International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.4 (2024) pp. 971-982
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

A Hybrid Probabilistic Graph Based Community Clustering Model for Large Social Networking Link Prediction Data

Rajasekhar Nennuri[1]*, S. Iwin Thanakumar Joseph[2], B. Mohammed Ismail[3], L.V. Narasimha Prasad[4]

[1]Research scholar, Department of computer science and engineering, Koneru Lakshmaiah education foundation, Vaddeshwaram 522302, AP, India
* **Corresponding Author Email:** rajasekharnennuri@gmail.com - **ORCID:** 0000-0001-5720-6358

[2]Assistant Professor, Koneru Lakshmaiah Education Foundation, Vaddeshwaram 522302, AP, India
Email: iwineee2006@kluniversity.in **ORCID:** 0000-0002-0452-4608

[3]P.A. College of Engineering Mangalore, Affiliated to Visvesvaraya Technological University Belgum.
Email: aboutismail@gmail.com **ORCID:** 0000-0003-4480-3801

[4]Professor, Institute of Aeronautical Engineering, Hyderabad
Email: lvnprasad@yahoo.com **ORCID:** 0000-0001-6514-1064

Article Info:

DOI: 10.22399/ijcesen.574
Received : 29 October 2024
Accepted : 31 October 2024

Keywords

Social network dataset,
Link Prediction,
Community Detection,
Dynamic clustering.

Abstract:

Dynamic community clustering is essential for online social networking sites due to the high dimensionality and large data size. It aims to uncover social relationships among nodes and links within the network. However, traditional models often struggle with community structure detection because of the extensive computational time and memory required. Additionally, these models need contextual weighted node information to establish social networking feature relationships. To address these challenges, an advanced probabilistic weighted community detection framework has been developed for large-scale social network data. This framework uses a filter-based probabilistic model to eliminate sparse values and identify weighted community detection nodes for dynamic clustering analysis. Experimental results demonstrate that this filter-based probabilistic community detection framework outperforms others in terms of normalized mutual information, entropy, density, and runtime efficiency (measured in milliseconds).

1. Introduction

In 1954, Barnes first used the term "social network" to characterize a social structure consisting of nodes connected by edges that symbolize various forms of interdependency [1]. To accommodate the demands of the user, OSNs are available in a range of sizes and forms. Some of the most popular services that people use on a regular basis are Facebook, Instagram, Twitter, Linkedin, Youtube, and Tumblr. Even if the services offered by different OSNs are intended for different reasons, like Online social networks (OSNs), encompassing platforms for networking, microblogging, video sharing, and more, all possess a set of fundamental features. The user base and the range of services provided by OSNs are expanding rapidly. These networks enable users to establish virtual connections with both familiar friends and strangers who share similar interests. Users are encouraged to share extensive information on OSNs for various reasons, such as the desire to connect with others, increase their number of connections, achieve recognition, trust within their affinity groups, and herd behavior. In today's digital landscape, sharing information is essential, and everyone, whether intentionally or unintentionally, contributes some data to the online environment. Intentional sharing occurs when users deliberately share information they deem appropriate for a specific audience and context. Social networking repositories are decentralized systems with no centralized control where each peer node exchanges and shares data across the network. Within the overlay network, each peer node is directly connected to a large number of peers. It is impractical to collect all of the data scattered across Social networking repositories into a centralized node or site and then use traditional data mining

techniques on it. In the client-server architecture, Social networking repositories have emerged as a promising mechanism for managing and sharing high-dimensional data[2]. Nodes communicating in large Social networking repositories can share and store data in a centralized manner in traditional client-server or centralized systems. These peer nodes act as information providers who don't want their clustered data or filtered information exposed to the distributed network. A centralized peer controls and performs clustering-based feature extraction schemes on a large collection of P2P node s in a distributed P2P system. For large collections of node sets in peer nodes, node clustering in Social networking repositories has been an active research field. One of the main goals of information retrieval research is to organize node s according to their congraph and content type. Due to the high dimensional data and its properties, traditional single pass node clustering algorithms do not produce the desired results. Furthermore, time and space complexity, as well as their incremental operation, are unappealing in a variety of situations involving large, multi-dimensional structured and unstructured node sets. As a node retrieval and visualization approach, node classification and clustering have been evaluated [3]. Node clustering techniques attempt to group node s together based on their common characteristics at the same time. Node s that are important to a specific concept will be grouped together in a unique cluster [4]. An automatic node clustering algorithm in any Social networking repository determines high quality node s more easily and efficiently at the peer end. The work's goal and purpose is to improve node clustering and feature extraction methods in order to make P2P node s in a distributed network easier to understand. Another model, a node clustering model based on k-trees, has been implemented [5]. The k-means clustering approach has been improved and approximated. The sparse representation problem in the node representation model is solved by using traditional k-means, which represents clusters in a hierarchical manner. They use a set of node s to compare the quality and efficiency of the Clustering Toolkit (CLUTO) algorithm. In the main memory, the tree structure allows for efficient space management. When dealing with high sparse vector spaces, the K-tree approach was originally implemented on high dense vectors, which causes efficiency and space complexity issues. On Wikipedia, an extended model was implemented as an ontology-based knowledge for node clustering [6]. The bag-of-words is used in most traditional clustering models. The bag-of-words model's main flaw is that it disregards the node 's contextual relationship between words or phrases. They used ontology to enrich the node representation model with background knowledge. Ontology is made up of three parts: attributes, concepts, and relationships between contexts. The main disadvantage of this approach is the ontology's limited coverage, which may result in data loss. For node clustering, the Harmony K-means approach has been proposed [7]. Despite the fact that most partition-based clustering approaches are designed for large node objects, they have two flaws that make them inefficient in many distributed environments. The first is sensitivity to node initialization, representation, and local convergence, and the second is global convergence. They used a Harmony Search-based Novel Harmony K-Means approach on multiple node clustering to solve these problems. Semantic, structural, and content characteristics are all present in XML data. Each feature is subdivided into a number of sub-features once more. In XML, clustering is done by selecting one feature at a time and ignoring the others in the meantime. As a result, the accuracy rate of the XML clustering algorithm has been compromised. Homogeneous and heterogeneous XML node s are divided into two categories. Homogeneous XML is defined as all XML node s originating from the same DTD[8]. It is determined not only by the structure, but also by the similarity of the contents. In the field of XML node clustering, a large number of proposals have been made. Traditional methods focus on either the structure or the content, but not both. As a result, all of these algorithms reduced clustering accuracy and quality. The model that takes into account both structure and content will have a higher accuracy and cluster quality rate. Both features are represented in a tree and a graph, and then a similarity index is calculated. Clustering algorithms have been plagued by a slew of issues for years, and these issues are divided into two categories: structure dimension and content dimension]. The structure dimension is calculated by comparing structural relationships between different elements. The content dimension, on the other hand, computes similarity between elements using textual contents. As a result, in order to achieve optimal performance and quality, clustering algorithms must consider both structure and content dimensions [8]. Furthermore, the results of clustering schemes differ depending on whether they are tree-based, path-based, or graph-based [9]. Similarity evaluation approaches differ depending on the clustering methods used. The clustering process uses the results of the similarity evaluation process as input. K-means, incremental clustering, fuzzy C-mean, and hierarchical agglomerative clustering approaches are some examples of such algorithms. [10] proposed the XCLS approach for combining XML node s into distinct groups based on structure dimension. For the

