IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 September 2022, accepted 11 October 2022, date of publication 3 November 2022, date of current version 14 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219107

RESEARCH ARTICLE

Rank-Label Anonymization for the Privacy-Preserving Publication of a Hypergraph Structure

DEBASIS MOHAPATRA[1], SOURAV KUMAR BHOI[1], KALYAN KUMAR JENA[1], KSHIRA SAGAR SAHOO[2], (Member, IEEE), ANAND NAYYAR[3], AND MOHD ASIF SHAH[4]

[1]Parala Maharaja Engineering College, Berhampur (Government), Berhampur, Odisha 761003, India
[2]Department of Computing Science, Umeå University, 901 87 Umeå, Sweden
[3]Faculty of Information Technology, Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[4]Bakhtar University, Kabul 1001, Afghanistan

Corresponding authors: Anand Nayyar (anandnayyar@duytan.edu.vn) and Mohd Asif Shah (ohaasif@bakhtar.edu.af)

This work was supported in part by the Kempe Fellowship, Sweden, under Project SMK21-0061; and in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) through the Knut and Alice Wallenberg Foundation.

ABSTRACT Social networks are often published in the form of a simple graph. The simple graph representation of a social graph shows the dyadic relationship among the social entities whereas it is unable to efficiently represent the relationship among more than two entities, such as the relationship found in the social groups. This type of relationship is called super-dyadic relationship, and it can be effectively represented by a hypergraph model. This work proposes an anonymization scheme called rank-label anonymization for the privacy-preserving publication of a hypergraph structure. Here, an attack model called rank-label attack is proposed, and an anonymization solution is provided to counter this attack. The percentage of disclosure risk shows that the rank-label attack is stronger than the existing rank attack. We propose a method based on sequential clustering to achieve rank-label anonymization called sequential rank-label anonymization (SA). Another algorithm called greedy rank-label anonymization (GA) is also proposed. The quality of the anonymization solution reported by SA and GA is compared with the help of normalized anonymization cost (*NCost*). Results show that the *NCost* reported by SA is less than that of GA for both Adult and MAG-10 datasets. In Adult dataset, approximately 58% and 62% reduction in the average execution time of GA and SA are obtained than that of a general-purpose computing system due to the use of a high-performance computing system. In MAG-10 dataset, this average reduction percentage is reported to be 56% for GA and 53% for SA. The time complexity of SA is found to be $O(n^4)$ whereas it is $O(n^3)$ in case of GA.

INDEX TERMS Anonymity, hypergraph, sequential clustering, privacy preservation.

I. INTRODUCTION

In this information age, social media plays a vital role in information dissemination where social network is the underlying interconnection structure among the social entities that work as a backbone for information propagation. The advancement in social networking and social media has attracted several industries to use such platforms as a medium to disseminate the information about their products and services. One of the applications of social media is found in e-governance service popularization [1]. Effective analysis of the data gathered by using social media platforms is essential to gain insight of information dynamics. The data gathered by social media platforms are available in text and graph forms. Although the type and quality of service are important for service popularity in such platforms, analyzing the graph structure of social connections is also crucial to understand the nature of response to a service. By this, the key players of service adoption, the category of the social entities using the service, and the non-responsive region of service can be identified. Furthermore, the strategic decision for effective advertising can be carried out to popularize the

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio.

service. However, this type of extensive data analysis may lead to the identity disclosure of the social entities. Hence, necessary steps must be taken and implemented to ensure privacy preservation of the social entities during the data analysis.

Most studies in the literature [5], [13], [34] are focused on a simple graph model where an edge can connect only two nodes. This type of relationship is known as dyadic relationship. A generalization of graph structure, known as hypergraph, can represent more complex connections than that of a simple graph [17], [23]. A hyperedge can connect more than two nodes. The relationship represented by the hypergraph is called as super-dyadic relationship [28]. In this work, we utilize a hypergraph model to represent the connections between the group of users, and hyperedge labels are used to show the users in a particular social group. A realistic view on the hypergraph publication and attack models is presented in [17]. The attack on hypergraph publication is possible as the network information is easily available with high utility, the background knowledge about the hypergraph properties along with target's behavior enables an adversary to disclose the identity of an entity in the network. The hypergraph properties like rank information, label information, etc. can be used by an adversary to disclose the identity of an entity. Herein, rank-label information is used as an attack model, and rank-label anonymization is used as a solution to obstruct this attack. The proposed rank-label attack is a stronger version of the existing rank attack [17] because it shows a higher disclosure risk than rank attack that is clearly depicted in the result. The proposed rank-label anonymization is an improvement of the existing rank anonymization proposed in [17]. Nonetheless, a rank attack was reported in the year 2013 because no other development on this model was being conveyed in the literature then. We have represented the idea of rank-label anonymization in [23]. However, this work presents an extended and more formalized framework with a detailed explanation on the rank-label attack and anonymization. The anonymization concept is formalized as a mathematical model where minimization in the cost of anonymization is represented as an objective function, and the privacy requirement is treated as a constraint.

We propose two approaches for rank-label anonymization: (i) Sequential rank-label anonymization (SA) (discussed in the Proposed methodology section) and (ii) greedy rank-label anonymization (GA) (discussed in the Result and discussion section). The anonymization cost/privacy preservation cost *PPCost* is used as a metric to measure the cost incurred due to anonymization. The solution that meets the privacy requirement with the least *PPCost* is the best solution. We propose a normalized version of the *PPCost* metric called *NCost*. The results show that SA performs better than GA because the *NCost* is found to be lesser in SA than GA. The time of execution of GA is less than that of SA because it uses some parts of SA. The experiment is extended to a high-performance computing system and reports a reduction in the average execution time for GA and SA than that of the general-purpose computing system.

The major contributions of this work are as follows:

- Disclosure of the rank-label information is used as an attack model wherein the identity of a person can be disclosed from the combination of rank and label information of the published hypergraph. Rank-label anonymization is proposed as a solution to handle a rank-label attack.
- The problem of anonymization is mapped to an optimization problem where the objective is to minimize the anonymization cost by satisfying the privacy requirement.
- Two anonymization algorithms SA and GA are proposed. The comparison between the two algorithms is depicted using *NCost* as a parameter.
- The implementation is extended to a high-performance computing device to reduce the response time.

Organization of Paper: The remaining part of this paper is organized as: Section II discusses the related work, followed by basic concepts and motivation in Section III. Section IV focuses on the system model and problem statement. Section V presents the proposed methodology. Section VI covers the results with analysis. Section VII concludes the paper with future scope.

II. RELATED WORK

Nowadays, social media plays a crucial role in information dissemination. Governments are also concerned about the effective use of social network platforms, such as Facebook and Twitter, for spreading e-governance service information [1], [8], [21], [33]. Landsbergen [15] presented the use of social media in many departments of the US government. Dwivedi et al. [10] presented a survey on the use of social media in e-governance. Magro [20] pointed out some important uses of social media, such as in policy making, disaster management, and digital divide. The prime objective of the government in e-governance is to enhance citizens' participation for achieving effective decision making. Kacem et al. [14] proposed a framework for investigating the users' and communities' profiles available in the social media to meet this requirement. The citizens' profiles are extracted from Facebook, and the interactions between the citizens are also established. The community profile is built using this information, and users who use the same e-governance service are placed under the same community. These two types of profiles are helpful in decision making for service adaptation in a much more effective manner. However, the uses of social media introduce some serious concerns regarding privacy and security of users' information. Bandy and Mattoo [4] elucidated the challenges in maintaining security and privacy of the users while social media is used in e-governance. Alguliyev et al. [2] pointed out the threats of targeting security and confidentiality of social network users. To meet these security requirements, new policies must be

set up, and modifications in the existing policies/guidelines must be dynamically conducted. Accordingly, the changes and upgrades in technical support, such as designing and implementing new algorithms/programs, are required to keep up with the modifications in the policies/guidelines. The security and privacy of users' data can be ensured by using different cryptographic techniques, such as like private-key cryptosystem, public-key cryptosystem, and authentication protocols. Some privacy preservation techniques are also used to ensure privacy [35], [36], [37], [38]. Anonymization is one of such techniques that is framed to ensure privacy-utility trade-off [22], [39]. This technique aims to meet the privacy requirement along with utility of the data by allowing effective data analysis [25], [26]. The concept of anonymization started with relational data [29] and later extended to graph data [5], [6], [7], [25]. Given that this work focuses on graph anonymization, we discuss some prominent contributions in this field. Naive anonymization is one of the earliest contributions in this domain. This method removes the label information from the nodes to hide the identity of concerned person. However, this method is an ineffective approach of dealing with identity disclosure; in such a scenario, disclosure of the identity is possible by simple structural information, such as the degree of the node [13]. K-degree anonymity is used to overcome the shortcomings of naive anonymization [19]. Several privacy threats are modeled, and solutions are proposed to counter those attacks. Majority of the works in this field are devoted toward anonymization of a simple graph model of the social networks [5], [7], [27], [30], [34], [40]. Few recent works address the issue of privacy preservation in the hypergraph model [3], [16], [17], [18], [31]. Asayesh et al. [3] used local recording and hypergraph model to meet k-anonymity in the relational data publication. Li and Shen [16], [17], [18] used a hypergraph model for representing social network. They proposed rank attack as a privacy threat and rank anonymization as a solution to this attack. The proposal of this work is based on rank-label attack, which is an improved version of rank attack. Rank-label anonymization of a hypergraph is proposed as a solution to counter rank-label attack. The salient features of some prominent contributions in the field of graph anonymization along with this work are presented in Table 1.

