular functions. The greedy algorithm repeatedly picks the node with maximum marginal gain and adds it to the present seed set until the budget k is reached. It can be shown that this algorithm approximates the optimal solution with a factor of the $(1 - \frac{1}{e})$ for the IM problem.

The optimization problem of selecting the influential nodes (i.e., the seed set) is $NP-hard$ while computing $\sigma(\cdot)$ exactly is $\#P-complete$ (for both LT and IC models). However, it is possible to calculate arbitrarily good approximations of $\sigma(\cdot)$ (i.e., $1 \pm e$ approximation for

any given e) with the help of a polynomial in $|V|$ number of times simulations of the process. However, this is inefficient for large networks. Various strategies have been proposed to handle the inefficiency of the greedy algorithm. We also observed in our experiments that the greedy algorithm takes a significant amount of time to run for large networks.

Influence Spread: The total number of users/nodes finally get influenced after the completion of the diffusion process in the network is termed as influence spread.

2.3.3 Other Important Definitions

Location-aware Dynamic Attributed Networks: At a particular timestamp t, the corresponding location-aware dynamic attributed network is represented as $G^t = (V^t, E^t, A^t, L^t)$, where vertices V^t denotes the set of users, $E^t \subseteq (V^t \times V^t)$ denotes the pairs of users having a friendship relationship at t, $A^t = [a^1, a^2, \ldots, a^{n_t}]^t$ denotes the node attributes and $L^t = [l_1, l_2, \ldots, l_{n_t}]^t$ denotes the nodes check-in information.
Here, the check-in information includes the "check-in time" (Date with time), "latitude", "longitude", and "location id" for each user. Here, there is a unique id for each location represented by "location id".

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 equation 2.2:

$$P_{v_i} = \frac{\Delta d_{v_i}(t_y, t_z - t_y)}{\Delta d_{v_i}(t_x, t_z - t_x)} = \frac{d_{v_i,fresher}}{d_{v_i,all}}, \tag{2.2}$$

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

2.3.4 Restricted Boltzmann Machine

In our proposed Link Prediction-based Influential Nodes Tracking (LPINT) model, we predict the appearance of new links by exploring the evolution pattern of the graph. We have used ctRBM [144], which adopts temporal variations (temporal connections) and neighbour opinions (neighbour connections) during the training phase and performs prediction dependent on the existing time window of snapshots and the local neighbour's predictions of each pair of nodes. A detailed explanation of the model is given in the next chapter. Here, the basic Restricted Boltzmann Machine (RBM) [144] is introduced and defined as follows:

Restricted Boltzmann Machine: is a particular case of Markov Random Field, which has two layers of variables, Visible layer variables are denoted as set V and Hidden layer variables are denoted as set H. A typical RBM is represented in figure 2.1. Here, set V and set H form a fully-connected bipartite graph with undirected edges. RBM defines a distribution over $(V, H) \in \{0,1\}^{|V|} \times \{0,1\}^{|H|}$, where $|V|$ and $|H|$ are the dimension of V and H layers. The joint probability distribution for RBM is defined as:

$$P(V, H) = \exp(V'WH + x'V + y'H)/Z, \tag{2.3}$$

here $Z = \sum_{V,H} \exp(V'WH + x'V + y'H)$, $W \in \mathbb{R}^{|V|\times|H|}$ is the weight between layers V and H, x, y is the biases for V and H, respectively. V', x', and y' are the transposes of V, x, and y, respectively.

Due to the bipartite nature of the RBM, there is no interaction between nodes in individual layer, so the conditional probability distributions are fully factorial and represented by:

$$\begin{aligned} P(H_j = 1|V) &= \omega(y_j + W'_{:,j}V), \\ P(\widetilde{V}_i = 1|H) &= \omega(x_i + W_{i,:}H), \end{aligned} \tag{2.4}$$

here, ω is the logistic function given as $\omega(a) = (1+\exp(-a))^{-1}$, $\widetilde{V}$ is the reconstructed data showing the model's evaluation, i and j are row and column index. The aim of learning is

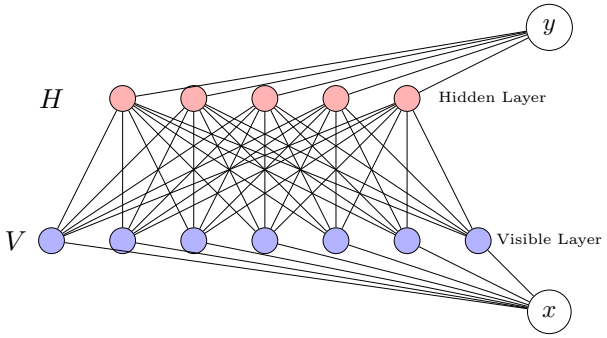

FIGURE 2.1: Restricted Boltzmann Machine

to minimize the gap between V and $\widetilde{V}$.

