IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 28, 2020, accepted August 18, 2020, date of publication August 24, 2020, date of current version September 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3018941

A Temporal User Attribute-Based Algorithm to Detect Communities in Online Social Networks

AMIN MAHMOUDI[1], AZURALIZA ABU BAKAR[2], MEHDI SOOKHAK[3], (Senior Member, IEEE), AND MOHD RIDZWAN YAAKUB[2]

[1]Department of Computing and Decision Sciences, Lingnan University, Hong Kong
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia
[3]School of Information Technology, Illinois State University, Normal, IL 61790, USA

Corresponding author: Amin Mahmoudi (amin_mahmoudy@yahoo.com)

ABSTRACT The world is witnessing the daily emergence of a vast variety of online social networks and community detection problem is a major research area in online social network studies. The existing community detection algorithms are mostly edge-based and are evaluated using the modularity metric benchmarks. However, these algorithms have two inherent limitations. Firstly, they are based on a pure mathematical object which considers the number of connections in each community as the main measures. Consequently, a resolution limit and low accuracy in finding community members in often observed. Whereas, online social networks are dynamic networks and the key players are humans whose main attributes such as lifespan, geo-location, the density of interactions, and user weight, change over time. These attributes tend to influence the formation of user communities in any category of online social network. Secondly, the output structure of existing community detection algorithms is usually provided as a graph and dendrogram. A graph structure, is, however, characterized by a high memory complexity, and subsequently exponential search time complexity. Implementing dendrogram such a complex structure is complicated. To address memory complexity and the accuracy rate of the community detection issues, this paper proposes a new temporal user attribute-based algorithm, namely the recently largest interaction based on the attributes of a typical online social network user. Experimental results show that the proposed algorithm outperforms eight well-known algorithms in this domain.

INDEX TERMS User attributes, online social network, community detection, gravity model, recently largest interaction.

I. INTRODUCTION

A. BACKGROUND

Nowadays, a large volume of transactions is stored in online social networks (OSNs) such as Facebook, Instagram, and Twitter. Understanding the characteristics of such networks, which are labyrinthine, to say the least, is very important because they represent a rich source of information and analysis of that information can contribute to cybersecurity, psychological studies [1], link prediction [2], event detection, marketing, recommender systems [3] and urban planning. Furthermore, the huge amount of transactions on OSNs provide a good opportunity to extract relations between two or more people, which has been termed the community detection (CD) problem. CD in networks is one of the most important problems currently being considered by numerous researchers working in the computer science field because it can be used for multitude of purposes, including recommender systems, cybersecurity, communication studies, and information science.

The main algorithm to overcome the CD problem was introduced by [4], which is called the Girvan and Newman (GN) algorithm. It was designed to work on static networks (wherein the nature of the nodes and edges will not change over time) and was tested on a physics collaboration network. CD algorithms for static networks find a group of similar nodes by identifying the nodes in each community that has the highest connections with each other compared to the rest of the network. GN, indeed, is a betweenness based algorithm, that calculate the number of times an edge is used by a node to reach others based on the shortest path. Girvan & Newman, (2002) measured the quality of the GN

The associate editor coordinating the review of this manuscript and approving it for publication was Mamoun Alazab.

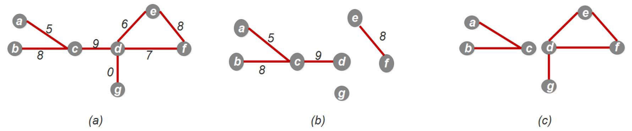

FIGURE 1. **Differences between existing CD algorithms and the proposed algorithm: (a) the network at the last time interval, (b) three communities detected by the proposed method, (c) two communities detected by the GN algorithm. (The numbers next to the edges represent the number of interactions in a recent time interval.)**

algorithm by using a modularity (Q) metric. As the GN algorithm considers time complexity and can function in a large-scale network, many researchers became interested in this domain [5]–[12].

B. MOTIVATION

In recent years, due to the emergence of dynamic networks such as OSNs, static algorithms have become incapable of detecting communities that are relevant to the real world. This is because the same weight can be applied to the nodes and edges in a static network, which makes it inapplicable to identify communities in a dynamic network. Although many researchers recently focused on overcoming the aforementioned issue by developing a variety of dynamic algorithms [13]–[17], the majority of such algorithms divide the network into a series of snapshots and then apply the modularity metric used by the GN algorithm to each snapshot (Fig. 1).

After introducing edge betweenness algorithms, some other CD algorithms have been proposed. For instance, Pons and Latapy presented the Walktrap algorithm [18]. Similarly, a label propagation algorithm (LPA) was proposed by [19]. Rosvall and Bergstrom introduced Infomap, which works on weighted and directed graphs [8]. Although these algorithms do not consider the betweenness between nodes, they are affected by the number of edges in a network, hence these algorithms are called edge-based algorithms.

According to Statista [20], there will be three billion OSN users in 2020, and based on Dunbar's Number [21] where each user could have 150 friends, which could create ~450 billion edges across all OSNs. The output of existing algorithms to detect communities in OSNs (as big data) takes the form of a graph or dendrogram [4]. However, the graph structure has memory complexity in this domain. In this paper, we address this issue by using a tree structure [22] and it has also been proven that solving the maximum modularity is NP-complete [23]. Moreover, according to [24], OSN users do not have interactions with 50% of their friends, and the most active users receive comments from just 5% of friends. This means that although some users make connections with each other, they do not necessarily interact with all users in their network. Therefore, considering the number of connections without considering the interactions in each time interval leads to false community detection. The existing CD algorithms thus cannot recognize meaningful communities due to lack of user attributes such as user weight and users and edges lifespan, where these attributes typically change over time.

C. CONTRIBUTIONS

In this paper, we propose a new CD algorithm based on user attributes such as users' weight, the density of interaction, geo-location, and users' lifespan. To achieve this goal, we leveraged an accurate time interval identification method to detect communities in the recent time interval in which the real picture of the network is presented. We also proved that the modularity metric is not a suitable fitness metric for CD algorithms in OSNs as the human-centric domain. The main contributions of the paper are listed as follow:

1) This study developed a new algorithm that addresses the time-varying nature of OSNs by taking into account some user attributes such as users' weight, the density of interaction, geo-location, and users' lifespan.

2) This study designed a Maximum Spanning Tree (MST) similar to the study in [25], as an output of the proposed algorithm to reduce the search complexity such as required to reduce the amount of memory necessary for storing information among the communities.

The rest of this paper is organized as follows: in the next section, the related work on this domain is reviewed. This is followed by section 3 in which the preliminaries and definition of the proposed algorithm are described in detail. In section 4, the problem is formulated while the proposed algorithm is presented in section 5. Next, in section 6, the results of some experiments are presented and the implications are discussed. Finally, in section 7, some conclusions are drawn.

II. RELATED WORKS

This section provides an overview of the existing works on community detections. So far, various researchers have proposed different categories of algorithms [26]–[30]. In [26], the authors presented a survey of CD for dynamic networks. In [27], CD algorithms are further categorized into two parts; topological and topical. A classification based on evolutionary algorithms is presented in [28]. Study in [29] and [30] categorized CD algorithms based on approaches and methods which are deployed to solve the CD problem. Although all classifications are correct, in this study we reviewed significant works and we categorized cited works as static and dynamic. However, regarding the nature of the CD problem in OSN, this study extrapolated a new category defined as Meta-heuristic algorithms.

A. STATIC ALGORITHMS

The basic algorithms designed to overcome the CD problem is the static networks. The basic algorithm for detecting communities is the GN algorithm which was proposed by Girvan and Newman [4]. The GN algorithm is based on the maximum betweenness between the nodes in each community and the lowest interconnection between the nodes in different communities. The authors also introduced the modularity measure ($Q = \sum_i (e_{ii} - a_i^2)$ where e_{ii} is the percent of edges in module i and a_i = percent of edges with at least

1 end in module i) for evaluating the results produced by the GN algorithm. This measure was widely used by some researchers over the next several years. However, modularity and the GN algorithm have some problems: i) the modularity function works on the number of connections, which means that it only computes the connections between the nodes in each community and only considers edges between nodes that have the same weight; ii) the algorithm is highly complex ($O(n^3)$); iii) the memory complexity is high at $O\left(n^2\right)$; and iv) the output of the algorithm is in the form of a dendrogram, which cannot reduce network complexity and has a high computational cost. Two years later, Newman in [5], presented a new algorithm with a better modularity value than the GN algorithm and which was also faster. The time complexity and modularity of this new algorithm was evaluated by applying it to the karate club dataset, an American college football team, a jazz musician collaboration and a network of physics scientists.

Clauset, *et al.* [7] presented the Clauset-Newman-Moore (CNM) algorithm to optimize the computational cost, reduce the time complexity, and detect meaningful communities in a very large network. The authors claimed that the algorithms developed in previous works were slow because of their structure. They, therefore, presented a new method by changing the modularity as the main factor in their algorithm. They used the ΔQ matrix instead of the adjacent matrix to save memory and time. They also used three data structures for their algorithm: (i) the sparse matrix for ΔQ, (ii) max-heap, and (iii) ordinary vector array. The proposed algorithm was applied to the amazon.com purchasing network and the result was evaluated by calculating the modularity.

Pons & Latapy [18] presented an algorithm based on the random walk, namely Walktrap. The idea for their algorithm was motivated by the fact that random walk tends to become trapped when it is applied to a graph. Consequently, they assumed that the nodes in the community are dense to reflect the number of edges. Therefore, in the random walk, the probability of arriving at node j from i can be computed, for nodes in the same community the probability value is high. However, this assumption is not always true. The probability function is calculated based on the node degree, which implies that the output of their algorithm is a dendrogram, or in other words, it is a tree-type structure. At the time of its publication, the Walktrap was a breakthrough in community detection algorithms in terms of addressing the issue of time complexity. However, it does not consider user attributes, is not time-based, and, more importantly, it is based on the number of edges.

The label propagation algorithm (LPA) was presented by [19] based on the number of neighbors instead of the betweenness. This means that a node is only given a label if that label is the most common among its neighbors. The basic idea behind their method is that each node in the network has a label that can propagate to other nodes. A node changes its label based on the number of neighboring nodes that have the same label. Thus, this algorithm can be categorized as an edge-based algorithm. The authors evaluated their method by using a modularity metric. However, in some cases, the LPA puts all the nodes in just one community [31]. Zhang *et al.* [32] assert that the randomness and uncertainly in classic LPA is the main reason for the poor result of LPA. The study overcame this limitation by finding the influential node (important node) in the social network. Chen *et al.* [33] utilizes information entropy as the measure for identifying the relationship between direct and indirect neighbors. The study further inferred that the classic LPA merely considered the directed relationship between neighbors. They calculated the mutual information for each pair of nodes as well as the weight of the label. The main idea behind their work is that the weight of friends is important to propagate the label. In [34], a new community detection algorithm based on the fire propagation model to deal with the time complexity and find the shared communities was developed. The designed method works based on two phases: (1) the algorithm starts with a random node and finds a two-radius neighborhood subgraph, and (2) finding the community structure around seed node from two-radius by using fire spread model. However, the study is limited to the R-radius neighborhood subgraph.

In [8], Rosvall and Bergstrom introduced Infomap, which works on weighted and directed graphs. The challenge of finding a community structure is similar to the coding compression problem. In Infomap, each node has a unique name that is selected by using a Huffman code. Then by using the random walk, the algorithm detects communities in the network. The algorithm maps the information flow in the network to detect communities. In this method, "modularity for a given partitioning of the network into m-modules is the sum of the total weight of all links in each module minus the expected weight" ([8], p.1122). The main advantages of Infomap include the use of a dynamic approach and considering the weight. However, it is the number of interactions that act as the basis for assigning weights to the links. Moreover, the time complexity of the complicated computational method is the main drawback of Infomap. Besides, this algorithm completely relies on connections and the output is in the form of a graph.

