An Adaptive Density Peak Clustering with Swarm Intelligence Algorithm for Detection of Overlapping Communities in Social Networks

R. Suganthi[1] and Dr.K. Prabha[2*]

[1]Ph.D. Research Scholar, Department of Computer Science, Periyar University, Centre for PG and Research Studies, Dharmapuri, Tamil Nadu, India. suganthi121110@gmail.com

[2*]Assistant Professor, Department of Computer Science, Periyar University, Centre for PG and Research Studies, Dharmapuri, Tamil Nadu, India. drprabha@periyaruniversity.ac.in

Received: August 14, 2022; Accepted: October 07, 2022; Published: November 30, 2022

Abstract

Community identification is an important technique for the investigation of complex networks because it makes it possible to examine mesoscopic features that are often connected to the organisational and functional properties of the underlying networks. In social media, there are billions of vertices and a variety of connections, thus community identification is a widely acknowledged approach of addressing the problem of grouping users. However, traditional methods are insufficient because of this. For the purpose of analysing social networks, overlapping community identification is crucial. On networks with complicated weight distributions, the current overlapping community recognition techniques seldom provide good results. Communities of any form may be easily and precisely found using density peaks clustering (DPC). Nevertheless, it also uses the truncation distance, and therefore is unable to automatically determine where the cluster centre is located. In this research work, an Adaptive density peak clustering (ADPC) with Modified Dragonfly Optimization (MDO) a suggested algorithm will decide the communities in a social network in an adaptable manner. Initially, the preprocessing methods such as Stemming, Stop-words removal, and Tokenization by bigrams, 1-to-3 grammars, and the unigram are three distinct types of data formats. Two feature extraction filters: Word Embedding Feature Extraction, and Term Frequency-Inverse Document Frequency (TF-IDF). In the Adaptive density peak clustering (ADPC) with Modified Dragonfly Optimization (MDO) algorithm, the clustering process is completed by determining the clustering centre by the MDO after determining the new local density based on the new neighborhood connection. ADPC-MDO adds a unique distance function based on common nodes to estimate the distance between nodes and takes weights into account to handle both weighted and unweighted social networks. A technique based on transitive consensus matrix building is used to generate a consensus matrix, which provides representative information of all dendrograms. Given a social networks dataset with a complicated weight distribution, the results show that the suggested ADPC-MDO performs better.

Keywords: Community Detection, Sentiment Analysis, Adaptive Density Peak Clustering (ADPC) with Modified Dragonfly Optimization (MDO) Algorithm, Dendrograms and Consensus Matrix.

Journal of Internet Services and Information Security (JISIS), volume: 12, number: 4 (November), pp. 204-223
DOI: *10.58346/JISIS.2022.I4.015*

*Corresponding author: Assistant Professor, Department of Computer Science, Periyar University, Centre for PG and Research Studies, Dharmapuri, Tamil Nadu, India.

1 Introduction

Online social networks (OSNs) are now very common, and this popularity growth has been rapid. A variety of Web 2.0-powered social networking services, such as Facebook, LinkedIn, and Twitter, are now available for use by both individuals and corporations [1]. The facilitation of the flow of substantial amounts of information among their users is a common feature of all of these services, despite the fact that they differ widely in both form and function. Analysis of social networks has grown to be a study area of special interest due to the enormous volumes of data that pass through them and the relative simplicity with which they can be accessed. Microblogging is the primary function of Twitter, which enables users to send messages, known as "tweets," that are no longer than 140 characters in length [2]. In addition to URLs, these messages often include Twitter-specific features like hashtags, mentions, responses, and retweets. When referring another Twitter user in a message, you do so by prepending their username with the @ symbol. Hashtags are words or phrases that signify a particular subject. Mentions and responses are ways to refer to other Twitter users. A user is able to re-post a tweet that was originally made by another user via the use of the retweet function [3]. This action often indicates that the user endorses or is interested in the content of the original tweet. Anyone who follows a certain user is referred to as that person's "follower," and Twitter users may "follow" other people to receive the messages other people publish to the service. Twitter users have unfettered access to each other's material through the website or its API since users' Tweets are by default publicly available.

Community identification and sentiment analysis are two common areas of research for social networks even though there are many more. Communities, or distinct groupings of things, often arise inside social networks. The main objective of community identification is to identify these groupings and characterise their dynamic interactions, which might provide insightful data across many disciplines. Sentiment analysis, often known as opinion mining, is a method that enables users' attitudes and views to be automatically determined based on the information that they have contributed. Sentiment analysis has grown in popularity as a method for social network analysis in addition to community discovery [5]. Determine if a given text is subjective or objective is the first stage of a challenge that is frequently presented as a two-part process for sentiment analysis. Subjective /objective polarity is the term used to describe this kind of thinking. After it has been established that a piece of writing is open to interpretation, it may be categorised according to whether it conveys a positive or negative attitude, which is referred to as the text's "positive/negative polarity" or "PN-polarity". There have been many different techniques employed in recent years to do sentiment analysis on a wide range of data formats. A lexicon of words tagged with their SO or PN polarity is a frequent technique for sentiment analysis [6]. One such tool is the Senti Word Net lexicon, which has shown effectiveness in evaluating a variety of text documents, including news headlines and product evaluations, and has even been used for sentiment analysis across many languages. The polarity scores for the words in a text are added when utilising a lexicon like Senti Word Net, and a prediction is then made based on the result. However, lexicon-based polarity ratings may be employed more precisely when combined with a machine learning algorithm. While this simple strategy can still provide good results.

Probably, one of the most well-liked study areas pertaining to OSNs is community detection. Social networks are described as mathematical graphs, often known as "social graphs," which include edges that reflect links between people and vertices (or nodes) that represent actors inside the network to help with community discovery [9].Graph clustering methods allow for the differentiation of

communities of users within a social network. Grouping comparable or connected nodes together is the fundamental concept behind graph clustering. Partitions are often designed to minimise connections across clusters while increasing the number of connections (edges) inside a cluster. A number of real-world communities on the Twitter OSN, like the Indie Mac developer community, have been successfully found using graph clustering in prior research [8].On the other hand, the majority of these studies are predicated on the assumption that the communities that make up complex networks are disjoint or isolated. More specifically, they assume that each node only belongs to precisely one community that does not overlap with any other communities. However, there is often some overlap between the groups in various real-world networks. To put it another way, due to their varied roles within network systems, certain nodes in networks may belong to many communities. An individual may play multiple roles in a social network, for instance, and is often associated with a variety of social groupings, including friends, family, and co-workers [9].In actual complicated networks that include concealed information of many types, there is lots of space for research on the community detection issue.

Community discovery in social networks has been used with a variety of graph clustering techniques. The hierarchical graph representations known as dendrograms are used in hierarchical clustering. Since each level of the hierarchy functionally represents a clustering of the network at a different degree of granularity, these topologies allow for simple control of clustering resolution [10]. Despite being widely used; this approach has drawbacks that have sparked interest in other approaches. Newman and Girvan's alternate method, which finds clusters inside a network structure, measures something termed "betweenness". The number of shortest routes linking any two nodes that pass via a certain edge is the definition of the betweenness metric, which is applied to the edges in a graph. Success has been achieved in the development of edge betweenness-based algorithms that are effective on both actual and artificial networks. The DENGRAPH algorithm, which is yet another one, is a density-based clustering method that was developed expressly for the purpose of analysing the topologies of social networks [11]. DENGRAPH, which is based on the incremental version of the DBSCAN algorithm, offers a fundamental advantage for the research of social networks: the capacity to manage the ongoing, dynamic changes in the structure of these networks. The Label Propagation Algorithm (LPA), yet another method, is predicated on the notion of a rapidly spreading epidemic of sickness. This technique starts out by giving each node in the network a special designation. Each node is updated to contain the label that is shared by the majority of its neighbours for each iteration following the first one. Numerous research have shown the efficacy of this strategy. In this study, a method for adaptively determining the communities in social networks is developed that combines adaptive density peak clustering (ADPC) with modified dragonfly optimization (MDO).