analysis of structured XML series, the level structure method is used. [11] demonstrated a new clustering approach that evaluates each and every existing path similarity to store data nodes in an XML storage system. The structure of XML node s is arranged as a tree in [12]. It also calculates similarity by measuring the distance between sub-trees. [13] developed another interesting method that implemented both the structure and content dimensions of XML node s. By using weights, two distinct similarity calculation approaches are merged together. [14] devised a new method known as SA-cluster (basically, it is a graph clustering method). The tree-edit distance method aids in the evaluation of the structure and styling sheet, whereas content necessitates the use of a graph similarity technique. [15] describes a new bitmap-based clustering technique, and the node s are indexed using various indexing techniques. [16] proposes an integrated approach for both homogeneous and heterogeneous node s. The new node s are either added to the newly created cluster or to the pre-existing clusters once the initial cluster is formed. [17] built a model that included the XML node s as well. Node Structure Model is the name of this model (DSM). [18] introduces a new clustering scheme that incorporates both structure and content dimensions in the XML node clustering process. The overall concept of their technique is to generate a rooted-ordered labelled tree using a tree-structural algorithm. To retrieve the tree's structural details, a structural-content data structure is used. This technique employs two approaches: nested-node reduction and repeated-node reduction. A non-leaf node with the same label as its parent nodes is known as a nested-repeated node. The path of a repeated node has already been traversed. Level structure and content representation are used to integrate structure and content dimensions. This algorithm is in charge of integrating XML nodes at all levels of the node . The name, ancestors, and content of each node are used to identify it. For these XML node arrangements, a new technique is proposed in [19]. The level structure algorithms are in charge of grouping XML nodes into levels and storing them as vector levels. In order to represent level structure, node tags are associated with integers. Every node in the rooted labelled tree is given an integer starting at the top and working its way down. It doesn't show the connections between nodes at different levels. A single level structure could potentially represent two different node s with the same nodes. This issue typically occurs in homogeneous node s or heterogeneous XML node s (where the files are originated from a common DTD). The names mentioned in abstracts may differ from the full-graph node 's representation. As a result, a new approach is proposed to address the shortcomings of the previous manual approach. The entire procedure is carried out automatically, taking into account the distribution of terms in both abstracts and full-graph node s. The Univariate scoring metrics play a significant role in the above-ranking process. In wrapper methods, the learning algorithm is used to extract the feature subsets without disturbing the classification mechanism. In contrast to both scenarios, the feature subset extraction and learning process are integrated with embedded methods. The mechanism of feature selection is combined with the classification process. The computation cost of embedded methods is less than wrapper methods. Examples of embedded feature selection approaches are decision trees and weighted Naive Bayes models. The Naive Bayes classifier is the most commonly used classifier in supervised learning, which is based on the Bayesian theorem. Statistical predictive models can be generated using this classifier. The Naive Bayes classifier presumes that the impact of an attribute on the target class is always independent of other attribute values. This is a probabilistic classifier that produces probability measure. In Naive Bayes classification, the prior probability, posterior probability and likelihood, were defined as follows: Initially, consider P is an instance whose target class is unknown and H is the hypothesis that Q belongs to class x. p(P/Q) = (p(P) P(Q/P))/(p(Q)) p(P/Q) is called likelihood which is the probability of sample P on the given hypothesis Q p(P) is called the probability of data sample P p(Q) is called is called the probability of data sample P P(Q/P) is called posterior probability which determines the probability of instance Q on the P. An instance-based machine learning model is the k Nearest Neighbour algorithm (kNN). It is simple to comprehend but extremely effective. The key idea behind this method is that it uses the closest training instances in the feature space to classify different objects [20]. The algorithm locates the k instances that are the most similar to the predefined instance. It comes to a conclusion about its class label by determining the most common target class label among the given training data that has the shortest distance between the query and training instances. The distance is defined by the distance metric. The supervised learning process is the learning process of the predicted labels that are already known. Unsupervised learning is performed when the class labels are unknown, for example, clustering. Feature selection is one of the most data mining activities. Feature selection is used to reduce the size of data, search for correlations, normalize data, and remove outliers. Including methods such as data cleaning, integration, conversion, and reduction, it includes several procedures. ML utilizes a relatively limited