B. DYNAMIC ALGORITHMS

With the advent of complicated networks such as OSNs researchers are interested in designing CD algorithms for dynamic networks. In [35], the authors presented an online algorithm based on the clique percolation method (CPM) and label propagation algorithm, namely OLCPM. Their proposed algorithm works on a temporal network with fine granularity and tries to update the local community structure. Addressing the community detection problem with respect to community evolve is the main concern of this work. The authors used a clique percolation method as a deterministic model. They also considered the communities independent of the rest of the network. Due to some limitations of the original CPM, they introduced a new CPM method in which a label propagation post-process is proposed. They claimed that with each change, the proposed method updates the community structure by inserting or removing edges, and nodes consider changes in the network behavior. Whilst it is generally assumed that community members can overlap, however, the current study assumes that a temporal network in each time point node can belong to one community. In [36], a dynamic approach is proposed for the community detection problem. The authors asserts that communities should update continuously by considering the results of the last time slice. The study developed an incremental update process for the community based on historical information. They assumed that new data continuously generate subgraphs which simultaneously join the dynamic network. In [37] a density and time-based approach for overlapping community detection and its evolutionary events is proposed. Their method considers only active nodes in new time-step instead of processing all nodes. Although the above-mentioned algorithms are proposed for dynamic networks and are time-based, they are designed for modularity metrics. These studies did not consider user attributes as an inherent component of the online social networks, given that OSNs is a typical human-centric network. Furthermore, these studies are unable to detect the recent time interval of an OSN as a critical measure to reveal the real picture of OSNs.

Bu, *et al.*, in [38] proposed a new algorithm for detecting communities in OSNs, named the fast parallel modularity optimization algorithm (FPMQA). The authors' main concern was to address the problem that the existing algorithms were incapable of analyzing- large OSNs. Their algorithm was agglomerative and also used ΔQ to evaluate the modularity in each stage for nodes clusterization, like the CNM algorithm. However, the difference between the FPMQA and the CNM algorithm is that the latter considers the global maximum for ΔQ, whereas the former, FPMQA, considers the local maximum. The authors evaluated the FPMQA by CPU time and modularity by using the karate club dataset. A key finding of their study was that by doing parallel strategy the time complexity could be reduced to (O $(\sum_{p=1}^{d} k_p^{max}(k_p^{max} + k_p * log k_p^{max}))$, where $k_p^{max}.k_p$ denotes the maximum degree and the average degree of the network in the *p-th* pass, respectively. Although the output of the FPMQA is an interest graph based on user opinions and other properties of an OSN, such as interactive times, it still worked like a static CD algorithm, and the algorithm was also modularity based.

Dev *et al.* [14] detected meaningful communities in OSNs by focusing on the interactions between users. Their algorithm considers the group behavior of every pair of users that are connected by common neighbors. Then, it computes the probability of two users belonging to a common community by using the interactions and user behavior. Finally, it employs hierarchical clustering to detect a community by considering the probability measure. The time complexity of their algorithm is $O((m+\Delta^2)n)$, where Δ is an average vertex degree, n is the number of nodes and m is the total number of edges of the social network. Study in [39] proposed a method for detecting communities in large weighted social networks based on density and attractiveness to reduce the time complexity of the algorithms proposed in previous works. They chose this approach because a typical real-world dataset has weighted nodes and edges. The basic algorithms for CD mostly ignore this feature and are more complex from the time standpoint. The time complexity of their CD algorithm is $O\ (nK + m^2)$ and exhibits better time complexity than the GN and CNM algorithms. It can be seen as a breakthrough in social network CD from the standpoint of assigning weights to edges and nodes, although their proposed algorithm was still designed to work in the static social network. The memory complexity of their algorithm is $O(n^2)$ because they used a N^*N matrix, while the time complexity is $O(nk + tm^2)$, where t is the maximum number of iterations, k is the average number of inter-interested nodes for all nodes, m denotes the number of clusters at the beginning of *i-th* iteration and n is the number of nodes.

Aston & Hu [13] made a significant contribution to address the CD problem by introducing two algorithms for dynamic social networks: the dynamic structural clustering algorithm for networks (DSCAN) and the genetic algorithm dynamic (GAD). The former is of particular interest in the context of the current study as the DSCAN employs the structural clustering algorithm for networks (SCAN) on the first time interval of a network (use constant time interval). Afterwards, for all consecutive time intervals, the differences in the edge between two-time intervals are computed. In this way, the network can be updated based on the edge changing in the network. When considering changes to the edges and nodes in a network, it is necessary to update the network and form a community for new nodes. The DSCAN is a density-based algorithm that uses CN notation. The original algorithm in this domain is the well-known density-based spatial clustering of applications with noise (DBSCAN), which is widely used for clustering problems. The difference between the DSCAN and DBSCAN is related to the definition of the similarity function, where the DBSCAN uses distance and the DSCAN uses CN notation. Although their proposed algorithm is modularity based and assigns the same weight to the edges and nodes, similar to static algorithms such as the GN algorithm, which is a drawback of their work, the authors presented a different methodology for analyzing the dynamic network based on time by dividing the network into several snapshots.

In recent years, many researchers have used the user characteristic approach to detect communities in social networks. For instance, Kanavos, *et al.*, [40] detected influential communities based on the emotional profile and the analytics profile of the users. This is a notable work in this domain, where the authors believed that it is not sufficient to use node connectivity alone when analyzing a social network; rather the special characteristics of the nodes can provide significant information for social network analyses such as community detection. They defined six main human emotions, namely surprise, sadness, happiness, fear, disgust, and anger. Consequently, the study identified the influence of each user to detect influential communities for which they use a modularity-based community detection algorithm [41]. Zhu, *et al.* [42] also supposed that user emotions can affect the formation of a community. They investigated this issue in three phases: (i) they created an emotional network, (ii) they applied the CNM and BGLL algorithms to detect communities in the emotional network, and (iii) they compared their results with those of four other networks to verify their method. From the viewpoint of using user behavior to detect communities, their work has value. However, the CNM method, which is suitable for static networks, is inconsistent in an environment in which the main players are human. In [43], a new method known as LED-based on structural clustering for detecting the overlapping communities in the social network is proposed which has linear time complexity. The idea behind their method is that two people with a mutual friend are more likely to belong to the same community. In trying to address the community detection process, a study in [44] proposed a divisive method, namely local edge centrality (LEC) for community detection, which uses the node dissimilarity and edge betweenness degree. The node dissimilarity in the proposed method is based on the neighbors of the node. In the first phase, the LEC value of each edge is computed based on the degree of node dissimilarity and edge betweenness, and its assigned weight. Thereafter, the network is divided into communities of isolated nodes. The process of joining a community in this approach is hinged on the following assumptions:

1) If all neighbors of node *i* are isolated, then *i* will be merged into a neighbor with a maximum degree.
2) When there are two competing communities, *i* will be added to a community with maximum density.

The density value is a fraction of edges in a specific community over the number of nodes in such a community. Their method is a neighbor-based method, where the joining and merging operation is based on the degree of the neighbor. However, the studies further opined that "*In a social network, people in the same group or community usually have common background and interests*" ([44], p.2), which they only considered as the structural properties of the network. In this regard, the process of assigning weight to edges is a core advancement of the study. Their method improved the modularity value as well as the general accuracy relative to the GN algorithm. A method hinged on the edge-betweenness, namely SocioRank, is developed in [45] to identify important nodes for making connections between communities. The authors claimed that their method is an extension of the GN algorithm capable of discovering a social role based on some fundamental properties of the network, such as degree, betweenness, and closeness centrality. Li, *et al.* [46] adopted the random walk algorithm for community detection (CD) in a multi-layer social network. Their method finds local core node based on a Trust-relationship in the network and then clusters the nodes based on the network information. This assumes that the result of the CD algorithm depends on the location of its initial node. Consequently, the study defined a new measure for identifying important nodes based on the node's degree in each layer. The important node (core node) is then identified using the number of neighbors at each level. Under specific criteria, their study generalizes the random walk method from single-layer networks to multi-layer networks where the Random Walker can walk in an adjacent layer. Their method utilizes modularity to evaluate the CD algorithm accuracy when the ground truth is unavailable. The proposed method is interesting from the viewpoint of finding the core node and assuming that the core node is important to form the community. A new function for measuring community detection in a social network is also presented in [47] to mitigate the limitation of modularity, in which smaller communities can be detected.

Sharma and Oliveira [48] presented a method that is suitable for dynamic networks. They focus on one of the main challenges in community detection in OSNs, namely memory size. To address this challenge, they propose a parallel hybrid algorithm to detect communities, where their method used the node with the highest centrality at each time and the weak edges are removed. On the other hand, Haji Seyed Javadi, *et al.*, [17] presented a novel algorithm to detect communities based on the role of community leadership, where they assume that communities are formed around the leadership of each community. They defined leadership as the node with the highest degree centrality. However, Mahmoudi, *et al.*, [49] showed that considering the node with the highest degree centrality without considering the weight of each user in each time interval cannot determine who among the nodes (users) is more influential. Nevertheless, the idea presented by Haji Seyed Javadi and colleagues is significant as they proved that the most important users can attract other users.

One core study that helped us to formulate our approach is that of Wilson *et al.* which made a significant contribution to this area of research by revealing the vital role that user interactions play in the study of social applications [24]. Their study resulted in some valuable findings as follows: i) users tend to communicate with a small group of their friends; ii) although Facebook evolves, user interactions do not vary with time. This means that even though many users join Facebook over a certain period, the growth rate of interactions between users over that time is relatively steady; iii) for the majority of users, about 70% of their interactions are with only 20% of their friends; iv) the most active users receive comments from 5% of their friends, and v) half of all interactions belong to 10% of the well-connected nodes based on their degree. Thus they showed that the social link is not a valid indicator for analyzing the user interactions in OSNs.

C. META-HEURISTIC ALGORITHMS

Some researchers adopted a different perspective and attempted to use the universal law of gravity to detect communities in networks. However, there are some main deficiencies in those attempts, which the current study will seek to address in its usage of this formula. The first study to mention here is that of [50] who introduced a new CD method based on local community gravitation. The discovery of overlapping communities was the main concern of their study. To achieve this goal, they used the in-degree and out-degree to define the fitness measure for their work. This fitness measure was designed to replace the modularity metric, however, their measure is the ratio of the internal degree of nodes to the summation of the internal and external degree of nodes. In their simulated gravitation formula, the weight of the nodes is employed to represent the degree of the nodes. However, this assumption is not correct unless the weight is assigned to each time interval. The study opined that the weight of the user in each stage of their lifespan in the network needs to be considered. In their work, distance is considered as the square of the shortest path between two intended nodes. This assumption inducts the betweenness measure. In sum, their method is completely edge-based, while in OSN user attributes and behavior influence the formation of community. Also, they did not discuss why they decided to use the square of the distance, particularly when Newton's original formula uses the geo-location of mass. Lastly, their method uses the initial step to categorize nodes into some communities. Moreover, an in-depth analysis of their method revealed that considering the shortest path between two nodes with no edges between edges leads to failure, where it is clear that nodes with a path length 1 are more likely to be in a common community than nodes with a path length of 2 and bigger.

Similarly, Yang, *et al.*, [51] used the gravitation formula for detecting communities. They computed the user weight as the node degree. Thus, they assumed that 2-hop nodes always have a certain influence on the 1-hop node and a rare influence on hop-3 nodes. However, they did not explain how they determined these levels of influence. Nevertheless, based on the above assumption, they created a function that computed the gravitation between a specific node and its neighbors. Besides, they did not explain how to compute the distance in their method or the role of the G constant. At the end of their study, they compared their method with some others by applying it to the Zachary, Dolphin, and Football datasets and using the modularity metric. Although the meta-heuristic approach shows promising concept, it however, follow similar limitation like the static and the dynamic approach, where human-centric composition is largely ignored. This limitation forms a core aspect of the current study. To do that, the next section presents a foundational basis on which OSN and human-centric components for community detection process, are defined.