Section 2 highlights some of the most current methods for detecting community and sentiment analysis in the remaining research effort. The methods suggested is described in section 3 in detail. Results and analysis are presented in section 4. The summary and follow-up efforts are covered in section 5.

2 Literature Review

The strategies for identifying sentiment-based communities using various clustering models are discussed in this section.

Inuwa-Dutse et al [12] introduced a multilevel clustering technique (MCT) that uses textual and structural data to find small, referred-to as microcosmic, groups. The effectiveness of the technique has

been evaluated experimentally using benchmark models and datasets. In order to identify coherent groups in social networks, this research adds a new component. This method provides a deeper knowledge as well as more clarity in articulating how low-level communities on Twitter form and function. Identification of such communities has advantages from an application standpoint, including improved recommendation. Wang et al [13] using semi-definite programming (SDP) optimization models, two techniques for finding sentiment communities are suggested. Great performances for the suggested strategies were shown by the experimental assessments. This research provides a useful framework for analysing SNS user sentiment for corporate decision-making. Awrahman et al [14] which stands for the Konstanz Information Miner (KNIME), was built in order to do statistical analysis on many major social networking websites. Twitter is a good example here since it has simple access to social network technologies and a wealth of data sources. The initial objective of this book is to provide readers a solid grasp of sentiment analysis and opinion mining based on current developments in the area. Next, several sentiment analysis components are described. The third part of this section outlines the different stages of the sentiment analysis process. In conclusion, many approaches that may be utilised for sentiment analysis are broken down, and the KNIME programme, which can function as a tool for sentiment analysis, is shown. Yang et al [15] grouped social posts and sentiment keywords using a technique known as self-organizing maps (SOM).The sentiment of a message was then ascertained using these linkages by using an association discovery approach to identify the relationships between a message and particular sentiment keywords. The accuracy of the findings beat that of a comparable technique after trials were done using gathered Twitter messages. It was suggested to use a SOM-based sentiment analysis technique. The suggested technique was used to infer the correlations between messages and keywords. The use of the association discovery procedure in sentiment analysis gives birth to the originality of this study.

Ding et al [16]a new trust model-based community discovery method known as the trust-based local overlapping community detection algorithm (TLCDA) has been suggested. TLCDA discovers communities via coarse-grained K-Mediods clustering by supplementing the conventional trust calculation with inter-node link strength and similarity in social networks. Our analysis of actual social networks demonstrates that the TLCDA-detected communities have higher preference cohesion and meet the topological cohesion requirements. Reihanian et al [17] presented a general architecture for rating-based social networks with overlapping community discovery in particular. In order to identify relevant communities, this approach takes into account both the subjects that users are interested in and the information they provide (ratings).As a result, we will arrive to topical communities whose members share an interest in a particular subject and whose members' ability to form strong bonds depends on the degree to which their points of view are congruent. Zheng et al [18] increase the performance of recommender systems' accuracy, a resonant emotional interest community-based recommendation model has been presented. In order to build semantic and emotional user profiles, we first learn the weighted semantics vector and sentiment vector. After that, resonance connection is calculated to assess users' resonance relationships by integrating semantic and emotional aspects. Finally, a resonance community is identified to find a resonance group and provide customised suggestions based on resonance associations. According to experimental findings, the suggested model is more successful than conventional approaches in identifying emotive interests associated to semantics.

Deitrick et al [19] a network of Twitter users to construct a Walktrap algorithm. The main purpose of Microsoft Corporation's Twitter accounts, including this one, is to communicate with information technology professionals. Once community recognition is complete, word sentiment scores from the

well-known Senti WordNet lexicon are used to examine the sentiment in the tweets sent by each of the communities identified in this network. The value of integrating these two approaches is shown by the ability to explore sentiment data at several levels using the combination of sentiment analysis and community recognition. Yang et al [20] to create a fresh community model, it was suggested integrating social linkages, content/topics, and sentiment data. The experimental findings on two distinct kinds of real-world datasets show that our model is capable of identifying communities with various topic-sentiment distributions in addition to achieving performance that is equivalent to that of a state-of-the-art community model. Shiet al [21] suggested a cutting-edge method to find overlapping groups. The suggested approach is based on link clustering instead of the more common node clustering used in traditional algorithms. Nodes inherently belong to many communities as a result. The method clusters on connections using genetic operation. The number of communities may be calculated automatically, and a useful encoding scheme is devised. The suggested algorithm's efficacy and efficiency have been verified in experiments using both artificial and actual networks. Gupta et al [22] developed a method for detecting overlapping communities based on rough set theory ideas and fine-grained knowledge about linkages. The first step is to create initial link subgroups using neighbourhood connections around each pair of nodes. The link subsets are then repeatedly calculated until convergence using the limited linkage upper approximation. Using the concept of reciprocal link reciprocity, the upper approximation subsets acquired at each iteration are limited and combined. The proposed algorithm's efficacy is shown by experimental findings on 10 real-world networks and by comparison with cutting-edge community discovery techniques.

3 Proposed Methodology

In this research work, an ADPC with Modified Dragonfly Optimization (MDO) algorithm is proposed to adaptively determine the communities in social network. Initially, the preprocessing methods such as Stemming, Stop-words removal, and Unigram, bigrams, and 1-to-3 grammes are used as three separate data forms for tokenization. Term Frequency-Inverse Document Frequency (TF-IDF) and Word Embedding Feature Extraction are two feature extraction filters. In the Adaptive density peak clustering (ADPC) with Modified Dragonfly Optimization (MDO) algorithm, to finish the clustering process, the MDO determines the clustering centre after determining the new local density based on the new neighborhood connection. ADPC-MDO adds a unique distance function based on common nodes to estimate the distance between nodes and takes weights into account to handle both weighted and unweighted social networks. A technique based on transitive consensus matrix building is used to generate a consensus matrix, which provides representative information of all dendrograms.