TABLE 1. Important features of some prominent graph anonymization approaches and proposal of this work.

Authors	Disclosure Risk	Type of Graph	Method Proposed
Campan, and Truta [5]	• Identity Disclosure • Link Disclosure	Simple Graph	SaNGreeA
Zheleva, and Getoor [34]	• Identity Disclosure • Link Disclosure • Attribute Disclosure	Simple Graph	Edge Anonymization
Tassa and Cohen [30]	• Identity Disclosure • Link Disclosure	Simple Graph	Sequential Clustering-based Anonymization
Casas-Roma et al. [7]	• Identity Disclosure	Simple Graph	Univariate Micro-aggregation for the Graph anonymization (k-degree anonymity)
Rousseau et al. [27]	• Identity Disclosure	Simple Graph	Core preserving k-degree anonymity
Li and Shen [16, 17, 18]	• Identity Disclosure	Hypergraph	Rank anonymization
Asayesh et al. [3]	• Identity Disclosure	Hypergraph	Hypergraph model using k-means clustering for relational data anonymization

III. BASIC CONCEPTS AND MOTIVATION

In this section, the basic concepts on hypergraph are explained. Moreover, the motivation behind the development of rank-label anonymization approach is presented. This work is based on a more generalized representation of simple graph called hypergraph. An edge in a simple graph can connect two vertices. Meanwhile, an edge in a hypergraph can connect any number of vertices. This type of edge is known as hyperedge. The hypergraph can represent a community structure in an effective manner than the simple graph [24]. In a simple graph, a community can be represented as a clique, where all the nodes are connected to each other by edges.

The security threat/attack considered in this work falls under the class of identity disclosure risk, where a disclosure of the identity of an actual social entity associated with his/her digital counterpart, such as Facebook and Twitter accounts, may occur from the publication of the graph. Rank attack is a kind of identity disclosure attack used to disclose the identity, and rank anonymization is a solution to this attack [17]. Our contribution in this work is to propose a new and stronger attack than rank attack called rank-label attack and to create a rank-label anonymization solution to counter this attack.

Definition 1 (Hypergraph): A hypergraph can be represented as H = (V, <E, L>), where V is the set of vertices, $< E, L >$ is the set of hyperedge, label pairs. If the hypergraph contains m number of hyperedges then $\mathrm{U}_{i=1}^{m} E_i = V$. Set L is the collection of labels that are assigned to the hyperedges. In Fig. 1, the vertices are V_1, V_2,…,V_8. The set $< E, L >$ contains four hyperedges with their labels, namely, <{1, 2}, a>, <{2, 3, 4, 6}, b>, <{6, 7, 8}, b>, and <{5, 7}, a>.

The degree of a node v in a hypergraph is the number of hyperedges incident on v. This notion of degree in a hypergraph is the same as that of a simple graph. The rank of a hyperedge is the number of nodes covered by a hyperedge. The rank of an edge in a simple graph is two.

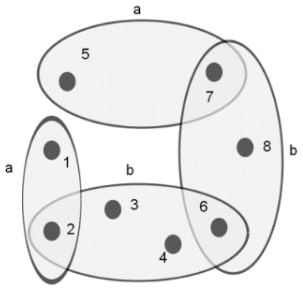

FIGURE 1. Original hypergraph.

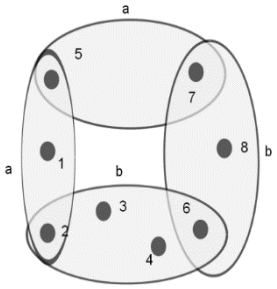

FIGURE 2. Two-rank anonymized version.

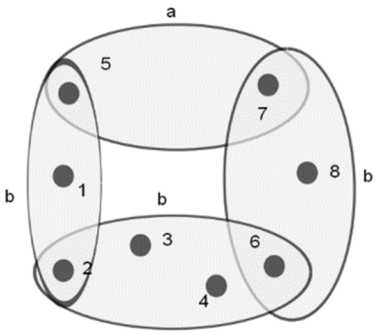

FIGURE 3. Two rank-label anonymized hypergraph.

TABLE 2. Rank-label information.

Vertices	<R|L> in Fig.1	<R|L> in Fig. 2	<R|L> in Fig. 3
V_1	<2|a>	<3|a>	<3|b>
V_2	<4,2|b, a>	<4,3|b, a>	<4,3|b, b>
V_3	<4|b>	<4|b>	<4|b>
V_4	<4|b>	<4|b>	<4|b>
V_5	<2|a>	<3,2|a, a>	<3,2|b, a>
V_6	<4,3|b, b>	<4,3|b, b>	<4,3|b, b>
V_7	<3,2|b, a>	<3,2|b, a>	<3,2|b, a>
V_8	<3|b>	<3|b>	<3|b>

Definition 2 (Vertex Rank-Label Tag): The rank-label tag of a vertex v in hypergraph H is denoted as <R|L>, where R denotes the ranks of the hyperedges arranged in a descending order, and L is the labels of hyperedges in the same order. For example, in Fig. 1, <R|L> of vertex 7 is <3, 2|b, a>.

The motivation behind the proposal of this work comes from the rank attack in a hypergraph proposed by Li and Shen [17]. The rank tag of a vertex v is the R part of the rank-label tag <R|L>. Li and Shen proposed a rank attack for identity disclosure. The <R|L> information of the vertices is presented in Table 2 for Figs. 1, 2, and 3. First, we consider the rank attack [17]. In Fig.1, vertices V_2, V_6, V_7, and V_8 is easy to uniquely identify from the rank information. The rank anonymized graph (Fig. 2) of the graph shown in Fig. 1 is obtained after applying the rank anonymization proposed in [17]. Fig. 2 is a two-rank anonymized graph of Fig. 1 because another vertex with same rank exists for each vertex in this graph. Hence, a node can be identified with 0.5 probability from the rank information of Fig. 2 (Table 2).

In this work, we propose a new attack called rank-label attack. In this attack model, <R|L> information is used for the attack. In Fig. 2, if we use <R|L> information, then vertices V_1, V_2, V_5, V_6, V_7, and V_8 are uniquely identified. Fig. 3 shows a two-rank label anonymized graph for the graph presented in Fig. 1. A k rank-label anonymized hypergraph can be obtained by applying rank-label anonymization proposed in this work. Suppose after applying this anonymization, we obtain Fig. 3, which is a 2 rank-label anonymized graph of Fig. 1. In Fig. 3, the probability of identifying a node from <R|L> information is 0.5 because one more node with same <R|L> exists for each node here.

IV. SYSTEM MODEL AND PROBLEM STATEMENT

This section discusses the overall system model along with the problem statement. The system model provides the system's detail for setting up the framework and the overall problem is stated as a problem statement.

A. SYSTEM MODEL

The system model contains five important entities: Social Media Service Provider (SMSP), Hypergraph Publisher (HP), Hypergraph Anonymizer (HA), External Party (EP), and Open Publishing Platform (OPP).

- SMSP: The SMSP provides a platform where the citizens/ users can create their profiles and interact with each other.
- HP: The HP represents the group interactions among the social entities through a hypergraph model. Moreover, HP publishes the hypergraph structure for understanding group dynamics.
- HA: The HA converts the original hypergraph representation/structure to an anonymized version that provides privacy preservation with minimum modifications in the original hypergraph.
- EP: The EP is an entity that asks the SMSP about the hypergraph structure to analyze the group dynamics. This EP may be another company seeking this information.
- OPP: The OPP is an open platform that publishes data openly to enable further analysis/research on the data.

The detailed system model is shown in Fig. 4. The sequence of operations carried out by the five entities is as follows:

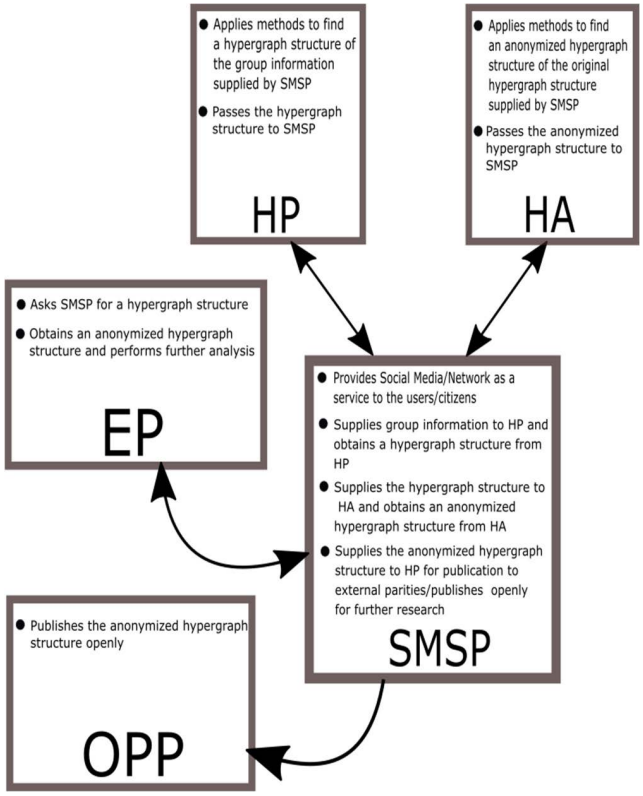

FIGURE 4. Proposed system model.

