Journal of Network and Systems Management (2020) 28:398–422
https://doi.org/10.1007/s10922-019-09511-1

Hybrid Approach to Speed-Up the Privacy Preserving Kernel K-means Clustering and its Application in Social Distributed Environment

P. L. Lekshmy[1] · M. Abdul Rahiman[2]

Received: 19 July 2017 / Revised: 17 December 2019 / Accepted: 24 December 2019 /
Published online: 16 January 2020

Abstract

In this most revolutionized world, the social network plays a vital role in each and everyone's life. Social networking is a pervasive communication platform where the users can search whole over the world via the Internet. Users have similar interest to connect and interact with one another and to share their private and personal interest. In this paper, we examine privacy concern for the social networking users by distributed clustering method. In the proposed scheme, to speed-up, the Kernel k-means algorithm, a prototype based hybrid kernel k-means algorithm is involved in distributing the users into the cluster. Since we are using a large data set, we use a hybrid approach to speed-up the kernel k-means clustering (*HSKK*). The clustering process used here is to partition a similar set of objects in a dataset. Additionally, in the clustering process, a cryptographic protocol such as homomorphic encryption is involved in every dataset to achieve the goal to protect the private data. To prove the efficiency of the proposed approach, the experiment is done on Movie lens dataset. The experimental study of *HSKK* shows that the proposed method can significantly reduce the computation time and the private data of users is hidden from the service provider.

Keywords Service provider · Social network · Kernel k-means · Distributed clustering · Encryption · Helper user · Cryptographic protocol

✉ P. L. Lekshmy
lekshmypl0682@gmail.com

[1] Computer Science and Engineering, L B S Institute of Technology for Women, University of Kerala, Trivandrum, India

[2] Kerala State Centre for Advanced Printing and Training, Trivandrum, India

1 Introduction

Nowadays, social media is expanded by increasing the number of users across the world, and industries also started advertising through social media to improve and broaden their businesses. A virtual community, in which people shared their interest such as a specific activity, can interact and socialize among themselves with the help of social media. The opportunity of mutual interest is provided by the social network and the members in a social network are united. Social Network applications are widely used by a number of people in the world and it is used to share information efficiently. A social community is a platform to build social relations among people who share interests, activities, backgrounds or real-life connections. People were using internet applications to share their personal and private data are in their groups. They expect the shared details to be secured. But the major problem faced by most of the users is information leakage to the service provider. A way to protect the privacy sensitive data of the user from the service provider is having a trusted third party that keeps the data and runs the algorithm instead of service provider [1]. By using advanced data storage capabilities of the computer, varieties of data mining algorithms were developed [2].

The way toward separating patterns from large information sets by joining techniques from insights and artificial consciousness with database administration is known as data mining. For modern business, to change information into business knowledge giving an instructive favorable position, data mining is utilized progressively [3]. Data mining is the development of models about aggregated data. More privacy issues are revealed by customers when they provide valuable knowledge for mining and data mining is one of the analyzing tools used for analyzing data. Then the clustering process is done in many research areas, including genetics, cybernetics, and marketing to cluster the known set of common entities without revealing any of the values [4]. Clustering is the process of grouping the set of data's into similar groups. For distributing the data among multiple participants, distributed data mining is used. Privacy-preserving data mining won't reveal any of the individual data information to any of the third party users among the system.

K-means clustering algorithm is used for finding the group of similar people based on their similarities and it is widely used to cluster the users in the social network. In the clustering method, k-means clustering is used widely because of its simplicity and ability to converge extremely quickly in practice [1]. The objects to be clustered dwells on various destinations are an assumption by distributed clustering. Standard clustering algorithms examine the data and then the data are clustered independently on the different local sites instead of transmitting all objects to a server [5]. In the k-means algorithm, the homomorphic property is used. Computing with encrypted data is enabled by Homomorphic Encryption technique. It means, without converting into the plain text one is able to perform the operations on this data. Mostly the data's which is stored in the cloud will be saved in the encrypted state [6]. Because of quadratic time complexity with respect to the size of the dataset, Kernel method is not suitable for large datasets.

To speed-up, the kernel k-means clustering method for large datasets and for reducing the time, simple prototype-based hybrid approach is provided. Non-linear extension of the k-means clustering method is Kernel k-means clustering method. To recognize clusters which are non-isotropic and straightly separable in the input space k-means has been turned out to be successful [7]. It is an iterative procedure where at first the data points are mapped from the input space to a higher dimensional element space through a non-straight transformation and after that minimizes the clustering error.

We have developed a hybrid approach to speed-up the kernel k-means clustering and its application in the socially distributed environment. The large dataset can be clustered by using a hybrid approach to speed up the clustering process. The service provider creates one helper user in each group to interact with the service provider and users. The helper user that is chosen by the service provider is trustworthy for users and the service provider. When the user sends his/her private message to another user, the helper user interacts with the user and converts the user's private message into random variables by using a homomorphic algorithm and send it to the service provider. The service provider receives the encrypted message and sends it to the particular user. Then the user decrypts the message by using the public key. The detailed proposed system is described for privacy-preserving the user data.

The paper is organized as follows: Sect. 2 presents the review of related work and Sect. 3 contains the motivation of this research. Section 4 presents the proposed technique of Hybrid approach to speed-up the kernel k-means clustering (*HSKK*) method. Section 5 provides the steps involved in the HSKK privacy-preserving data clustering method. Section 6 provides the results and discussion of the technique. Here the data set parameters are analyzed and the experimental results are noted. Also the comparison between existing Kernel k-means and proposed *HSKK* method is plotted. Finally, the conclusion of the proposed *HSKK* method is given in Sect. 7.

2 Review of Related Works

Literature presents several works for privacy preserving in distributed clustering method. Here, we review some recent works related to distributed clustering privacy method.

This paper describes Privacy preserving user clustering in a social network. Erkin et al. [1] have explained a solution based on secure multi-party computation techniques in a semi-honest environment. Here, proposal group's users in a social network for protecting their privacy-sensitive data against the server by encrypting the private data of the users. The server obtains neither the identity of the users nor the content of user data and a user gets a cluster identifier at the end of the protocol. To a genuine situation of a unified social network, this proposition gives an answer that is computationally effective and versatile. By achieving more privacy this also shows that communication cost of protocol reaches the same performance of the most similar work in the field. This protocol was also implemented and tested on the Movie Lens dataset. Experimental results of this algorithm show that this paper is both reliable and efficient for practical use.

Additionally, Ying-Hua et al. [8] have presented the State-of-the-art distributed privacy preserving data mining. The authors show how to mine the potential knowledge without revealing the sensitive data. Here the distributed privacy preserving data mining is proposed by three methods. Perturbation method focused on adding noise and random response. In this method, a study on blocking and condensation is done. The efficiency of an algorithm based on Secure Multiparty Computation combine the disturbance methods and restrict query methods to reduce the computation time and communication cost. Here restricted queries in a dynamic environment are needed.

There are some works by other researchers to synthesize and classify existing privacy-preserving distributed clustering. Erkin et al. [9] have presented a method based on encryption and secure multiparty computation techniques for clustering users in a centralized system. In that work, for accomplishing better execution as far as run-time and data transmission, the author kept the inclination vector of every user in the system hidden from every single other user and the service provider and uncovers the centroid location to the service provider. This strategy requires the participation of all users and here the normal correspondence and calculation cost is high as a result of homomorphic encryption. Here they examined an improved version of K-means clustering by proposing a three-party setting. In that work, user's private data were stored by one party and the decryption key by the other. A third party helps with the computations. While the overall system is highly efficient, this depends on trusting three separate parties that may not collude.

