Computing and Informatics, Vol. 39, 2020, 537–567, doi: 10.31577/cai_2020_3_537

IMPROVED K-ANONYMIZE AND L-DIVERSE APPROACH FOR PRIVACY PRESERVING BIG DATA PUBLISHING USING MPSEC DATASET

Priyank Jain, Manasi Gyanchandani, Nilay Khare

Maulana Azad National Institute of Technology
Bhopal MP, India
e-mail: priyankjain88@gmail.com

Abstract. Data exposure and privacy violations may happen when data is exchanged between organizations. Data anonymization gives promising results for limiting such dangers. In order to maintain privacy, different methods of k-anonymization and l-diversity have been widely used. But for larger datasets, the results are not very promising. The main problem with existing anonymization algorithms is high information loss and high running time. To overcome this problem, this paper proposes new models, namely Improved k-Anonymization (IKA) and Improved l-Diversity (ILD). IKA model takes large k-value using a symmetric as well as an asymmetric anonymizing algorithm. Then IKA is further categorized into Improved Symmetric k-Anonymization (ISKA) and Improved Asymmetric k-Anonymization (IAKA). After anonymizing data using IKA, ILD model is used to increase privacy. ILD will make the data more diverse and thereby increasing privacy. This paper presents the implementation of the proposed IKA and ILD model using real-time big candidate election dataset, which is acquired from the Madhya Pradesh State Election Commission, India (MPSEC) along with Apache Storm. This paper also compares the proposed model with existing algorithms, i.e. Fast clustering-based Anonymization for Data Streams (FADS), Fast Anonymization for Data Stream (FAST), Map Reduce Anonymization (MRA) and Scalable k-Anonymization (SKA). The experimental results show that the proposed models IKA and ILD have remarkable improvement of information loss and significantly enhanced the performance in terms of running time over the existing approaches along with maintaining the privacy-utility trade-off.

Keywords: Big data, privacy, k-anonymization, l-diversity, multi dimensional generalization, improved symmetric and asymmetric k-anonymization, Apache Storm, MPSEC dataset

1 INTRODUCTION

Data analytics and data stockpiling process is mostly connected with big data. Presently there is an exponential growth of information, which is gathered, put away, and passed on within organizations and over the web. The sudden ascent of information has brought interest towards big data use, analytics, and raised scholastic intrigue. A brief check of google trends on the search interest has been an overall increase in activity since January 2011, with maximum attention being reached around October 2017, as shown in Figure 1. Big data is considered having massive information volume and complex information structures [1]. Few illustrations of big data are social and business site information, cell phone call records, geological data, web search tool information, smart card information, and so forth.

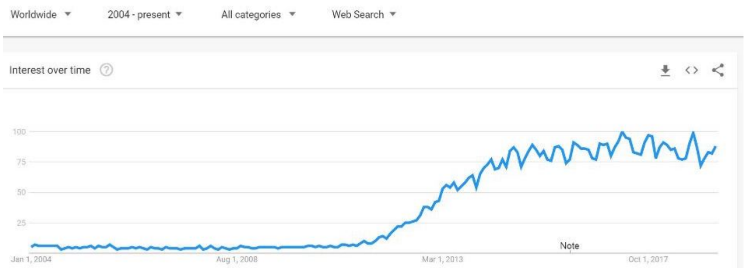

Figure 1. Search popularity of big data, source: Google trends (January 31, 2019)