1. SMSP supplies the group information to HP. HP converts this group information to a hypergraph and supplies the same to the SMSP.
2. SMSP verifies the hypergraph representation.
3. SMSP supplies the original hypergraph representation to HA by defining the objective and constraints of anonymization.
4. HA coverts the original hypergraph to an anonymized version.
5. SMSP verifies the anonymized version. If SMSP finds that the anonymized version is not as per the requirement, then it sends back the feedback to HA. This process is repeated until a desired version is obtained.
6. SMSP provides the anonymized hypergraph for publication to EP as per its requirement.
7. If the process is not initiated by any EP, then SMSP may go for open publishing of data through OPP to enable further research on group dynamics.
3. Analysis on the anonymized hypergraph enables essential data analysis without identity disclosure.

B. PROBLEM STATEMENT

The problem statement is all about the anonymization operation performed by HA. Given the original hypergraph H, the HA converts it to an anonymized hypergraph H^* that satisfies k rank-label anonymity with minimum information loss (*PPCost*).

Definition 3 (Rank-Label Attack and k Rank-Label Anonymity): Given a hypergraph $H = (V, <E, L>)$, if the <R|L> tag of a vertex v is unique in H then v can be identified by an adversary with a prior background knowledge of <R|L> tags of all vertices of H. A hypergraph H is k rank-label anonymized if it contains at least $k-1$ other vertices having the same rank-label <R|L> tag for each vertex v.

Definition 4 (Problem Definition): The problem statement can be mathematically stated as $H \rightarrow H^*$: *min* (*PPCost*). Here, the original hypergraph H is transferred to an anonymized hypergraph H^*. Accordingly, the cost of anonymization incurred by the anonymization process is minimized. An optimal solution in polynomial time is not possible because this problem is computationally hard [17]. Hence, our objective is to find a better possible approximate solution that minimizes *PPCost*.

V. PROPOSED METHODOLOGY

The proposed methodology works in two folds: (i) Developing a hypergraph structure representation of the social connections (Social Graph/Network) and (ii) anonymization of the hypergraph structure to ensure privacy preservation. The overall approach discussed in this section is sequential rank-label anonymization (SA). Meanwhile, greedy rank-label anonymization (GA) is discussed in the Result and discussion section for a comparative study.

A. HYPERGRAPH STRUCTURE OF THE SOCIAL CONNECTIONS

Representing the social graph/network in the form of hypergraph structure helps the SMSP in analyzing the citizens' adoption to services, understanding the group dynamics of a particular service, and setting the target to popularize the service. The hyperedges in the hypergraph can be constructed from the user profiles by looking at their service commonalities. Fig. 5 shows the conversion of a simple graph/network to a hypergraph representation. In this figure, we assume two services, for example a and b. A node (representing citizen) in the simple graph is labelled with a number and service name used by that citizen. Here, the hypergraph presented in Fig. 5 is obtained where the group/cluster density is fixed to 50%. This type of grouping evolves in the graph/network due to triadic closure, focal closure, and membership closure [11]. According to triadic closure, two persons become friends if they have maximum friends in common. In per focal closure, two users using the same type of services become friends with each other. In the per the membership closure, if majority of the friends of a person are using service x, then that person also uses service x after certain period. Hence, all the three closures are set with predefined probability thresholds where the closure relationship evolves if the probability exceeds the threshold value. The simple graph of Fig. 5 is assumed to be obtained after a few evolutions of the initial network. The hypergraph below is obtained after the evolution of friendships.

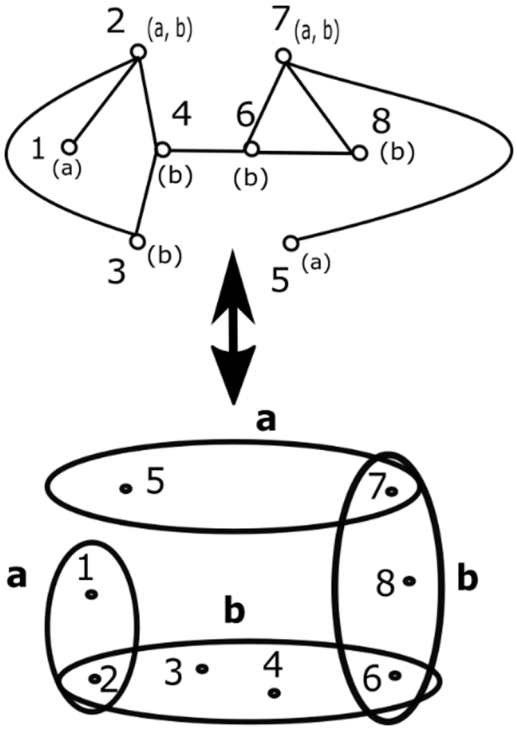

FIGURE 5. Simple graph to a hypergraph and vice versa.

Lemma 1: The conversion from simple clique graph (Gaifman Graph) to Hypergraph is a one-to-one relationship and so is the reverse.

Proof: The nodes are labelled 1,2, …, n in the clique graph G. The following steps are used to convert a simple clique graph G to a hypergraph H.

i. V← Set of vertices of G
ii. X← V
iii. i←1
iv. Repeat
    a. Pick a node with label 'i' from X
    b. Select a group of interconnected neighbors of 'i' and make them a group with 'i' and represent them as a hyperedge in H if no such hyperedge exists. Repeat it for all interconnected groups.
    c. Remove 'i' from X if all its neighbors are included in H
    d. i ← i + 1

Until (i == n + 1)

As the hypergraph H preserves the same labelling of G, the node to node mapping from simple clique graph to hypergraph is one-to-one. Likewise, the conversion from Hypergraph to simple clique graph with same labelling is one-to-one. An illustration is shown in Fig. 5 that depicts this one-to-one mapping.

B. HYPERGRAPH ANONYMIZATION

The second phase converts a hypergraph to an anonymized hypergraph. Here, the whole objective is to obtain an anonymized hypergraph that satisfies k rank-label anonymity.

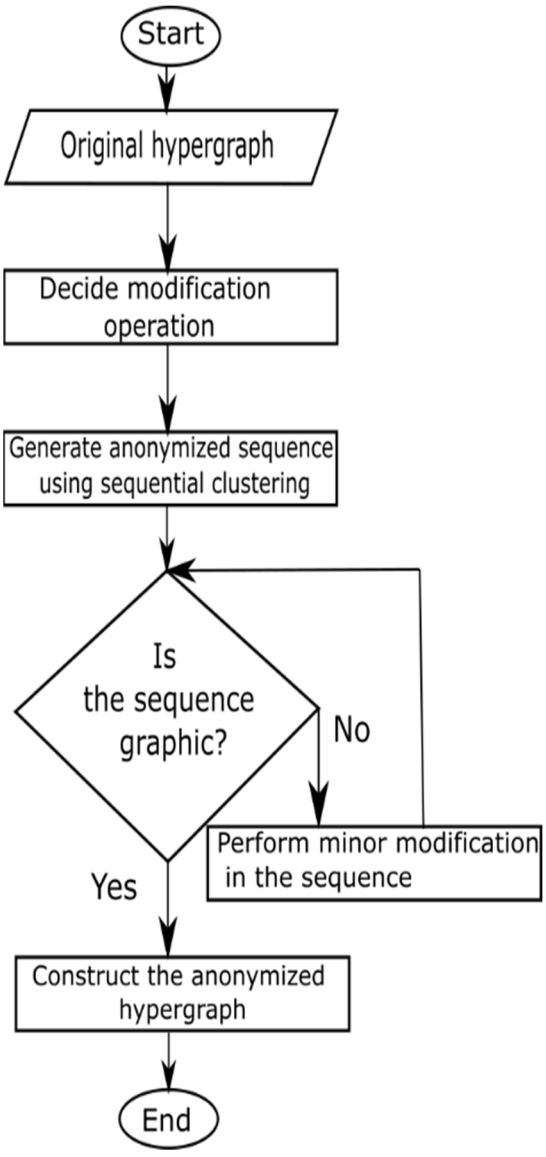

FIGURE 6. Flow of hypergraph anonymization.

This phase is again divided into three steps. First, the modification operation is considered because the complexity of the conversion procedure and the cost of anonymization depend upon the modification operation. Thereafter, sequential clustering is used to generate a k rank-label anonymized sequence from the original rank sequence. Finally, the rank-label anonymized hypergraph is constructed from the k rank-label anonymized sequence. The flowchart of the overall approach is shown in Fig. 6. The steps are elaborately explained in the subsequent subsections.

1) FIXING MODIFICATION OPERATION

Modification operations on the graph are required for transforming an original graph to an anonymized graph. The modifications on the graph are carried out after generating

an anonymized sequence. However, the operations need to be decided beforehand.