Moreover, Privacy-Preserving K-Means Clustering over Vertically Partitioned Data was proposed by Vaidya and Clifton [10]. Here k-means clustering method is presented for a common set of independent existence when different sites contain different attributes. Here each entity learns nothing about the attributes at other sites but learns the cluster of each entity. The secure permutation algorithm used here simultaneously computes a vector sum and permutes the order of the elements in the vector. Some of this work keeps up the provable security of individual data and limits on disclosure makes exchange offs amongst effectiveness and data disclosure and all disclosure was restricted to data that is probably not going to be of practical concern.

In addition, Javaid et al. [11] explained an energy-efficient distributed clustering algorithm for heterogeneous wireless sensor networks. Here they used enhanced developed distributed energy-efficient clustering method for heterogeneous wireless sensor networks. Both energy consumption of nodes and impact of radio environment gave careful consideration of this energy consumption.

Moreover, target coverage through distributed clustering in directional sensor networks was presented by Islam et al. [12]. In this paper, the authors have planned distributed clustering and target coverage algorithms. This paper determines the active sensing nodes and their directions for solving target coverage problems in Directional Sensor Networks. The system outperforms a various cutting edge approaches in the extensive simulation study. In directional sensor networks, this paper has presented a cluster head based distributed target coverage algorithm. The proposed target covering distributed clustering system is the main way to deal with address the most extreme target coverage with a minimum number of sensor nodes issue

utilizing cluster heads. In the target covering distributed clustering system, the cluster heads, and gateways are determined first, and then each cluster head coordinates selection of the active sensor nodes and their sensing direction.

Another author Chen et al. [13] has explained an Improved Distributed Clustering Algorithm Based on Density. Here the authors explained an improved density based distributed clustering algorithm. To improve the efficiency of the implementation of the local clustering, this algorithm used a data grid mapping method which mapped data object to the space grid first in the local level. In the global clustering level of the new algorithm, they proposed a global clustering method based on representative points intersection and uses the central point of a representative point to reduce the clustering error. Here first distributed clustering algorithm was improved based on density. Then based on the representative intersecting points they proposed a global clustering algorithm in the clustering phase of the improved one.

Additionally, Massin et al. [14] have presented a distributed clustering algorithm in dense group-based Ad Hoc Networks. Here the authors proved the theoretical convergence of dynamic clustering with operational groups clustering algorithm to be used in group-based ad-hoc networks. The algorithm has been developed by numbers of users in that group and by the closest distance of each cluster size. It is operated in large scale dense networks based on groups and it ensures good performance in relation to end-to-end delay and network stability.

In addition, Zhang et al. [15] have explained Improvement of distributed clustering algorithm based on min-cluster. Here in privacy-preserving distributed clustering arithmetic, the min cluster concept is adopted by data string and joined in multi-security protocols. Additionally min-cluster, as another idea, is proposed when the edge node of the dataset is clustering, which gains the information of edge node ground the refinement of clustering frequently, and exchanges the outcome to the inside node specifically. The center node again integrates the clustering data. This methodology has the advantages of network overhead decrease and less iteration in local stations. Here the authors concluded that the distributed clustering algorithm is proved to be with high accuracy, with low time complications, and the last but most important security.

3 Motivation of the Research

To share interests, activities, backgrounds, or real-life connections, people use the social community as a platform to build social relations. Besides, Social Network (*SN*) ties the relationships between two users who were connected in *SN* and this is the foundation for effective collaboration among users. In *SN*, the strength of social ties can be used to facilitate effective data forwarding and service recommendation. In *SN*, social community implies trust relationships and helps *SNUs* build trust relationships in a distributed way. However, the challenge of such applications is privacy and security. In fact, the privacy-sensitive destinations and healthcare symptoms are unlikely shared in *SNs* with strangers. Without a trusted mediator, the privacy is easily violated, and thus the *SNUs* are probably uncooperative. *SN* still faces many security and privacy challenges, including private information leakage, cheating

detection, Sybil attacks etc. The use of Social Networking websites and applications is also increasing day by day.

SN security and privacy are urgent research issues since various *SN* applications are widely launched in an insecure *SN* environment. The *SNs* are tightly related to the specific application design and a user's unique requirements such as security and privacy issues. However, in practice, most *SNUs* choose to ignore the privacy settings and put themselves in potential danger. Many users are not properly informed of the risks associated with using these sites and application. Users can choose to have their check-ins posted on their accounts on Twitter or Facebook. The location-based application collects and utilizes locations, which are most privacy-sensitive to *SNUs*. Inappropriate disclosure of locations to potential attackers may put the *SNUs* lives in danger or cause property loss. Considering these risks and challenges, privacy preservation should be addressed to avoid potential loss of private and personal information. The very accomplishment of applications based on discovering similar individuals relies on upon the exactness of grouping users which is specified relative to the measure of gathered user data. Since the content of the data is mostly privacy sensitive, the protection of the data is a growing concern among users [1]. Even if the service provider (*SP*) protects its database against a common security problem of identity theft, there is no guarantee to prevent information from being passed on without consent.

A trusted third party that keeps the data and runs the algorithm instead of the *SP* is a possible solution to protect the privacy sensitive data and it should be trusted by both user and *SP*. Unfortunately, doing every bulky computation is not sensible by having a third party that is completely trusted and willing. Deploying cryptographic protocols is a genuine solution to protect the privacy sensitive data of the users. Accepting that the server and the customers are semi-honest and they share a huge number of information. It is possible to have a protected system where no information is revealed beside the outcome of the estimation run. With such an arrangement, fraud and manhandle of user data by the organization is impossible without having the secret key that is used by customers for security purpose.

In a social network city, there is a lot of peoples connected to the service provider in order to review for social media sites. The users were interacting among themselves by using wireless connections such as Bluetooth, Wi-Fi, and some people through cellular network etc. The users (U) have various communication technologies to reach *SP.* Mobile or Computer applications enable the users to watch online videos (YouTube), update personal information (Facebook), share photos (Flickr), search for information (Foursquare) and talk to neighbors (Say Hello) anytime anywhere. In the social network city, each and every user have their own properties i.e. characteristics. They are rating according to their interest from different places. But their properties, that is the user similarities must be hidden from the *SP* and the ratings that are given by the users can be revealed.

There are a huge number of users $U = \left(u_1, u_2, u_3, \ldots, u_n\right)$ in a social network connected with the (*SP*). The number of groups $G = \left(g_1, g_2, g_3, \ldots, g_n\right)$ depends on the number of user in a system. Inside the group, the users are clustered based on their similarities according to their properties. The cluster is denoted by (C). Our aim is to protect the user's privacy message from the *SP*. So the *SP* chooses

one helper user (H_u) from each group. H_u that is chosen by the service provider is fully trusted by both the user and service provider. While the user sends a message to another user, H_u interacts with the user message and encrypt the message using the private key (P_k) and sends a message to the service provider. Then the encrypted message (E_m) passes through the service provider as random variables. Then the message will be decrypted by using public key (P_k) while reaching the destination. So another user will receive the decrypted message (D_m). Thus the message sends by the user is more privacy protected by using encryption method.

The user behavior based classification for the privacy concern is a difficult task in distributed clustering method. In general, for user clustering in *SN* k-means algorithm has been applied. Here in distributive clustering method, a k-means clustering algorithm is used to cluster data into groups via an iterative process. For privacy preservation in user clustered group, we use cryptographic protocol and our aim is to protect our private data from the *SP*. For that, the *SP* chooses one helper user (H_u) from each group. That H_u hides the private data from an *SP* by using homomorphic encryption method. Hence our proposed method will provide more privacy for transferring private data's. The social network architecture is given below in Fig. 1.