Big data gives us benefits in various fields, for example in medicine, biology [2], banking [3], and social websites, and so on where a huge amount of data is collected. Now challenges are rising regarding its privacy and usage. One of the noteworthy utilization of big data use is offering data to various associations and analysts to comprehend social changes and make forecasts [4]. Approved associations, for example government organizations, banking sectors, medicinal research, have sensitive attributes in their dataset, where data distributing may cause data leakage for research purposes [5]. Differential privacy [6, 7, 21] with its expansions [8] seemed ten years back, which drives another bearing for protection saving. Subsequently, protection and privacy have to turn into an overall issue for present-day scientists. The significant issue in such a distribution is the disclosure of sensitive information, which is very bothersome [9]. Therefore, before the distribution of this information, adequate caution must be taken to conceal the sensitive details. To accomplish this, there should be a balance between privacy and utility of information. For this various algorithms has been proposed, but for a smaller dataset, like k-anonymization [16, 19, 20, 36], l-diversity [22] and t-closeness [10]. All these algorithms are based on the basic assumption that the records are free from each other and anonymization can be entirely autonomous. The widely used algorithm for data anonymization is k-anonymity [11]. For example, consider the multidimensional patient dataset shown in Table 1 that contains personal data of the patient. The personal data of the patient consists of four disjoint sets of data: explicit identifier (EI), quasi-identifiers (QI), sensitive data (SD), and non-sensitive data (NSD). In Table 1, EI attribute is name, QI attributes are age, pin code, SD attribute is a disease and NSD attribute is job. QI are those which alone cannot provide information about an individual, but when QI is linked with the external information, it can recognize the individual by connecting them. Generalization and suppression play a crucial role in anonymization. Generalization is a technique of replacing more specific values with generic and semantically similar values. Generalization can be applied at cell or the tuple or the attribute levels. Generalization uses the concept of the domain generalization and value generalization. Each attribute in the multidimensional database is a domain. In suppression, quasi-identifiers are replaced by *, and thereby it increases the privacy of database. Thus the size of the database and content of the database is reduced. k-anonymity checks that if one record in the dataset has some value of QI then at least $k-1$ other records also have the same QI values [12, 17, 18]. The equivalency among the data tries to maintain anonymity by k times [10]. Table 2 represents patient dataset after anonymization. As an instance of patient C = $\langle$"Carl"; 52; "flu"$\rangle$, this instance is generalized and suppressed to gc = $\langle *; [50-60];$ "Respiratory infection"$\rangle$.

Name	Age	Pincode	Job	Disease
Anand	45	400052	Writer	Flu
Bharti	47	400058	Writer	Pneumonia
Carl	52	400032	Lawyer	Flu
Diana	53	400045	Artist	Stomach ulcers
Emily	64	100032	Lawyer	Stomach infection
Fatima	67	100053	Lawyer	Hepatitis
Garvin	62	200045	Writer	Stomach cancer

Table 1. Patient table

Name	Age	Pincode	Job	Disease
$*$	40 > && > 50	40****	Writer	Respiratory infection
$*$	40 > && > 50	40****	Writer	Illness
$*$	50 > && > 60	40****	Lawyer	Respiratory infection
$*$	50 > && > 60	40****	Artist	Stomach disease
$*$	60 > && > 70	10****	Lawyer	Stomach disease
$*$	60 > && > 70	10****	Lawyer	Liver disease
$*$	60 > && > 70	20****	Writer	Illness

Table 2. After anonymization of patient table

One of the essential clarifications for the big data find the difficulty in k-anonymization. It works on a single-dimensional function [1]. The k-anonymity and l-diversity follow one group for all information, which significantly diminishes the obtained information, and sometimes the anonymized data is not replaced by an immediate parent; instead, it is replaced by a super parent. Implementing k-anonymity generalization in big dataset gives weak anonymization.

The multi-dimensional operation is supported by top-down generalization (TDG). The TDG strategy was proposed because of LKC privacy where L, K, C are thresholds [28]. That is used for centralization and distributed anonymization in multi-dimensional activity [13]. So applying multi-dimensional operation on it becomes multi-dimensional top-down generalization (MDTDG) [14, 15]. As a major task of anonymization, all k-anonymity methods implement the grouping process. Information typically assembled into proportionate or comparative records, known as compressions. The information loss rate decreases by using this technique.