In the literature, various studies have been reported on the simple graph modification operations, such as vertex addition/deletion and edge addition/deletion [6], [7]. However, very few studies have discussed about the hypergraph operations [17]. The complexity of implementing hypergraph operations is more than that of simple graph operations. Among the different hypergraph operations, we choose EXPAND EDGE operation. In this operation, a hyperedge can be expanded to cover more vertices. This operation has two variants (i) EXPAND EDGE with an addition of a new vertex and (ii) EXPAND EDGE without addition of a new vertex. In the first variant, the new vertices are first added. Then, hyperedge is expanded. In the second variant, the hyperedge is expanded to cover more vertices from the existing vertices. Hence, the first variant is much costlier than the second one. In this work, we consider EXPAND EDGE without vertex addition as our modification operation.

2) GENERATION OF RANK-LABEL ANONYMIZED SEQUENCE

This step is crucial in the anonymized hypergraph construction. Here, we discuss how to measure the distance between the rank-label tags followed by the rank-label anonymized sequence generation using sequential clustering.

a: DISTANCE BETWEEN RANK-LABEL TAGS

The vertices of the hypergraph are assigned with their rank-label tags. The anonymized k rank-label sequence can be generated by the transformation of some rank-label tags to another tag, ensuring that all vertices have at least $k-1$ counterpart equivalents according to the rank-label tag. We evaluate the distance between the rank-label tags to meet this transformation. On this basis, the anonymization cost is formulated in the form of an optimization function.

Equation 1 presents the formulation that computes the distance between two vertices v_i and v_j with tags $< R_i, L_i >$ and $< R_j, L_j >$, respectively.

$$D\left(v_i, v_j\right) = \left(\sum\nolimits_k \left(R_{ik} - R_{jk}\right)^2\right)^{1/2} + \varphi\left(L_i, L_j\right), \qquad (1)$$

where R_{ik} represents the k^{th} component of the rank of vertex v_i, and L_i is its label. Meanwhile, R_{jk} represents the k^{th} component of the rank of vertex v_j, and L_j is its label. Equation 2 defines function $\varphi(L_i, L_j)$, which finds the distance between the labels.

$$\varphi\left(L_i, L_j\right) = \sum\nolimits_k h\left(L_{ik}, L_{jk}\right), \qquad (2)$$

where $h(L_{ik}, L_{jk})$ defines the lowest level of the concept hierarchy, where the k^{th} label of L_i and L_j meet, divided by the number of levels in the concept hierarchy. If two labels are same, then they are assigned with zero directly without any computation.

The same idea can be extended to measure the distance between a vertex and a group. Moreover, the distance between two groups is measured, considering complete linkage function [12].

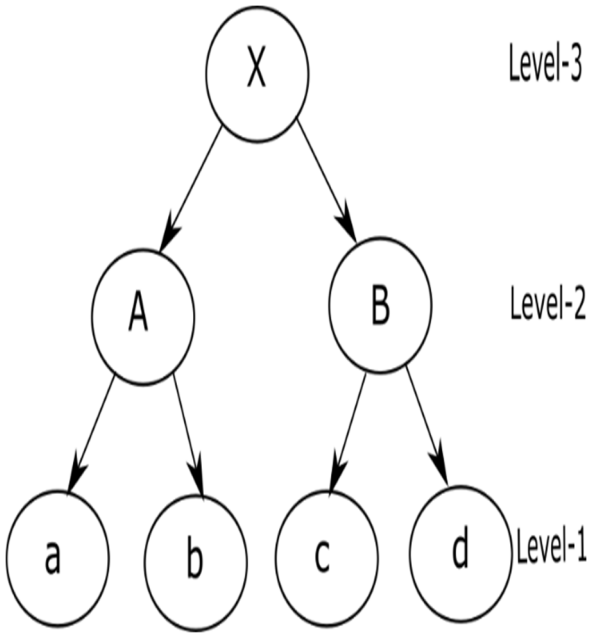

FIGURE 7. Concept hierarchy of some hypothetical groups.

Illustration: Fig. 7 shows a concept hierarchy of some hypothetical groups (i.e., a, b, c, and d). A and B are the upper-level abstraction or the categories under which the groups are present. X is the topmost category of the groups. The groups are generalized from the bottom to the top. Let us consider two vertices v_2 and v_6 of Fig. 5 with rank-label tags <4,2|ba> and <4,3|bb>, respectively. $D(v_2, v_6 = ((4\text{-}4)^2+(3\text{-}2)^2)^{1/2} + (0+2/3) = 1.66$. Meanwhile, the distance between v_6 and v_7 of Fig. 5 with tags <4,3|bb> and <3,2|ba> can be computed as $D(v_6, v_7 = ((4-3)^2 + (3\text{-}2)^2)^{1/2} + (0+2/3) = 2.08$.

b: GENERATION OF THE k RANK-LABEL ANONYMIZED SEQUENCE

In this step, the k rank-label anonymized sequence is generated from the original rank sequence by using sequential clustering [30]. The distance measurement is used for formulating the cost of anonymization. The sequential algorithm is better than the greedy approach [5], [30] because the readjustment of cluster assignment is possible in the later phase. Algorithm 1 discusses the overall process of k rank-label anonymized sequence generation.

The objective of the anonymized sequence generation is to generate such a sequence that minimizes the cost of anonymization/privacy preservation cost (*PPCost*) given in Equation 3. The constraint limits the group size in the range $[k, 2k)$.

$$PPCost = \sum_{i=1}^{m} \sum_{j=1}^{G(i)} D(< R_{ij}|L_{ij} >, < R_i|L_i > *),$$

$$\text{subject to } k \leq Size\,(G\,(i)) < 2k, \quad 1 \leq i \leq m, \tag{3}$$

where m defines the number of groups, and $Size(G(i))$ denotes the number of elements in each i^{th} group. The optimal rank-label tag $< R_i|L_i > *$ is the anonymized rank-label tag of the i^{th} group to which other rank-label tags are mapped, the *PPCost* is incurred due to this transformation. Hence, the job of the sequential clustering is to find the $< R_i|L_i > *$ that minimizes the *PPCost*.

Algorithm 1 k Rank-Label Anonymized Sequence (RS, k)

Input: Rank-label Sequence (RS) of the original hypergraph H with parameter k
Output: k rank-label anonymized sequence AS

```
1:  Begin
2:    Consider that each v ∈ V is an individual group
3:    for each i^th group
4:    {
5:            if(k ≤ Size (G(i))
6:            {
7:               Merge two groups G(i) and G(j) using
                 minimum distance D(G(i), G(j)) and
                 complete linkage function
8:            if (Size(merged group) ≥ 2k)
9:               First k vertices are assigned to one group,
                 and the rest are assigned to another group
10:           }
11: }
12:   do
13:   {
14:     for each vertex v ∈ V
15:     {
16:        Evaluate the distance D from v to all groups by
           using complete linkage
17:        Assign v to group G(i) if D(v, G(i)) is minimum
18:           if (Size (merged group) ≥ 2k)
19:              The first k vertices are
                 assigned to one group, and the rest are
                  assigned to another group
20:     }
21:   } while (swapping possible)
22: AS← Substitute the rank-label of all vertices by the rank-
    label of the vertex at maximum distance from others
23: if (Realize(AS ))
24:   {
25:     return(AS)
26:   }
27: else
28:   {
29:   Perform minor modification in AS
30:   go to step-23
31:   }
32: End
```

In Algorithm 1, steps 2–11 convert the vertex set into groups of vertices where each group has the group size defined by the constraint given in Equation 3. Distance D is computed using Equation 1. Steps 12–21 are used for cluster reassignment to satisfy the objective function shown in Equation 3. Step 22 generates anonymized sequence AS that substitutes the rank-labels of all vertices of a group with the rank-label of the vertex with maximum distance to make it compatible with EXPAND EDGE without vertex addition operation. Steps 23–31 check the realizability by using a hypergraph realizability Algorithm 2, Realize (AS). The realize (AS) algorithm returns 1 if the anonymized sequence is realizable; otherwise, it returns 0. If AS is not realizable, then minor modification in AS is performed to make it realizable. Hypergraph realization is explained in detail in the next subsection.

c: HYPERGRAPH REALIZATION

The anonymization of a hypergraph is possible only when realization is guaranteed. The anonymized degree and anonymized rank sequences must be realizable to generate an anonymized hypergraph. Realization plays a vital role in ensuring hypergraph anonymity. The following steps are used for hypergraph realization:

1. The hypergraph is converted to Gaifman graph [28] that represents each hyperedge as a clique. Gaifman graph is a simple graph representation of hypergraph.
2. Havel–Hakimi algorithm [32] is used to test the realizability.

These steps are represented in Algorithm 2.

Algorithm 2 Realize (AS)

Input: $AS \leftarrow$ k rank-label anonymized sequence
Output: 1 denotes realizable

0 denotes not realizable

```
1: Begin
2:      Convert the rank sequence of the anonymized
        hypergraph H* to degree sequence (DS) of the
        equivalent Gaifman graph.
3:      Test the graphic property of DS using the
        Havel-Hakimi method
4:   if(DS is graphic)
5:         return(1)
6:   else
7:         return(0)
8: End
```

Algorithm 2 tests whether the anonymized rank sequence AS is graphic or not. If the sequence is graphic, then it leads to the construction of a hypergraph. Step 2 of the algorithm converts the rank sequence AS of the anonymized hypergraph to the DS of the equivalent Gaifman graph. Gaifman graph is a simple graph where the edge of a hypergraph is represented as clique. The labels of all simple edges of Gaifman graph are

equal to the label of the corresponding hyperedge. Generating the *DS* from *AS* is intuitive. Once *DS* is generated, the rest of the steps (3–7) are used to verify if this sequence is graphic or not. We use the Havel–Hakimi method [32] to test the graphic property.