2.3.5 Topic modelling

To capture the users' interest and attribute similarity, we used the topic modelling technique from the field of Natural Language Processing. Here, we give the introduction of the topic modelling technique and also discuss a basic topic modelling algorithm, i.e., Latent Dirichlet Allocation (LDA) [145]. We have used these concepts in chapters 4 and 5 of this thesis.

Topic modelling: It is an unsupervised Bayesian model, which presents each document in a document set as a probability distribution with an unsupervised learning approach [146]. The main objective of the topic model is to identify topics from large document collections by exploiting the word distribution in a corpus. It is a typical Bag of Words (BOW) model which assumes that a document is a collection of words and there is no ordering relationship between words. Here a topic is a probability distribution with all the words in the document as a support set, indicating how often the word appears in the topic.

Latent Dirichlet Allocation: LDA is capable of clustering words, documents, and other related entities based on latent topics [145]. To be specific, given a document d, a multinomial distribution θ_d over topics T is sampled from a Dirichlet distribution with

parameter α. For each word w_{d_i} from document d_i, a topic t_{d_i} is picked from a topic multinomial distribution ϕ_t sampled from a Dirichlet distribution with parameter β. Thus, we can calculate the probability of a word w from a document d as follows:

$$P(w|d,\theta,\phi) = \sum_{t \in T} P(w|t,\phi_t)P(t|d,\theta_d), \tag{2.5}$$

then, the likelihood of corpora C is

$$P(T,W|\Theta,\Phi) = \prod_{d \in D}\prod_{t \in T} \theta_{d_t}^{n_{d_t}} \times \prod_{t \in T}\prod_{w \in W} \phi_{t_w}^{n_{t_w}}, \tag{2.6}$$

where n_{d_t} is the number of times the topic t has been mentioned in document d, W is the number of words in a given document and n_{t_w} represents the number of times that the word w has been associated with topic t.

2.4 Evaluation Metrics

In this section, we present the evaluation metrics that we used to capture the necessary features of our proposed models in this thesis. The literature and our experiments show that the chosen evaluation metrics are the best ones to present the efficiency and effectiveness of the proposed models of this thesis.

2.4.1 Quality Metric for Evaluation of Link Prediction

The link prediction task can be treated as a binary classification problem [147]. Here, the presence of a link is the positive data element, and the absence of a link is the negative data element. The evaluation of the binary classification process can be represented as a confusion matrix. In the confusion matrix, we use the following terms:

- *True Positive(TP)*: positive data element predicted as positive
- *True Negative(TN)*: negative data element predicted as negative
- *False Positive(FP)*: negative data element predicted as positive

- *False Negative(FN)*: positive data item predicted as negative

Many metrics can be derived using the confusion matrix; some of them are as follows:

- *True Positive Rate(TPR)/Recall/Sensitivity*: Out of all the positive classes, how much we predicted correctly.

$$TPR = \frac{\#TP}{\#TP + \#FN}. \tag{2.7}$$

- *False Positive Rate(FPR)*

$$FPR = \frac{\#FP}{\#FP + \#TN}. \tag{2.8}$$

- *True Negative Rate(TNR)/Specificity*: The proportion of actual negatives that are correctly identified.

$$TNR = \frac{\#TN}{\#TN + \#FP}. \tag{2.9}$$

- *Precision*: Out of all the positive classes, we have predicted correctly, how many are actually positive.

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP}. \tag{2.10}$$

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 as follows:

- **Precision**: Precision quantifies the number of positive class (existence of links in this case) predictions that actually belong to the positive class. It can be evaluated using equation 2.10.

- **Recall**: The recall metric finds all positive samples (existence of links in this case) in the data and can be computed using equation 2.7.

- **F1-Measure**: The F1-Measure is the harmonic mean of precision and recall. It can be computed using the following formula:

$$\text{F1-Measure} = 2 \cdot \frac{Precision \cdot Recall}{Precision + recall}. \tag{2.11}$$

- **AUROC**: The receiver operating characteristics curve represents a graph plot between the TPR(Sensitivity) on the y-axis and the FPR(1-sensitivity) on the x-axis. The TPR and FPR can be calculated using equations 2.7 and 2.8, respectively. The AUROC can be evaluated using the trapezoidal rule, which is the sum of all trapezoids under the curve. The value of the AUROC of a predictor should be greater than 0.5, and the higher value of AUROC shows the better performance of the predictor [150].

The values of these metrics lie between $0.5 - 1.0$. Better prediction results have greater precision, recall, F1-measure, and AUROC values.