Zakerzadeh et al. [30] introduced a new cluster-based algorithm for anonymizing numerical data streams using window processing known as fast anonymizing algorithm for numerical streaming data (FAANST). The main drawback of FAANST is that some tuples may remain in the system more than allowable time constraint. In addition, the time complexity of the algorithm is $O(n^2)$ and not efficient for data streaming [7]. Another weakness of FAANST is that it does not support categorical data. To remove this drawback another algorithm was introduced by Guo et al., FADS algorithm [31] for data stream anonymization, in which the time complexity of the approaches is $O(s)$, which is linear to the stream size s, also the space complexity is $O(c)$, which is constrained by a constant c. The main drawback of the FADS is that the algorithm does not check the remaining time of tuples that hold in the buffer in each round and that are outputted once they are probably taken into consideration to have expired. The other critical weakness of FADS is that it is not parallel and cannot handle a large number of data streams in tolerable time. Mohammadian et al. proposed FAST [32] to overcome the drawbacks of FADS. FAST protects the privacy of big data stream using parallel anonymization algorithm. It speeds up anonymization of data streams. A proactive heuristic approach was proposed in order to publish data before a specific expiration time passed. Proactive time expiration heuristic is applied to publish data before they are being expired. It works efficiently on a smaller dataset. Drawbacks of the FAST algorithm is that for the larger dataset, it results in high information loss, and the time complexity of the algorithm is comparatively high, i.e. $O(n \log n)$. Another drawback of this algorithm is that for anonymization purpose it takes super parent node for replacement instead of the current parent node to enhance running time which also causes high information loss. Zakerzadeh et al. [35] discusses the multidimensional k-anonymization Mondrian algorithm [34] and then proposes an anonymization technique for MapReduce framework: MRA. They proposed two versions of MRA. In the first version, a single global file is shared between all the nodes. The size of this file becomes larger and larger after each iteration as each node uses the same global file to update the equivalence class after each iteration. In the second version, there is no shared global file, but instead, it generates chunks of files distributed among all the nodes. Multiple iterations and file management are the major drawbacks of this technique, and as the number of iterations increases the performance decreases. In addition, the time complexity of the algorithm is $O(n^2)$. To overcome this, Mehta et al. proposed an SKA approach using MapReduce [36]. SKA divides the input dataset into smaller equivalence classes based on all the attributes of the dataset. Classes are merged gradually (one at a time) in order to make it large, enough to fulfill k-anonymity condition. These steps were repeated for all classes. SKA takes advantage of Hadoop's data distribution (Map) phase in the class division and sort and shuffling phase in class merging; hence, it works with lesser number of iterations, compared with the existing approach [35]. The time complexity of the algorithm is $O(n \log n)$. Lack of diversity and high information loss are the major drawbacks of this work.

As per the literature review of various big data privacy mechanisms, it is observed that existing privacy mechanisms are suffering the issues of high information loss and high running time for big data. Privacy on streaming data is still a challenge and needs to be solved. Thus in this work, the focus is on the development of privacy-preserving mechanism to reduce information loss and to reduce the time taken for streaming/batch big data.

1.1 Contribution

1. To improve the time efficiency of privacy preservation algorithms in comparison with the existing approaches (FADS, FAST, MRA, and SKA).
2. Proposed Improved Symmetric k-Anonymization (ISKA) and proposed Improved Asymmetric k-Anonymization (IAKA) reduce the information loss in comparison with existing approaches.
3. Achieving higher k-value guaranteed the strongest privacy.
4. Achieving high data utility with the same level of privacy compared with existing approaches.

1.2 Organization of the Paper

The flow of paper after the introduction is, initially, Section 2 discusses the proposed model. Section 3 presents an understanding of different datasets which are used in the experiment, Section 4 covers results and discussion, and Section 5 concludes the paper with a future scope.