By leveraging other optimization processes, Cai, *et al.* [52] used particle swarm optimization (PSO) algorithm to detect communities in a signed network. PSO is originally designed for continuous optimization. However, the study modified the operators of PSO in a manner that applies to the CD problem. Firstly, they defined the community based on the characteristics of a signed network, which implies that a community consists of nodes with more positive links than negative links. Then, the authors discretely redefined the velocity and position and considered everyone as a particle. Position vector indicates the partition of the node in a signed network and the velocity shows a vector, where each element can be either a 1 or 0, according to the value of the position (changed or unchanged). In advancing this logic, a study in [53] used the quantum-inspired evolutionary algorithm (QIEA). The observed that social networks are affected by the collective latent behavior, while it is immune to individual behavior.

III. PRELIMINARIES AND DEFINITIONS

In this section, an online social network is first modeled, and then maximum spanning tree and community detection are defined.

A. ONLINE SOCIAL NETWORK

Definition 1 (Online Social Network): Online Social Network (OSN) are online services that facilitate communications in a social network. Each OSN can be presented as $G(U, C, T)$ where U is a set of users' ID, C is a set of triplets of the form of (u_i, u_j, t) where $u_i, u_j \in U, t \in T$. T is a set of timestamps in which two users u_i and u_j communicate with each other, mostly T be considered as the time of making a connection between two users.

According to the above definition, OSN is a time-varying and dynamic domain in which the behavior of the network changed. OSN behavior changes refer to the specific states of OSNs in which the number of user activities changes.

Definition 2 (Network Behavior): Network behavior of OSN is a snapshot of G in t $(G(U, C, T)_t)$ which shows the number of users and connections in $t \in T$. The network behavior changes if the number of users and connections changed markedly, so the network behavior is a function of the number of users and connections (1)

$$f\left(N_u, N_c\right) = \begin{cases} \mathbf{1}, & \boldsymbol{N_u\left(t\right) + N_c(t) > \textit{threshold value}} \\ \mathbf{0}, & \boldsymbol{N_u\left(t\right) + N_c\left(t\right) < \textit{threshold value}} \end{cases} \tag{1}$$

$N_u(t)$ is the number of users at time t and $\in N$
$N_c(t)$ is the number of connections at time t and $\in N$
$f(N_u, N_c) \rightarrow [0, 1]$

It is clear, the number of connections is related to the number of users, thus the number of users plays an important role in network behavior. For further information [54].

The main player in OSNs is human which is a complicated creature in the world and the OSNs presented as the user. The users' attributes such as user weight, geo-location, lifespan, and density of interaction show the user behavior in OSN.

Definition 3 (User Behavior): The user behavior (UB) in G is a state of a user in $t \in T$, and is a set of quintuplets of the form of $UB(u_i, w_i, l_i, \rho_i, d_i)_t$. Where $u_i \in U$, w_i is the weight of user *i*, l_i is the lifespan of user *i* and is a triplet with the form of (u_i, t_s, t_e) where t_s, t_e represent the first time of activity and last time of activity of user i respectively $(t_s < t_e)$. ρ_i is the density of interactions of user i in a specific time interval. d_i refers to geo-location of user *i*.

OSNs usually are undirected, meaning that when a user sends a message to another user, they accepted the friendship, although one side does not send any reply (e.g. WhatsApp). However, in some OSNs the relationship is directed (e.g. Twitter).

B. MAXIMUM SPANNING TREE

The maximum spanning tree (MST) is retrieved from the minimum spanning tree (MinST) concept, where instead of minimum edge weigh we consider maximum edge weight. MinST is a problem in a graph, According to [55] this problem defined as, in graph G "nodes represent cities and edges represent possible communication links and whose edge weights represent the cost of construction or lengths of the links" [37, pp: 1] therefore the problem is finding a set of edges with minimum costs and cover all nodes.

Definition 4 (MST): Given an online social network G, nodes represent users and edges represent the connection between users and whose edges weights represent the number of interaction in the recent time interval, it is clear that we try to find the link with the highest number of interaction. The MST is a tree that covers all users in G with a maximum weight of connections.

C. COMMUNITY DETECTION

Definition 5 (Community): Cambridge dictionary defines community as "the people living in one particular area or people who are considered as a unit because of their common interests, social group, or nationality" [56]. However, in network science, Newman & Girvan in [57] community is defined as "the division of network nodes into groups within which the network connections are dense, but between which they are sparser". Thang N. Dinh & My T. Thai in [10] defined community as "nodes in the network are naturally clustered into tightly connected communities with only sparser connections between them". Radicchi, *et al.*, (2004) in [58] Defined community as "a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network". These definitions also used and repeated by recent researchers in different types of networks such as social networks, OSN, complex networks, collaboration networks, and biological networks. In OSNs we define the community as a set of users $c_k = u_1, u_2, \ldots u_j$ who have the maximum dependency among each other as a group. In other words, the community is a group of users that exhibits the largest gravity among its members.

Definition 6 (Community Detection): The community detection problem in a given online social network G is to find a set of communities $C = c_1, c_2, \ldots c_r$ of disjoint subsets of users in a meaningful manner relevant to the definition of community.

IV. PROBLEM FORMULATION

The result of the literature review showed that the existing CD algorithms have several drawbacks for solving this problem in OSNs. First of all, the definition of community and community detection is not relevant to a human-centric domain. Second, they are mostly modularity based and designed for CD in a static environment. Third, they do not consider time or at least do not present an accurate definition of the time interval, and fourth, the complexity is a major concern for them. Finally, and above all, they are not human-centric. When the term human appears in a research study, identifying the characteristics and analyzing the behavior of human beings is, of course, essential.

This study deploys the gravitational search method proposed in [59], [60]. In this type of search method, two main parameters are influenced such as user weight and distance. This type of search is based on the force of gravity law which described as below:

- Force of gravity: According to Newton's law of gravity [61] "Gravity is the force that attracts two bodies toward each other. The more massive an object is, the stronger its gravitational pull". This law is expressed as follows: The gravitational force between two particles directly proportional to the product of the mass and the square of their distance from each other two particles is inversely proportional (2).

$$\boldsymbol{F = G\frac{m1 * m2}{r^2}} \tag{2}$$

However, the proposed method also considers the density of interaction as discussed in the following.

Before discussing the proposed algorithm in detail, it is essential to describe the important variables for the CD algorithm based on user attributes. The proposed algorithm deploys four main attributes: time, user weight, the density of interactions, and geo-location.

Time: Any attempt to address the problems associated with a dynamic environment will fail if the user and edge lifespan are not considered because the time dimension is an integral part of a dynamic network. Online social networks can be considered dynamic networks, thus OSNs also rely on the time dimension. The role of the user and edge lifespan is important because the community of OSN changes in each time frame. Therefore, this study considered the time dimension to be a key component in developing the proposed CD algorithm. According to [24], a meaningful community in an OSN is one that is apparent in a recent time interval. In other words, the CD algorithm should act by the most recent interactions between users to detect active communities in each time interval.

Definition 7 (Time Interval): A time interval is a series of periods that contain different amounts of user behavior (according to definition 3). A time interval i showed as TI_i which include a range between two timestamp t in G such as $[t_i, t_{i+k}]$. *TI* computed based on (3). Further information available in [54].

$$\boldsymbol{TI} = (\boldsymbol{t_1} = [\boldsymbol{D}, \boldsymbol{D} + \boldsymbol{SD}], \ldots, \boldsymbol{t_{k-1}} = [\boldsymbol{D} + (\boldsymbol{k} - \mathbf{1}) * \boldsymbol{SD}], \\ \boldsymbol{t_k} = [\boldsymbol{D} + (\boldsymbol{k} * \boldsymbol{SD})]) \quad (3)$$

where D is the time of the first connection in the network per day or an hour and SD is the standard deviation, $k = \frac{\text{R}}{\text{SDvalue}}$ and R is the range between creating a new connection per unit (day or hour) and the last connection per unit.

Geo-Location: A variety of measures can influence the community forming among nodes (in this case, OSN users), such as job, interests, and activities. However, some algorithms use geo-location because it has been proved that users who are near to each other are more likely to form relationships with each other than with users who are further away from them [62]–[69].

Density of Interactions: The density of interactions is relatively self-explanatory and represents the number of interactions between users in a specific time interval (4).

$$\rho = \frac{\boldsymbol{Number\ of\ interaction}}{\boldsymbol{timeinterval}} \quad (4)$$

User Weight: Here, the term user weight refers to the influence of a user in an OSN. In recent years, many researchers have sought to identify the influential users and user weight [49], [70]–[76]. This is because "in an OSN, each user has a specific weight, which refers to the influence of the user on the OSN, and the weight of each user is different. A user's weight is a key indicator of the user's influence on the OSN; where the weight of the user is greater, the more influence that the user has on the OSN as compared to other users. An accurate understanding of the role of users is fundamental to the solving of many online social network (OSN) domain problems, such as community detection, event detection, and marketing" [49]. We use the method presented in [49] and deploys the simple exponential smoothing (5) is a method to estimate the current value based on previous values in a time series with a coefficient α. This method is widely used in time series data mining and is expressed as:

$$\boldsymbol{W} = (\mathbf{1} - \boldsymbol{\alpha})\, \boldsymbol{w_{tn}} + \boldsymbol{\alpha}\, (\mathbf{1} - \boldsymbol{\alpha})\, \boldsymbol{w_{tn-1}} + \boldsymbol{\alpha}^{\mathbf{2}}(1 - \boldsymbol{\alpha})\boldsymbol{w_{tn-2}} \quad (5)$$

where α is a smoothing constant between 0 and 1 and W is the simple exponential smoothed statistic at time t.

In Newton's formula (2), the attraction between two bodies is directly proportional to their weights and is inversely proportional to the square of the distance between them (Fig. 2(a)). Figure 2(b) provides a computer science simulation of the universal gravity formula, which shows that users tend to move toward heavier users. In other words, the total lifespan of the connection between two users that know each other could be overcome by the gravitational pull of a more famous "heavyweight" user with a larger number of connections and who may not even have made a direct connection with those users. In OSNs, people mostly tend to interact with famous users such as artists, athletes, and celebrities. Hence the number of connections of famous users becomes heavy. Also, the weight/fame of some users is relatively heavy that their first interaction quickly attracts other users and resolves into a community; drawing in all of the OSN users within its radius of influence. On the other hand, it has been proved that distance has an inversely proportional effect on attracting users in OSNs to each other [62], [63], [77].

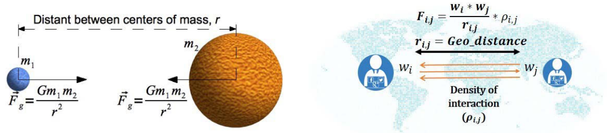
Distant between centers of mass, r
m_1
m_2
$\vec{F}_g = \frac{Gm_1 m_2}{r^2}$
$\vec{F}_g = \frac{Gm_1 m_2}{r^2}$
$F_{i,j} = \frac{w_i * w_j}{r_{i,j}} * \rho_{i,j}$
$r_{i,j} = Geo\_distance$
w_i
w_j
Density of interaction ($\rho_{i,j}$)

FIGURE 2. (left) the universal gravity formula; (right) the universal gravity formula simulation in OSN.