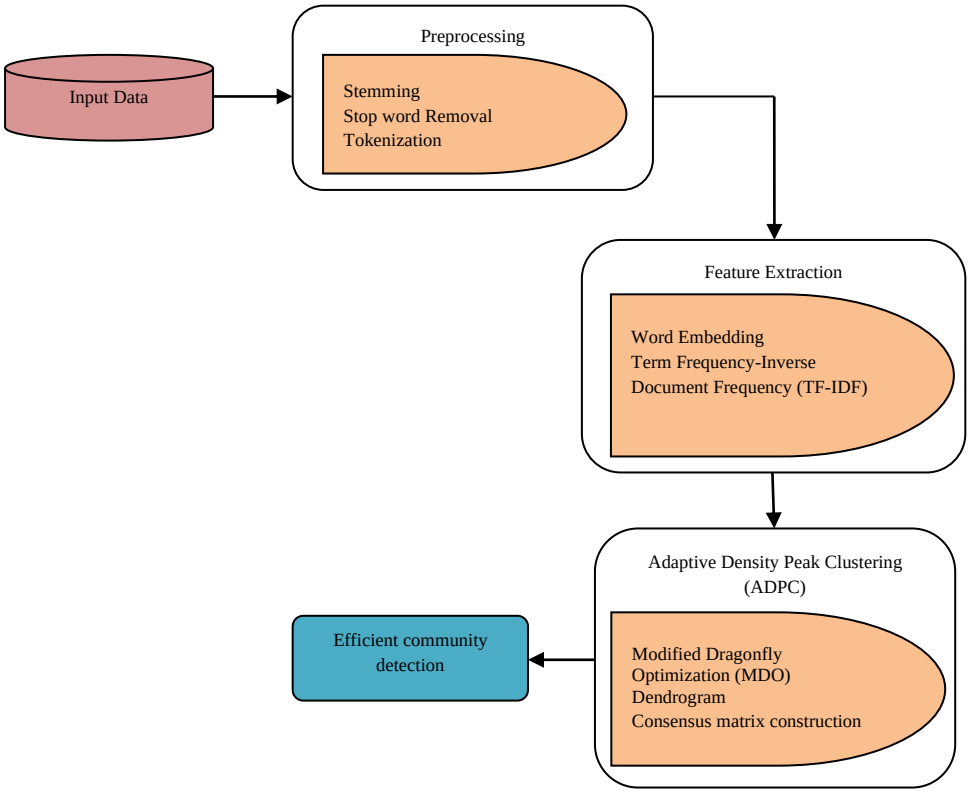

Figure 1: The process of the proposed Adaptive Density Peak Clustering with Modified Dragonfly Optimization (MDO) Algorithm Approach for the Community Detection

3.1. User Discussion Model

We provide a directed user web graph G in this work as a model of user conversation.

$$G_D = G(V, E) \quad (1)$$

where V = (V_G,V_R, V_{GR})includes all participants in the conversation at hand, $V_G$the participants in the conversation who alone contributed content D, $V_R$are the individuals who have solely responded in reaction to the material in the conversation D, and $V_{GR}$are those who have not only contributed material but also replied to other users' work [23]. In addition, E refers to the relationships between debate participants that are represented by likes, comments, or reposts. Figure 2 shows a visual depiction of a networked conversation broken down by kind of involvement in accordance with (1).

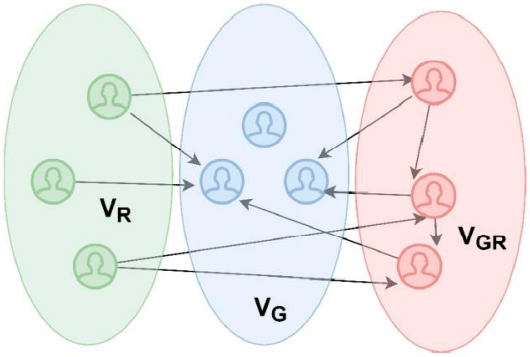

Figure 2: Visual Representation of the Structure of a Networked Discussion, by User Type.

The assumption here is that user A from$V_G$posted a message on social media about a particular incident, and user B from $V_R$liked, commented on, or reposted this post. An unweighted directed graph will therefore have a link B - A. Consequently, in this study, analysing (1)—that is, locating sub-structures in a discursive community—is the problem of searching for communities in social graphs.

A critical stage in sentiment analysis is data preparation, which may enhance the number of instances that are successfully identified by using the optimal pre-processing techniques. This study examines the importance of text pre-processing in sentiment analysis, the expanded evaluation of sentiment polarity classification techniques for Twitter text, and the experimental findings showing that feature extraction and representations may improve categorization accuracy. Three distinct data representations—bigrams, 1-to-3 grammes, and unigrams—as well as two feature extraction techniques were assessed as part of the current study's pre-processing approaches. TF-IDF and Word Embedding, respectively.

3.2. Preprocessing

For sentiment classification, preprocessing is a crucial stage in data preparation. Use WEKA's String to Word Vector filter to carry out the preprocessing. The following setups are supported by this filter:

- TF-IDF weighting scheme: The creation of feature vectors may be done using this strategy, which is a conventional method. TF-IDF is an abbreviation that stands for "term frequency-inverse document frequency." It is a numerical statistic that represents how significant a word is to a particular document within a corpus.
- Stemming: In accordance with certain grammatical norms, stemming algorithms remove the word's suffix.
- Stop-words removal: It is a method for removing meaningless and ineffective terms for the purpose of text categorization that are often used and have high frequency. This brings down the overall size of the corpus without sacrificing any essential information.
- Tokenization: In this situation, the documents are divided into words and terms, creating a word vector known as a "bag-of-words." To compare word bigram, unigram, and 1-to-3-gram, I also propose the N Gram Tokenizer.

A vast variety of characteristics are produced by the aforementioned pre-processing, many of which are unrelated to categorization.

3.3. Feature Extraction

Considering that it directly affects the model's accuracy, text feature extraction is vital for text categorization. A text is seen as a dot in an N-dimensional space for the purposes of feature extraction, which is based on the vector space model. In digital form, each dot's dimension corresponds to a different text characteristic. A keyword set is often used by feature extraction techniques. The feature extraction method determines the weights of the text's words based on these specified keywords, creating a digital vector that represents the text's feature vector. One strategy for extracting characteristics is the use of the Term Frequency Inverse Document Frequency (TF-IDF) weighted method [24]. This technique may provide feature vectors with a lot of dimensions if the text corpus is big, which may boost the likelihood of successful results. Different dimensionality reduction approaches may be used to resolve this. The reasoning behind this technique is that the shortened

dimensions need to reflect something that is conceptually closer to the ideas that are being discussed in the paper.

To begin, convert each tweet into a word vector form by making use of the TF-IDF weighting model and making use of the Snowball stemmer library and the Rainbow list for stop-words removal as fixed variables, while also experimenting with tokenization and feature extraction. Choose the term unigram, bigram, or 1-to-3-gram as the tokenization setting, and then compare the results. Following the term-weight vector representation of the tweets, feature extraction is used to determine if the removal of characteristics with weak character will improve classification accuracy. Tokenization and feature extraction techniques used together to create each vector model result.

A keyword is weighed in any context as part of the TF-IDF weighting system, and its value is determined by how often it occurs in the document. The relevancy of the term throughout the corpus is also checked by this information retrieval method. This feature extraction method's general process operates as follows. If a word w, a document alone, and a collection of documents D are all provided

$d \in D$, we calculate:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (2)$$

Here $w_{i,j}N$ documents make up the corpus, and is the weight for term i in document j, df_i j is the term frequency of term i in document j the same as the word frequency of term $jdf_i$is the phrase I in the corpus's document frequency. Whether a term is pertinent to a document's subject or not, the principle behind TF-IDF is that words in a text may be split into two classes: those that are unique and those that are not. One of the most significant drawbacks is the size of the feature set in TF-IDF for text data, which is equivalent to the size of the vocabulary across the entire corpus. This leads to enormous computation on weighing all of the words in the data set, which is one of the main reasons why this technique isn't widely used.

3.4. Adaptive Density Peak Clustering (ADPC)

ADPC with MDO algorithm, in order to complete the clustering process, the MDO determines the clustering centre after determining the new local density based on the new neighbourhood connection. ADPC-MDO adds a unique distance function based on common nodes to estimate the distance between nodes and takes weights into account to handle both weighted and unweighted social networks.