amount of human involvement, as well as pre-programmed automatic techniques that have been shown to reduce human biases. The Univariate scoring metrics play a significant role in the above-ranking process. In wrapper methods, the learning algorithm is used to extract the feature subsets without disturbing the classification mechanism. In contrast to both scenarios, the feature subset extraction and learning process are integrated with embedded methods. The mechanism of feature selection is combined with the classification process. The computation cost of embedded methods is less than wrapper methods. Examples of embedded feature selection approaches are decision trees and weighted Naive Bayes models. The Naive Bayes classifier is the most commonly used classifier in supervised learning, which is based on the Bayesian theorem. Statistical predictive models can be generated using this classifier. The Naive Bayes classifier presumes that the impact of an attribute on the target class is always independent of other attribute values. This is a probabilistic classifier that produces probability measure. In Naive Bayes classification, the prior probability, posterior probability and likelihood, were defined as follows: Initially, consider P is an instance whose target class is unknown and H is the hypothesis that Q belongs to class x. p(P/Q) = (p(P) P(Q/P))/(p(Q)) p(P/Q) is called likelihood which is the probability of sample P on the given hypothesis Q p(P) is called the probability of data sample P p(Q) is called is called the probability of data sample P P(Q/P) is called posterior probability which determines the probability of instance Q on the P. An instance-based machine learning model is the k Nearest Neighbour algorithm (kNN). It is simple to comprehend but extremely effective. The key idea behind this method is that it uses the closest training instances in the feature space to classify different objects . The algorithm locates the k instances that are the most similar to the predefined instance. It comes to a conclusion about its class label by determining the most common target class label among the given training data that has the shortest distance between the query and training instances. The distance is defined by the distance metric. The supervised learning process is the learning process of the predicted labels that are already known. Unsupervised learning is performed when the class labels are unknown, for example, clustering. Feature selection is one of the most data mining activities. Feature selection is used to reduce the size of data, search for correlations, normalize data, and remove outliers. Including methods such as data cleaning, integration, conversion, and reduction, it includes several procedures. ML utilizes a relatively limited amount of human involvement, as well as pre-programmed automatic techniques that have been shown to reduce human biases.

2. Material and Methods

Meta heuristic optimization techniques are based on exploration and exploitation. Exploring is critical to ensure search over every part of the solution space to provide an accurate global optimum, and exploitation is key to finding better solutions by applying local search[21]. Different kinds of strategies such as evolutionary strategy, information transition and social behavior are implemented to ensure that the whole population moves towards the global optimum iteratively and protect from falling in the local optimum. Many authors have used meta-heuristic techniques to select the best feature subset to maximize their model's maximum effectiveness. In modern days, researchers are using hybrid meta-heuristic techniques to improve the mixing of features of two or more meta-heuristic approaches in their work. As a result, the focus of the analysis is on the structural regularities in a static network's graph structure. Dynamic structure mining, on the other hand, is concerned with data that changes over time. In this case, the analysis is aimed at identifying patterns of change in the social network as time passes. Furthermore, the study of the social network can be socio-centric, in which the entire set of relationships between actors is taken into account for analytical purposes, or ego-centric, in which the focal actors (called egos) are identified and their corresponding relationships with connected nodes are taken into account. In general, community or group detection in OSNs is based on analysing the social network structure to identify individual nodes in the network that are more closely related to each other than to other related groups of users [22]. Intra-communities (users or nodes belonging to the same community) are more likely to be connected or associated than inter-communities as a result of their discovery (users or nodes belonging to different community). This type of grouping aids in making further assumptions about users in the network, such as their likes and interests, tastes, and future activities. This, in turn, will aid in determining the likelihood of which products he or she will purchase, which songs or movies he or she will watch, which services he or she will be interested in, and so on. It's worth noting that OSNs are rapidly expanding, with millions of users participating in them. As a result of the network's massive growth, the question of which network users can be trusted and which are the distrusted ones who may cause a disruption in the near future arises. Building such a "reputation system" in a social network has become critical in order to avoid any kind of malicious or harmful