Thus, it seems possible and potentially fruitful to simulate the universal gravity formula (2) as an OSN CD problem (6), where heavyweight users have much more gravity than lightweight users, and consequently, the lightweight users tend to have more interactions with heavyweight users and stay in those communities [49].

The current study's conceptualization of user interactions is represented by (6). We consider the social network as an undirected graph because when a friendship is made between two persons they have a bipartite relationship with each other.

Here we assume that users in OSNs exert a gravitational force on each other. For instance, two users are attracted to each other relative to their respective weight and influence. It is clear that the amount of gravity between them is a function of the directly proportional relationship between their weights and densities, and is inversely proportional to their distance. Newton's formula (2) uses the square of the distance to show the effect of distance, whereas we use a power of 1 distance because other studies have shown that friendship has an inverse relationship with distance [63], [68].

$$\boldsymbol{F_{i,j}} = \frac{\boldsymbol{w_i} * \boldsymbol{w_j}}{\boldsymbol{r_{i,j}}} * \boldsymbol{\rho_{i,j}} \quad (6)$$

where w_i is the weight of the user (node) i, w_j is the weight of user (node) j, $r_{i,j}$ is the geo-distance between the current location of node i and node j, and $\rho_{i,j}$ is the density of interaction between users i and j.

We use the ρ (density of interaction) value in place of the gravity constant in the universal gravity formula (2) to thwart the effect of gravity between users without interaction. This is because the proposed method assumes that connections that remain passive during the time should be overlooked. Equation 7 shows that the density of interaction of user i is equal to the ratio of summation of a number of its interaction to the time range.

$$\boldsymbol{\rho_i} = \frac{\sum_{j=t+sd}^{j=t+(k*sd)} \boldsymbol{number\ of\ interaction_j}}{\boldsymbol{time\ range}} \quad (7)$$

However, Computer Scientists believe the relationships between people are virtual, which means that people interact with each other independently of location. Nevertheless, in this study, the measure used for computing the distance between users is geo-distance because it is clear that people in the same place have more chance of communicating with each other than people who are far from each other [62], [63], [68], [77]. Also, the stated variables in the proposed formula only consider the most recent time interval in which the largest number of interactions takes place among users. Hence we name the proposed algorithm, the recently largest interaction (RLI) algorithm.

The distance should be normalized, according to [78] average area of a city considers as 40km, so the distance in (8) defined as a fraction of 40, it means that for two users who live in 40km area the r value defines as 1, and for distance 80 define as 2 and so on (8).

$$r = \left\lceil \frac{distance}{40} \right\rceil \tag{8}$$

In following the maximum value as gravity considers for each node to assign pair nodes among their neighbors, (9).

$$\textit{assign node } v_i \textit{ to node } u = Max\left(G_{uv1}, G_{uv2}, \ldots, G_{uvn}\right) \tag{9}$$

V. PROPOSED ALGORITHM (RECENTLY LARGEST INTERACTION)

Figure 3 shows the flowchart of the procedure followed in the proposed algorithm. In the first step, we compute the duration of the recent time interval required to detect communities because, according to [24] and [49], users in OSNs do not interact with around 50% of old users (friends) and communities change over time. Therefore, the proposed algorithm needs to be able to detect new communities dynamically. We consider that the amount of interactions in recent time intervals indicates the presence of meaningful communities and is a measure that is relevant to the real world. Then in the next step, we compute the weight of each user based on the proposed method by [49]. After that, the algorithm computes the density of interactions in the recent time interval and the distance between each pair of users, respectively. Next, it calculates the gravity based on (6). Thereafter, the maximum amount of gravity considered as the best measure to form the community was defined as a measure for selecting the connections in an MST. Finally, the pair-users assign in a community was extracted. For next pair of the user in next iteration, a pair-user is assigned to a community in which one of them already joins that community, otherwise, they formed a new community. The pseudocode for the algorithm is presented in Algorithm 1and 2. It is noted that the stated variables in the proposed formula only consider the most recent time interval in which the largest number of interactions takes place among users. Hence, we defined the proposed algorithm as the recently largest interaction (RLI) algorithm, which consists of two sub algorithms (algorithms 1 and 2). Algorithm 1 computes the recent time interval and inserts the communication data of recent time intervals in a new data table namely *RintTbl*. Then, the number of interactions of users in the recent time interval can be computes based on *RintTbl* data while the result along with users' names are stored in *IntTbl* as the output of algorithm 1. Algorithm 2 computes the gravity value and then finds the maximum value of gravity, finally, assigns each user to the relevant community.

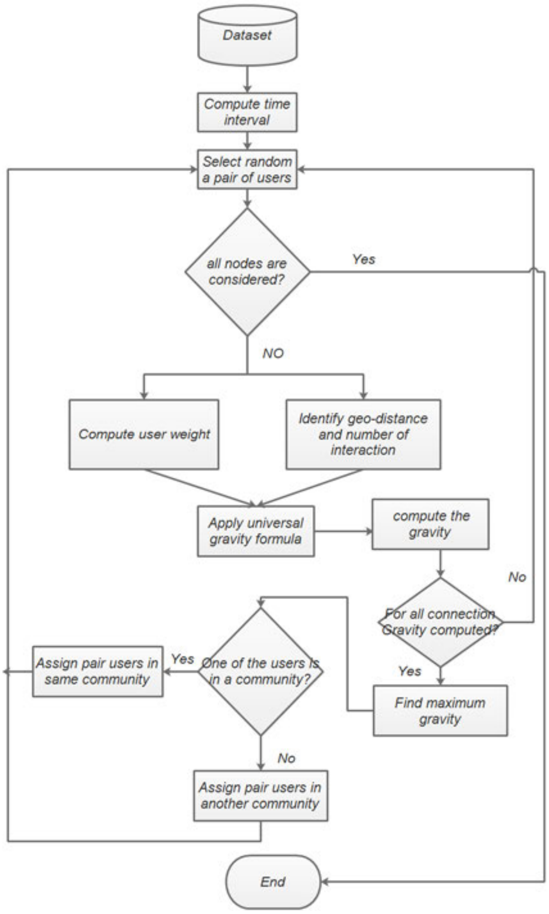
Dataset
Compute time interval
Select random a pair of users
all nodes are considered?
Yes
NO
Compute user weight
Identify geo-distance and number of interaction
Apply universal gravity formula
compute the gravity
No
For all connection Gravity computed?
Yes
Find maximum gravity
One of the users is in a community?
Yes
Assign pair users in same community
No
Assign pair users in another community
End

FIGURE 3. Flowchart of the proposed algorithm.

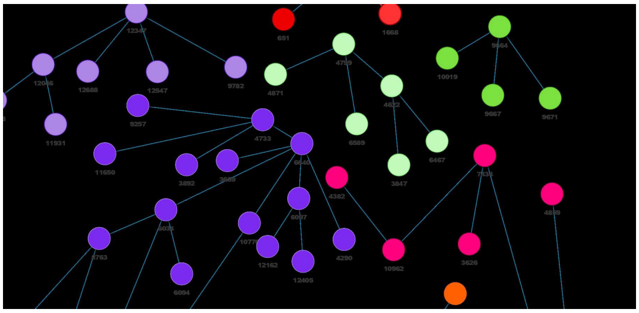

FIGURE 4. A part of the proposed CD algorithm' output for Travian dataset.

The pseudocode for the algorithms is presented below.

Figure 4 shows a part of the Gravity CD algorithm which experiments on the Travian dataset, the different color shows the different communities. As inference from the figure, the structure of the output of the algorithm is a tree.

A. CONVERT TO MAXIMUM SPANNING TREE (MST) STRUCTURE

After computing the gravity value based on (6), the proposed algorithm finds the maximum value of gravity for each pair

Algorithm 1 Procedure Recent Time Interval (Recenttbl)

Input: a time labeled dataset consisting of user communication (ComTbl)
Output: a dataset includes interaction and number of communication in recent time interval (IntTbl)

```
Sort ComTbl based on timestamp
$Std \leftarrow$ standard deviation of ComTbl based on timestamp field
$Last \leftarrow$ timestamp of last row
$First \leftarrow$ timestamp of first row
$Rangetime \leftarrow last - first$
$K \leftarrow ceiling(rangetime/std)$
$recentT \leftarrow first + (std * k - 1)$
$CominRecTime \leftarrow$ communication between recentTI and last# this is the communication in recent time interval
$N \leftarrow$ length (CominRecTime)
for   $i = 1$ to $N$ do
    Inset $user_1, user_2$ timestamp of row i of CominRecTime into RintTbl
end
$N \leftarrow length(RintTbl)$
for $i$ to $N$ do
    $user_1 \leftarrow RintTbl[i, 1]$
    $user_2 \leftarrow RintTbl[i, 2]$
    if  $(user_1$ & $user_2)$ is not in IntTbl then
        $Data \leftarrow$ rows in RintTbl where $user_1$ & $user_2$ are there
        $ctime \leftarrow NROW(data)$
        Insert $user_1, user_2$ and ctime into IntTbl
    end
end
```

of users. This is a fitness measure for selecting the edge for each pair of nodes, and this action converts a graph into an MST, so the output of the *RLI* algorithm is an MST.

Sub graph T of G is a spanning tree for G if T is a tree T includes all vertexes V of G

Theorem 1: Suppose G is an undirected and connected graph and for each edge in G a weight is assigned as gravity value. If T is the spanning tree of algorithm 2 then T is an MST.

Proof: It is proved by contradiction, suppose that $|U| = n$, T is spanning tree of algorithm 2 (*RLI*). The edges of T is labeled such as $e_1, e_2, e_3, \ldots e_{n-1}$ based on the order of output of *RLI* and gravity value are $grv_1, grv_2, grv_3, \ldots grv_{n-1}$ which are the maximum value of gravity for each node.

Suppose that T is not a MST, let $\acute{T}$ denotes the MST. Consider the edge $e_k = (a, b) \exists$ in $\acute{T}$ and in T and grv_k is the gravity value of e_k, it cannot be that $\acute{T}$ includes e_k since *RLI* only omit edges if they already add to T and the Gravity value between a and b is bigger than any other pair edges of a or b, so e_k in $\acute{T}$ create a cycle thus $\acute{T}$ cannot be a spanning tree. Therefore T includes e_k but $\acute{T}$ did not. In $\acute{T}$ there is a unique path p from a to b, edge e_k is not on this path. Further there is an edge $\acute{e_k}$ on p such that $grv_{\acute{e_k}} > grv_{e_k}$. The tree $\acute{T} \cup \{\acute{e}_k e_k$ is also a spanning tree and the gravity value strictly is heavier than $\acute{T}$. This contradicts the MST of $\acute{T}$, thus the $\acute{T} = T$.

Then a sequence of nodes in the new MST structure which connect formed a community. In this structure, the time of community traversal is reduced markedly. Also, the memory complexity is reduced because many passive connections are removed in the MST structure. Figure 4 shows how an MST structure is made by the *RLI* algorithm.

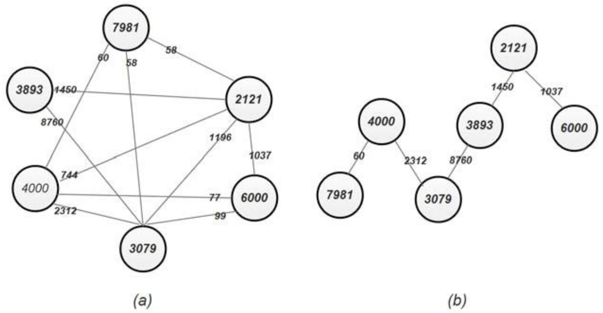

FIGURE 5. Convert graph structure (a) to MST structure (b).