3.4.1. Density Peak Clustering

DPC is predicated on the idea that cluster centres differ from their neighbours in terms of density and are located a distance that is disproportionately far apart [25]. The local density ρ and separation distance are its two constituents δ. The local density may be determined in a number of ways ρ. Using the cutoff distance stated in eq is one method (3)

$$\rho_i = \sum_j \chi(dist(i,j) - d_c) \quad (3)$$

$$\chi(x) = \begin{cases} 1\ x \le 0 \\ 0\ otherwise \end{cases} \quad (4)$$

Where $\rho_i$is the neighbourhood node densityv_i, $dist$ (i, j) the space among$v_i$and v_j, and $d_c$is the splitting distance. a node's node density$v_i$the number of nodes whose distance to the centre of the computed using this technique is equal to$v_i$is less than d_c. Eq. (5) illustrates a further use of the

Gaussian kernel. The contribution that node v_j makes to the density of v_i increases with the distance between them.

$$\rho_i = \sum_j exp\left(-\left(\frac{dist(j,j)}{dc}\right)^2\right) (5)$$

Eq. (6) is used to get the separation distance δ_i. The shortest distance between a node with a greater density $v_i$and any others is δ_i.

$$\delta_i = \min_{j:\rho_j>\rho_i} dist(i,j) (6)$$

Following calculation ρand δ The decision graph will display a choice for each data node. as a δ vertical axis as well as a ρ horizontal axis. This graph's observation will be used to determine which data nodes should serve as the cluster centres. Last but not least, the closest and highest density data node to each of the remaining data nodes will be allocated to that cluster.

3.4.2. Adaptive Density Peaks Clustering for Overlapping Community Detection in Social Networks

The work described below is mostly focused on extending DPC such that it can detect overlapping communities in social networks that are both weighted and unweighted: (1) Create a function for measuring the distance between nodes in social networks, and call it the distance function. (2) Describe how to use adaptive cluster centre selection with linear fitting. (3) Make adjustments to DPC's distribution plan so that communities may overlap. After that, we'll go through each of the works that make up the method's foundation in turn.

a) Distance Function

The neighbouring matrix is often used as the social network's input, from which it is possible to derive the connecting information, including the connections and weights between links. The neighbouring matrix's components for an unweighted network are either 1 or 0. A connection exists if the value is 1 and not if the value is 0.The components for a weighted network are nonnegative real numbers that indicate connection weights [26]. It is necessary to convert the neighbouring matrix to the distance matrix in order to use density peaks approaches in social networks. Setting connecting weights' reciprocals as distances is one straightforward concept. The neighbouring matrix, however, will have an excessive number of zero values since social networks are often sparse. Consequently, the distance matrix will include an excessive number of infinite values. In this study, a new distance function that is generalised for weighted networks as well as unweighted networks is defined.

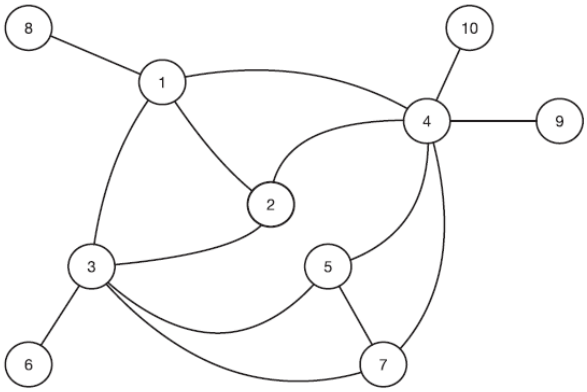

Figure 3: A Sample Network

Graph theory applied to social networks$G = (V, E_w)$, where V represents the collection of nodes that make up the users and $E_w$consists of the collection of user edges. A_{ij} is the connecting weight between node v_i and node v_j, and it is used to generate a n n neighbouring matrix for a network with n nodes. We use common nodes instead of just the two nodes that make up the pair to determine how far apart they are, as inspired by HOC Tracker. Two factors are involved. One is that despite the fact that they do not now engage in any explicit contacts, two individuals may one day become friends via their shared friends. The second factor is that two individuals will have more opportunities to engage with one another if they have a large number of friends in common. As a result of their shared friends, they will get closer. Here, we use the network shown in Figure 3. Because of their shared friends, nodes 5 and 7, they shouldn't have an infinite distance between them even if nodes 3 and 4 do not explicitly communicate with one another. Therefore, there will eventually be a connection between them. Because node 3 and node 4 are common friends and will improve the likelihood of their contacts, we take their mutual friends into account when estimating the distance between nodes 1 and 2.First, we compute *c c* (*i j*) the separation between v_i and v_j, the strength of their relationship as a result of the shared nodes.*cc*(*i*,*j*) is defined as Eq. (5), where v_{ij}a group of shared nodes between$v_i$and v_j, $v_i$and $w_{ipj}$is the minimum of $A_{ip}$and A_{jp}, *maxw* is the highest possible weight throughout the whole network, while r represents the weight distribution, $t \in [0, 1]$ is a factor determining how much a common node will increase$v_p$affects the node-to-node connection strength$v_i$and v_j, η is a little positive amount to prevent the numerator from becoming zero. Increasing thew_{ipj}, the contribution of connection strength increases. It implies that v_p will contribute more to their link strength the more often they interact with common node v_p.

$$cc(i,j) = \sum_{p \in V_{ij}} W_{ipj} * \exp\left(-\left(\frac{W_{ipj} - maxw}{r*t+\eta}\right)^2\right) (7)$$

The connection strength between the two$v_i$and $v_j$with Eq. (6). It is indicated that the connection strength is$ls(i,j)$. V_{ij} is the collection of nodes that both $v_i$and v_j in the network connect to that are shared by both of them, and $|V_{ij}|$ is how many nodes they share. I_i and $I_j$are, for $v_i$and v_j respectively, their respective outgoing connections' total weights. Link strength among $v_i$and $v_j$depends on more than just the contributions of their related nodes$cc(i,j)$, nonetheless, it is also impacted by the number of common nodes that they share. More interactions and a closer distance result from more frequent nodes, as was previously described. Remember that a close distance does not necessarily imply a significant direct connection weight. In the event that, for example, there are two people who interact frequently with one another, but both of them also interact frequently or even more frequently with their other friends, then it is impossible for us to simply conclude that these two people are sufficiently close due to the high direct linking weight between them. Instead, we should think about the proportion of their contacts that were directly caused by one another. Link strength increases with increasing fraction.

$$ls(i,j) = \frac{(cc(i,j) + A_{ij}) * (|v_{ij}| + 1)}{\min(I_i, I_j)} (8)$$

Eq. (9) is then used to determine the distance between nodes v_i and v_j, making ensuring that the denominator is not zero and that is a non-negative real value.

$$dist(i,j) = \frac{1}{ls(i,j) + \varepsilon} (9)$$

The distances between isolated nodes and any other nodes will be automatically adjusted to the greatest distance, it should be noted$1/\varepsilon$ instead of calculating. The neighbouring matrix *A* might be used with the distance function to generate the distance matrix *D*.

b) Modified Dragonfly Optimization (MDO) Algorithm

A population-based optimizer with recent success is DA. The DA algorithm is built on the dragonfly's movement and hunting patterns. Static swarms are a kind of hunting strategy where all swarm members may fly in close formation over a constrained area to find food sources. The migratory method used by dragonflies is thought to be dynamic swarming [27]. The swarm may move because the dragonflies are anxious to take flight in larger groups at this phase. Figure 4 displays static and dynamic groupings. Additionally, the operators of DA accomplish two key ideas in other swarm-based methods: intensification, which is prompted by the dynamic swarming activities, and diversification, which is inspired by the static swarming activities.