Definition 5 (Rank Summation (RSum) of a Vertex): The RSum of vertex v is the summation of all the ranks present in the rank tag of v. Let RS be the rank tag of a vertex v, RS = $\{R_1, R_2, \ldots, R_m\}$, where v is a member of m hyperedges. Then, RSum = $\sum_{k=1}^{m} R_k$.

Lemma 2: If the rank tag of a vertex v in hypergraph H is RS, then the degree of v in the equivalent Gaifman graph G is *RSum*-Deg(v, H).

Proof: Given hypergraph H, the rank tag of a vertex v is RS=<R_1, R_2, ……, R_m >, where m is the degree of node v in H (i.e., Deg(v, H)). The rank sum *RSum* is the summation of ranks $RSum = \sum_{i=1}^{m} R_i$. The hypergraph can be represented as an equivalent Gaifman graph, where each hyperedge is represented as a clique. Hence, the degree of v in Gaifman graph is $\sum_{i=1}^{m} (R_i - 1)$ (i.e., *RSum*−Deg(v, H)). ■

Lemma 3: In a hypergraph H, the number of vertices with odd(*RSum*−Deg(v, H)) is even.

Proof: This Lemma is useful in proving the realizability of a hypergraph from its rank sequence (collection of rank tags). Hypergraph can be easily converted to a simple graph where each hyperedge is represented as a clique in the equivalent simple graph. A hyperedge with n number of vertices is equivalent to a clique with n vertices. According to the previous result of graph theory, the number of vertices with odd degree is even, and it is true for all the simple graph equivalent of hypergraph. The degree of a node N in an equivalent simple graph can be interpreted, as shown in Lemma 2. Hence, the number of vertices with odd *RSum* is even. ■

In the hypergraph shown in Fig. 3, the rank sequence is RS = {<3>, <4, 3>, <4>, <4>, <3, 2>, <4, 3>, <3, 2>, <3>}. Sequence *RSum* = {7, 7, 5, 5, 4, 4, 3, 3} is in a descending order and the degree sequence *DS* = {5, 5, 4, 4, 3, 3, 3, 3} in an equivalent simple graph. According to the Havel–Hakimi result, the *DS* is graphic. Hence, *RS* is graphic. This notion means that we can construct a hypergraph from *RS*.

3) CONSTRUCTION OF A RANK-LABEL ANONYMIZED HYPERGRAPH

We use the incidence matrix for the construction of an anonymized hypergraph $H*$ from the original hypergraph H. Incidence matrix can more accurately represent the hypergraph because it uses an edge–vertex relationship. The vertices present in the hyperedge are marked 1 for that hyperedge, and the rest are marked as 0. The construction takes place in two phases. In the first phase, the anonymized incidence matrix $IM*$ is constructed from *IM* by EXPAND EDGE. In the second phase, $H*$ is constructed from $IM*$. These steps are represented in Algorithm 3.

Algorithm 3 Hypergraph Construction

Input: Original hypergraph H with rank-label sequence *RS* and anonymized rank-label sequence *AS*
Output: Hypergraph $H*$ satisfying *AS*

1: **Begin**
2: $IM \leftarrow$ Incidence matrix of H (labels are assigned to columns of *IM*)
3: $IM* \leftarrow IM$
4: Obtain Gaifman degree sequence *DS*(*RS*) and *DS*(*AS*) using Lemma 2.
5: Find residual sequence $Res \leftarrow DS(AS)\text{-}DS(RS)$
6: **while** (*Res* not null)
7: {
8: In *Res,* if two vertices V_i and V_j have +ve residue ranks, then add 1 in the equivalent Gaifman graph. Reflect the changes in $IM*$.
9: Reduce the corresponding rank by 1. Adjust the labels accordingly.
10: }
11: Construct $H*$ from $IM*$
12: **return** ($H*$)
13: **End**

In Algorithm 3, step 2 finds the incidence matrix *IM* of the original hypergraph H. Step 3 assigns *IM* as the initial instance of the anonymized incidence matrix $IM*$. Step 4 finds the Gaifman graph degree sequence of *RS* and *AS* (i.e., *DS*(*RS*) and *DS*(*AS*)). Step 5 computes the residual sequence *Res*, that is, the difference between *DS*(*RS*) and *DS*(*AS*). Steps 6–10 compute $IM*$ by hyperedge expansion that establishes new relationships. Step 11 constructs the hypergraph $H*$ from the $IM*$. The +ve residue rank of a vertex is the remaining positive rank that can be included in the hyperedges. The Step 12 returns $H*$.

VI. RESULT AND DISCUSSION

This section discusses the results obtained from the implementation of the proposed approaches. At first, we consider the Adult dataset [9] for this implementation and analysis. A greedy-based approach of hypergraph anonymization is proposed to set a comparison framework for the SA approach. The experiment is extended further to a real labelled hypergraph dataset MAG-10 [41].

System Specification: Experiments are carried out in a 2.40 GHz Intel(R) Core (TM) i7-4770 processor with a memory support of 4 GB and Microsoft windows 8.1 professional operating system (System 1).

Dataset Details: We prepare a synthetic hypergraph structure by considering the Adult dataset [9] that is used for the experiment and analysis because the original hypergraph structure is not available. This dataset contains 48,842 records with 15 attributes. Among these attributes, 14 are independent

TABLE 3. Independent attributes of the Adult dataset and their type.

Continuous independent attributes	Categorical independent attributes
Age, Fnlwgt, Education-num, Capital-gain, Capital-loss, and Hours-per-week	Workclass, Education, Marital-status, Occupation, Relationship, Race, Sex, Native-country, and Dominican-Republic

attributes, and 1 is a dependent attribute (represents classification labels). The dataset details of the independent attributes are illustrated in Table 3. This table shows the name of 14 independent attributes and separates them into two groups: Continuous and Categorical. All the continuous attributes are chosen for the experiments, and the categorical attributes are not used. Furthermore, the dependent attribute is not used.

A. IMPLEMENTATION STRATEGY

The following steps are used to generate an original synthetic hypergraph structure from the Adult dataset and for the subsequent application of the proposed approaches:

1. We consider four groups (a, b, c, and d), as shown in Fig. 7.
2. The similarity between the vertices is obtained by using the continuous attributes presented in Table 3. Euclidean distance is used to measure the similarity between two vertices. A similarity matrix that contains the similarity between all pairs of vertices is created.
3. The four labels (a, b, c, and d) are assigned to all 48,842 vertices (each record of the dataset is considered a vertex). The distribution of labels is set to be uniform (25% of the total number of vertices).
4. A vertex v is picked at random and assigned with label a. Then, $n-1$ (where $n = 25\%$ of the total number of vertices) number of similar nodes of v are obtained and assigned with the same label. Then, the group is assigned with a hyperedge. The same process is continued for the rest of the three labels. This step continues until all the vertices are labeled. This labeling procedure assigns multiple labels to 10% of the vertices that are intentionally incorporated. Subsequently, we generate a hypergraph structure.
5. The proposed SA and GA are applied on the structure obtained in step 4.

B. GREEDY RANK-LABEL ANONYMIZATION (GA)

The greedy based approach is proposed to set a comparison framework with SA approach. This approach adopts the same method as discussed in SA, but with some changes. In the greedy approach, steps 12–21 of Algorithm 1 are removed because this part uses sequential clustering for the readjustment of the cluster assignment. The rest of the steps are the same as those of SA. In GA, cluster adjustment is not possible in the later phase. Hence, GA takes less time than SA, but the quality of solution is better in the case of SA as it allows readjustment in the later phase.

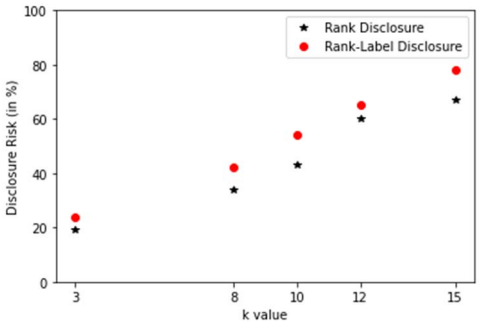

FIGURE 8. Rank vs rank-label disclosure (in %).

C. RANK VS RANK-LABEL DISCLOSURE

Rank disclosure occurs when a vertex can be identified with a disclosure probability p if it does not have at least a $k-1$ counterpart with the same rank [17]. The same idea is applicable for the rank-label disclosure of a vertex with a disclosure probability p if it does not have at least a $k-1$ counterpart with the same rank-label tag. We consider five different k values (3, 8, 10, 12, and 15) and observed the disclosure risk % (the percentage of records suffers from a disclosure risk) according to rank and rank-label disclosure. Both the disclosure risks are observed on the synthetic hypergraph structure obtained from the Adult dataset. Fig. 8 shows that the risk of disclosure increases when k increases. Furthermore, the disclosure risk due to rank-label disclosure/attack is higher in all cases compared with the rank disclosure/risk, as shown in Fig. 8.