Figure 5 (a) shows a graph that consists of the connections between six users in the Travian dataset, where the labels on the edges show the gravity values computed by (6). In the next step, an MST structure is made by selecting the max gravity between each pair of users (nodes); for instance, the gravity values between user 3893 and their friends (2121, 3079) are 1450 and 8760, respectively. Thus, for user 3893, user 3079 is selected. However, Fig. 5 (b) shows that there is a connection between user 3893 and 2121. This is because this connection has the highest gravity value for user 2121. This

Algorithm 2 RLI (Recent Largest Interaction)

Input: a time labeled dataset consisting of user interactions (IntTbl)
Output: a list of communities (GravityCommunityTbl)

```
$Icount = length(IntTbl)$
for $i = 1$ to $Icount$ do
    $user_1 \leftarrow InteractionTbl[i, 1]$
    $user_2 \leftarrow InteractionTbl[i, 2]$
    $dens \leftarrow InteractionTbl[i, 3]$
    $w_1 \leftarrow weight\ of\ user_1$
    $w_2 \leftarrow weight\ of\ user_2$
    $r_{1.2} \leftarrow$ distance between user$_1$ and user$_2$
    $\rho_{i.j} \leftarrow \frac{the\ communication\ times}{recent\ time\ interval}$ for $user_1$ and $user_2$
    $Grv \leftarrow ((w_1 * w_2) * \rho_{1.2})/r_{1.2}$
    Insert $user_1$, $user_2$ and $Grv$ into $GravityTbl$
end
$Gcount \leftarrow length(GravityTbl)$
for $i = 1$ to $Gcount$ do
    $user1 \leftarrow GravityTbl[i, 1]$
    $maxgrv \leftarrow Find\ max\ value\ of\ gravity\ for\ user1$
    $Pairuser \leftarrow pairofuser1$
    if ($user1\ and\ pairuser\ is\ not\ in\ MaxGrvTbl$) then
        Insert $user1$ and $pairuser$ into $MaxGrvTbl$
    end
    $user2 \leftarrow GravityTbl[i, 2]$
    $maxgrv \leftarrow Find\ max\ value\ of\ gravity\ for\ user2$
    $Pairuser \leftarrow pair\ of\ user2$
    if ($user2\ and\ pairuser\ is\ not\ in\ MaxGrvTbl$) then
        Insert $user2,\ pairuser$ and $maxgrv$ into $MaxGrvTbl$
    end
end
$Gcount \leftarrow length(MaxGrvTbl)$
for $i = 1$ to $Gcount$ do
    $nc() \leftarrow as\ an\ array$
    $Id1 \leftarrow 1\ \#id1\ is\ counter\ for\ array$
    $n[id1 \leftarrow MaxGrvTbl[i, 1]$
    While $id1 > 0$ do
        $node1 \leftarrow n[id1]$
        $id1 \leftarrow id1 - 1$
        If $node1\ is\ not\ in\ GravityComTbl$ then
            Insert $node1$ and $i$ as community number in $GravityComTbl$
            $NodeMaxGrvTbl \leftarrow find\ a\ list\ of\ rows\ of\ MaxGrvTbl\ in\ which\ one\ of\ the\ user\ is\ node1$
            $id = 1$
            While $(id <= length($NodeMaxGrvTbl$))$ do
                If $(NodeMaxGrvTbl[id, 1] = node1)$ then
                    $pairnode \leftarrow NodeMaxGrvTbl[id, 2]$
                    $PairGravityComTbl \leftarrow find\ a\ row\ of\ Gravity\ ComTbl\ in\ which\ node\ is\ pairnode$
                    If $(NROW(PairGravityComTbl) = 0)$ then
                        $id1 \leftarrow id1 + 1$
                        $n[id1] \leftarrow NodeMaxGrvTbl[id, 2]$
                    end
                else
                    Append other pairnode of $NodeMaxGrvTbl$ into $n$
                end
                $id \leftarrow id + 1$
            end
        end
        Erase $n$
    end
end
```

set of users (nodes), which have connections to each other, are thus considered to be a community.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the proposed method is experimented on two different datasets and the results are compared with eight different and well-known CD algorithm. Also, the study used six different datasets to show how the time interval is computed.

A. EXPERIMENT SETUP

In this section, we describe the datasets which is deployed in this study and a description of metrics and algorithms used to evaluate the proposed CD algorithm.

1) DATASETS

- The Facebook-like dataset [79] is available in [80]. ''The Facebook-like Social Network originate from an online community for students at the University of California, Irvine''. This dataset consists of 59,835 messages that were sent or received by 1899 users. Each row of the dataset contains the following information: (i) the time and date of sending or receiving the message, (ii) a number that shows the ID of the senders, (iii) the ID of the targeted users and (iv) the weight of each tie The data in this dataset are in the range of *4th* 2004 to *10th* 2004.
- ''The UKM dataset is a new synthetic dataset that consists of 234 users and 1079 interactions that encompass the interactions of a foreign student with their family, friends, colleagues, and university members, where the interactions between the student and other users cover the period 1/1/2016 to 1/8/2016. This dataset is time labeled, we consider 234 real users and make a random connection between them and other users based on their role.'' [49].
- Email-EU-core temporal datasets: the dataset generated by Jure Lescovec an associate professor at Stanford University [81]. The dataset consists of email data from a large European research institution. The publisher ''have anonymized information about all incoming and outgoing email between members of the research institution. The emails only represent communication between institution members (the core), and the dataset does not contain incoming messages from or outgoing messages to the rest of the world. A directed edge *(u, v, t)* means that person u sent an e-mail to person v at time t. A separate edge is created for each recipient of the e-mail. Node IDs in the sub-networks do not correspond to the same node ID in the entire network.'' [82] These datasets consist of 986,162, 89 and 142 users and 61046, 46772, 12216, and 48141 connections in department 1, 2, 3, and 4 of the institution respectively. The period of these datasets is 803 days.
- Travian network datasets: ''This network dataset was designed for studying multiplex network problems, in particular, community detection since the ground truth for alliance membership is available. It was extracted from data from a study of virtual organizations conducted by researchers across several institutions, including the University of Arkansas. Data was collected from a *3x* server (with a 3.5-month game cycle) based in Germany. These networks are from a 30-day game period.'' 83]. This dataset is time-based and also human-based. Besides, the Travian dataset is a real-world dataset and the ground truth of community structure is available. Hence, it is a suitable dataset for this study. This dataset consists of three files: (i) Attacks, which contains information about raids that have occurred between players, (ii) Messages, which contains information on the communications between players, and (iii) Trades, which represents the trades between players. We use the Messages file in our experiment as it best represents a social network between the users in this dataset.

A brief statistical review is presented in Table 1.

TABLE 1. Datasets statistic.

Dataset	Nodes	Temporal edges	Time Span
Facebook-like	1899	59,835	210 days
UKM	234	1,079	240 days
Email-EU dep1	309	61,046	803 days
Email-EU dep2	162	46,772	803 days
Email-EU dep3	89	12,216	802 days
Email-EU dep4	142	48,141	803 days
Travian (messages)	3092	451,589	1 month

2) MEASURES ADOPTED FOR COMPARING COMMUNITY DETECTION ALGORITHMS

We compared our proposed method with eight well-known algorithms, namely GN (edge-betweenness) [4], Infomap [8], label propagation [19], fast greedy [84], Walktrap [18], Louvain [6], leading eigenvector [85], and Dynamic Structural Clustering Algorithm for Network (DSCAN) [13]. The reasons for selecting these eight algorithms are: (i) the first seven algorithms are standard, meaning that they can be implemented using R programming language, so they can be considered as benchmark algorithms and are widely used in other studies. And the DSCAN is selected to compare our work with a time-based method. (ii) They represent a variety of approaches that have been used to solve the CD problem such as the LPA which used the propagation method, Walktrap [18], which uses a random walk, the fast greedy algorithm which uses the greedy solution, the GN algorithm [4] which uses betweenness centrality, Infomap [8] which uses a dynamic approach and can work on dynamic and weighted graphs, and the leading eigenvector algorithm [85], which uses the eigenvectors of the Laplacian graph. In addition, the RLI algorithm compares to the DSCAN algorithm [13], which is a time-based algorithm.

The results of these algorithms are compared using three standard measures, namely normalized mutual information (NMI), adjusted rand index (ARI) and Pairwise F measure (PWF). The NMI value is between 0 and 1, where the higher value shows a greater similarity between the two partitions (10).

$$\boldsymbol{NMI\,(A,B) = \frac{-2\sum_{i=1}^{R}\sum_{j=1}^{S} C_{ij}\log(C_{ij}n/C_{i.}C_{.j})}{\sum_{i=1}^{R} C_{i.}\log(C_{i.}/n)+\sum_{j=1}^{S} C_{.j}\log(C_{.j}/n)}} \tag{10}$$

A community structure on V is a partitioning $A = \{A_1 \ldots .A_R\}$ of V in several R subsets and V is the set of n nodes. C_{ij} denotes the number of nodes that clusters A_i and B_j share. If $A = B$ then $NMI(A, B) = 1$ and if A and B are completely different then $NMI(A, B) = 0$.

The ARI lies between -1 and 1. When two partitions agree perfectly, the ARI achieves the maximum value of 1. A larger ARI denotes a higher agreement between two partitions (11).

$$\boldsymbol{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right] - [\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}]/\binom{n}{2}} \tag{11}$$

The F-measure metric is widely used in experimental methods to determine the accuracy of methods designed to solve clustering problems. PWF is computed as follows (12):

$$\boldsymbol{PWF = \frac{2 * precision * recall}{precision + recall}} \tag{12}$$

The precision is the number of correct results divided by the number of all returned results (dimensionless) (13)

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \tag{13}$$

The recall is the number of correct results divided by the number of results that should have been returned (dimensionless) (14)

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \tag{14}$$

In community detection, the result of the classifier is TP if a node is proven to exist in the specific community and the test also shows that the node exists in that community. The result of the classifier is FP if the test for a node identifies as a member of the community which is not a real community member. The test result is FN if the result of the classifier shows that there is no community member for a set of users which contains community members.

The main reason for selecting these measures is that they are widely used for comparing CD algorithms so they can be used as benchmark measures.

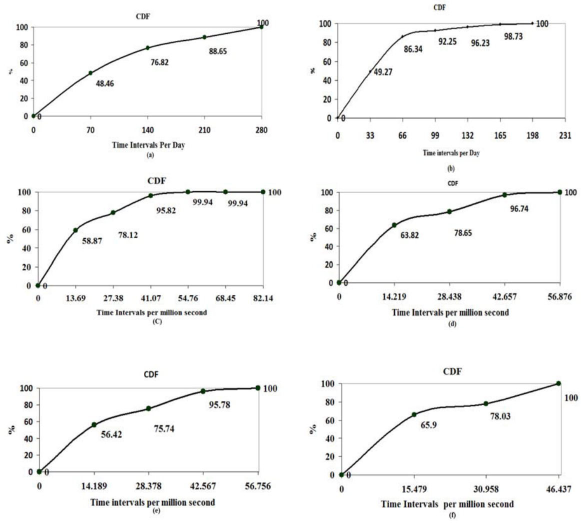

FIGURE 6. The CDF of the data distribution in each time interval based on the standard deviation for data sets, (a) UKM; (b) Facebook-like; (c) EU department 1 e-mails; (d) EU department 2 e-mails; (e) EU department 3 e-mails; (f) EU department 4 e-mails.

3) TIME INTERVAL COMPUTATION

The proposed algorithm was tested with respect to its ability to reduce memory and time complexity and its accuracy. However, before we discuss the results, it is necessary to describe the method of computation used to define the time interval.