2 PROPOSED MODEL

Data protection is a key factor; everyone wants data to be secured as far as privacy would not incline towards losing the prominence of information [23, 24, 25, 27]. Privacy here implies hiding the actual data in such a way that analytics operations can still be performed on the data but without losing the utility of data. The privacy breach of users in any organization can be prevented using the proposed IKA and

ILD model. This model can be used for both batch and streaming dataset. The results obtained using this model are more optimized in terms of running time and information loss as compared to that of the results obtained using the existing FADS, FAST, MRA, and SKA algorithms. From protection and security point of view, it guarantees that data subjects (i.e., people) have maintainable control over their data. Figure 2 represents the proposed model, in the pre-processing phase of data, the data is cleaned, and all the missing values and irrelevant values are expelled. In further steps, the data is anonymized by using IKA, which is categorized into two parts ISKA and IAKA. Here higher values of k represent the strongest privacy and these algorithms also resolve the suppression issue of the information loss, i.e., the child node is directly replaced by a super parent instead of replacing it by an immediate parent. Then the anonymized dataset is diversified using proposed ILD model. ILD applied to the result obtained after anonymization so that it certifies that there are at least two or more unique sensitive values in each equivalence class with no attribute disclosure.

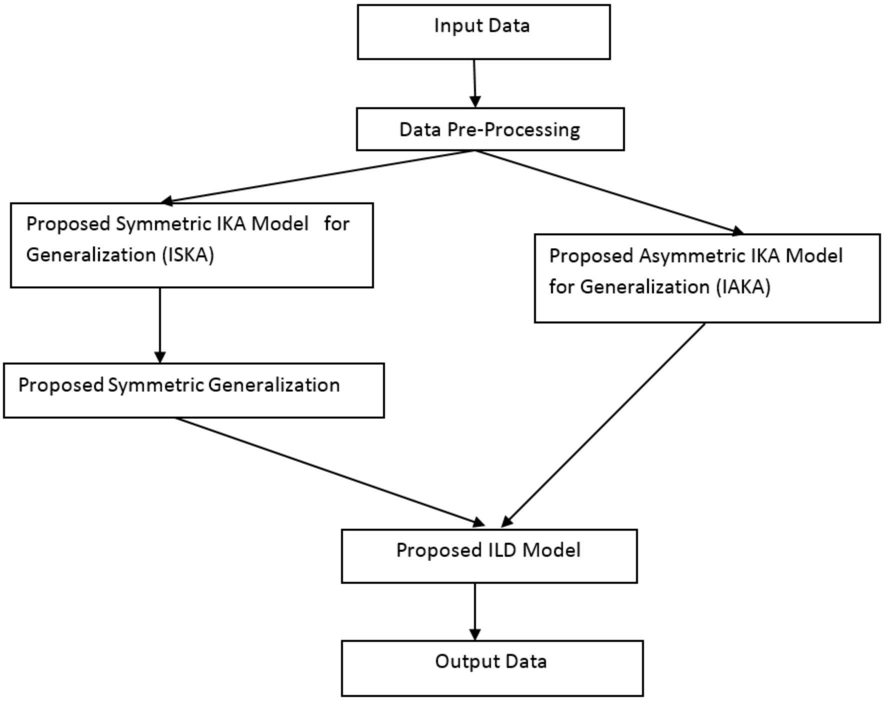
Input Data
Data Pre-Processing
Proposed Symmetric IKA Model for Generalization (ISKA)
Proposed Asymmetric IKA Model for Generalization (IAKA)
Proposed Symmetric Generalization
Proposed ILD Model
Output Data

Figure 2. Proposed model

2.1 Data Pre-Processing

The data obtained contains duplicate values, additional data of the same individual or missing values. It makes the data pre-processing a vital task. The primary goal of data pre-processing is to create an appropriate analysis suitable for the dataset. Data pre-processing maintains a strategic distance from the duplicate data and the missing values as indicated by the past recorded information. Likewise, it lessens the memory and normalizes the values that are put away in a database. For achieving anonymity, there is a need to erase the identifiers and adjust the quasi-identifiers and keep the sensitive attribute. The exactness of the attributes must be considered to choose which property is a sensitive attribute, identifier, or quasi-identifiers and care must be taken to select which feature is the sensitive attribute, which is the identifier and which is a quasi-identifier. Likewise, immaterial characteristics and qualities with no significance have to be erased. In the information pre-processing, the goal is to accomplish more advancement. Few attributes should be erased as they are neither material nor essential. The first procedure in this model is to eliminate data uncertainty by using information pre-processing. By breaking down the data, there is a realization that information has no noisy value.