D. COMPARISON BETWEEN SA AND GA

The two approaches are compared by considering the cost of anonymization as an evaluation parameter. The optimal cost of anonymization obtained by Equation 3 is a non-normalized one. Scaling down the cost of anonymization ($PPCost$) to [0, 1) is possible by applying Algorithm 4.

Algorithm 4 Normalized_Cost ($PPCost$)

Input: $PPCost \leftarrow$ Anonymization cost/privacy preserving cost
Output: $NCost \leftarrow$ Normalized cost

1: **Begin**
2: **if**($PPCost == 0$)
3: $NCost \leftarrow 0$
4: **else**
5: $NCost \leftarrow (1 - \frac{1}{PPCost+0.1})$
6: **return** ($NCost$)
7: **End**

Algorithm 4 returns 0 when $PPCost$ is 0. In such a scenario, the rank-label tags are equal and do not need any

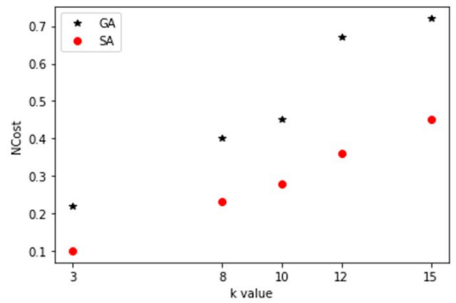

FIGURE 9. Normalized cost (*NCost*) of anonymization in GA and SA.

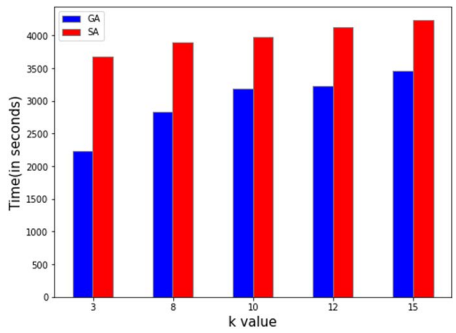

FIGURE 10. Time of execution (in seconds) of GA and SA in system 1.

modifications; otherwise, the algorithm returns a value in between (0, 1) (i.e., a value greater than 0 and less than 1). The normalized cost never attains 1 because $\frac{1}{PPCost+0.1}$ never turns to 0. This situation occurs because we added 0.1 as a constant value. If this constant is not added, then the algorithm returns a value of 1 when the *PPCost* value is 1, which violates the requirement when the *PPCost* value exceeds 1, the *NCost* goes down. In most cases, *PPCost* is greater than 1. Hence, we add constant 0.1. Any other constant value may also be considered on behalf of 0.1.

We compute the *NCost* incurred by SA and GA for the five k values (3, 8, 10, 12, and 15). This comparison is clearly depicted in Fig. 9. *NCost* increases with the increase in the k value. *NCost* must be less in SA than in GA in all the cases. Fig. 10 shows the time (in seconds) required by GA and SA in System 1. GA takes lesser time than SA because it executes some steps of SA only.

E. EXTENDING THE EXPERIMENT TO A HIGH-PERFORMANCE COMPUTING (HPC) SYSTEM

The implementation takes a long time in a typical system, such as System 1, as shown in Fig. 10. To reduce the time and enhance the efficiency, we extend the experiment to an HPC system (i.e., PARAM Shavak HPC system with CPU@2.60 GHZ, 28 cores, and 96 GB memory) (System 2). A comparison between the execution time (in seconds) in Systems 1 and 2 is shown in Figs. 11 and 12 for GA and SA, respectively. The percentage reduction in time due to the use

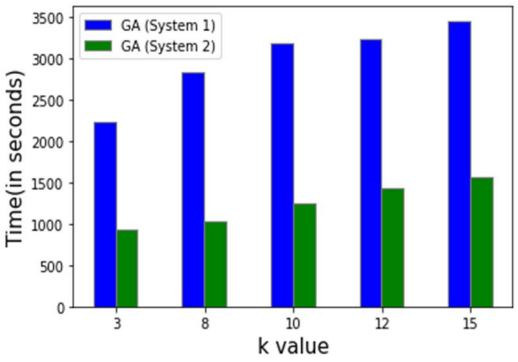

FIGURE 11. Time of execution (in seconds) of GA in Systems 1 and 2.

TABLE 4. Time of execution (in seconds) of GA with % reduction.

k value	GA (System 1) (in seconds)	GA (System 2) (in Seconds)	% reduction in time (GA)
k=3	2234	923	58%
k=8	2832	1029	63%
k=10	3192	1256	60%
k=12	3234	1435	55%
k=15	3456	1567	54%

TABLE 5. Time of execution (in seconds) of SA with % reduction.

k value	SA (System 1) (in seconds)	SA (System 2) (in seconds)	% reduction in time (SA)
k=3	3675	1231	66%
k=8	3897	1345	65%
k=10	3984	1545	61%
k=12	4123	1678	59%
k=15	4234	1706	59%

of an HPC system (System 2) can be computed for GA and SA by using Equations 4 and 5, respectively.

$$\%\ reduction\ in\ time\ (GA) = \frac{TGA\ (System1) - TGA\ (System2)}{TGA\ (System1)} \times 100, \quad (4)$$

$$\%\ reduction\ in\ time\ (SA) = \frac{TSA\ (System1) - TSA\ (System2)}{TSA\ (System1)} \times 100, \quad (5)$$

where TGA and TSA denote the execution time (in seconds) of GA and SA, respectively. Tables 4 and 5 show the percentage reductions in time for GA and SA, respectively. The percentage value is rounded up to a whole number. We observe an average of 58% reduction in time in GA and an average of 62% reduction in time in SA due to the use of System 2.

F. EXTENDING THE EXPERIMENT TO A REAL-WORLD DATASET

The MAG-10 dataset is considered from [41] where the nodes are authors and the hyperedges represent publications. The labels of the hyperedges are the conferences in computer

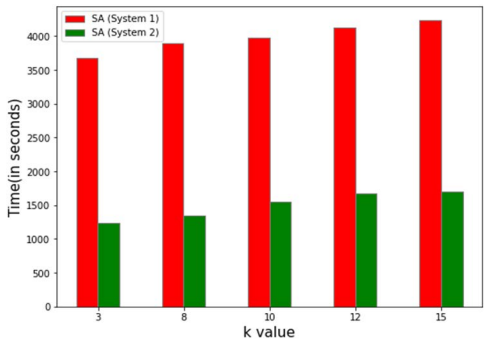

FIGURE 12. Time of execution (in seconds) of SA in system 1 and system 2.

TABLE 6. Essential information about the MAG-10 dataset.

Number of nodes (n)	Number of hyperedges (m)	Number of labels with description	Maximum hyperedge rank	No. of publications in each conference
80198	51889	10 1-KDD 2-WWW 3-ICML 4-NeurIPS 5-STOC 6-FOCS 7-CVPR 8-ICCV 9-VLDB 10-SIGMOD	25	1-4675 2-5468 3-4803 4-7060 5-2284 6-2487 7-11006 8-5366 9-4448 10-4291

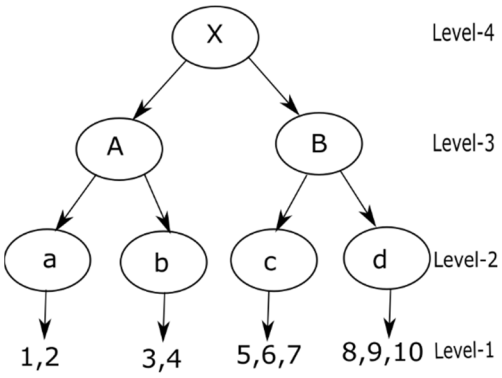

FIGURE 13. Concept hierarchy of 1-10 labels of MAG-10.

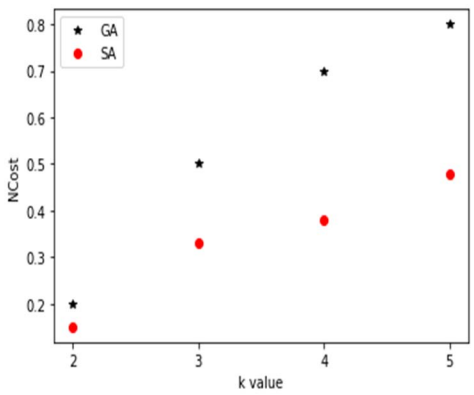

FIGURE 14. NCost comparison between GA and SA in MAG-10.

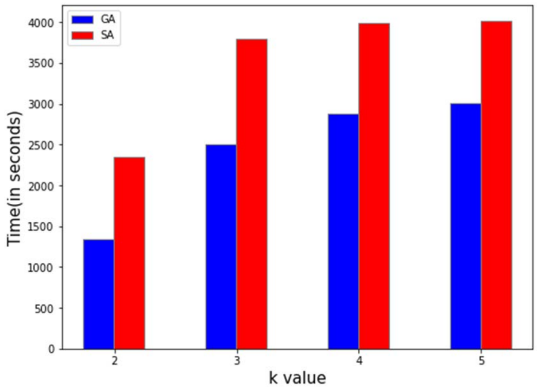

FIGURE 15. Time of execution (in seconds) of GA and SA in System 1 for MAG-10 dataset.