Following study in [49], we compute the time interval based on a computation of the standard deviation of the data. We also assume that making a new connection with another user is a measure of a change in the community. We set a threshold value for defining recent time intervals (15), where time intervals that are more than the threshold are considered the latest time intervals and communities should be detected in those time intervals. In this study, we define the threshold value as 75% because a simulation of the proposed method on the six datasets used by [49] showed that this value can cover the abnormal connections made by users (Fig. 6).

$$\begin{aligned}\mathbf{P(x \le threshold) = CDF(X \le threshold)}\\ = \sum\nolimits_{x_i \le threshohld} p(X = x_i)\end{aligned} \tag{15}$$

B. EXPERIMENT TO EVALUATE ACCURACY AND SPACE COMPLEXITY

In this section, the proposed method in terms of accuracy and space complexity is evaluated.

1) EXPERIMENT TO FIND THE BEST ALGORITHM TO IDENTIFY THE COMMUNITIES

The development of a CD algorithm that is more accurate than existing standard algorithms is one of the main concerns of this study. As mentioned above, previous studies do not consider user attributes; they only consider the number of connections between the nodes to detect communities. In this

TABLE 2. Comparing of deferent cd algorithms based on NMI, ARI, and F-measure for Travian dataset.

Algorithm	ARI	NMI	F-measure
RLI	0.514	0.774	0.179
Leading Eigenvector	0.000	0.005	0.018
Fast greedy	0.123	0.508	0.072
Label propagation	0.105	0.513	0.20
Edge-Betweeness	0.060	0.679	0.139
Walktrap	0.259	0.712	0.20
Infomap	0.342	0.713	0.168
Louvain	0.218	0.629	0.072
DSCAN(μ=2)	0.020	0.578	0.171

TABLE 3. Comparing of deferent cd algorithms based on NMI, ARI, and F-measure for UKM dataset.

Algorithm	ARI	NMI	F-measure
RLI	0.616	0.78	0.462
Walktrap	0.317	0.653	0.362
Fast greedy	0.328	0.666	0.381
Leading Eigenvector	0.320	0.649	0.346
Label propagation	0.319	0.659	0.387
Edge-Betweeness	0.337	0.687	0.420
Infomap	0.340	0.693	0.418
Louvain	0.326	0.661	0.370
DSCAN(μ=2)	0.006	0.514	0.119

section, we evaluate the proposed method from the viewpoint of the accuracy of the algorithm in detecting communities. To do this, we compute the user weight based on a simple exponential smoothing model for every user in the UKM dataset and Travian dataset.

The communities identified by the proposed RLI algorithm are then compared with those detected by the GN (Edge-betweenness), Walktrap, Label propagation, Infomap, Fast greedy, and Leading Eigenvector, Louvain and DSCAN algorithms. In this part, we only use the UKM dataset, which is synthetic, and the Travian dataset, which is a real-world dataset, because the ground truth of the other datasets is not available. Tables 2 and 3 show the results of the RLI algorithm and the above-mentioned eight comparable algorithms on the Travian and UKM datasets, respectively. The results show that the proposed algorithm outperforms the eight other algorithms with regards to NMI, ARI, and PWF value for both datasets.

It should also be noted that we used the default community detection functions in the R programming language that were released in 2019 (Action of the Toes version).

Tables 4 and 5 show the results of the proposed RLI algorithm without considering the four important features highlighted by this research, such as the recent time interval, distance, user weight, and density of interaction. Interestingly, when the recent time interval is considered, the result is better, specifically, the ARI value. However, when the RLI algorithm is applied to the UKM dataset without knowledge of distance, it cannot find a unique edge for each connection. The RLI algorithm performed significantly better without these features when it is applied to a larger dataset, such as the Travian dataset. These results indicate that the RLI algorithm can be effectively applied to both small and large datasets by considering different features. Moreover, in the case of small datasets, the RLI algorithm needs to consider all the features (Time, user weight, the density of interaction, and geo-distance) to obtain a result. On the other hand, when it is applied to large datasets, given that the algorithm leverages the gravity model, higher accurate time interval and an accurate user weight can be reliably achieved. These two tables show that the effect of the density of interaction is more prominent rather than others, where without this attribute, the NMI and ARI value decrease markedly for both datasets.

Furthermore, Table 4 also shows that when the user weight in the RLI is computed by user degree centrality, lesser accuracy is obtained relative to the time-based user weight proposed in this research. Consequently, the features (i.e., the three user attributes and time) that are included

TABLE 4. Comparison of the performance of the proposed algorithm with and without considering the four important features when applied to the Travian dataset.

Algorithm	**ARI**	**NMI**	**Community number**
RLI	0.51	0.77	68
RLI without considering recent time interval	0.29	0.63	52
RLI by considering user weight based on degree centrality	0.27	0.65	59
RLI without considering the density of interaction	0.1	0.47	47

TABLE 5. Comparison of the performance of the proposed algorithm with and without considering the four important features when applied to the UKM dataset.

Algorithm	**ARI**	**NMI**	**Community number**
RLI	0.61	0.78	10
RLI without considering recent time interval	0.23	0.41	5
Use weight based on degree centrality	0.21	0.45	6
RLI without considering the density of interaction	0.15	0.35	4
RLI without considering distance	Cannot find unique maximum gravity (leads to fail to detect communities)		

TABLE 6. Comparing RLI and gravity method based on shortest path.

Dataset	ARI	NMI	PWF
UKM (based on geo-location)	0.61	0.78	0.462
UKM (based on shortest path)	0.44	0.62	0.19
Travian (without shortest path and 894,560 sample connection of Travian)	0.55	0.79	0.17
Travian (shortest path consideration and 894,560 sample connection of Travian)	0.53	0.77	0.15

in the proposed RLI algorithm contribute to improving the accuracy of the CD. Besides, the number of communities decreases when we run the proposed algorithm without each above-mentioned feature.

As mentioned in section IV of this manuscript, geo-location is used to identify the distance between nodes. In fact, propinquity is the basis of the RLI algorithm. The term propinquity is used in social psychology in which it is understood that people who live near to each other are more likely to be friends [86]. This measure is important because it shows the effect of geo-distance on the formation of communities in OSNs, which deviate from several studies that use the shortest path as the main measure for identifying distance in the network [4], [87]. Table 6 shows the comparison between the RLI algorithm when it considers the shortest path and when it considers geo-distance. Due to the size of the Travian dataset (4, 778, 686 unique edges), the experimental process was based on sample data (894,560 random possible connections between nodes) of this dataset. The shortest path between all pair of nodes is computed using the expression in (16)

$$\textit{number of possible connection} = \frac{n*(n-1)}{2} \tag{16}$$

There are 4, 778, 686 possible edges (connections) between nodes in Travian dataset, where n is 3092 (nodes).

Table 6 shows that when we consider the shortest path as the measure for distance, the NMI, ARI, and PWF value for the UKM dataset decrease significantly. Similarly, the value of these three measures also decreases for the Travian dataset. In addition, the complexity of computing the shortest path is very high, while the computational complexity of the geo-distance is significantly low.

The results of the RLI algorithm is statistically significant given that the attributes used by the RLI algorithm represent the user behavior. Therefore, this study posits that an OSN community can be defined as follows: A community is a group of users that exhibits the highest gravity among its members relative to the members in the network.

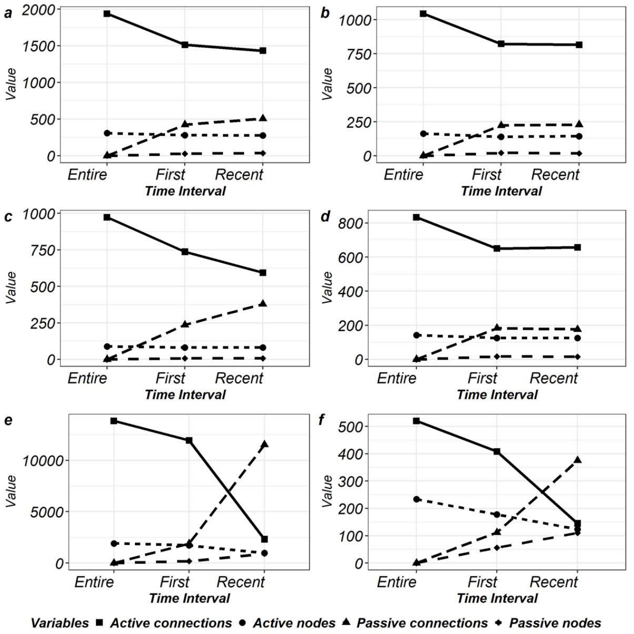

FIGURE 7. Number of passive connections and nodes; (a) EU department 1 e-mails; (b) EU department 2 e-mails; (c) EU department 3 e-mails; (d) EU department 4 e-mails; (e) UKM; (f) Facebook-like.

2) MEMORY COMPLEXITY EVALUATION

The result produced by the proposed algorithm based on active and passive nodes and edges is shown in Fig 7 (a)-(f). The charts in the figure show the number of passive connections and users for each of the six datasets. During the period covered, many users are passive and do not have any interactions with each other. For instance, if we consider the Facebook-like dataset (Fig. 7(e)), there is a total of 13838 connections between users: in the initial time intervals (first and second) the number of connections between users is 11945 and the number of active users is 1712, which means that the algorithm has removed 1893 passive connections. In other words, 183 nodes have been removed because they do not have interactions with each other in those two-time intervals. A trend of increasing inactivity among the user community continues over the period covered to reach 11522 passive connections in the more recent time intervals (from 22 June 2004 to 30 October 2004) with only 990 users having interactions with each other. Hence it is clear that when attempting to detect meaningful communities we can remove a lot of passive connections in the recent time intervals, which leads to a reduction in memory complexity. This result proves that the modularity measure, which considers all the edges, has high memory complexity when applied to OSNs. This complexity can be reduced by the proposed algorithm where, in the worst-case scenario, only around 50% of the nodes have interactions with each other in recent time intervals.

Figure 8 shows the number of connections in each of the six datasets after running the proposed algorithm.

It can be seen from the figure that, in the case of the Facebook-like dataset, for example, roughly 93% of connections are removed by the *RLI* algorithm. Overall, the result proves that many connections are pruned by the proposed algorithm because many users do not have any interactions with each other in recent time intervals. In other words,

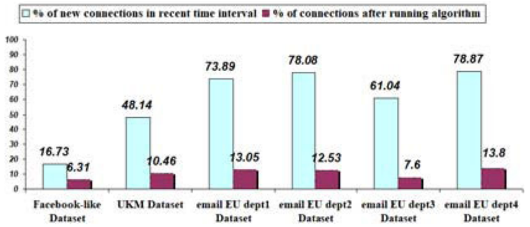

FIGURE 8. Percentage of connections pruned after running max-gravity function.

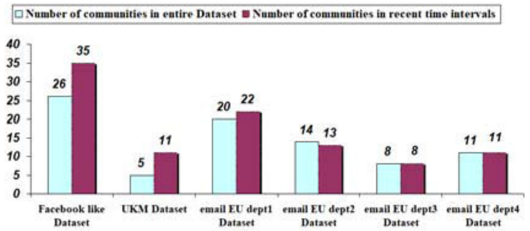

FIGURE 9. Number of communities in each time interval.

the proposed algorithm reduces the memory complexity in each community. This indicates that the connections in the recent time intervals are more true to life than those in other time intervals especially when considering an entire dataset.

Figure 9 shows the number of communities in the recent time interval that are detected by the proposed algorithm.

The result in the figure shows that the number of communities in the entire dataset is somewhat similar to that in the recent time intervals. In other words, although the proposed algorithm prunes the passive connections and nodes in each time interval, the number of communities in each time interval is to some extent the same.