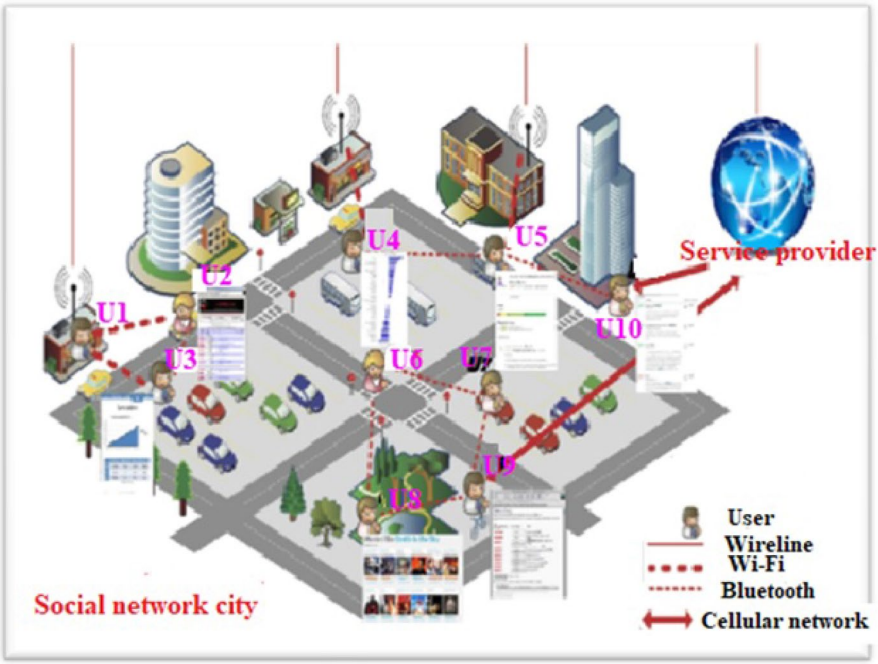

Fig. 1 Social network Architecture: The users (U) have various communication technologies to reach Service provider (SP). Mobile or Computer applications enable the users to watch online videos (YouTube), update personal information (Facebook), share photos (Flickr), search for information (Foursquare) and talk to neighbors (Say Hello) anytime anywhere

4 Hybrid Approach To Speed-Up the Kernel K-Means Clustering (HSKK)

In this paper, for the clustering process, we utilize leader based kernel k-means clustering algorithm. The normal k-means clustering algorithm is used to cluster the input data. To speed up the k-means clustering process, in this paper hybrid approach is introduced.

4.1 K-means Algorithm

Data is partitioned into smaller subgroups with their members sharing a common property by using a common data clustering technique for statistical data analysis. Recent years [9], in distributed data mining for security purpose K-means clustering algorithm is used. For cluster analysis in signal processing, the popular method used is vector quantization. While using k-means clustering algorithm K value is difficult to predict and it did not work well with global clustering and in different final clusters. It does not work properly in original data of different size and different density. Linearly inseparable clusters are also not effective in the input space. So in order to deduce these limitations, we choose Kernel k-means clustering method.

4.2 Kernel K-means Clustering Method

The non-linear extension of k-means is Kernel k-means algorithm [16]. Clusters in the input space which are non-isotropic and linearly inseparable can be effectively identified using Kernel k-means. In the feature space, it is an iterative method of input space where the clustering error is minimized in data points through a non-linear transformation $\phi(.)$ that is mapped from a higher dimensional feature space as same as the k-means clustering method. The process that is used to increase the relationship between variables is known as non-linear transformation. Some standard kernel functions are given as follows [17];

- Linear kernel:

$$\mathrm{K}\left(u_i, u_j\right) = u_i.u_j \tag{1}$$

- The polynomial kernel of degree a:

$$\mathrm{K}\left(u_i, u_j\right) = \left(u_i.u_j + 1\right)^a \tag{2}$$

- Radial kernel:

$$\mathrm{K}\left(u_i, u_j\right) = \exp\left(-\frac{\left\|u_i - u_j\right\|^2}{2\omega^2}\right) \tag{3}$$

These kernel functions are involved in the iterative process for the value of $\mathrm{K}\left(u_i, u_j\right)$. The value of i and j is $(1, 2, \ldots, n)$.

The number of clusters is denoted by k. The data set of size n is given by $S = \left\{u_1, u_2, u_3, \ldots, u_n\right\}$ and initial seed point is given by $\varepsilon^{(0)}$. Let the resultant initial partition be C. Kernel k-means produces entire data set and its input parameters are k and $\rho^{(0)}$,ρ_D is the output.

The objective function is to minimize the criterion function;

$$F = \sum_{r=1}^{k} \sum_{x_i \in C_r} \left\|\phi\left(X_i\right) - m_r\right\|^2 \tag{4}$$

Here, m_r is the mean of the cluster C_r, for $r = \{1, 2, 3, \ldots, k\}$ in the induced space and m_r is given by

$$m_r = \sum_{x_i \in C_r} \frac{\phi\left(X_i\right)}{\left|C_r\right|} \tag{5}$$

The distance between two data points $\phi\left(u_i\right)$ and $\phi\left(u_j\right)$ in the induced space is induced as

$$\begin{aligned} \left\|\phi\left(u_i\right) - \phi\left(u_j\right)\right\|^2 &= \phi\left(u_i\right) \cdot \phi\left(u_i\right) - 2\phi\left(u_i\right) \cdot \phi\left(u_j\right) + \phi\left(u_j\right) \cdot \phi\left(u_j\right) \\ &= \mathrm{K}\left(u_i, u_i\right) - 2\mathrm{K}\left(u_i, u_j\right) + \mathrm{K}\left(u_j, u_j\right) \cdot \end{aligned} \tag{6}$$

$\left\|\phi\left(u_i\right) - m_r\right\|^2$ can be calculated without the transformation $\phi(\cdot)$ explicitly is given by

$$\begin{aligned} \left\|\phi\left(u_i\right) - m_r\right\|^2 &= \left\|\phi\left(u_i\right) - \sum_{u_l \in C_r} \frac{\phi\left(u_i\right)}{\left|C_r\right|}\right\|^2 \\ &= \mathrm{K}\left(u_i, u_i\right) - J\left(u_i, C_r\right) + L\left(C_r\right) \cdot \end{aligned} \tag{7}$$

Here $J\left(u_i, C_r\right)$ is given by,

$$J\left(u_i, C_r\right) = \frac{2}{\left|C_r\right|} \sum_{u_l \in C_r} \phi\left(u_i\right) \cdot \phi\left(u_i\right) = \frac{2}{\left|C_r\right|} \sum_{u_i \in C_r} K\left(u_l, u_i\right) \tag{8}$$

$$L\left(C_r\right) = \frac{1}{\left|C_r\right|^2} \sum_{u_l \in C_r} \sum_{u_s \in C_r} \phi\left(u_l\right) \cdot \phi\left(u_s\right) = \frac{1}{\left|C_r\right|^2} \sum_{u_i \in C_r} \sum_{u_s \in C_r} K\left(u_l, u_s\right) \tag{9}$$

The iterative process of kernel k-means gives the output $\rho_{S^{\cdot}}$. Even though Kernel k-means is advantageous than k-means, that too has drawbacks. Because of quadratic time complexity, for large dataset Kernel, k-means method is not suitable. So, here we are introducing Hybrid approach to speed-up the kernel k-means clustering.

4.2.1 Leaders Clustering Method and its Modifications

In order to use Kernel k-means algorithm in large data sets, we developed a Hybrid approach to Speed-up the Kernel K-means clustering (*HSKK*). Here, the leaders are chosen for the clustering process. It is a single scan method which divides a large number of data sets into the groups in the input space in linear time. To partition the dataset into a number of clusters; the leader clustering method takes the size of each cluster, called the threshold T as an input parameter. Clusters are mentioned by a pattern as a leader and other patterns in the cluster are mentioned as followers. Set of leaders A are maintained initially empty and is incrementally built. If there is a leader $a \in A$ such that distance between u and a is less than or equal to T, for each pattern in the dataset S and then the pattern is assigned to the cluster represented by a. In this case, we call patterns as a follower of the leader and leader is a follower to itself. The first user which is at a distance less than or equal to T, is chosen as a follower of the leader. The pattern u becomes a new leader if there is no such leader and is added to A. The set of leaders A is provided as output by the algorithm. Modifications used in this proposed method are as follows:

- The clusters are not found in input space and it can be found only in kernel space.
- According to the pattern in the input space, each cluster is represented by its leader.
- All the patterns in each cluster can be retrieved easily when the datasets are re-indexed according to these clusters.