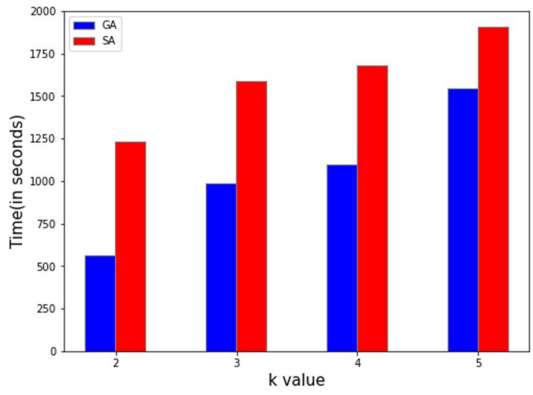

FIGURE 16. Time of execution (in seconds) of GA and SA in System 2 for MAG-10 dataset.

science. The detail information regarding the dataset is given in Table 6.

A concept hierarchy shown is Fig.12 is constructed by considering the 1 to 10 labels of hyperedges. This concept hierarchy is used in the process of rank-label anonymization.

Fig. 14 shows the NCost comparison between SA and GA for dataset MAG-10. The observation in this dataset is also the same as the Adult dataset. The SA reports less NCost than GA for k values 2, 3, 4, and 5. As per the execution time shown is Fig. 15 and 16, GA takes less time than the SA in both System 1 and System 2.

The reduction in time for SA and GA is computed using Equations 4 and 5 respectively for MAG-10 dataset. An average of 56% reduction in time of execution is reported in GA whereas it is 53% in SA due to the use of System 2.

G. ANALYIS ON TIME COMPLEXITY

The time complexity of the proposed SA (Algorithm 1) is $O(n^4)$. In Algorithm 1, steps 3-11 take $O(n^3)$ time as the outer loop is executed for O(n) times, in step 7, complete linkage function is computed from a distance matrix that takes $O(n^2)$ time. Hence, steps 3-11 take $O(n)xO(n^2) = O(n^3)$ time. Steps 12-21 take $O(n^4)$ time. Here, the do…while loop is executed for O(n) times. The for-loop in steps 14-20 is executed for O(n) times. Step 16 takes $O(n^2)$ time for the distance matrix computation. Hence, steps 12-21 take $O(n)xO(n)xO(n^2) = O(n^4)$ time. Steps 23-31 depend on hypergraph realization i.e., Algorithm 2. The process of hypergraph realization takes $O(n^2)$ time as it applies Havel-Hakimi method. Therefore, the complexity of the Algorithm 1 is $O(n^3)+ O(n^4)+ O(n^2) = O(n^4)$. GA adopts the same strategy as SA but without cluster readjustment in the later phase. GA uses all the steps of SA (Algorithm 1) except steps 12-21. Hence, the time complexity of GA is $O(n^3)$. Due to less time complexity, GA is more scalable as compared to SA.

H. ANALYSIS ON DE-ANONYMIZATION PROBABILITY

Here, we discuss the possibility of de-anonymization of a published k rank-label anonymized hypergraph. As EXPAND EDGE without vertex addition operation with relabeling of hyperedges is considered for the construction of anonymized hypergraph from the original hypergraph, the original hypergraph can be reconstructed from the anonymized one by considering all possible hyperedges through SHRINK EDGE without vertex deletion operation i.e., opposite of EXPAND EDGE without vertex addition operation with all possible hyperedge labels. Let us consider the anonymized hypergraph has m hyperedges say $< E_1, L_1 >, < E_2, L_2 >, \ldots\ldots\ldots, <\underline{E}_m, L_m >$. Assume that it contains l number of distinct labels such that $l \leq m$. An original hyperedge can be reconstructed from an anonymized hyperedge $< E_i, L_j >$ by considering one among all subsets of E_i excluding the null set i.e., one from $(2^{Rank(Ei)}$ -1) subsets, combined with one among all l labels. Hence, the probability of reconstructing the original hyperedge is $\frac{1}{(2^{Rank(E_i)}-1)*l}$. By combining all hyperedge reconstruction probabilities, we find the probability of reconstructing the original hypergraph to be $\frac{1}{l^m \prod_{i=1}^{m}(2^{Rank(E_i)}-1)}$. As this reconstruction probability is less, the reconstruction of original hypergraph from the published k rank-label anonymized hypergraph is very difficult.

Lemma 4: The probability of de-anonymization in proposed rank-label anonymization is less than that of rank anonymization.

Proof: Let us assume that both the rank-label anonymization and rank anonymization use EXPAND EDGE without vertex addition operation to convert a given hypergraph to its anonymized version. As discussed above, the probability of de-anonymization in k rank-label anonymization is $\frac{1}{l^m \prod_{i=1}^{m}(2^{Rank(E_i)}-1)}$. In case of rank anonymization, the probability of de-anonymization is $\frac{1}{\prod_{i=1}^{m}(2^{Rank(E_i)}-1)}$ as the label information is not used in rank anonymization. As $\frac{1}{l^m \prod_{i=1}^{m}(2^{Rank(E_i)}-1)} < \frac{1}{\prod_{i=1}^{m}(2^{Rank(E_i)}-1)}$, the de-anonymization is difficult in rank-label anonymization than that of rank anonymization. ■

VII. CONCLUSION AND FUTURE SCOPE

This work proposes a hypergraph model of social structure according to social connections and label information. Here, we propose a stronger attack model than the existing rank attack called rank-label attack. We have proposed SA and GA to counter the rank-label attack. SA is found to be better than GA according to normalized anonymization cost metric *NCost* for both Adult and MAG-10 datasets. In both datasets, GA takes less time than SA, but the quality of solution is not better than that of SA. Furthermore, GA and SA are implemented in two systems, namely, Systems 1 and 2. The averages of percentage reduction in time are 58% and 62% in GA and SA in System 2 for Adult dataset, respectively. In MAG-10 dataset, average reduction percentage is reported to be 56% and 53% for GA and SA respectively. GA and SA report $O(n^3)$ and $O(n^4)$ time complexity respectively. This work can be further extended to address some other types of attacks. In the near future, a stronger attack model performing better than the rank-label attack can be modeled, and the corresponding anonymization solution can be developed.

REFERENCES

[1] A. Addo and P. K. Senyo, "Advancing e-Governance for development: Digital identification and its link to socioeconomic inclusion," *Government Inf. Quart.*, vol. 38, no. 2, Apr. 2021, Art. no. 101568.

[2] F. Yusifov, R. Alguliyev, and R. Aliguliyev, "Role of social networks in E-government: Risks and security threats," *Online J. Commun. Media Technol.*, vol. 8, no. 4, pp. 363–376, Nov. 2018.

[3] A. Asayesh, M. A. Hadavi, and R. Jalili, "(t, k)-hypergraph anonymization: An approach for secure data publishing," *Secur. Commun. Netw.*, vol. 8, no. 7, pp. 1306–1317, May 2015.

[4] M. Tariq Banday and M. M. Mattoo, "Social media in e-governance: A study with special reference to India," *Social Netw.*, vol. 2, no. 2, pp. 47–56, 2013.

[5] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Proc. 2nd ACM SIGKDD Int. Workshop Privacy, Secur., Trust KDD (PinKDD)*, 2008, pp. 33–54.

[6] J. Casas-Roma, "DUEF-GA: Data utility and privacy evaluation framework for graph anonymization," *Int. J. Inf. Secur.*, vol. 19, no. 4, pp. 465–478, Aug. 2020.

[7] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "k-degree anonymity and edge selection: Improving data utility in large networks," *Knowl. Inf. Syst.*, vol. 50, no. 2, pp. 447–474, Feb. 2017.

[8] P. Dey, and S. Roy, "Governance in smart city: An approach based on social network," in *Smart Cities: A Data Analytics Perspective*. Cham, Switzerland: Springer, 2021, pp. 63–87.

[9] D. Dua, and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2019.

[10] Y. K. Dwivedi, N. P. Rana, M. Tajvidi, B. Lal, G. P. Sahu, and A. Gupta, "Exploring the role of social media in e-government: An analysis of emerging literature," in *Proc. 10th Int. Conf. Theory Pract. Electron. Governance*, Mar. 2017, pp. 97–106.

[11] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[12] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[13] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," Comput. Sci. Dept. Fac. Publication Ser., Univ. Massachusetts Amherst, 2007, p. 180.