At the end of this section, we evaluated the time complexity of the proposed algorithm. The RLI algorithm has five main sections. Algorithm 1 shows that the table of the interactions between users in the recent time interval is computed; this is done in $O(E)$. The second loop of this algorithm computes the number of interactions between nodes of recent time interval in the worst case this is done in $O(E)$. Algorithm 2 at the first section computes the gravity value for all the connections in the recent time interval; this is also done in $O(E)$. The second loop of algorithm 2 computes the maximum value of the gravity for each pair of users in the recent time interval, which is done in $O(E)$, and finally, we identify the community, which is done in $O(n \log_{10} E)$, where the experimental results show that the $O(n \log_{10} E)$ roughly equates to the number of connections in the output of the second loop of algorithm 2 which computes the maximimum value of gravity. This is because it removes a significant number of connections. The proposed algorithm has two phases. In the first phase, it removes some edges by considering the connections that are present only in the recent time interval, and in the second phase, it removes some edges by considering the connection with the highest value of gravity. The output of the *RLI* algorithm is a maximum spanning tree (MST), where the number of edges in the MST structure is *n-1*. Thus the total complexity of the RLI algorithm is $O(4E + n \log_{10} E)$. The time complexity of some other existing CD algorithms is provided in Table 7 [88].

TABLE 7. Time complexity of existing community detection algorithms.

Algorithm	Time complexity
Eigenvector	$O(N^2)$
Walktrap	$O(N^2 \log_2 N)$
Fast greedy	$O(N \log_2 N)$
RLI	$O(E + N \log_{10} E)$
Infomap	$O(E)$
Betweeness	$O(E^2 N)$
Label propagation	$O(E)$
DSCAN	$O(N^2)$
Louvain	$O(N^2)$

VII. CONCLUSION

This study sheds light on the importance of considering user attributes in attempts to find solutions to problems such as the CD problem in OSNs. Existing studies mostly overlooked the role of user attributes (lifespan, distance, interaction density, and user weight). Interestingly, these attributes are similar to the universal gravity formula, which provided a substratum for the development of our novel approach: the recently largest interaction (RLI) algorithm, which consists of five sub algorithms. The results showed that the proposed RLI algorithm can detect communities better than the eight standard existing algorithms and that it can reduce memory complexity significantly. Memory complexity was reduced by using only the active connections in recent time intervals, whereas the edge-based existing algorithms and others like it, which use the modularity metric, consider all the connections between nodes and therefore have high memory complexity.

The proposed algorithm also reduced time complexity compared to the Infomap, Louvain, edge-betweenness and label propagation algorithms and similar other algorithms, which have a time complexity of $O(N^3)$. The study also revealed that the gravity metric is more suitable than the modularity metric for detecting communities in OSNs. Also, the output of the proposed algorithm is an MST, which improves the search of communities compared to a graph structure output. The results also showed that the recent time interval communities are more realistic than in other time intervals. The findings of this study could be used to overcome the problem of big data in OSNs and could be used to make an efficient recommender system. A future research direction could involve trying to detect life events based on communities, which may results in an improvement over existing event detection methods that mostly try to detect events based on a set of keywords because the text in an OSN data stream is noisy and often contains non-standard acronyms that are very hard to utilize in a meaningful way. Thus, this study paves the way for further research on network science in which the most important players are humans because it demonstrates that user attributes must be considered to improve the accuracy of proposed methods.

REFERENCES

[1] A. Mahmoudi, M. R. Yaakub, and A. A. Bakar, "The relationship between online social network ties and user attributes," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 3, pp. 1–15, Jul. 2019, doi: 10.1145/3314204.

[2] A. Mahmoudi, M. R. Yaakub, and A. A. Bakar, "A new real-time link prediction method based on user community changes in online social networks," *Comput. J.*, vol. 63, no. 3, pp. 448–459, Mar. 2020, doi: 10.1093/comjnl/bxz050.

[3] N. Jamil, S. A. M. Noah, and M. Mohd, "Collaborative item recommendations based on friendship strength in social network," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 3, pp. 437–443, 2020.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002, doi: 10.1073/pnas.122653799.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066133, doi: 10.1103/PhysRevE.69.066133.

[6] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, pp. 1–12, 2008, doi: 10.1088/1742-5468/2008/10/P10008.

[7] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066111, doi: 10.1103/PhysRevE.70.066111.

[8] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[9] F. Hu and Y. Liu, "A novel algorithm infomap-SA of detecting communities in complex networks," *J. Commun.*, vol. 10, no. 7, pp. 503–511, 2015, doi: 10.12720/jcm.10.7.503-511.

[10] T. N. Dinh and M. T. Thai, "Community detection in scale-free networks: Approximation algorithms for maximizing modularity," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 997–1006, Jun. 2013, doi: 10.1109/JSAC.2013.130602.

[11] T. You, H.-M. Cheng, Y.-Z. Ning, B.-C. Shia, and Z.-Y. Zhang, "Community detection in complex networks using density-based clustering algorithm and manifold learning," *Phys. A, Stat. Mech. Appl.*, vol. 464, pp. 221–230, Dec. 2016, doi: 10.1016/j.physa.2016.07.025.

[12] C. Liu and Q. Liu, "Community detection based on differential evolution using modularity density," *Information*, vol. 9, no. 9, p. 218, Aug. 2018.

[13] N. Aston and W. Hu, "Community detection in dynamic social networks," *Commun. Netw.*, vol. 6, no. 2, pp. 124–136, 2014, doi: 10.4236/cn.2014.62015.

[14] H. Dev, M. E. Ali, and T. Hashem, "User interaction based community detection in online social networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 8422, 2014, pp. 296–310, doi: 10.1007/978-3-319-05813-9_20.

[15] T. Hecking, T. Göhnert, S. Zeini, and U. Hoppe, "Task and time aware community detection in dynamically evolving social networks," *Procedia Comput. Sci.*, vol. 18, pp. 2066–2075, Jan. 2013.

[16] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai, "Dynamic social community detection and its applications," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e91431, doi: 10.1371/journal.pone.0091431.

[17] S. H. S. Javadi, S. Gharani, and P. Khadivi, "Detecting community structure in dynamic social networks using the concept of leadership," in *Sustainable Interdependent Networks*, P. Amini, M. Boroojeni, K. Iyengar, S. Pardalos, Ed. Cham, Switzerland: Springer, 2018, pp. 97–118.

[18] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006, doi: 10.7155/jgaa.00124.

[19] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, pp. 1–12, Sep. 2007, doi: 10.1103/PhysRevE.76.036106.

[20] *Statista*. Accessed: Oct. 20, 2017. [Online]. Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[21] T. McCue. (2013). Social media maximum: 150 friends. Forbes. Accessed: Dec. 19, 2018. [Online]. Available: https://www.forbes.com/sites/tjmccue/2013/01/15/social-media-maximum-150-friends/#42b2fec83aec

[22] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, "The complexity of searching a graph," *J. ACM*, vol. 35, no. 1, pp. 18–44, Jan. 1988, doi: 10.1145/42267.42268.

[23] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized Louvain method for community detection in large networks," in *Proc. Int. Conf. Intell. Syst. Design Appl. (ISDA)*, 2011, pp. 88–93, doi: 10.1109/ISDA.2011.6121636.

[24] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proc. 4th ACM Eur. Conf. Comput. Syst. (EuroSys)*, 2009, pp. 205–218, doi: 10.1145/1519065.1519089.

[25] J. Wu, X. Li, L. Jiao, X. Wang, and B. Sun, "Minimum spanning trees for community detection," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 9, pp. 2265–2277, May 2013, doi: 10.1016/j.physa.2013.01.015.

[26] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–37, Jun. 2018.

[27] Y. Ding, "Community detection: Topological vs. topical," *J. Informetrics*, vol. 5, no. 4, pp. 498–514, Oct. 2011, doi: 10.1016/j.joi.2011.02.006.

[28] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 2, pp. 84–98, 2016, doi: 10.1504/IJBIC.2016.076329.

[29] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 6, no. 3, pp. 115–135, May 2016.

[30] M. Planti and M. Crampes, "Survey on social community detection," in *Social Media Retrieval*. London, U.K.: Springer, 2013, pp. 65–85. [Online]. Available: https://doi.org/10.1007/978-1-4471-4555-4_4

[31] A. Rezaei, S. M. Far, and M. Soleymani, "Controlled label propagation: Preventing over-propagation through gradual expansion," 2015, *arXiv:1503.04694*. [Online]. Available: http://arxiv.org/abs/1503.04694

[32] X.-K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, "Label propagation algorithm for community detection based on node importance and label influence," *Phys. Lett. A*, vol. 381, no. 33, pp. 2691–2698, Sep. 2017, doi: 10.1016/j.physleta.2017.06.018.

[33] N. Chen, Y. Liu, H. Chen, and J. Cheng, "Detecting communities in social networks using label propagation with information entropy," *Phys. A, Stat. Mech. Appl.*, vol. 471, pp. 788–798, Apr. 2017, doi: 10.1016/j.physa.2016.12.047.

[34] H. S. Pattanayak, A. L. Sangal, and H. K. Verma, "Community detection in social networks based on fire propagation," *Swarm Evol. Comput.*, vol. 44, pp. 31–48, Feb. 2019, doi: 10.1016/j.swevo.2018.11.006.

[35] S. Boudebza, R. Cazabet, F. Azouaou, and O. Nouali, "OLCPM: An online framework for detecting overlapping communities in dynamic social networks," *Comput. Commun.*, vol. 123, pp. 36–51, Jun. 2018, doi: 10.1016/j.comcom.2018.04.003.

[36] Z. Zhao, C. Li, X. Zhang, F. Chiclana, and E. H. Viedma, "An incremental method to detect communities in dynamic evolving social networks," *Knowl.-Based Syst.*, vol. 163, pp. 404–415, Jan. 2019, doi: 10.1016/j.knosys.2018.09.002.

[37] S. Y. Bhat and M. Abulaish, "HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 4, pp. 1019–1031, Apr. 2015, doi: 10.1109/TKDE.2014.2349918.

[38] Z. Bu, C. Zhang, Z. Xia, and J. Wang, "A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network," *Knowl.-Based Syst.*, vol. 50, pp. 246–259, Sep. 2013, doi: 10.1016/j.knosys.2013.06.014.

[39] R. Liu, S. Feng, R. Shi, and W. Guo, "Weighted graph clustering for community detection of large social networks," *Procedia Comput. Sci.*, vol. 31, pp. 85–94, Jan. 2014, doi: 10.1016/j.procs.2014.05.248.

[40] A. Kanavos, I. Perikos, I. Hatzilygeroudis, and A. Tsakalidis, "Emotional community detection in social networks," *Comput. Electr. Eng.*, vol. 65, pp. 449–460, Jan. 2018, doi: 10.1016/j.compeleceng.2017.09.011.

[41] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, ''Fast unfolding of communities in large networks,'' *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008, doi: 10.1088/1742-5468/2008/10/P10008.

[42] J. Zhu, B. Wang, B. Wu, and W. Zhang, ''Emotional community detection in social network,'' *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 10, pp. 2515–2525, 2017.

[43] T. Ma, Y. Wang, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, ''LED: A fast overlapping communities detection algorithm based on structural clustering,'' *Neurocomputing*, vol. 207, pp. 488–500, Sep. 2016, doi: 10.1016/j.neucom.2016.05.020.

[44] X. Li, S. Zhou, J. Liu, G. Lian, G. Chen, and C.-W. Lin, ''Communities detection in social network based on local edge centrality,'' *Phys. A, Stat. Mech. Appl.*, vol. 531, Oct. 2019, Art. no. 121552, doi: 10.1016/j.physa.2019.121552.

[45] W. Rafique, M. Khan, N. Sarwar, and W. Dou, ''SocioRank*: A community and role detection method in social networks,'' *Comput. Electr. Eng.*, vol. 76, pp. 122–132, Jun. 2019, doi: 10.1016/j.compeleceng.2019.03.010.