online activity. As a result, trust and distrust prediction in OSNs is an important research area being investigated in this field of social network mining in order to detect and/or prevent malicious activities from occurring via the network. The users' role in OSNs is implicitly understood to be the most important in the development and success of these networks. As a result, studying user behaviour or interactions in these networks is an important research area that is gaining traction. Clicking on a specific advertisement, selecting the like option for an image, accepting a friend's request, joining a group or discussion forum, dating with a person, and so on are examples of human behaviour. Studying the mood or behaviour of users on a social network aids in the network's growth and success. In viral marketing, precise models of user behaviour in OSNs are also critical. Despite the numerous advantages, however, measuring such related user activities has received little attention thus far. Two types of data must be studied in this context: "clickstream" data, which is generated from all visible user activities and interactions, and "silent" data, which is generated from silent user actions such as viewing a photo or browsing another user's profile page. Furthermore, the analysis' goal itself may be tainted with uncertainty. For instance, the information request may be ambiguous and unclear at times. Because data is frequently incomplete, a lack of information is also a source of uncertainty. Only a portion of the network structure is provided, as is common in real-world social data; typically, not all of the links are known. Furthermore, because there are multiple sources of social network data, there may be inconsistencies and conflicts. When collecting social data from various online social media sites such as Facebook and Linkedin, for example, we may encounter inconsistencies. A user can, for example, put different birthday dates or hometowns on both networks. Svenson (2008) drew attention to the issue of representing uncertainty in social network analysis and offered some potential solutions. According to Svenson, there are two types of uncertainty in social networks: I We can be unsure whether a particular node is distinct or not, i.e. whether two social entities are the same or not, and (ii) we can be unsure whether a link exists between two nodes. Should we assume that person X and person Y are related because they attended the same college? Furthermore, intelligence data obtained through signals intelligence is subject to natural uncertainty. For example, information provided by a camera about a specific person seen talking to a suspicious person may not be able to confirm the real identities of the individuals or that they actually conversed. To deal with uncertainty, Svenson proposed combining Monte Carlo simulation, modelling via random set, and Bayesian analysis. With this in mind, researchers must be able to manipulate ambiguous information and formulate and test hypotheses based on the data available. Uncertain information must be represented and reasoned with in analysis frameworks. According to Adar and Re (2007), collecting social network structure, as well as the shift in scale, comes with a higher degree of imprecision, which must be considered when using social network analysis techniques. The authors proposed using probabilistic databases to manage and manipulate uncertain social data. The majority of the time, uncertainty is linked to the data being analysed. Direct observations and questionnaires were used to collect data manually. As a result, data sets were typically small, and social entities were generally well-known. However, due to the emergence of online networking applications and the development of scalable databases, current data are extremely large in comparison to previous techniques. This prompted researchers from various fields to investigate the properties of large-scale social data sets. Despite this, little attention has been paid to examining uncertainty in social networks. Social network analysis (SNA) is the study of a social network's structure, based on the idea that the pattern of social connections or ties formed within the network contains vital information for the network's nodes. Content-rich Facebook, which is derived from explicit social interactions, or Instagram, the photo-sharing service, which allows content sharing via networks, have given birth to a remarkable outburst of network-centric data. In general, social networks are a network of interactions in which nodes represent actors or users and edges represent interactions or relationships between them. The linkage structure between nodes in a network, demographic or other feature-based information of nodes, and product ratings are all included in social networks datasets. The link prediction problem is studied in multiplex networks in [23], with two platforms - the Twitter network, a microblogging site, and Foursquare, a location-based social network - being used for experimental evaluations. Feature-based classification, matrix factorization models, kernel-based models, and probabilistic graphical models are some of the other major link prediction techniques [24]. Each pair of nodes is labelled as positive (if there is a link connecting the two nodes) or negative (if there is no link connecting the two nodes) in feature-based classification (if there is no link connecting the two nodes). Several factors, such as model complexity, prediction accuracy, scalability, and time complexity, can be used to compare all of these techniques. A link between each pair of nodes in a social network is assigned a probability value in a

probabilistic graph model, such as a transition probability in a random walk or topological similarity. Link prediction can also be thought of as a matrix completion problem, and research in this area can be expanded to look into the matrix factorization method as a way to explain the problem. The link prediction problem has also been investigated in the context of relational data [25], where graph mining is less important. Link prediction in relational data takes into account real-world domains that are highly structured and involve multiple types of related entities, which may be difficult to study using machine learning. In this regard, a Relational Markov network is a method that can be used to solve the problem of link prediction. For any social network data, a link predictor can be built that studies links in the graph and converts graph features into flat features. [26] discusses a review of standard community detection algorithms. Several traditional community detection algorithms, such as hierarchical clustering [27], spectral clustering [28], and partitional clustering [29], have been developed over time. This work describes a sophisticated approach to community detection in online social networking datasets using a combination of data filtering and machine learning techniques. Here's a more detailed breakdown of the methodology.

Data Pre-processing and Filtering

Pre-processing Algorithm: Initially, the input data is subjected to a pre-processing algorithm. This step is crucial for cleaning the data and making it suitable for further analysis.

Hybrid Data Filtering Approach: Given that the social networking datasets contain numerical attributes, a hybrid data filtering approach is used to handle sparse values. This ensures that the data is more robust and reliable for subsequent analysis.

Community Detection Using Clustering:

Density Community Clustering: In the second phase, the pre-processed and filtered data is fed into a machine learning model that employs a density-based clustering approach. This method is effective for identifying densely packed groups of data points, which correspond to communities within the social network. Probabilistic Clustering Approach: A probabilistic clustering method is applied to the filtered dataset to detect communities. This approach allows for a more flexible and accurate identification of community structures within the data.

Intra-cluster and Inter-cluster Analysis:

Intra-cluster Variance: Within each cluster, data points are analyzed for intra-cluster variance, which measures the variability of data points within the same cluster. Inter-cluster Variation: The differences between clusters, or inter-cluster variation, are analyzed to understand the variations in data attributes between different communities.

Machine Learning on Sensitive Attributes:

Profile Information Analysis: Finally, the machine learning approach is applied to sensitive attributes, such as users' profile information, to gain deeper insights into the community structures. This step helps in understanding how different attributes contribute to the formation and characteristics of communities. The overall proposed framework is presented in Figure1. Instead of traditional community detection algorithms, proposed machine learning approach analyse the profile attribute using the community detection approach. In Algorithm 1, every data instance from the distributed data source is pre-processed using the min-max measure and max probabilistic measure. An attribute of either nominal or numerical type is pre-processed for each instance in each data source. If the attribute type is continuous, the missing value of the numerical attributes is filled up using equation (1). Similarly, the conditional probability of the attribute is used to replace the sparsity values for nominal or categorical qualities.