2.4.2 Quality Metric for Evaluation of Influence Maximization

- **Spread of Influence:** It is the total number of nodes in the network which gets influenced by the seed set S after the diffusion process stops. The algorithm with a higher value of spread of influence is termed as a higher quality algorithm.

- **Speedup:** We consider the speedup percentage (%) in terms of time taken for influence spread. It represents how efficiently the influence spreads using a selected seed set S by the proposed algorithm is compared to the considered baseline algorithms. A higher value of speedup % shows that the proposed algorithm performs better than the considered baseline algorithm in terms of time taken to spread the influence.

2.5 Datasets

In this section, we present the datasets used in the experimental parts of this thesis. These are standard datasets that are publically available and are commonly used in various papers in the literature.

2.5.1 Datasets used for Link Prediction based Influence Maximization

In the link prediction based influence maximization task, we performed our experiments on four real-world dynamic networks: College, Mathoverflow, Askubuntu, and Wikitalk. The datasets are available on the web at https://snap.stanford.edu/data/.

- The **College** [151] is a transient network dataset that comprises private messages sent on an online social network at the University of California, Irvine. Here, clients search the network for different clients, and after that, start a discussion based on profile data. An edge (p, q, t) signifies that client p sent a message to client q at time t.
- The **Mathoverflow** [152] is a temporal network of interactions on the stack-exchange web site mathoverflow. The data consists of a directed edge (p, q, t), where user p interacts with user q at time t.
- The **Ask-ubuntu** dataset used in [152] is an online social network of interactions on the stack-exchange website askubuntu. This network data also consists of a directed edge (p, q, t), where user p interacts with user q at time t.
- The **Wiki-talk** dataset used in [153] is an online social network representing Wikipedia users editing each other's talk page. A directed edge (p, q, t) shows that user p edited user q's talk page at time t.

The basic statistics of the dataset networks are given in Table 2.1.

TABLE 2.1: Statistics of Datasets 1

Dataset	Nodes	Time span (days)
college	1,899	193
mathoverflow	21,688	2,350
ask-ubuntu	137,517	2,613
wiki-talk	1,140,149	2,320

2.5.2 Datasets used in Multifeature Analysis based Link Prediction and Context-aware Influence Maximization

For Multifeature Analysis-based Link Prediction and Context-aware Influence Maximization tasks, we used six real-world network datasets for the performance evaluation of our proposed models. These datasets are from online social networks and coauthor networks. The basic statistics of the datasets are indexed in Table 4.1. These datasets are publically available at Stanford Large Network Dataset Collection (http://snap.stanford.edu/index.html). The description of these networks is given below.

- **Facebook** [154]: This dataset consists of friend lists from Facebook. The dataset includes node features, circles, and ego networks.
- **Epinions** [155]: This is a who-trust whom, an online social network of a general consumer review site Epinions.com. Members of the site can decide whether to trust each other.
- **Brightkite** [156]: This dataset was taken from a location-based social networking service provider where users shared their locations by checking in. The dataset was collected using their public Application Programming Interface (API).
- **DBLP** [157]: The DBLP computer science bibliography dataset provides a comprehensive list of research papers in computer science. This is a co-authorship network in which two authors are connected if they publish at least one paper together.

- **Gowalla** [156]: This dataset was taken from a location-based social networking website where users share their locations by checking in. The dataset was collected using their public API.

- **Twitter** [158]: This dataset has been built after monitoring the spreading process on Twitter before, during, and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson on the 4th July 2012.

TABLE 2.2: Statistics of Datasets 2

Dataset	Nodes	Edges
Facebook	4,039	88,234
Epinions	5,261	23,915
Brightkite	50,686	194,090
DBLP	101,836	873,256
Gowalla	107,067	456,760
Twitter	256,626	3,991,895

2.6 Baseline Methods

In this section, we briefly discuss the state-of-the-art techniques for Link prediction and Influence Maximization used to compare the effectiveness and efficiency of our proposed models in this thesis. We use these baselines either because they are standard techniques or because our work is an extension of the chosen baseline methods.

2.6.1 Baseline Methods used for Link Prediction based Influence Maximization

Basic approaches used for comparison of influence maximization in evolving networks with and without link prediction are:

- **DegreeDiscount** [67]: A degree discount heuristic algorithm developed for the Independent Cascade model with uniform propagation probability.