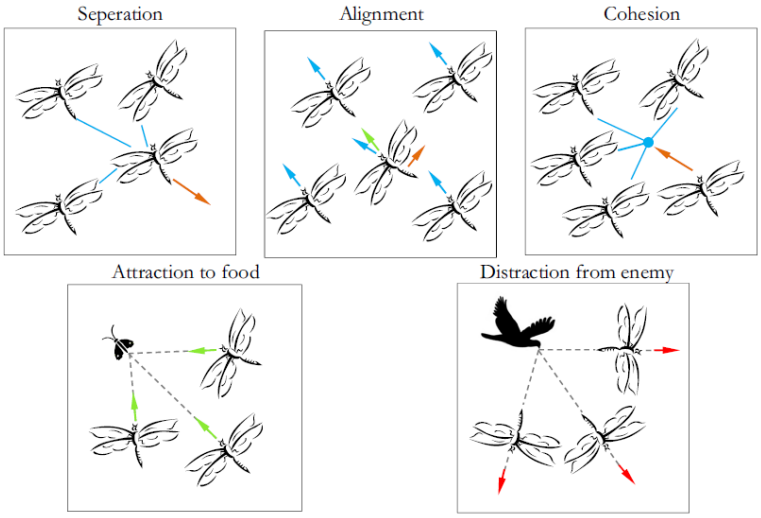

Figure 4: Dynamic and Static Dragonflies.

DA, where X is the position vector, X j is X's neighbour at position j, and N is the size of the neighbourhood, exhibits the following five behaviours:

Separation: To distinguish themselves from other agents, dragonflies use this tactic. The following steps make up this procedure:

$$S_i = \sum_{j=1}^{N} X - X_i \quad (10)$$

Alignment: Demonstrates how an agent will adjust its velocity to be the same as the velocity vector of other dragonflies that are near to it. The mathematical representation of this idea is given by the equation (10): where Vj is the velocity vector of the j-th neighbour:

$$A_i = \frac{\sum_{j=1}^{N} V_i}{N} \quad (11)$$

Cohesion: Demonstrates individuals' propensity to migrate toward the closest bulk centre. This stage is described as follows in (12):

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \quad (12)$$

Attraction: shows how members tend to go in the direction of the food supply. Based on (13) where Floc is the location of the food supply, the attraction tendency between the food source and the i-th agent is carried out:

$$F_i = F_{loc} - X \quad (13)$$

Distraction: shows how dragonflies tend to stay away from a potentially dangerous adversary. Where Eloc is the adversary's location: The enemy gets distracted by the i-th dragonfly in accordance with (14)

$$E_i = E_{loc} + X \quad (14)$$

The fittest agent yet discovered is used to update the fitness of the food supply and position vectors in DA. Additionally, depending on the worst dragonflies, the fitness values and locations of the opponent are computed [28]. Due to the existence of this information, DA will be able to move closer to solution spaces that include more promising regions, therefore avoiding areas that do not contain such regions. According to two rules, the position vector and the step vector, the position vectors of dragonflies are updated (X). The step vector is computed as shown in (15), and it represents the direction of the dragonflies' motion.

$$X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + wX_t \quad (15)$$

Where the weighting vectors for various components are represented by the letters s, w, a, c, f, and e. In Eq. (16), where t is the number of iterations, the position vector of members is calculated:

$$X_{t+1} = X_t + X_{t+1} \quad (16)$$

Different types of local and global searches may often be provided by the DA during optimizations using separation, alignment, and cohesion. Additionally, the dragonflies may benefit from the best solutions and avoid the bad ones by being attracted to them and distracted by them. The DA algorithm is better because of these five swarming behaviours.

To determine changing probabilities for the location of dragonflies, DA uses a V-shape transfer function. As contrast to other binary metaheuristics, BDA allows the dragonflies to pick values other than 1 and 0, using this transfer function. In order to find the potential search space, BDA has a high level of exploration.

c) Mutation Learning Strategy (MLS) based Dragonfly Algorithm

When revising positions, the MD uses an MLS that incorporates the ideas of personal best and personal worst solutions (Fig. 5). For attraction and distraction in the traditional DA, the dragonflies coordinate the world's finest and worst solutions. The possibility for food hunting and opponent fleeing behaviours, however, is thought to be boosted by including the personal best and personal worst dragonflies into these activities.

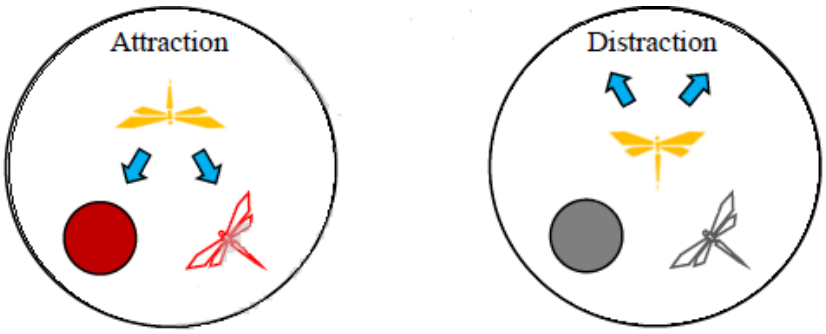

Figure 5: The Attraction and Distraction Behaviors of the Proposed MLS-DA.

In contrast to DA, the following formulae are used to determine the attraction and distraction of MLS.

$$F_i = \frac{(Xpb_i - X_i) + (Xf - X_i)}{2} \quad (17)$$

$$E_i = \frac{(Xpw_i + X_i) + (Xe - X_i)}{2} \quad (18)$$

where $Xpb_i$is the position held by the individual's most impressive dragonfly, $Xpw_i$the dragonfly consider to be the worst, *X* is the location of the dragonfly, *Xf* and Xe is the adversary, is the source of nourishment. Additionally, the dragonflies may benefit from both their own and the world's greatest answers by using the mutation learning technique while they are searching. The overall idea of the learning technique is shown in Fig. 6. The dragonfly strives to replicate from its best experiences both locally and globally rather than adjusting its location in response to swarming behaviours.

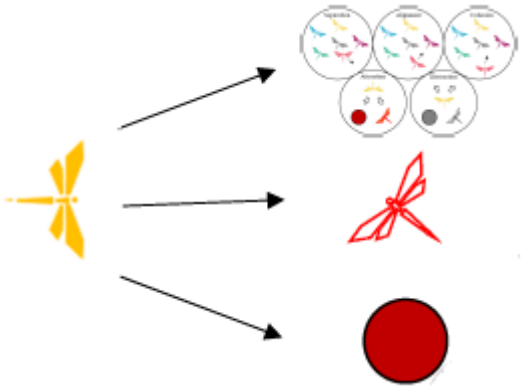

Figure 6: General Concept of the Learning Strategy

The following information is updated for the position of a dragonfly in the planned MLS-DA:

$$X_i^d(t+1) = \begin{cases} \bar{X}_i^d & 0 \le r_1 < pl \\ Xpb_i^d(t) & pl \le r_1 < gl \\ Xf^d(t) & gl \le r_1 < 1 \end{cases} \quad (19)$$

$$\bar{X}_i^d = \begin{cases} 1 - X_i^d(t) r_2 < TF(\Delta X_i^d(t+1) \\ X_i^d(t) r_2 \ge TF(\Delta X_i^d(t+1) \end{cases} \quad (20)$$

In which X represents where the dragonfly is located, *Xpb* is the title of the finest dragonfly you've ever seen, *Xf* a source of food, *i* is the dragonfly order, *d* shows how many variables there are in a choice. (dimension), *t* displays the most recent version, *r1* and *r2* in the range between 0 and 1 are two separate random values. The *pl* and *gl*are the same for both the individual learner and the population as a whole, and their values always fall between 0 and 1. The equation is as follows: (19 & 20), the *pl* and *gl* played significant parts in the education that was received. The algorithm will mostly seek around the individual best and overall best solutions if *pl* and *gl* values are too low, which increases the likelihood that it will get stuck in the local optima. If the values of *pl* and *gl*are excessively high, on the other hand, the position update procedure will resemble DA. Therefore, choosing the right *pl* and *gl* is crucial.