Algorithm 2: Weighted Probabilistic Community detection model for social networking graph data

Probabilistic weighted measure for community detection:

In this study, critical relational graph nodes for the community detection process are identified using a hybrid probabilistic weighted measure based on their attributes. The central tendency of the weighted measure between these attributes is presented as

$$\lambda 1 = \frac{|(\mu_{A1} - \mu_{A2})|}{2.\sqrt{\min\{\sigma_{A1}, \sigma_{A2}\}}} \quad \text{-----(1)}$$

Where M_{A1} is the average of the attribute A! wrt class samples

M_{A2} is the average of the attribute A2 wrt class samples

The maximized weighted probabilistic measure for the community detection is given as:

$$\lambda 2 = \text{Min}\{\frac{\text{Max}(\text{Prob}(A1/Cm))}{2.|\sum|A1||}, \frac{\text{Max}(\text{Prob}(A2/Cm))}{2.|\sum|A2||}\} \quad ..(2)$$

Maximized weighted probabilistic measure is given by : MPWM=T=max{ $\lambda 1$, $\lambda 2$ }

In the algorithm2, initially, all the data objects are filtered using the algorihthm1. These filtered data are used for data machine learning and community detection process.

Algorithm 1: *OSN Graph data pre-processing*

Input : Multi source datasets MD={D-1,D-2…D-n}, Attribute: $A\tau$,Max feature value M_x, Minimum feature value M_n, frequency of the maximum feature value that contains class c Max_c, frequency of the minimum attribute value that contains class c Min_c,

1: Load input datasets DS
2. For each training data D[i]
3. Do
4. For each record I[r]
5. Do
6. For each attribute in I[r]
7. Do
8. If($A\tau$ [I]==Continuous && $A\tau$ [I]== ϕ)
9. then
10. Replace $A\tau$ [I] using the eq .(1)
11. $$A\tau[I] = \frac{|A\tau[I] - (\mu_x(A\tau) + Min_n(A\tau)/2)|}{2.\sigma_{A\tau}((Max_c(A\tau) - Min_c(A\tau))} \quad \text{--(1)}$$
12. End if
13. If($A\tau$ [I]==Categorical && A[I]== ϕ)
14. Then
15. Replace $A\tau$ [I] using the eq.(2)
16. $$A\tau[I] = \frac{\sum_{i=1,m=1} Max\{(A\tau[i]/c_m)\}}{Prob(A\tau[i]/c_m)} \quad \text{-----(2)}$$
$$i = 1..n; m = 1..k(classes)$$
17 Done
18 Done
19 done

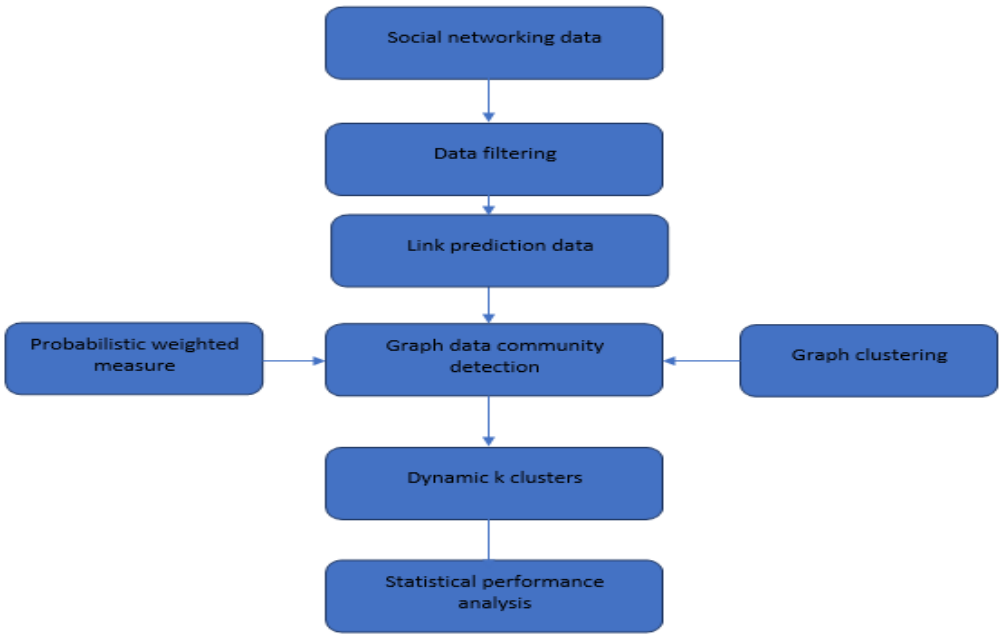

Figure 1: *Proposed framework*

Input: dataset D

Output: Filtered dataset D'

1. Read the input dataset.
2. To each numerical attribute A in the dataset D.
3. Apply algorithm 1 for data filtering.
4. Done
5. If the dataset contains heterogeneous attributes in the list FD[].
6. To each attribute s in FD
7. Do
8. Link prediction using model1.
9. Done
10. To each instance in the local community objects
11. do
12. For each isntance O_i in the KNN objects KNN[]
13. Do
14. For each instance O_j in IPG[] // where i!=j
15. Computing the Chebyshev distance N_m^k on the KNN objects.
16. Done
17. To each Chebyshev distance objects in the local density modelling, find the k nearest objects in the sorted as
18. N_m^k []=TopKNN(k);
19. Apply local density estimation probability on the filtered local objects.
20. To each reducer in the MR framework
21. do
22. Find the nearest density objects using the proposed probabilistic KNN method.
23. Construct initial graph clustering with k nodes as clusters.
24. Compute local and global density estimation by using the following weighted measures as

$$Dist_c = mean^K + \lambda_1 \cdot \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\phi_i^K / mean^K)}$$

Here, N is total number of filtered knn objects

ϕ_i^K is the average of the kth nearest neighbour to instance i.