[14] A. Kacem, R. Belkaroui, D. Jemal, H. Ghorbel, R. Faiz, and I. H. Abid, "Towards improving e-Government services using social media-based citizen's profile investigation," in *Proc. 9th Int. Conf. Theory Pract. Electron. Governance*, Mar. 2016, pp. 187–190.
[15] D. Landsbergen, "Government as part of the revolution: Using social media to achieve public goals," *Electron. J. e-Government*, vol. 8, no. 2, pp. 135–147, 2010.
[16] Y. Li and H. Shen, "Anonymizing hypergraphs with community preservation," in *Proc. 12th Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, Oct. 2011, pp. 185–190.
[17] Y. Li and H. Shen, "On identity disclosure control for hypergraph-based data publishing," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1384–1396, Aug. 2013.
[18] Y. Li, Y. Li, B. Zhang, and H. Shen, "Preserving private cloud service data based on hypergraph anonymization," in *Proc. Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, Dec. 2013, pp. 192–197.
[19] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Jun. 2008, pp. 93–106.
[20] M. J. Magro, "A review of social media use in e-Government," *Administ. Sci.*, vol. 2, no. 2, pp. 148–161, 2012.
[21] P. Manocha, S. Som, and L. Chanana, "Technological trends, impact, and analysis of quality service parameters on e-Governance applications," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim., Trends Future Directions (ICRITO)*, Jun. 2020, pp. 1179–1184.
[22] M. Mohammady, M. Oqaily, L. Wang, Y. Hong, H. Louafi, M. Pourzandi, and M. Debbabi, "A multi-view approach to preserve privacy and utility in network trace anonymization," *ACM Trans. Privacy Secur.*, vol. 24, no. 3, pp. 1–36, Aug. 2021.
[23] D. Mohapatra and M. R. Patra, "A graph based approach for privacy preservation of citizen data in e-governance applications," in *Proc. 20th Annu. Int. Conf. Digit. Government Res.*, Jun. 2019, pp. 433–438.
[24] D. Mohapatra, and M. R. Patra, "Rank consensus between importance measures in hypergraph model of social network," in *Evolution in Computational Intelligence*. Singapore: Springer, 2021, pp. 305–314.
[25] G. A. Requena, R. Mayer, and A. Ekelhart, "Anonymisation of heterogeneous graphs with multiple edge types," in *Proc. Int. Conf. Database Expert Syst. Appl.* Cham, Switzerland: Springer, 2022, pp. 130–135.
[26] A. Majeed, S. Khan, and S. O. Hwang, "Toward privacy preservation using clustering based anonymization: Recent advances and future research outlook," *IEEE Access*, vol. 10, pp. 53066–53097, 2022.
[27] F. Rousseau, J. Casas-Roma, and M. Vazirgiannis, "Community-preserving anonymization of graphs," *Knowl. Inf. Syst.*, vol. 54, no. 2, pp. 315–343, Feb. 2017.
[28] S. Roy and B. Ravindran, "Measuring network centrality using hypergraphs," in *Proc. 2nd ACM IKDD Conf. Data Sci.*, 2015, pp. 59–68.
[29] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, 2002.
[30] T. Tassa and D. J. Cohen, "Anonymization of centralized and distributed social networks by sequential clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 311–324, Feb. 2013.
[31] X. Wang, Y. Li, Y. Jin, and W. Wang, "Spectrum-centric differential privacy for hypergraph spectral clustering," in *Proc. Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, Aug. 2018, pp. 3–14.
[32] D. B. West, *Introduction to Graph Theory*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[33] E. S. Zeemering, "Functional fragmentation in city Hall and Twitter communication during the COVID-19 pandemic: Evidence from Atlanta, San Francisco, and Washington, DC," *Government Inf. Quart.*, vol. 38, no. 1, Jan. 2021, Art. no. 101539.
[34] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proc. Int. Workshop Privacy, Secur., Trust KDD*, Aug. 2007, pp. 153–171.
[35] H. Zhang, L. Lin, L. Xu, and X. Wang, "Graph partition based privacy-preserving scheme in social networks," *J. Netw. Comput. Appl.*, vol. 195, Dec. 2021, Art. no. 103214.
[36] M. Kiranmayi and N. Maheswari, "A review on privacy preservation of social networks using graphs," *J. Appl. Secur. Res.*, vol. 16, no. 2, pp. 190–223, Apr. 2021.
[37] S. Shakeel, A. Anjum, A. Asheralieva, and M. Alam, "k-NDDP: An efficient anonymization model for social network data release," *Electronics*, vol. 10, no. 19, p. 2440, Oct. 2021.
[38] K. Macwan, and S. Patel, "Privacy preservation approaches for social network data publishing," in *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities*. Cham, Switzerland: Springer, 2021, pp. 213–233.
[39] M. Bewong, L. Jixue, L. Lin, and L. Jiuyong, "Utility aware clustering for publishing transactional data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2017, pp. 481–494.
[40] A. Majeed and S. Lee, "Anonymization techniques for privacy preserving data publishing: A comprehensive survey," *IEEE Access*, vol. 9, pp. 8512–8545, 2021, doi: 10.1109/ACCESS.2020.3045700.
[41] *Dataset: Cat-edge-MAG-10*. Accessed: Aug. 9, 2022. [Online]. Available: https://www.cs.cornell.edu/~arb/data/cat-edge-MAG-10/

DEBASIS MOHAPATRA received the Ph.D. degree in computer science from Berhampur University, India. He is currently working as an Assistant Professor with the Department of CSE, PMEC Berhampur (Government). He has published nearly 30 research papers in reputed conferences and journals. His research interests include graph anonymization, privacy preserving data publication, complex networks, and e-governance. He received the Prestigious UGC JRF Award, in 2012.

SOURAV KUMAR BHOI received the Ph.D. degree in computer science and engineering from the National Institute of Technology, Rourkela, India. He is currently working as an Assistant Professor with the Department of CSE, PMEC Berhampur (Government). He has published nearly 100 research papers in reputed conferences and journals. His research interests include the IoT, deep learning, machine learning, edge and fog computing, and information security. He received the Prestigious IET Premium Award, in 2016.

KALYAN KUMAR JENA received the Ph.D. degree in computer science engineering from Utkal University, Bhubaneswar, India. He is currently working as an Assistant Professor with the Department of CSE, PMEC Berhampur (Government). He has published nearly 60 research papers in reputed conferences and journals. His research interests include image processing, machine learning, deep learning, and the IoT. He received the Bhubananda Das Award (Gold Medal Recipient 2013).

KSHIRA SAGAR SAHOO (Member, IEEE) received the M.Tech. degree in information and communication technology from the IIT, Kharagpur, India, in 2014, and the Ph.D. degree in computer science and engineering from the National Institute of Technology Rourkela, India, in 2019. He is currently a Kempe Fellow at the Autonomous Distributed Systems Laboratory, Umeå University, Sweden. He has more than five years teaching experience, two-year industry experience, and four-years of research experience. He has published more than 80 research papers in various top international journals and conferences, including the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE SYSTEMS JOURNAL, IEEE INTERNET OF THINGS JOURNAL, *ACM TOMM*, *FGCS* (Elsevier), and *JSS* (Elsevier). His research interests include future generation network infrastructure, such as SDN, edge computing, the IoT, and the industrial IoT. He is a member of the IEEE Computer Society and an Associate Member of the Institute of Engineers (IE), India.

ANAND NAYYAR received the Ph.D. degree in computer science in the area of wireless sensor networks, swarm intelligence and network simulation from Desh Bhagat University, in 2017. He is currently working with the School of Computer Science, Duy Tan University, Da Nang, Vietnam, as an Assistant Professor, a Scientist, and the Vice-Chairperson (Research), where he is also the Director of the IoT and Intelligent Systems Laboratory. He is a Certified Professional with more than 125 professional certificates from CISCO, Microsoft, Amazon, EC-Council, Oracle, Google, Beingcert, EXIN, GAQM, Cyberoam, and many more. He published more than 150 research papers in various high-quality ISI-SCI/SCIE/SSCI impact factor journals cum Scopus/ESCI indexed journals, more than 60 papers in international conferences indexed with Springer, IEEE Xplore, and ACM Digital Library, more than 40 book chapters in various Scopus, Web of Science indexed books with Springer, CRC Press, Wiley, IET, and Elsevier with citations more than 7500, H-index: 45, and i-index: 155. He has authored/coauthored cum edited more than 40 books of computer science. He associated with more than 500 international conferences as a program committee/chair/advisory board/review board member. He has 18 Australian patents, seven Indian design cum utility patents, three Indian copyrights, two Canadian copyrights, three German patents, two Japanese patents, and one U.S. patent to his credit in the area of wireless communications, artificial intelligence, healthcare informatics, digital twins, cloud computing, the IoT, and image processing. He has reviewed more than 2000 articles for various Web of Science indexed journals. His research interests include wireless sensor networks, the IoT, swarm intelligence, cloud computing, artificial intelligence, drones, blockchain, cyber security, network simulation, and wireless communications. He is a member of more than 50 associations as a Senior Member and a Life Member, including ACM. He was awarded 36 awards for Teaching and Research—Young Scientist, Best Scientist, Young Researcher Award, Outstanding Researcher Award, Excellence in Teaching, and many more. He is acting as an Associate Editor of *Wireless Networks* (Springer), *Computer Communications* (Elsevier), *International Journal of Sensor Networks* (IJSNET) (Inderscience), *Frontiers in Computer Science*, *Computer Science* (PeerJ), *Human Centric Computing and Information Sciences* (HCIS), *IET Quantum Communication*, *IET Wireless Sensor Systems*, *IET Networks*, *IJDST*, *IJISP*, *IJCINI*, and *IJGC*. He is acting as the Editor-in-Chief of IGI-Global, USA, journal titled *International Journal of Smart Vehicles and Smart Transportation* (IJSVST).

MOHD ASIF SHAH is currently working as an Associate Professor at Bakhtar University, Kabul, Afghanistan. He has been earlier working as an Assistant Professor of economics at the Forbes Business School, India, and LPU, India. He worked as a Lecturer at the JCE, Jammu and Kashmir, India, and also helped his department with teaching assistance during his Ph.D. He has published more than 20 research articles (SCI/WOS/UGC indexed) with more than 30 citations.