[46] X. Li, G. Xu, and M. Tang, ''Community detection for multi-layer social network based on local random walk,'' *J. Vis. Commun. Image Represent.*, vol. 57, pp. 91–98, Nov. 2018, doi: 10.1016/j.jvcir.2018.10.003.

[47] H. Zardi and L. B. Romdhane, ''An O(n2) algorithm for detecting communities of unbalanced sizes in large scale social networks,'' *Knowl.-Based Syst.*, vol. 37, pp. 19–36, Jan. 2013, doi: 10.1016/j.knosys.2012.05.021.

[48] R. Sharma and S. Oliveira, ''Community detection algorithm for big social networks using hybrid architecture,'' *Big Data Res.*, vol. 10, pp. 44–52, Dec. 2017, doi: 10.1016/j.bdr.2017.10.003.

[49] A. Mahmoudi, M. R. Yaakub, and A. A. Bakar, ''New time-based model to identify the influential users in online social networks,'' *Data Technol. Appl.*, vol. 52, no. 2, pp. 278–290, Apr. 2018, doi: 10.1108/DTA-08-2017-0056.

[50] T. Pei, Y. Cao, Z. Li, and G. Zhu, ''Overlapping community detection by local community gravitation in social networks,'' *J. Netw.*, vol. 9, no. 9, pp. 2360–2364, Sep. 2014, doi: 10.4304/jnw.9.9.2360-2364.

[51] C. Yang, M. Li, and Y. Wang, ''Overlapping community detection algorithm based on the law of universal gravitation,'' in *Proc. MATEC Web Conf.*, vol. 22, 2015, pp. 1–5, doi: 10.1051/matecconf/20152201056.

[52] Q. Cai, M. Gong, B. Shen, L. Ma, and L. Jiao, ''Discrete particle swarm optimization for identifying community structures in signed social networks,'' *Neural Netw.*, vol. 58, pp. 4–13, Oct. 2014, doi: 10.1016/j.neunet.2014.04.006.

[53] S. Gupta, S. Mittal, T. Gupta, I. Singhal, B. Khatri, A. K. Gupta, and N. Kumar, ''Parallel quantum-inspired evolutionary algorithms for community detection in social networks,'' *Appl. Soft Comput.*, vol. 61, pp. 331–353, Dec. 2017, doi: 10.1016/j.asoc.2017.07.035.

[54] A. Mahmoudi, M. R. Yaakub, and A. A. Bakar, ''A new method to discretize time to identify the milestones of online social networks,'' *Social Netw. Anal. Mining*, vol. 8, no. 1, p. 34, Dec. 2018, doi: 10.1007/s13278-018-0511-4.

[55] R. L. Graham and P. Hell, ''On the history of the minimum spanning tree problem,'' *IEEE Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, Jan./Mar. 1985, doi: 10.1109/MAHC.1985.10011.

[56] Cambridge, U.K. (2019). *Cambridge Dictionary*. Accessed: Feb. 24, 2019. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/community

[57] M. E. J. Newman and M. Girvan, ''Finding and evaluating community structure in networks,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[58] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, ''Defining and identifying communities in networks,'' *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004, doi: 10.1073/pnas.0400054101.

[59] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, ''GSA: A gravitational search algorithm,'' *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[60] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, ''BGSA: Binary gravitational search algorithm,'' *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.

[61] T. Ghose. (2013). *Livescience*. Accessed: Oct. 29, 2018. [Online]. Available: http://www.livescience.com/37115-what-is-gravity.html

[62] Z. Huang and Y. Liu, ''Community detection from location-tagged networks,'' 2015, *arXiv:1501.04675*. [Online]. Available: https://arxiv.org/abs/1501.04675

[63] M. Allamanis, S. Scellato, and C. Mascolo, ''Evolution of a location-based online social network: Analysis and models,'' in *Proc. ACM Conf. Internet Meas. Conf. (IMC)*, 2012, pp. 145–158, doi: 10.1145/2398776.2398793.

[64] E. Cho, S. A. Myers, and J. Leskovec, ''Friendship and mobility: User movement in location-based social networks,'' in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 1082–1090, doi: 10.1145/2020408.2020579.

[65] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, ''Bridging the gap between physical location and online social networks,'' in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, Sep. 2010, pp. 119–128, doi: 10.1145/1864349.1864380.

[66] A. Kaltenbrunner, S. Scellato, Y. Volkovich, D. Laniado, D. Currie, E. J. Jutemar, and C. Mascolo, ''Far from the eyes, close on the Web: Impact of geographic distance on online social interactions,'' in *Proc. ACM Workshop Online Social Netw. (WOSN)*, 2012, pp. 19–24, doi: 10.1145/2342549.2342555.

[67] Á. B. Lengyel, A. Varga, B. Ságvári, and Á. Jakobi, ''Distance dead or alive: Online social networks from a geography perspective,'' Int. Bus. School, Budapest, Hungary, Tech. Rep. 1/2013, 2013. [Online]. Available: https://www.ibs-.hu/data/downloads/2015/09/10/OSON_WP2_iWiWdistance.pdf

[68] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, ''Geographic routing in social networks,'' *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 33, pp. 11623–11628, 2005, doi: 10.1073/pnas.0503018102.

[69] C. Scellato, S. Noulas, A. Lambiotte, and R. Mascolo, ''Socio-spatial properties of online location-based social networks,'' in *Proc. ICWSM*, 2011, pp. 329–336.

[70] F. Erlandsson, P. Bródka, A. Borg, and H. Johnson, ''Finding influential users in social media using association rule learning,'' *Entropy*, vol. 18, no. 5, pp. 1–15, 2016, doi: 10.3390/e18050164.

[71] S. Hangal, D. MacLean, M. S. Lam, and J. Heer, ''All friends are not equal: Using weights in social graphs to improve search,'' in *Proc. 4th SNA-KDD Workshop*, vol. 10, 2010, pp. 1–7. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.298.1136

[72] J. Heidemann, M. Klier, and F. Probst, ''Identifying key users in online social networks: A pagerank based approach,'' in *Proc. 31st Int. Conf. Inf. Syst. (ICIS)*, 2010, pp. 1–21.

[73] M. U. Ilyas and H. Radha, ''Identifying influential nodes in online social networks using principal component centrality,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5, doi: 10.1109/icc.2011.5963147.

[74] Z. Jianqiang, G. Xiaolin, and T. Feng, ''A new method of identifying influential users in the micro-blog networks,'' *IEEE Access*, vol. 5, pp. 3008–3015, 2017.

[75] M. Z. Shafiq, M. U. Ilyas, A. X. Liu, and H. Radha, ''Identifying leaders and followers in online social networks,'' *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 618–628, Sep. 2013, doi: 10.1109/JSAC.2013.SUP.0513054.

[76] R. Zafarani, M. Abbasi, and H. Liu, ''Network measures,'' in *Social Media Mining: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2014, pp. 51–79, doi: 10.1017/CBO9781139088510.004.

[77] P. Lambiotte, R. Blondel, V. D. D. Kerchove, C. Huens, E. Prieur, C. Smoreda, and Z. V. Dooren, ''Geographical dispersal of mobile telecommunication networks,'' *Phys. A, Stat. Mech. Appl.*, vol. 387, no. 21, pp. 5317–5325, 2008.

[78] (2018). *Worldometers*. Accessed: Jul. 3, 2018. [Online]. Available: http://www.worldometers.info/population/largest-cities-in-the-world/

[79] T. Opsahl and P. Panzarasa, ''Clustering in weighted networks,'' *Social Netw.*, vol. 31, no. 2, pp. 155–163, May 2009.

[80] T. Opshal. (2009). *Tore Opsahl*. Accessed: Feb. 18, 2018. [Online]. Available: https://toreopsahl.com/datasets/#online_social_network

[81] A. Paranjape, A. R. Benson, and J. Leskovec, ''Motifs in temporal networks,'' in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 601–610.

[82] J. Leskovec. (2017). *Stanford Network Analysis Platform (SNAP)*. Accessed: Oct. 5, 2017. [Online]. Available: https://snap.stanford.edu/data/email-Eu-core-temporal.html

[83] N. Hajibagheri, A. Sukthankar, G. Lakkar, K. Alvari, H. Wigand, and R. T. Agarwal, ''Using massively multiplayer online game data to analyze the dynamics of social interactions,'' in *Social Interactions inVirtual Worlds*. Cambridge, U.K.: Cambridge Univ. Press, 2018, pp. 375–416.

[84] M. E. J. Newman and M. Girvan, ''Finding and evaluating community structure in networks,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, pp. 1–16, Feb. 2004, doi: 10.1103/PhysRevE.69.026113.

[85] M. E. J. Newman, ''Finding community structure in networks using the eigenvectors of matrices,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104, doi: 10.1103/PhysRevE.74.036104.

[86] R. Reagans, "Close encounters: Analyzing how social similarity and propinquity contribute to strong network connections," *Org. Sci.*, vol. 22, no. 4, pp. 835–849, Aug. 2011, doi: 10.1287/orsc.1100.0587.

[87] S. Bhattacharyya and P. J. Bickel, "Community detection in networks using graph distance," Jan. 2014, *arXiv:1401.3915*. [Online]. Available: http://arxiv.org/abs/1401.3915

[88] Z. Yang, R. Algesheimer, and C. J. Tessone, "A comparative analysis of community detection algorithms on artificial networks," *Sci. Rep.*, vol. 6, no. 1, Aug. 2016, Art. no. 30750, doi: 10.1038/srep30750.

AMIN MAHMOUDI received the master's degree in management information system from Shiraz University, in 2014, and the Ph.D. degree in computer science from UKM University, Malaysia, in 2019. From 2018 to 2019, he was a Research Assistant with the Sentiment Analysis Laboratory, UKM University. From October 2019 to December 2019, he was a Visiting Researcher with the Czech Academy of Science. He is currently a Research Assistant Professor of data science with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong. His research interests include the network science, social network analysis, and data science.

AZURALIZA ABU BAKAR received the Ph.D. degree in artificial intelligence from Universiti Putra Malaysia, in 2002. She has been a Professor of data mining with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, since 2011. She has served as an Advisor for the Data Mining and Optimization Lab and a member of the Sentiment Analysis Lab. She has lead 13 research projects (including three in progress) and a member of 42 research projects. She is a member of the IEEE Computational Intelligence Society. She has also served on roughly 30 conference and workshop program committees and served as the Program Chair. Her main research interests include data analytics and artificial intelligence specifically in rough set theory, feature selection algorithms, nature inspired computing, and sentiment analysis.

MEHDI SOOKHAK (Senior Member, IEEE) received the Ph.D. degree in computer science, major in information security, from the University of Malaya (UM), in 2015.

He was an Active Researcher with the Center of Mobile Cloud Computing Research (C4MCCR), UM. From 2016 to 2017, he was with Carleton University, Canada, as a Postdoctoral Fellow. He is currently an Assistant Professor of cybersecurity with Illinois State University, Normal, IL, USA. He has authored more than 30 articles in high ranking journals and conferences. His areas of interest include cloud and mobile cloud computing, fog computing, vehicular cloud computing, the IoT and smart cities, computation outsourcing, access control, network security, wireless sensor & mobile Ad Hoc network (architectures, protocols, security, and algorithms), big data security and analytic, distributed systems, and cryptography and information security.

MOHD RIDZWAN YAAKUB is currently a Senior Lecturer with the Center for Artificial Intelligence and Technology (CAIT), Faculty of Information Science and Technology, National University of Malaysia (FTSM, UKM). His expertise is Sentiment Analysis/Opinion Mining, Feature Selection, Feature Extraction, Ontology, and Data Mining. A Ph.D. holder from the Queensland University of Technology (QUT), Australia. He is also the Head Researcher with the Sentiment Analysis Lab, CAIT. In administration, he is a Postgraduate Coordinator with CAIT.