The principle behind proposed kernel based leaders clustering method is its linear time complexity. Based on the size of the input the running time increases linearly and its working principle is as follows:

For a given threshold T, a set of leaders A and the number of followers of each leader a is maintained by the kernel based leaders clustering method, which is count a. A is initially empty and is incrementally built. For each pattern u in the dataset S, if there is a leader $a \in A$, such that distance between $\phi(u)$ and $\phi(a)$ is less than or equal to T, then u is assigned to the cluster that is represented by a count (a) and the value is incremented by 1. Otherwise u becomes a new leader and is added to A, and count (a) becomes 1. The output of the algorithm is the set of leaders A, the number of followers of each leader i.e., count (a) and the set of followers of each leader a i.e., followers (a). This output is denoted by A^*. The proposed kernel based leaders clustering method is given in Table 1.

Table 1 Algorithm for Kernel-based leaders clustering method

Algorithm 1 : Kernel-based leaders clustering method (S,T)

Leader $A=0$

for each $u \in S$ **do**

Find a leader $a \in A$ such that $\|\phi(a)-\phi(u)\| \le T$ /* where

$\|\phi(a)-\phi(u)\|$ can be computed using the Equation

if there is no such a or when $A=0$ **then**

$A = A \cup \{u\}$;

$count(u)=1$;

$followers(u)=\{u\}$;

else

$count(a)=count(a)+1$;

$followers(a)= followers(a)\cup\{x\}$

end if

end for

Output:

$A^* = \{< a, count(a), followers(a) > a\ is\ a\ leader\}$

4.2.2 Proposed Hybrid Approach to Speed-up the Kernel K-means Clustering (HSKK)

A hybrid approach is proposed to speed up the kernel k-means clustering method. This method functions in two stages.

Stage 1: First, the kernel-based leaders clustering method is used to find A^*, as explained in Sect. 4.2.1

Stage 2: Later, to derive a partition of the set of leaders ρ_A, in the set of leaders A which is taken from A^*, the kernel k-means clustering method is applied over. In all iterations, each leader a_i is assigned to the cluster C_r such that $\left\|\phi(a_i) - m_r\right\|^2$ is minimized. Assume that the patterns in the cluster are very

close to the leader where it exists. Hence $\left\|\phi(a_i) - m_r\right\|^2$ is computed as follows:

$$\begin{aligned}\left\|\phi(a_i) - m_r\right\|^2 &= \left\|\phi(a_i) - \sum \frac{\phi(a_r)}{\left(\sum_{a_r \in C_r} count(a_r)\right)}\right\|^2 \\ &= \phi(a_i).\phi(a_i) - J(a_i, C_r) + L(C_r)\end{aligned} \tag{10}$$

where,

$$J(a_i, C_r) = \frac{2}{\left(\sum_{a_r \in (C_r)} count(a_r)\right)} \sum_{a_i \in C_r} \left\{count(a_r) K(a_i, a_r)\right\}, \tag{11}$$

$$L(C_r) = \frac{1}{\left(\sum_{a_r \in (C_r)} count(a_r)\right)^2} \{B_1 + B_2\}. \tag{12}$$

where,

$$B_1 = \sum_{a_r \in C_r} \left\{count(a_r)^2 K(a_r, a_r)\right\}, \tag{13}$$

$$B_2 = \sum_{a_r \in C_r} \sum_{a_s \in C_r} \left\{count(a_s) K(a_r, a_s)\right\}, \quad for\ a \neq s \tag{14}$$

Finally, each leader is replaced by all of its followers to get a partition of the entire dataset at the end of the iterative process and it is denoted by ρ_S^*. The proposed method is explained below in Table 2.

Table 2 Proposed prototype based hybrid kernel k-means

Prototype-based Hybrid Kernel k-means $(D, k, \varepsilon^{(0)}, T)$
Step 1: A*is generated by using the Kernel-based leaders clustering method that is given in Algorithm
Step 2: Using the given initial seed points $\varepsilon^{(0)}$ compute the initial partition $\rho_A^{(0)}$ of the leader set A
Step 3: Apply Kernel k-means clustering method $\left(A, k, \rho_A^{(0)}\right)$ and find the nearest cluster for a leader. Let ρ_A be the output
Step 4: To get the partition for the entire dataset, say ρ_S^* replace each leader $a \in \rho_A$, by its cluster
Step 5: Output is ρ_D^*

5 Proposed Model for Privacy-Preserving User Clustering through HSKK

Our aim is to protect the private data from the service provider. For that, we present a cryptographic protocol that clusters users in a social network using the k-means algorithm. The data of users must be transferred through the service provider and it is an unavoidable process. By using the k-means clustering technique, the users are separated into similar groups by the service provider in Fig. 2.

1. The *SP* creates *G* groups by using u_i and in each *G*, *SP* selects a random user H_u for all iteration. Here a total number of users in a social network are equal to a total number of users in all groups. Then the *SP* informs every user about the key distribution that is used for encryption.
2. The users receive encrypted cluster centroids from the *SP*. For every cluster centroid, each user computes cluster encrypted Euclidean distance and sends it to the *SP*. Here *HSKK* approach is used to speed up the process of clustering.
3. The *SP* interacts with the encrypted vector and encrypted matrices of each user for finding the partial inputs from cluster encrypted values. For updating the cluster centroids these values are used.

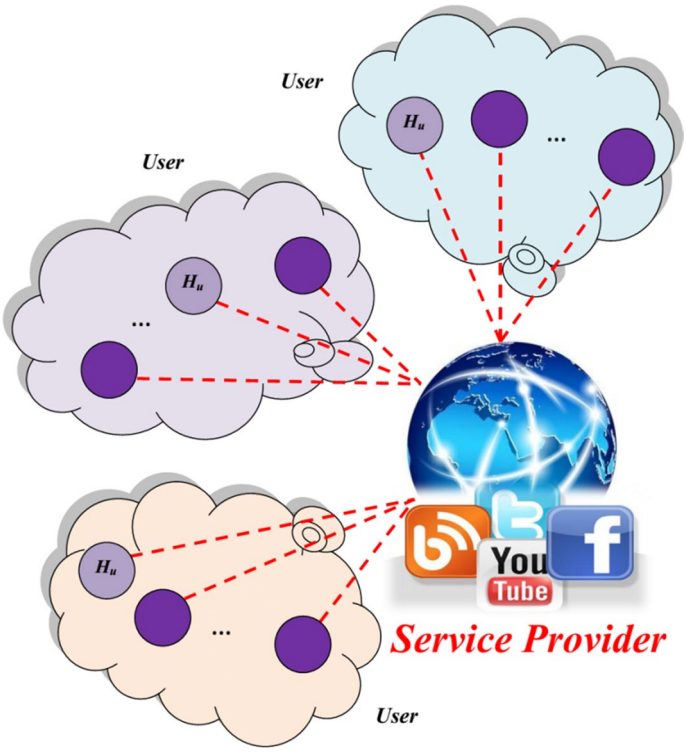

Fig. 2 The user groups of the secure *HSKK* clustering

4. The partial inputs of all users in the group interact with helper user H_u which is chosen by *SP* to obtain the clustering result. By using encrypted vector and encrypted matrices the cluster centroids are updated.
5. The *SP* combines the clustering results from all groups and new centroids are obtained for the iteration.

Step 1 to 5 is repeated for all iteration till the cluster centroids do not change significantly. In all iteration, every user is sent to the closest cluster by using the H_u that is chosen by *SP* in Table 3.