$\phi_t^K = max_j\{KNN_i(\text{Dist}_{ij})\}$, and

$mean^K$ is the average of ϕ_i^K ,

computed as $\text{mean}^K = \lambda_1 \frac{1}{N}\sum_{i=1}^{N}\phi_i^K$

Prior estimation probability $=\kappa = \text{Prob}(I,C_k)=\text{Max}\{\text{Prob}(I/C_k$); k:k-nearest objects}

Pr*oposed* local density estiamtion is given as

$$\text{P}LDE(v_i) = \frac{\kappa}{\max\{\lambda_1,\lambda_2\}} \text{e}(-\frac{\| \log(v_{ij}) - Dist_c \|^2}{2\sigma^2})$$

25. Repeat this procedure till k graph clusters.
26. Done
27. Done

3. Results and Discussions

Utilizing third-party graph and similarity packages, experimental results are simulated within a Java framework. Four datasets, including the Zachary, dolphin, football, and Yelp datasets, were used to assess the performance of the suggested model. Figures 2, 3, and 4 demonstrate the loading of Facebook OSN datasets, various community clusters in the visualization form, and dynamic display of Facebook clusters for OSN. Various criteria are employed in this experimental investigation to assess the outcomes. On the training datasets, metrics like density, NMI (normalized mutual information), and entropy (variation of information) are utilized to assess the outcomes.

$$Density = \sum \left(e_{ii} - a_i^2 \right) \quad \text{..... (3)}$$

$$NMI(\mathrm{P}|\ \mathrm{Q}) = \frac{\mathrm{e}(\mathrm{P}) + e(\mathrm{Q}) - e(\mathrm{P},\mathrm{Q})}{(\mathrm{e}(\mathrm{P}) + e(\mathrm{Q}))/2}, \quad \text{.....(4)}$$

where Y represents the expected communities and X represents the original value. The related communities' entropy values are denoted by e(P) and e(Q).

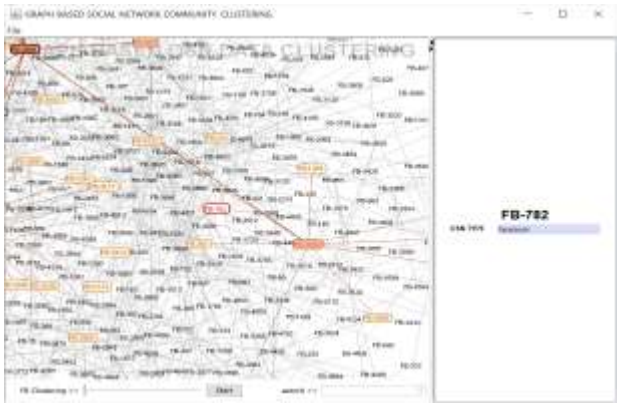

Figure 2: *Loading Facebook OSN dataset*

Figure 3: *Different community clusters in the visualization form*

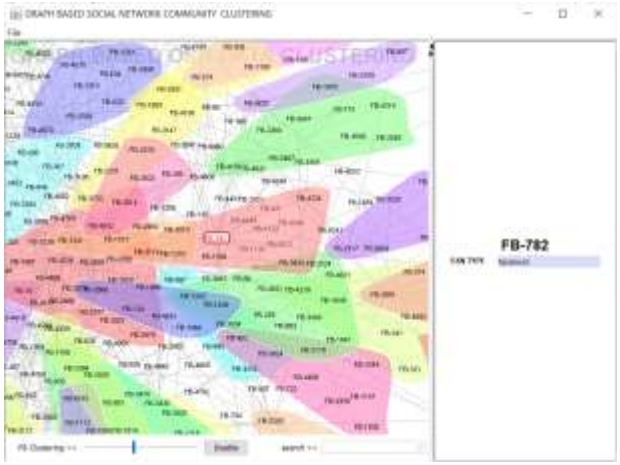

Figure 4: *Dynamic visualization of Facebook clusters for OSN*

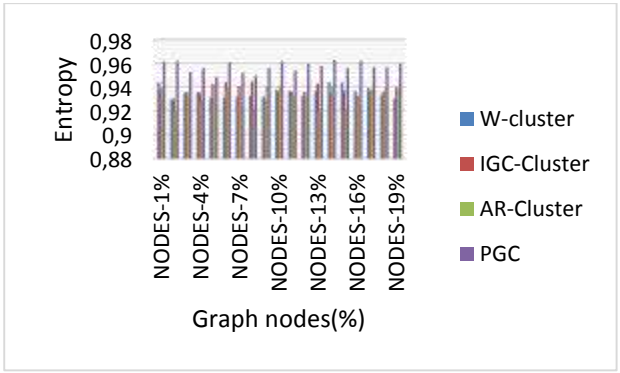

Figure 5: *Comparative analysis of proposed model Entropy value to existing models on facebook data*

Table 1. *Represents the comparative analysis of present probabilistic model to the conventional models on Facebook dataset.*

Sample Graph Nodes	W-cluster	IGC-Cluster	AR-Cluster	PGC
NODES-1%	0.937	0.94	0.925	0.964
NODES-2%	0.941	0.939	0.932	0.962
NODES-3%	0.934	0.937	0.943	0.95
NODES-4%	0.943	0.944	0.935	0.951
NODES-5%	0.935	0.933	0.931	0.95
NODES-6%	0.931	0.931	0.938	0.959
NODES-7%	0.941	0.94	0.926	0.951
NODES-8%	0.933	0.94	0.926	0.957
NODES-9%	0.934	0.941	0.919	0.956
NODES-10%	0.933	0.932	0.929	0.958
NODES-11%	0.933	0.934	0.931	0.959
NODES-12%	0.935	0.931	0.927	0.96
NODES-13%	0.93	0.93	0.933	0.964
NODES-14%	0.933	0.942	0.94	0.956