2.2 Proposed IKA Model

The proposed model works in the direction of falsifying the generalized data, which will make data more generalized as well as distorted. Existing work has a significant drawback of a higher degree of suppression in case of categorical attribute where values were replaced by their super parent instead of immediate parent that causes more information loss. Referring to Figure 3, the value “Local govt.” should be replaced by the class of immediate parent (govt.), but instead, the superclass (work-class) is considered for generalization in the existing algorithms. There are following main reasons for this kind of high degree generalization that the individual (end-user) has specified the work-class as just local (instead of Local govt.) to the algorithm and it did not perceive it as a given workplace (because it is not matching with any of the nodes in the tree, i.e., Private, Govt. (Local gov., State gov., Federal gov.), Self emp., Without pay) so it directly went to the superclass which is “work-class” class. The proposed symmetric and asymmetric IKA model using the MDTDG technique overcomes those drawbacks. MDTDG generalizes a table which satisfies the anonymity requirement along with preserving its utility for classification. MDTDG compresses data to the topmost level, which is generalized by QI attributes [26, 29]. It takes various possible cases into account, for example if a user enters only local as its work class the algorithm will consider different possible values for same work-class (keeping account of different possible values for a single domain such as “local”, “local government”, “local gov” for “Local Gov.” work-class domain). The information loss rate decreased by using this technique. The proposed model generalizes k-anonymity into two different types of generalization for getting accurate results.

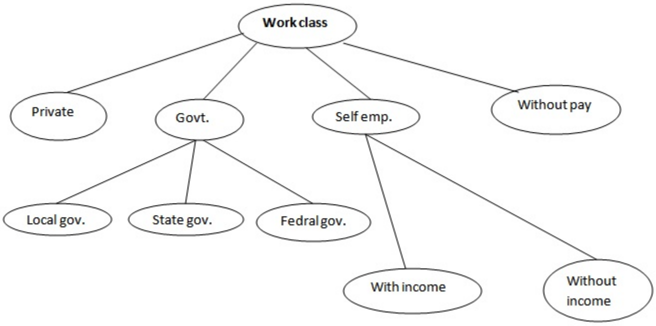

Figure 3. Taxonomy tree for work-class

2.2.1 Symmetric Anonymization

In any given dataset there is always one possible k value at a time symmetrically applied to the whole dataset. Symmetric anonymization takes equal k intervals to achieve privacy.

2.2.2 Asymmetric Anonymization

In this type of anonymization, the value of k will vary at a time. Asymmetric anonymization takes unequal k intervals to achieve privacy. The greater value of k is directly proportional to higher privacy. Asymmetric anonymization is able to achieve a higher k value as compared to symmetric anonymization. So the proposed framework endeavors to accomplish optimal k value as the higher is the k value; the more is the privacy.

The proposed Algorithm 1 is designed for both symmetric and asymmetric anonymization and tested for batch data and real-time stream data as well. The following topology used in proposed work, in which one spout (data source) and two Bolts $Bolt_1$ and $Bolt_2$ are used in *FIS* algorithm. In Figure 4 initially, input data stream s is sent into a spout that emits the data stream tuples. These tuples from spout are then sent to $Bolt_1$. It then makes Set of "delta" tuples at a time and then it inserts into Set named SOT. This set is fed as output to next bolt, i.e. $Bolt_2$. In $Bolt_2$, removal of k-anonymized clusters takes place, which is present longer than Tkc. In $Bolt_2$ function named $Publish(SOT)$ is called. After several steps, the output received from $Bolt_2$ is the k-anonymized tuples on which further processing will take place in Algorithms 2, 3 and 4.

In $Bolt_2$, pick one tuple T from SOT and publish that tuple by calling $PublishTuple()$ with SOT and T as parameters.