The following steps are the detailed explanation of the block diagram for the overall steps involved in distributed clustering method as shown in Fig. 3.

5.1 Step 1: Grouping into Clusters

Service provider chooses cluster points C with dimensional space M as the initial cluster centroid. Here we have to consider one-dimensional space according to the users U that we have taken. Next, the *SP* creates groups consisting of users and from each group service provider selects one H_u randomly. Helper user interacts between *SP* and user to transfer the data. The *SP* will treat helper user as an ordinary user in a group. The helper user encrypts the data, which is sent to the user in that group before the data reaches the *SP*. The *SP* informs all users in group G about the public key to be used for encryption, which is the public key of H_u. The *SP* interacts with the H_u to obtain an encrypted vector for each user, whose element indicates the closest cluster to that user. Then *SP* sends the vector to the user. The *SP* gets the partial inputs from all users in G and H_u is used to obtain the clustering results of group G in plain text. After completing the iterations, the *SP* runs a protocol with H_u to send the index of the closest cluster to each user. H_u applies masking by adding random variables to avoid information leakage and sends the remaining masked values to

Table 3 List of symbols

Symbol	Description	Symbol	Description
U	Users in social network	N_i	Encrypted Matrices
G	Group consists of users	S	Data Set
SP	social provider	$\varepsilon^{(0)}$	Seed point
H_u	Helper user	F	Objective function
SN	Social network	M_r	Mean of cluster C_r
C_r	Cluster	ρ_D	Resultant initial partition
K	Number of clusters	T	Threshold
E_m	Encrypted message	A	Set of leaders
D_m	Decrypted message	a	Leader
P_k	Public key	x	Follower
N_k	Private key	ρ_a	Output of Kernel k-means
M	Dimensional space	B	Bit length of the values used to compare
M_i	Encrypted vector	V	Numbers of movies in the subset

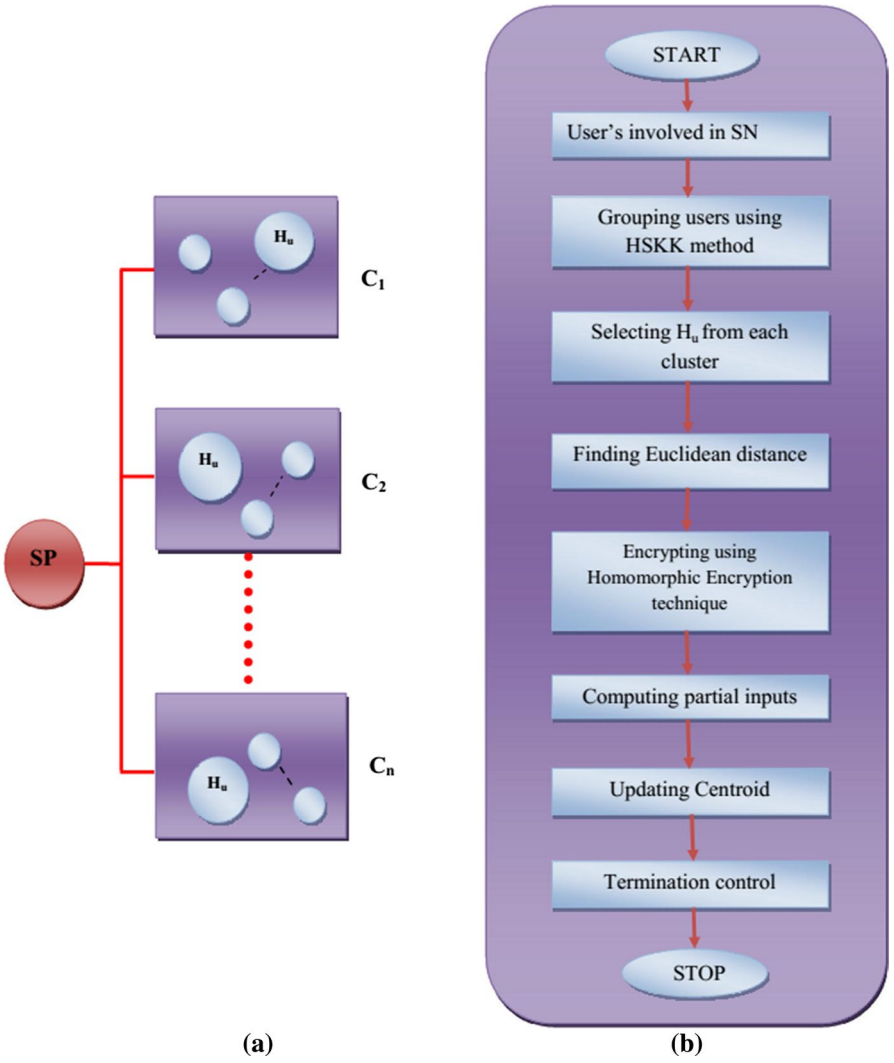

Fig. 3 Illustration of the proposed approach. **a** Block diagram to show the distributed clustering; **(b)** overall steps involved in distributed clustering method

the *SP*. These random values are encrypted by a public key of corresponding H_u. Finally, each H_u sends random values to the *SP*. The *SP* sends the ciphertext to H_u to be decrypted. After receiving the plain text, the *SP* sends the values to the users.

5.2 Step 2: Finding Encrypted distance

To assign the users in the closest cluster, Euclidean distance is to be found between the user u_i and C_r in a M dimensional space. Since a large number of

users is involved, it may take a lot of time. In order to improve the speed of clustering, *HSKK* method is included in this stage. The leaders were involved here to form the clusters based upon the user similarities.

$$E^2_{(i,r)} = \|\phi(U_i) - C_r\|^2 = \left\| \phi\left(u_i^l\right) - \sum_{u_r \in C_r} \frac{\phi\left(u_r^l\right)}{\left(\sum_{u_r \in C_r} count\left(u_r^l\right)\right)} \right\|$$
$$= \phi\left(u_i^l\right).\phi\left(u_r^l\right) - \frac{2}{\left(\sum_{u_r \in C_r} count\left(u_r^l\right)\right)} \sum_{u_r \in C_r} \left\{count\left(u_r^l\right) K\left(c_i^l, c_r^l\right)\right\}$$
$$+ \frac{1}{\left(\sum_{u_r \in C_r} count\left(u_r^l\right)\right)^2} \left\{B_1 + B_2\right\} \tag{15}$$

Assume,

$$\frac{2}{\left(\sum_{u_r \in C_r} count\left(u_r^l\right)\right)} \sum_{u_r \in C_r} \left\{count\left(u_r^l\right)\right\} = \delta$$

To obtain $[-2c_{i,r}]$ for all i and r, the server encrypts (-2) times its centroid location with the public key and publishes them. To compute the sum in the first term of the equation, the user calculates the encrypted Euclidean distance to each centroid and encrypts the value. Then to find the encrypted second term, homomorphism property of the Paillier cryptosystem is used. The user needs each encrypted centroid value $-2\left(c_i^l, c_r^l\right)$ to the user's location in the mth dimension and these values are multiplied. The encryption of the squares of the centroids is required for the calculation of the last term. At last, the user multiplies these values to obtain the Encrypted distances $E^2_{(i,r)}$ for their final value.

$$E^2_{(i,r)} = \phi\left(u_i^l\right).\phi\left(u_r^l\right)..\Pi_{u_r \in C_r}\left(-2C_i^l, C_r^l\right)^{\delta}.\frac{1}{\left(\sum_{u_r \in C_r} count\left(u_r^l\right)\right)^2}\left\{B_1 + B_2\right\} \tag{16}$$

Finally, each user possesses encrypted distance $(E_{i,1}.....E_{i,c})$ from its location U_i to the C centroids.