NODES-15%	0.945	0.937	0.926	0.952
NODES-16%	0.937	0.936	0.933	0.96
NODES-17%	0.936	0.933	0.93	0.958
NODES-18%	0.938	0.945	0.92	0.951
NODES-19%	0.939	0.932	0.935	0.96

As shown in the Table 1, proposed NML has better efficiency than the traditional models on facebook dataset. The NLM value represent the quality of the inter and intra community detection process on the facebook dataset. The comparison of the current probabilistic model with the traditional models on the Facebook dataset is shown in Figure 5. The suggested entropy value outperforms the conventional models on the Facebook dataset, as seen in Figure 5. The quality of the intra- and inter-community detection method on the Facebook dataset is represented by the entropy value.

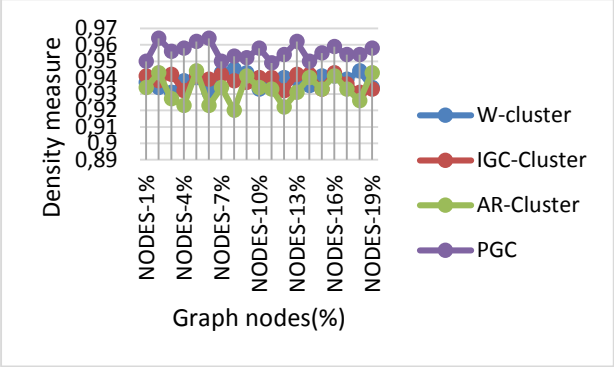

Figure 6: *Comparative analysis of proposed model density value to existing models on yelp data*

Figure 6 shows a comparison of the current probabilistic model with traditional models using the Yelp dataset. The suggested density value outperforms the conventional models on the Yelp dataset, as seen in Figure 3. The density value on the Yelp dataset indicates how well the inter- and intra-community detection procedure performed. In this table 2, the runtime(ms) comparison of proposed model to the conventional models are presented. From the table 2, it is noted that the runtime of the probabilistic machine learning model has better efficiency than the conventional models on the OSN datasets.

4. Conclusions

In the online social networking databases, dynamic community clustering is essential. Due

Table 2: *Performance of proposed runtime(ms) to conventional models on different facebook networkdata samples*

Sample Graph Nodes	W-cluster	IGC-Cluster	AR-Cluster	PGC
Test1%	4289	4117	4168	3523
Test2%	4778	3888	4148	3394
Test3%	4388	4473	3941	3282
Test4%	4694	4243	3892	3512
Test5%	4712	4072	4032	3504
Test6%	4073	4042	4175	3283
Test7%	4085	4217	4021	3374
Test8%	4138	4260	4055	3404
Test9%	4009	3966	3949	3468
Test10%	4070	4045	4161	3413
Test11%	4597	4170	4144	3462
Test12%	4158	4192	3925	3444
Test13%	4127	4422	3969	3473
Test14%	4298	4013	4019	3358
Test15%	4372	3896	3976	3344
Test16%	4523	3943	4132	3554
Test17%	4516	4351	4058	3288
Test18%	4724	3884	3995	3398
Test19%	4006	4080	4015	3388

to the static character of most traditional community clustering modes, they are only useful for non-link prediction approaches. An innovative probabilistic weighted based community discovery method is created on the massive social networking data to address these problems. To eliminate the sparse data and identify the weighted community detection nodes for dynamic clustering analysis, a filter-based probabilistic model is created in this model. According to experimental findings, the probabilistic community detection framework based on filters is more efficient in terms of density, entropy, normalized mutual information, and runtime (ms).

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.

- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1]M. Sattari and K. Zamanifar, (2018). A cascade information diffusion based label propagation algorithm for community detection in dynamic social networks, *Journal of Computational Science*, 25;122–133, doi: 10.1016/j.jocs.2018.01.004.

[2]X. Zhao, J. Liang, and J. Wang, (2021). A community detection algorithm based on graph compression for large-scale social networks, *Information Sciences*, 551;358–372, doi: 10.1016/j.ins.2020.10.057.

[3]R. George, K. Shujaee, M. Kerwat, Z. Felfli, D. Gelenbe, and K. Ukuwu, (2020). A Comparative Evaluation of Community Detection Algorithms in Social Networks, *Procedia Computer Science*, 171;1157–1165, doi: 10.1016/j.procs.2020.04.124.

[4]Z. Liu and Y. Ma, (2019). A divide and agglomerate algorithm for community detection in social networks, *Information Sciences*, 482;321–333, doi: 10.1016/j.ins.2019.01.028.

[5]N. R. Smith, P. N. Zivich, L. M. Frerichs, J. Moody, and A. E. Aiello, (2020). A Guide for Choosing Community Detection Algorithms in Social Network Studies: The Question Alignment Approach, *American Journal of Preventive Medicine*, 59(4);597–605, doi: 10.1016/j.amepre.2020.04.015.

[6]M. M. D. Khomami, A. Rezvanian, and M. R. Meybodi, (2018). A new cellular learning automata-based algorithm for community detection in complex social networks, *Journal of Computational Science*, 24;413–426, doi: 10.1016/j.jocs.2017.10.009.