3.4.3. Dendrogram and Consensus Matrix

The presence of the dendrogram is implied if a similarity relation is min-transitive (i.e., t = min). A relation matrix R's min-transitive closure may be calculated with ease, and Algorithm 1 describes the

whole procedure. A dissimilarity relation is the last component needed to complete an agglomerative clustering. Here, we took the following outcome into account:

R should be a relation of similarity between the items $R\langle x, y\rangle \in [\,0, 1]$and assuming that D is a relation of dissimilarity, from which R provides by consequently, if R is minimal, D is ultra-metric.

$$D(x, y) = 1 - R\langle x, y\rangle \quad (21)$$

Algorithm 1: Min-transitive closure

Input: relation S_i
Output: transitive relation $C_i = S_i^T$
Elaborate:
1. compute $S_i^* = S_i \cup (S_i . S_i)$
2. if $S_i^* \neq S_i$ replace $S_i$with S_i^* and go to step 1
else $C_i = S_i^T = S_i^*$ and the algorithm terminates.

Alternatively, a 1:1 relationship between dendrograms and ultra-metric dissimilarity matrices and minimal transitive similarity matrices. The consensus matrix, or the representative information of all the dendrograms, is then created by merging the transitive closures after the dendrograms have been collected each time (i.e., max-min operation). Algorithm 2 outlines the general strategy. Fig. 7 summarises the entire process of the suggested technique.

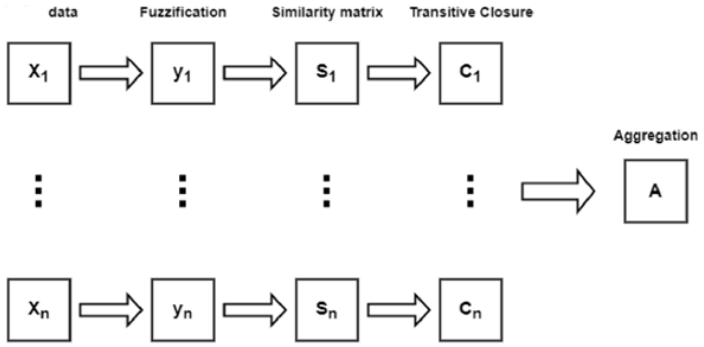

Figure 7: Workflow of the Fuzzy Based Hierarchical Clustering

Algorithm 2: combination of dendrograms

1: **Input C***i*, 1 ≤ *i*≤ *L L* input similarity matrices (dendrograms)
2: **Output** similarity matrix (dendrogram) **A**
1. The similarity matrices should be combined to create a single matrix
A = *Aggregate (***C**1, **C**2, ..., **C***L)*
a. Let **A*** be the identity matrix
b. For each **C***i* calculate e **A***= **A***∪ *(***A***◦ **C***i)*
c. If **A***is not changed **A** = **A*** and goto step 3
else goto step 1.b
3: Using A, create the last dendrogram

In specifically, a fuzzification process is used to produce the new data set *Yi* for the data set *Xi*. The transitive closure of the matrix is ensured by computing a new matrix *Ci* after computing the similarity matrix *Si* using a fuzzy similarity measure (see Algorithm 1). The consensus matrix [30] A and the ultimate final dendrogram are obtained by compiling all the Ci matrices (see Algorithm 2).

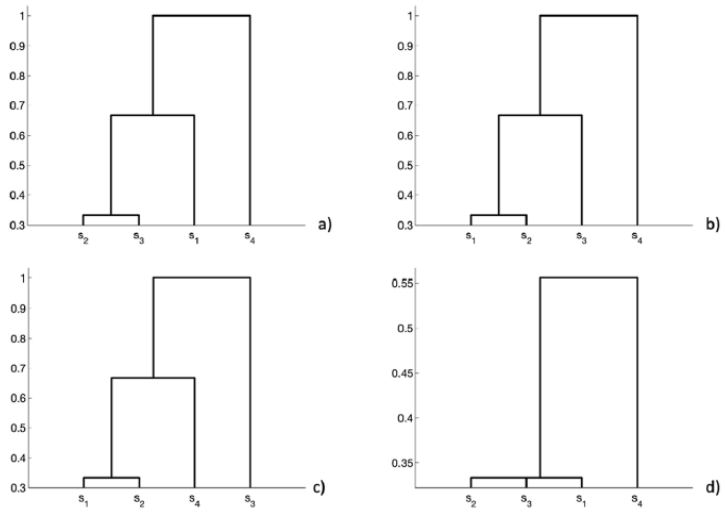

Figure 8: Combination Algorithm: a-b-c input Dendrograms; d Combined Hierarchy

An illustration of a realistic agglomeration outcome should be shown in Fig. 8. Additionally, display the three input hierarchies derived from the combined datasets in Figures 8a through 8c. Four data sequences, s1, s2, s3, and s4, are taken into consideration in this situation. Agglomerate dendrograms in Fig. 8d to display the result. Observe that each of the clusters (such as (s1, s2, s3)) is repeated at least twice in each of the three input dendrograms, and that the output hierarchy comprises clusters (s1, s2, s3) and (s1, s2, s3, s4) at various levels. In addition to this, it is important to emphasise that the method that has been presented, which is based on the aggregation of dendrograms, may also be used with metrics that are typically utilised (e.g., Euclidean distance).

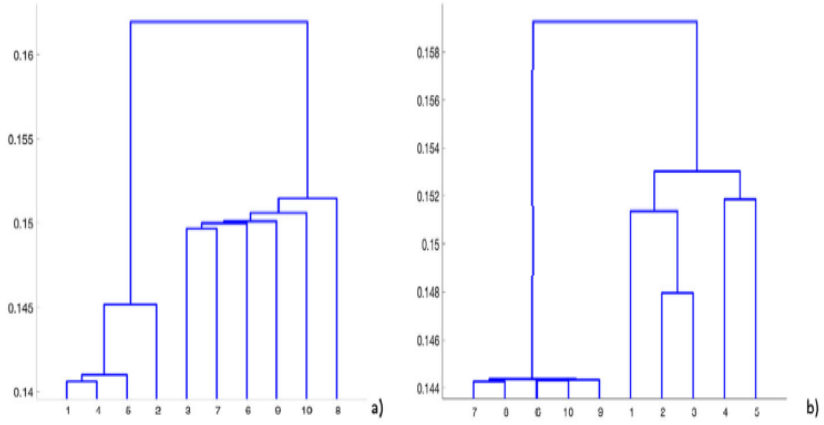

Fig 9: Crisp Hierarchical Clustering vs Fuzzy based Hierarchical Clustering: a Dendrogram of Euclidean Based Hierchical Clustering; b Dendrogram of Similarity based Hierachical Clustering

Comparing the dendrograms produced by a similarity-based technique with the Euclidean metric is shown in Figure 9. Utilize data sets with 10 rows and 100 columns to replicate this actual situation. Additionally, divide the single datasets into two halves such that the first five rows are representative samples drawn from a regular normal distribution with variance 1 and the following five rows are drawn from the same distribution with variance 0.5 to create a type of overlap. The similarity-based strategy in Figure 9b, however, allows for a flawless separation of the source divisions, despite the fact that both approaches uncover two separated clusters.