5.3 Step 3: Computing Partial Inputs

After finding the encrypted distance $(E_{i,1}.....E_{i,c})$ to each centroid, user i, needs to find the minimum of C encrypted values. For comparing two encrypted values, the cryptographic protocol is required. The encrypted vector $\left[M_i\right] = \left(\left[m_{i,1}\right] \ldots \left[m_{i,C}\right]\right)$ where $m_{i,j}$ is 1 if and only if $E_{i,r}$ is the minimum distance and 0 otherwise. For updating the cluster centroids, encrypted matrix $[N_i]$ is calculated. This value can be obtained by multiplication of $\left[M_i\right]^T$ and user point U_i in encrypted domain.

$$[N_i] = \begin{bmatrix} [m_{i,1}]^{U_{i,1}} & \cdots & [m_{i,1}]^{U_{i,M}} \\ [m_{i,2}]^{U_{i,1}} & \cdots & [m_{i,2}]^{U_{i,M}} \\ \cdot & \cdot\ \cdot & \cdot \\ \cdot & \cdot\ \cdot & \cdot \\ \cdot & \cdot\ \cdot & \cdot \\ [m_{i,C}]^{U_{i,1}} & \cdots & [m_{i,C}]^{U_{i,M}} \end{bmatrix} \tag{17}$$

The jth row of $[N_i]$ equals U_i when the user *is* in cluster j and 0 otherwise.

5.4 Step 4: Updating Centroid

After completing the calculation on forming their encrypted vector $[M_i]$ and encrypted matrix $[Y_i]$, they jointly form the protocol for updating the centroid. Here the user chain is created which will explain the procedure of N_i matrices and accumulation of vectors X_i and the bit of X_i is one. The users generate a random number r for each value in the matrix Ni and these values are used as a blinding factor, then the server won't know the value of individual matrices. Note that the initial user computes $(M_i)_{j,n} - (M_Q)_{j,n}$, Q being the number of users. The matrix M_i is sent to the left neighbor of the user chain. As a blinding value for $(N_i)_{j,n}$ each user computes $(M_i)_{j,n} - (M_{i-1})_{j,n}$. For keeping the blinding factors uniformly distributed, every user should compute $M_i - M_{i\text{-}1}$ modulo $2^{k'}$, before adding these to N_i. The user can't compute $N_i + (M_i - M_{i-1})$ modulo $2^{k'}$ since the matrix Ni is encrypted. Here extra random numbers are needed to mask a possible overflow of modulo $2^{k'}$. The extra random number should be k bits where it is a security parameter that is enough to mask the overflow to the server.

$$\left[\left(N_i'\right)_{j,n}\right] = \left[\left(N_i'\right)_{j,n}\right] \cdot \left[2^{k'}.\left(M_i'\right)_{j,n} + \left(M_i\right)_{j,n} - \left(M_{i-1}\right)_{j,n} \bmod 2^{k'}\right] \tag{18}$$

It is more efficient to use the homomorphic paillier property for the server to decrypt the matrix elements. The server will compute the matrix N'^{sum} by adding all matrix elements $\left[\left(N_i'\right)_{j,n}\right]$ overall user i. The server computes the actual Y^{sum} by decrypting $[Y'^{sum}]$ and computing Y'^{sum}modulo $2^{k'}$ as shown in the equation. This is the sum of all user points per cluster.

$$\begin{aligned} N_{j,n}'^{sum} &= \sum_{i=1}^{Q} \left(N_i\right)_{j,n} + 2^{k'}\left(M_i'\right)_{j,n} + \left(\left(M_i\right)_{j,n} - \left(M_{i-1}\right)_{j,n}\right) \\ &= \sum_{i=1}^{Q} \left(N_i\right)_{j,n} = \sum_{i=1}^{Q} 2^{k'}\left(N_i'\right)_{j,n} \\ &= N_{j,n}'^{sum} + 2^{k'}\left(M_i'\right)_{j,n} + \left(\left(M_i\right)_{j,n}\right) \\ &= N_{j,n}^{sum} \bmod 2^{k'} \end{aligned} \tag{19}$$

The same procedure is followed to calculate M^{sum} which is the number of users per cluster. The sum simply counts the number of users assigned to each cluster. The centroid can be updated by computing $c_{j,n} = N_{j,n}^{sum} / M_j^{sum}$ and rounding the result to the nearest integer by the server.

5.5 Step 5: Termination Control

The server checks whether the predetermined termination condition is reached at the end of iterations. Here the termination control cost is less because the centroid location and number of iterations are known by *SP*. The label information of the user which is the index of the non-zero element in the encrypted vector $[M_{\mathrm{i}}]$ should be revealed to the user after the termination condition is reached. For this purpose, every user performs the following operation to obtain cluster label information.

$$\begin{aligned} [Cd] &= \left[\sum_{j=1}^{C} m_{i,j} \times j \right] \\ [Cd] &= \prod_{j=1}^{C} \left(m_{i,j} \right)^{j} \end{aligned} \tag{20}$$

where $[Cd]$ denotes the cluster number that the user belongs to. Before sending the value to the server to be decrypted, the user additively blinds this encrypted value with a uniformly random element k of size $log(C)+c$ to get $[Cd+k]$ and re-randomize it. By subtracting k from the decrypted value sent by the server, the user can easily obtain his/her corresponding cluster label. Then the privacy-preserving K-means clustering algorithm is completed.

6 Results and Discussion

This section presents the results and discussion of the proposed prototype based hybrid Kernel k-means algorithm for privacy preservation in the social network. The prototype based hybrid Kernel k-means algorithm using cryptographic protocol is executed using JAVA and the experimentation is carried out on Movie Lens dataset.

6.1 Dataset Description

For our experiments, Movie Lens dataset [18] was used. The dataset contains 100,000 integer ratings in the range of [0.5] for 1682 movies by 943 users. As the minimal condition of the dataset is extraordinary (94%), a subset containing the movies rated by most users was considered. The corresponding row is filled with null entries of this subset were the user mean vote rounded to the nearest integer value. The number of movies in this subset, represented by *V*, also determines the parameter *b* which is the bit length

of the values to be compared. We set the number of clusters C to a single value, to show the equivalent accuracy between the proposed, privacy-preserved k-means clustering and Expectation maximization algorithm. In this paper, the privacy property of the proposed method has been proved theoretically, the evaluation focuses on comparing the efficiency and the accuracy of the proposed method with different methods. This data set mainly used for privacy preserving.

6.2 Quality Metrics

6.2.1 Clustering Accuracy (CA) (Fig. 4)

In the class, each cluster is assigned to compute CA and then the accuracy of this assignment is measured by counting the number of correctly assigned data and dividing it by N.

$$CA(\Gamma, \mathrm{P}) = \frac{1}{N} \sum_{k} \max_{j} \left| \theta_k \cap p_j \right| \tag{21}$$

where, $\Gamma = \{\theta_1, \theta_2, \ldots, \theta_k\}$ is the set of clusters and $\mathrm{P} = \{p_1, p_2, \ldots, p_j\}$ is the set of classes. We interpret θ_k as the set of data in θ_k and p_j as the set of data in p_j.

Reduction in Cluster Accuracy (RCA): Percentage of reduction in clustering accuracy

$$\frac{CA_{HSKK} - CA_K}{CA_K} \times 100 \tag{22}$$

where, CA_K—Clustering accuracy obtained using the conventional kernel k-means, CA_{HSKK}–Clustering accuracy obtained using the *HSKK* method.