[7]S. Ahajjam, M. El Haddad, and H. Badir, (2018). A new scalable leader-community detection approach for community detection in social networks, *Social Networks*, 54;41–49, doi: 10.1016/j.socnet.2017.11.004.

[8]X. Chen, C. Xia, and J. Wang, (2018). A novel trust-based community detection algorithm used in social networks, *Chaos, Solitons & Fractals*, 108;57–65, doi: 10.1016/j.chaos.2018.01.025.

[9]A. Rekik, S. Jamoussi, and A. B. Hamadou, (2020). A recursive methodology for radical communities' detection on social networks, *Procedia Computer Science*, 176;2010–2019, doi: 10.1016/j.procs.2020.09.237.

[10]M. Sattari and K. Zamanifar, (2018). A spreading activation-based label propagation algorithm for overlapping community detection in dynamic social networks, *Data & Knowledge Engineering*, 113;155–170, doi: 10.1016/j.datak.2017.12.003.

[11]V. Moscato and G. Sperlì, (2021). A survey about community detection over On-line Social and Heterogeneous Information Networks, *Knowledge-Based Systems*, 224;107112, doi: 10.1016/j.knosys.2021.107112.

[12]S. Aghaalizadeh, S. T. Afshord, A. Bouyer, and B. Anari, (2021). A three-stage algorithm for local community detection based on the high node importance ranking in social networks, *Physica A: Statistical Mechanics and its Applications*, 563;25420, doi: 10.1016/j.physa.2020.125420.

[13]X. You, Y. Ma, and Z. Liu, (2020). A three-stage algorithm on community detection in social networks, *Knowledge-Based Systems*, 187;104822, doi: 10.1016/j.knosys.2019.06.030.

[14]M. Naderipour, M. H. FazelZarandi, and S. Bastani, (2020). A type-2 fuzzy community detection model in large-scale social networks considering two-layer graphs, *Engineering Applications of Artificial Intelligence*, 90;103206, doi: 10.1016/j.engappai.2019.07.021.

[15]M. Qin, D. Jin, K. Lei, B. Gabrys, and K. Musial-Gabrys, (2018). Adaptive community detection incorporating topology and content in social networks☆, *Knowledge-Based Systems*, 161;342–356, doi: 10.1016/j.knosys.2018.07.037.

[16]M. Azaouzi and L. B. Romdhane, (2017). An evidential influence-based label propagation algorithm for distributed community detection in social networks, *Procedia Computer Science*, 112;407–416, doi: 10.1016/j.procs.2017.08.045.

[17]Y. Wang and X. Han, (2021). Attractive community detection in academic social network, *Journal of Computational Science*, 51;101331, doi: 10.1016/j.jocs.2021.101331.

[18]P. Pham, L. T. T. Nguyen, B. Vo, and U. Yun, (2021). Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks, *Information Systems*, 101771, doi: 10.1016/j.is.2021.101771.

[19]X. Li, S. Zhou, J. Liu, G. Lian, G. Chen, and C.-W. Lin, (2019). Communities detection in social network based on local edge centrality, *Physica A: Statistical Mechanics and its Applications*, 531;121552, doi: 10.1016/j.physa.2019.121552.

[20]R. Sharma and S. Oliveira, (2017). Community Detection Algorithm for Big Social Networks Using Hybrid Architecture, *Big Data Research*, 10;44–52, doi: 10.1016/j.bdr.2017.10.003.

[21]J. Fumanal-Idocin, A. Alonso-Betanzos, O. Cordón, H. Bustince, and M. Minárová, (2020). Community detection and social network analysis based on the Italian wars of the 15th century, *Future Generation Computer Systems*, 113;25–40, doi: 10.1016/j.future.2020.06.030.

[22]X. Li, G. Xu, and M. Tang, (2018). Community detection for multi-layer social network based on local random walk, *Journal of Visual Communication and Image Representation*, 57;91–98, doi: 10.1016/j.jvcir.2018.10.003.

[23]S. Guesmi, C. Trabelsi, and C. Latiri, (2019). Community detection in multi-relational social networks based on relational concept analysis, *Procedia Computer Science*, 159;291–300, doi: 10.1016/j.procs.2019.09.184.

[24]P. Chunaev, (2020). Community detection in node-attributed social networks: A survey, *Computer Science Review*, *3*7;100286, doi: 10.1016/j.cosrev.2020.100286.

[25]P. Chunaev, T. Gradov, and K. Bochenina, (2020). Community detection in node-attributed social networks: How structure-attributes correlation affects clustering quality, *Procedia Computer Science*, 178;355–364, doi: 10.1016/j.procs.2020.11.037.

[26]H. S. Pattanayak, A. L. Sangal, and H. K. Verma, (2019). Community detection in social networks based on fire propagation, *Swarm and Evolutionary Computation*, 44;31–48,doi: 10.1016/j.swevo.2018.11.006.

[27]Y. Du, Q. Zhou, J. Luo, X. Li, and J. Hu, (2021). Detection of key figures in social networks by combining harmonic modularity with community structure-regulated network embedding, *Information Sciences*, 570;722–743, doi: 10.1016/j.ins.2021.04.081.

[28]M. Xu, Y. Li, R. Li, F. Zou, and X. Gu, (2019). EADP: An extended adaptive density peaks clustering for overlapping community detection in social networks, *Neurocomputing*, 337;287–302, doi: 10.1016/j.neucom.2019.01.074.

[29]A. Kanavos, I. Perikos, I. Hatzilygeroudis, and A. Tsakalidis, (2018). Emotional community detection in social networks, *Computers & Electrical Engineering*, 65;449–460, doi: 10.1016/j.compeleceng.2017.09.011.