4 Results and Discussion

The experiment's findings for the provided database of various volumes are presented in this section. We choose to employ multiple quality metrics, which describe how close a network's link structure is to that of a community, in order to assess partitions. The metrics are predicated on the concept that communities are groupings of nodes that have more connections among themselves than they do with nodes that are located outside the community. They were chosen from a standard set of known metrics, which are typically used to evaluate the graph attributes of discovered communities in a directed graph. This was done in order to ensure that the best results were obtained. The ability to differentiate between large communities in a graph is one example of such a property. Other examples include the density of internal connections within formed communities, the number of edges in a graph that are located outside of the community, and the ratio of incoming edges to the total number of community edges. We also followed the approach outlined below while choosing the metrics: (1) The graph attributes that each metric measures must be different; (2) Both density-based and pattern-based clustering must be accommodated by the metrics; and (3) In terms of community discovery, the metrics as a whole must describe the primary graph attributes. The proposed Fuzzy Similarity based Hierarchical Clustering (FS-HC) model is compared with the existing GANX is and Interlinked Spatial Clustering Model (ILSCM).

1. Directed Modularity

The modularity measure for directed graphs [32] has been expanded to include directed modularity [31]. Better results are obtained with greater values.

$$Q = \frac{1}{m}\sum_{ij}\left[A_{ij} - \frac{k_i^{in}k_j^{out}}{m}\right]\delta_{c_ic_j} \quad (22)$$

Comparing the random distribution of linkages between all nodes regardless of communities to the concentration of edges inside communities, or "modularity". Additionally, modular demonstrates how well the technique works for identifying huge groups.

2. Normalized Mutual Information

Creating a criterion to measure how similar the algorithm's supplied partition is to the partition one wants to recover is necessary to measure the performance of a community discovery method. Within the context of information theory, normalized mutual information (NMI) might be employed as a metric of similarity. The following is a definition of the normalised mutual information $N(X|Y)$:

$$N(X|Y) = \frac{H(X)+H(Y)-H(X,Y)}{(H(X)+H(Y))/2} \quad (23)$$

Where H(X), H(Y) is the joint entropy and X(Y) is the random variable associated with partitions $C'$and C''.Only when the two partitions $C'$and C'' perfectly coincide does this variable, which has a range of [0, 1], equal 1.

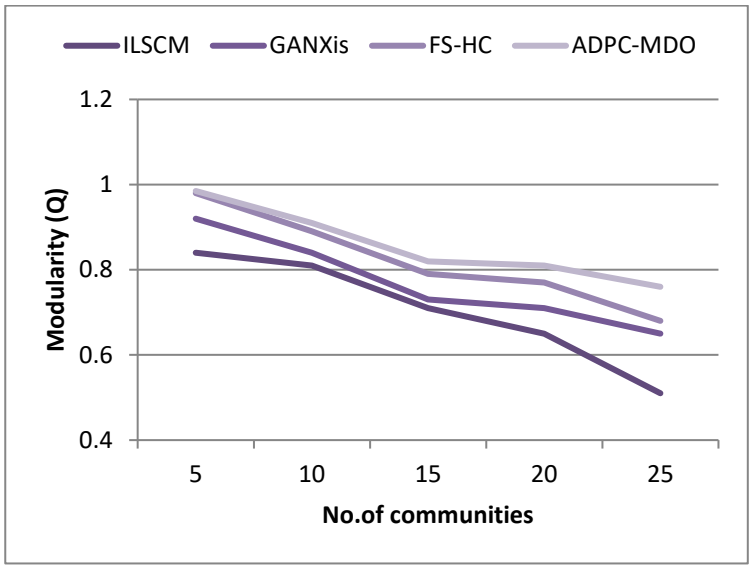

Figure 10:Modularity Comparison Between the Proposed and Existing Methods

Fig.10. shows the modularity comparison between the suggested and current techniques for neighborhood detection. This approach differs from the "ground truth" approach in that it takes four discussions of varying volumes and sets the sociological perspective for expectations to see which algorithm(s) comes closest to them on datasets of varying volumes, as opposed to using just one pre-analyzed discussion with known modularity as the "ground truth".From the simulation result, it is observed that the proposed ADPC-MDO technique provides higher modularity rate when compare to the existing FS-HC, GANX is and ILSCM methods.

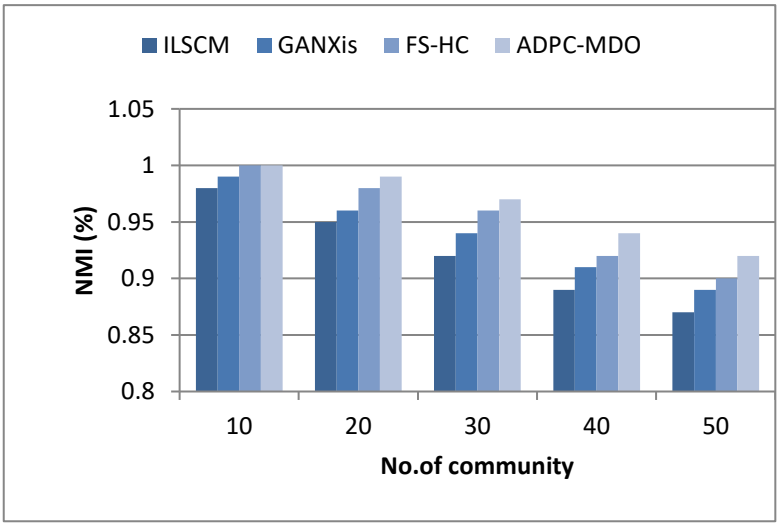

Figure 11: NMI Comparison Between the Community Members in a Network of Different Techniques

Fig.11. shows the NMI comparison between the community members in a network of different techniques for community detection. In most cases, the ideal situation involves a small number of thriving communities. Because of this, we believe that these four algorithms bring social science one step closer to the potential ground truth. From the simulation result, it is observed that the proposed ADPC-MDO technique provides better NMI results when compare to the existing FS-HC, GANX is and ILSCM methods.

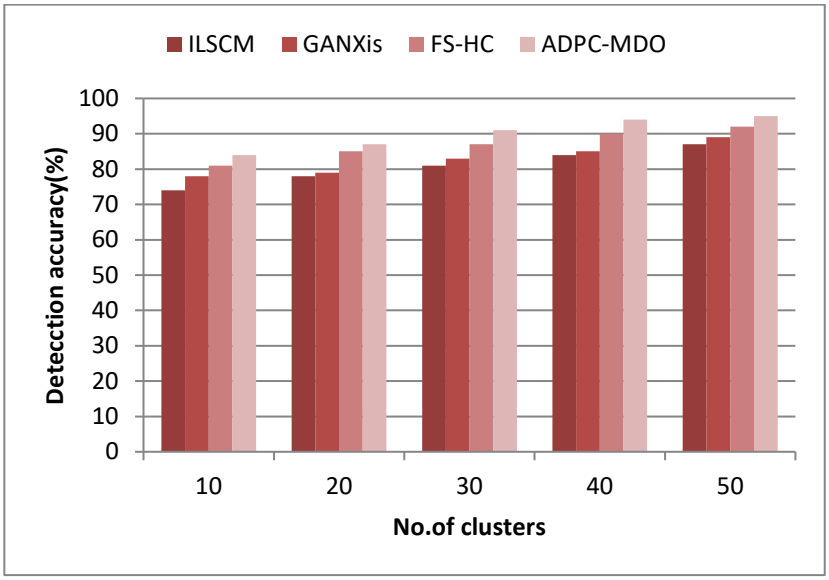

Figure 12: Detection Accuracy Comparison Between the Proposed and Existing Methods

The figure 12. provides the comparison results of the community detection accuracy between the proposed and existing methods. The suggested model discovers the biggest groupings, which might be regarded as the best outcome, out of the four methods with the best (lowest) number of communities. A conclusion drawn from the experimental data is that the suggestedADPC-MDO model provides the high community detection accuracy results than the existing FS-HC, GANX is and ILSCM techniques.