Reduction in running time: Percentage of reduction in running time is

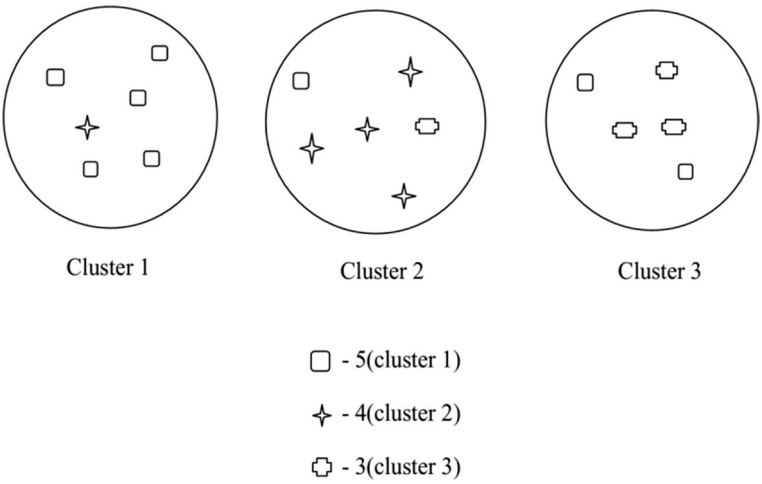

Fig. 4 Number of users in the majority class of three clusters. CA is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

$$\frac{RT_{HSKK} - RT_K}{RT_K} \times 100 \tag{23}$$

where RT_K-Running time for conventional kernel k-means, RT_{HSKK}-Running time of the *HSKK* method.

6.3 Experimental Results

The clustering accuracy *CA* and running time *RT* values of the proposed method are recorded for the best value of *T* in the specified range. The initial seed point is used to produce the final clustering point. To show the superiority of the *HSKK* method over the kernel k-means method, we calculate the percentage of reduction in *CA* and *RT* in ms for each Kernel k-means method.

For each value of *T*, the above range is repeatedly executed the proposed *HSKK* method with various orderings of each dataset to analyze the value of *CA* and *RT* value. The same set of initial seed points $\varepsilon^{(0)}$ is used for execution that is used in conventional kernel k-means method. Here different such parameters are used to calculate *CA* and *RT* value. In the linear kernel, the parameters are not used and then the value of *CA* and *RT* is calculated. When using the polynomial kernel, the parameter (a) such as 2,3,4,5 and 10 are used. The values of *CA* and *RT* vary in each parameter. In Gaussian (RBF) kernel, the parameters used are $\omega = 1, 2, 3, 4$ and 5. In each parameter the values of *CA* and *RT* are varied. *CA* and *RT* value for proposed *HSKK* method by varying parameters is given in Table 4. The linear, polynomial and RBF or Gaussian kernel are simply different in case of making the hyperplane decision boundary between the classes. Among three clusters RBF kernel attain the better result.

6.4 Performance Evaluation

In this section, the performance evaluation of the proposed prototype based HSKK method is compared with the existing Kernel k-means method and Expectation maximization (EM) clustering algorithm. EM is a probabilistic clustering which computes probabilities of cluster memberships based on one or more probability distributions. Here, we utilize the RBF kernel for the clustering process. Because RBF is given the better result compare to other two kernels. The runtime comparison between users and helper users are plotted and the clustering accuracy is also plotted below.

6.4.1 Runtime Comparison vs. Helper User

There are some parameters used to evaluate the performance by using a hybrid approach in Kernel k-means. By using the helper user, we can increase the performance of the proposed system because the H_u interacts between the *SP* and the

Table 4 *CA* and *RT* value for proposed *HSKK* method by varying parameters

Kernel	Parameter	CA (%)	RT(in ms)
Linear kernel	$a=1$	65.96	14,753,216
Polynomial kernel	$a=2$	62.34	7,126,342
	$a=3$	64.15	13,654,781
	$a=4$	65.23	16,365,642
	$a=5$	69.08	21,022,419
	$a=10$	67.23	25,647,412
Gaussian (RBF) kernel	$\omega = 1$	69.05	17,453,219
	$\omega = 2$	67.30	18,293,694
	$\omega = 3$	68.13	21,022,410
	$\omega = 4$	70.02	25,647,410
	$\omega = 5$	67.48	23,719,267

users. We measured the performance evaluation with the real and synthetic datasets to evaluate the effectiveness of the algorithm. In Fig. 5, the variation of time is plotted between Kernel k-means, EM and proposed HSKK method. The variation of time is mentioned in milliseconds and the time difference of the HSKK method is low while compared with Kernel k-means method and EM. The variation of users from 8 to 10 gives the time difference of 102794 ms. Hence by increasing the number of H_u, the time consumption is high. That means the proposed method will take only the less time when compared with the Kernel k-means and EM method.

6.4.2 Clustering Accuracy vs. Number of Users

Then the performance of accuracy is compared with the Kernel k-means, EM with the same datasets to HSKK method. The accuracy performance is mentioned in Fig. 6 with the number of users. In our dataset, 100,000 users are used in the performance of accuracy. As distributed clustering method is used here, the users are

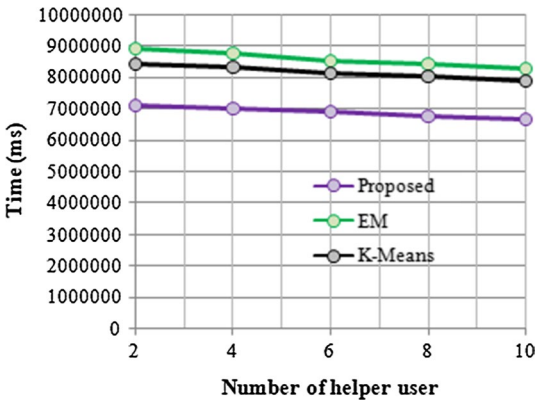

Fig. 5 Runtime comparison vs. helper user

partitioned in each process. First, the process is done between 20,000 users and the accuracy performance is compared between Kernel k-means, EM and the proposed HSKK method. While comparing the accuracy difference between kernel k-means, EM and the proposed technique in Fig. 2 for 20,000 users, the accuracy is varied by nearly 1.22%. In our proposed method, by increasing the number of users the clustering accuracy is high while compared with Kernel k-means method. In our proposed method the accuracy difference between 20,000 users and 40,000 users, the accuracy difference is 1.81%. Then by increasing the users, the accuracy level will be a little high. The user is 100,000 means, our proposed method attain the 2.85% better than k-means and 5.38% better than EM method. And in each and every set of users, the accuracy difference is high in HSKK while compared Kernel k-means and EM method.

6.4.3 Privacy Performance

In this section, we show the privacy preservation level in distributed clustering process between existing and proposed method using Man-in-the-middle (MITM) attack and Denial of service (DOS). Here, we compare our proposed work with existing kernel k-means clustering based privacy preserving an EM based clustering. While taking a number of clusters, the hacking percentage difference between the existing and proposed method is calculated.

A MITM attack is a kind of cyber attack where a noxious performer embeds him/her into a discussion between two gatherings, impersonates both sides and accesses data that the two gatherings were attempting to send to each other. A MITM attack permits a malevolent on-screen character to capture, send and get information implied for another person, or not intended to be sent by any means, without either outside gathering knowing until it is past the point of no return. Hacking percentage using man in the middle attack is shown in Fig. 7.