- **LPDegreeDisount**: A degree discount heuristic algorithm implemented on the link prediction based predicted snapshot for seed selection.
- **PageRank** [159]: A link analysis algorithm that positions the priority of pages in a Web graph.
- **LPPageRank**: The PageRank algorithm for IM implemented on the link prediction based predicted snapshot for seed selection.
- **UBLF**: The Upper Bound based Lazy Forward (UBLF) algorithm [127] for IM derive an upper bound to reduce the number of spread estimations in the initialization step of influence maximization.
- **LUBLF**: The UBLF algorithm implemented on the link prediction based predicted snapshot for seed selection.
- **EXCHANGE**: Our proposed upper bound based algorithm computes the upper bound of marginal gain while evaluating the upper bound of node replacement gain.

Approaches used for comparison of Online Influence Maximization with the proposed LPINT model are:

- **CIM** [160]: CIM (Continuous Influence Maximization) adopted the IC model as an influence model and pp=0.01 as propagation probability.
- **OIM** [161]: OIM (Online Influence Maximization) uses explore-exploit strategies for IM problems in dynamic networks. Here the propagation probability is taken as 0.01, and we have evaluated the result for varying seed sets using the greedy approach.
- **INT** [131]: INT (Influential node tracking) algorithm using for influence estimation with propagation probability 0.01. The initial seed set S_0 is set up by the Greedy algorithm. Then by using the UBI algorithm, we calculate an upper bound of marginal gain and an upper bound of node replacement gain.
- **LPINT**: Our LPINT algorithm using ctRBM for link prediction to predict the upcoming snapshot of the graph G^{t+1} and then uses the above IM algorithms for

predicting the probable seed set S^{t+1} for efficient Influence Maximization in the dynamic social network.

2.6.2 Baseline Methods used for Multifeature Analysis based Link Prediction

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

- **Common neighbours (CN)** [27]: For link prediction, CN evaluates score based on common neighbours between pair of nodes.
- **Adamic-Adar (AA)** [27]: AA refines the counting of common neighbours by penalizing them with high node degrees. It is an extension of CN.
- **Jaccard Coefficient (JC)** [27]: JC calculates the similarity score for link prediction between a pair of nodes by using a Jaccard coefficient, which is defined as the size of the intersection divided by the size of the union of the common neighbours of the nodes.
- **Rooted Pagerank (RP)** [162]: RP evaluates node proximity from the root node to other nodes by performing a random walk, which starts from the root node.
- **AttriRank** [163]: It evaluates a score for each node by performing PageRank on the attributed network and then uses the product of scores of two end nodes for link prediction.
- **Streaming Link predIction for Dynamic attributEd networks (SLIDE)** [54]: SLIDE maintains and updates a low-rank sketching matrix to summarize all observed network data (structure/attributes) and further uses this matrix to predict the missing interactions in a dynamic network.

- **Local Naive Bayes based Common neighbour (LNBCN)** [164]: This method is based on the Naive Bayes theory and arguments that different common neighbours play a different role in the network and hence contributes differently to the score function computed for non-observed node pairs.

- **SemiGraph (SG)** [118]: This method of link prediction adopts the semi-supervised graph embedding approach. The learned embedding reflects information from both the temporal and cross-sectional network structures.

- **Subgraphs, Embeddings, and Attributes for Link-prediction (SEAL)** [49]: This method is based on GNN to learn general structure features from local enclosing subgraphs, embeddings, and attributes; it then uses this framework for the link prediction task. In SEAL, the hop number h is an important hyperparameter. Here, we select h only from $\{1, 2\}$.

- **Three-level Hidden Bayesian link Prediction (3-HBP)** [165]: The 3-HBP method uses a hidden Naive bayesian based algorithm and LDA topic modelling technique to predict the links using inferred interests of the pair of users.

2.6.3 Baseline Methods used for Context-aware Influence Maximization

We compare our proposed MINT algorithm with the following baseline methods.

- **Random**[59]: In this algorithm, randomly, any k users are selected as the influential users.

- **MaxDegree**[59]: In this algorithm, top k users with the highest out-degree are selected as influential users.

- **TIM** [83]: This algorithm uses topic-of-interest-based queries for efficient seed selection. It selects k seeds from social networks to maximize the topic-aware influence spread in the network.

- **MCIM** [84]: A set of influential nodes are selected as the initial core in the spreading process with the Technique for Order of Preference by Similarity to Ideal Solution

(TOPSIS) [140] method, such that the seed set featured maximal influence spread and minimal overlap.

- **IMUD** [82]: This algorithm aims to select a set of k seed nodes for initiating the information spread process, such that the seed nodes and their neighbours, collectively represented as S_Coverage (SC), are maximally interested in the contents of the marketing message.

2.7 Hardware and Software Used

All experiments of this thesis are conducted on a server machine running CentOS-7 with a Quad-Core 2.1 GHz Intel Xeon Silver 4110 processor and 64 GB memory. All the algorithms are implemented in the Python programming language.