5 Conclusion

In particular, public sentiment monitoring, opinion leader discovery, and personalised suggestion have all benefited from community detection as a crucial method for researching social networks. In this research work, an Adaptive density peak clustering (ADPC) with Modified Dragonfly Optimization (MDO) algorithm is proposed to adaptively determine the communities in social network. It is possible to quickly evaluate the sentiment expressed by various communities and compare and contrast this data with the network as a whole by integrating sentiment analysis with community identification findings. This highlights the biggest advantage of combining sentiment analysis with community identification; doing so enables more accurate sentiment analysis. It has been determined, based on the findings, that the suggested model has good accuracy results for the identification of overlapping communities in the data that was provided for the social network. The way individuals cooperate and communicate is changing as a result of modern social networks. To account for changing user needs and preferences, it is crucial to create an adaptive community detection technique. While having a strong track record in social networks, trust model-based community discovery is still in its infancy. We will include additional network data and examine other network topologies in our succeeding work.

References

[1] Bedi, P., & Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(3), 115-135.

[2] Deitrick, W., & Hu, W. (2013). Mutually enhancing community detection and sentiment analysis on twitter networks.
[3] Parau, P., Stef, A., Lemnaru, C., Dinsoreanu, M., & Potolea, R. (2013). Using community detection for sentiment analysis. *In IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 51-54.
[4] Xu, K., Li, J., & Liao, S.S. (2011). Sentiment community detection in social networks. *In Proceedings of the 2011 iConference*, 804-805.
[5] Wang, D., Li, J., Xu, K., & Wu, Y. (2017). Sentiment community detection: exploring sentiments and relationships in social networks. *Electronic Commerce Research*, *17*(1), 103-132.
[6] Bhatnagar, S., Dixit, M., & Prasad, N. (2020). A review of common approaches to sentiment analysis and community detection. *International Journal of Computer Application*, 975, 8887.
[7] Phyu, K.S., & Min, M.M. (2019). Graph-based community detection in social network. *In IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, 12-17.
[8] Lam, A.J. (2016). Improving Twitter Community Detection through Contextual Sentiment Analysis of Tweets. *ACL 2016*, *30*.
[9] Lam, A.J. (2016). Improving Twitter Community Detection through Contextual Sentiment Analysis. *In Proceedings of the ACL 2016 Student Research Workshop*, 30-36.
[10] Varsha, K., & Patil, K.K. (2020). An overview of community detection algorithms in social networks. *In International Conference on Inventive Computation Technologies (ICICT)*, 121-126.
[11] Khatoon, M., & Banu, W.A. (2018). An effective way of detecting communities in social network. *Communities*, *200*.
[12] Inuwa-Dutse, I., Liptrott, M., & Korkontzelos, I. (2021). A multilevel clustering technique for community detection. *Neurocomputing*, *441*, 64-78.
[13] Wang, D., Li, J., Xu, K., & Wu, Y. (2017). Sentiment community detection: exploring sentiments and relationships in social networks. *Electronic Commerce Research*, *17*(1), 103-132.
[14] Awrahman, B., & Alatas, B. (2017). Sentiment analysis and opinion mining within social networks using konstanz information miner. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(1), 15-22.
[15] Yang, H.C., Lee, C.H., & Wu, C.Y. (2018). Sentiment discovery of social messages using self-organizing maps. *Cognitive Computation*, *10*(6), 1152-1166.
[16] Ding, S., Yue, Z., Yang, S., Niu, F., & Zhang, Y. (2019). A novel trust model based overlapping community detection algorithm for social networks. *IEEE Transactions on Knowledge and Data Engineering*, *32*(11), 2101-2114.
[17] Reihanian, A., Feizi-Derakhshi, M.R., & Aghdasi, H.S. (2018). Overlapping community detection in rating-based social networks through analyzing topics, ratings and links. *Pattern Recognition*, *81*, 370-387.
[18] Zheng, J., & Wang, Y. (2018). Personalized recommendations based on sentimental interest community detection. *Scientific Programming*, *2018*.
[19] Deitrick, W., Valyou, B., Jones, W., Timian, J., & Hu, W. (2013). Enhancing sentiment analysis on twitter using community detection.
[20] Yang, B., & Manandhar, S. (2014). Stc: A joint sentiment-topic model for community identification. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 535-548. Springer, Cham.
[21] Shi, C., Cai, Y., Fu, D., Dong, Y., & Wu, B. (2013). A link clustering based overlapping community detection algorithm. *Data & Knowledge Engineering*, *87*, 394-404.

[22] Gupta, S., & Kumar, P. (2020). An overlapping community detection algorithm based on rough clustering of links. *Data & Knowledge Engineering*, *125*.
[23] Blekanov, I., Bodrunova, S.S., & Akhmetov, A. (2021). Detection of Hidden Communities in Twitter Discussions of Varying Volumes. *Future Internet*, *13*(11), 295.
[24] Christian, H., Agus, M.P., & Suhartono, D. (2016). Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, *7*(4), 285-294.
[25] Yaohui, L., Zhengming, M., & Fang, Y. (2017). Adaptive density peak clustering based on K-nearest neighbours with aggregating strategy. *Knowledge-Based Systems*, *133*, 208-220.
[26] Jiang, D., Zang, W., Sun, R., Wang, Z., & Liu, X. (2020). Adaptive density peaks clustering based on K-nearest neighbour and Gini coefficient. *IEEE Access*, *8*, 113900-113917.
[27] Mafarja, M., Heidari, A.A., Faris, H., Mirjalili, S., & Aljarah, I. (2020). Dragonfly algorithm: theory, literature review, and application in feature selection. *Nature-inspired optimizers*, 47-67.
[28] Rahman, C.M., & Rashid, T.A. (2019). Dragonfly algorithm and its applications in applied science survey. *Computational Intelligence and Neuroscience*, *2019*.
[29] Forina, M., Armanino, C., & Raggio, V. (2002). Clustering with dendrograms on interpretation variables. *Analytica Chimica Acta*, *454*(1), 13-19.
[30] Escobar, M.T., Aguarón, J., & Moreno-Jiménez, J.M. (2015). Some extensions of the precise consistency consensus matrix. *Decision Support Systems*, *74*, 67-77.
[31] Leicht, E.A., & Newman, M.E. (2008). Community structure in directed networks. *Physical review letters*, *100*(11), 118703.
[32] Newman, M.E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577-8582.
[33] Giorgi, G., Abbasi, W., & Saracino, A. (2022). Privacy-Preserving Analysis for Remote Video Anomaly Detection in Real Life Environments. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *13*(1), 112-136.