DOS is a type of attack on a network that is intended to push the network to the edge of total collapse by flooding it with futile traffic. Numerous DoS

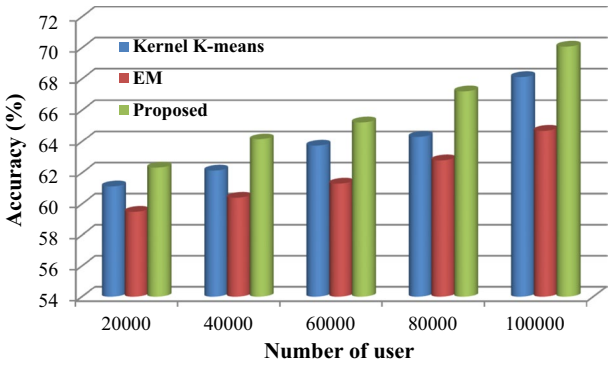

Fig. 6 Clustering accuracy vs. number of users

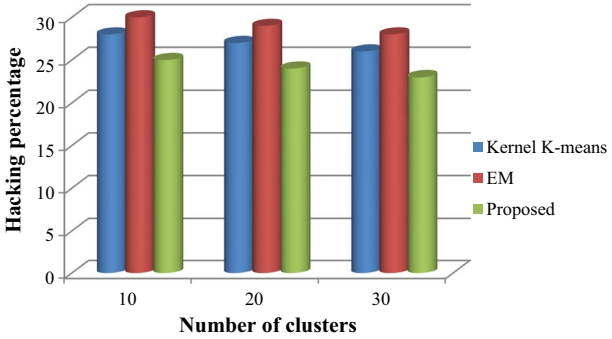

Fig. 7 Hacking percentage using man in the middle attack

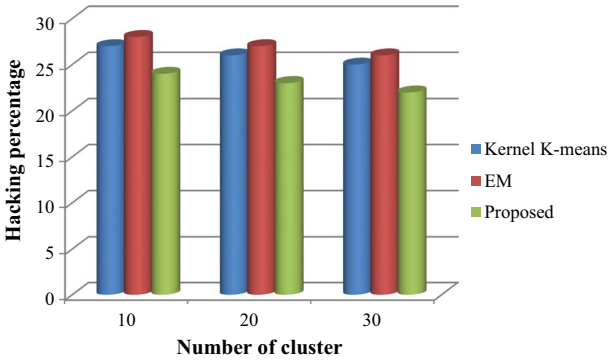

Fig. 8 Hacking percentage using denial of service

attacks, for example, the Ping of Death and Teardrop attacks, abuse constraints in the TCP/IP protocols. For every known DoS attacks, there are programming fixes that system administrators can introduce to restrain the harm brought about by the attacks. In any case, as like infectious, new DoS attacks are persistently being brainstormed by hackers. The plot of hacking percentage using man in middle denial of service attack is shown in Fig. 8.

7 Conclusion

In this paper, we present effective privacy-preserving distributed clustering approach in the social network applications. Initially, we have reduced the time in this large movie lens dataset by using a hybrid approach to speed up the clustering process in conventional Kernel k-means clustering method. Here we have proposed that while using social networking applications the private data of the users are kept hidden from the service provider by means of encryption using homomorphic encryption

technique. Here the helper user interacts with the service provider and users to convert the private data of users into random variables and send it to the service provider. This proposed method works with a set of leaders, instead of using large datasets in the entire iteration process. The size of the leaders kept depends on the threshold value. In our proposed method, the accuracy difference is 1.81% and for a maximum of 100,000 users in our dataset, the time difference between Kernel k-means and *HSKK* method is 5.27 min. The proposed hybrid approach to speed up Kernel k-means algorithm is more efficient in accuracy and time. The result based on homomorphic encryption in privacy-preserving K-means clustering algorithms can be improved further on a real system and this encourages the deployment.

References

1. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.:. Privacy-preserving user clustering in a social network. In First IEEE International Workshop on Information Forensics and Security (WIFS), pp. 96–100. IEEE, New York (2009)
2. Qi, X., Zong, M.: An overview of privacy preserving data mining school of technology. In: International Conference on Environmental Science and Engineering (ICESE 2011). Harbin University, Harbin, 150086
3. Sachan, A., Roy, D., Arun, P. V.: An analysis of privacy preservation techniques in data mining. In: Advances in Computing and Information Technology. Springer Berlin Heidelberg, pp. 119–128, (2013)
4. Vaidya, J., Clifton, C.W.: Privacy-preserving kth element score over vertically partitioned data. IEEE Trans. Knowl. Data Eng. **21**(2), 253–258 (2009)
5. Januzaj, E., Kriegel, H.P., Pfeifle, M.: Towards effective and efficient distributed clustering. In: Workshop on Clustering Large Data Sets (ICDM2003). (2003)
6. Dhote, C.A.: Homomorphic encryption for security of cloud data. Procedia Comput. Sci. **79**, 175–181 (2016)
7. Sarma, T.H., Viswanath, P., Reddy, B.E.: Speeding-up the kernel k-means clustering method: A prototype based hybrid approach. Pattern Recogn. Lett. **34**(5), 564–573 (2013)
8. Ying-hua, L., Bing-ru, Y., Dan-yang, C., Nan, M.: State-of-the-art in distributed privacy preserving data mining. In: 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 545–549. IEEE, New York, (2011)
9. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Privacy-preserving distributed clustering. EURASIP J. Inf. Secur. **2013**(1), 1–15 (2013)
10. Vaidya, J., Clifton, C.: Privacy-preserving k-means clustering over vertically partitioned data. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 206–215. ACM, New York, (2003)
11. Javaid, N., Rasheed, M.B., Imran, M., Guizani, M., Khan, Z.A., Alghamdi, T.A., Ilahi, M.: An energy-efficient distributed clustering algorithm for heterogeneous WSNs. EURASIP J. Wirel. Commun. Netw. **2015**(1), 1–11 (2015)
12. Islam, M.M., Ahasanuzzaman, M., Razzaque, M.A., Hassan, M.M., Alelaiwi, A., Xiang, Y.: Target coverage through distributed clustering in directional sensor networks. EURASIP J. Wirel. Commun. Netw. **2015**(1), 167 (2015)
13. Chen, J., Li, Y., Sun, P., Sun, M., Mao, R., Dong, L.: An improved distributed clustering algorithm based on density. In 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 133–136, IEEE, New York (2015)
14. Massin, R., Le Martret, C. J., Ciblat, P.: Distributed clustering algorithm in dense group-based ad hoc networks. In 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), pp. 1–7. IEEE
15. Zhang, Hao, Dai, GuangLong: Improvement of distributed clustering algorithm based on min-cluster. Optik Int. J. Light Electron Opt. **127**(8), 3878–3881 (2016)

16. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neur Comput. **10**(5), 1299–1319 (1998)
17. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines and Other Kernel Based Learning Methods. Cambridge University Press, Cambridge (2000)
18. Harper, F. M., & Konstan, J. A.: The movielens datasets: history and context. ACM trans. interact. intell. syst. **5**(4), 1–19 (2015)

P. L. Lekshmy obtained her Bachelor's degree in Information Technology from M S University, Tamilnadu in 2004. Then she obtained her Master's degree in Computer Science and Engineering from Karunya Deemed University, Coimbatore in 2006. Currently, she is working as Assistant Professor in Computer Science and Engineering, L B S Institute of Technology for Women, Trivandrum, University of Kerala (since 2008). Her current research interests are Privacy Preserving Datamining and Big Data analytics.

M. Abdul Rahiman received the Doctor of Philosophy (Ph.D.) degree in Computer Science & Engineering from Karpagam University. He obtained his Master of Technology from Kerala University in 2004, and Bachelor of Technology from Calicut University in 1998. He achieved Post Graduate Diploma in Human Resource Management from Kerala University in 2006 & Master of Business Administration (MBA) in 2008. He is an eminent academician and an able administrator. is currently the Pro Vice chancellor of APJ Abdul Kalam Technological University, (since September 2014). He has joined AICTE in February 2012 as its Director; he has been at the forefront of bringing in some radical changes for transparency and accountability in its administration through e-Governance. He was also appointed as Director Vocational Higher Secondary Education to the Government of Kerala. He has also served as a Faculty of Engineering at LBS Institute of Technology for Women, Trivandrum. He specializes in Digital Image Processing & Pattern Recognition and he taught for more than 10 years having a rich teaching experience and current research areas are Image and Computer Vision, Data Mining and Networking. He is also serving as Member of many professional & technical bodies; he has chaired many Technical Conferences. Also serving in the Editorial board of many International Journals. He was also a Member of Advisory body of Technical Education UT of Daman Diu, which guides the Technical & Higher Education area.