Algorithm 4 describes the Procedure of $PublishTuple(SOT, T)$. It is attempting to anonymize tuple T. At first, the system discovers its $k - 1$ closest tuples in SOT and embeds them in the new cluster called NEW and generalizes it into $gNEW$.

Algorithm 1 Proposed IKA and ILD algorithms

```
INPUT: Given dataset
OUTPUT: Improved $k$-anonymized and $l$-diversity data file
1. Step 1: cleaning(data, $A$) // Read the data from input file row-wise in a loop
2. If ((len(row) < actual_row_length) OR ( '?' in row) OR (' ' in row))
      Continue
   Else
      Writerow(row)
3. Step 2a: Asymmetric_ Anonymization($data, A$)
   a1: Sort according to the values of attribute $A$
   a2: $FIS(S, k, \delta, Tkc, Te, NumofExecutors)$
   Goto step 5
        Or
   Step 2b: Symmetric_ Anonymization($data, A$)
   b1: Sort according to the values of attribute $A$
   b2: $K =$ kgen($data, A$)
   b3: $FIS(S, k, \delta, Tkc, Te, NumofExecutors)$
   Goto step 5
4. Step 3: kgen($data, A$)
   $D =$ distinct values for attribute $A$
   For each value in $D$
   (a) Count[value] = 0
       For each row in data
   (b) Count[row[A]]+ = 1
   (c) Max = count[D[0]]
       For each value in $D$
   (d) Max = max(Max, count[value])
   (e) Return Max
5. Step 5: $ILD$ Model(data) For each equivalence class in data {
   If every value of a sensitive attribute in an equivalence class is equal
   {
   Add tuple from next equivalence class to current equivalence class, change some
   values for an attribute to achieve anonymity
   }}
```

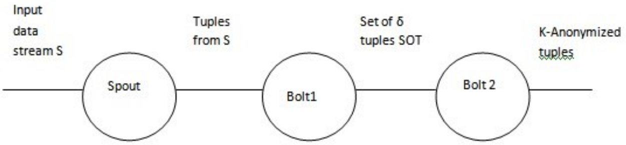

Figure 4. Topology for FIS

Algorithm 2 $FIS(S, k, \delta, Tkc, Te, NumofExecutors)$

```
While |S| ≠ 0 do
    1. In $Bolt_1$ $\delta$ tuples are read from $Spout$ and insert them into $SOT$;
    2. Output this $SOT$ to $Bolt_2$;
    3. In $Bolt_2$, all the clusters which exist longer than $Tkc$ are removed;
    4. Publish ($SOT$);
End while
```

Then a reusable cluster with minimum information loss *Ckbest* that covers tuple T, is chosen from SKC. If *Ckbest* exists and has smaller information loss compared to NEW, tuple T is published with *Ckbest* generalization and time of *Ckbest* is updated. Then other $k-1$ tuples that remain in SOT are checked whether they can be processed in another round or must be suppressed and published immediately.

Algorithm 3 Publish (SOT)

```
1. Pick the first tuple from $SOT$ and call it $T$;
2. $PublishTuple(SOT, T)$;
```

If tuple T does not match with any cluster in SKC which has less information loss than NEW, tuple T and its neighbors are published with NEW generalization $gNEW$. Then, $gNEW$ is inserted in SKC. Alternate tuples in SOT are checked for remaining time. If they have enough time to process, they are passed to SOT otherwise they will be suppressed and published. Figure 5 represents the flow chart of a streaming algorithm.

2.3 Proposed ILD model

Another motivation behind the proposed model is to accomplish variety in the sensitive attribute. Here to achieve the privacy, information is classified. Initially, the IKA model has been applied in the dataset and then the sensitive attribute is diversified by the proposed ILD model. The proposed ILD model is an improvement of l-diversity. For each equivalence class in data value of a sensitive attribute are less

Algorithm 4 *PublishTuple*(SOT, T)

```
1. Step 1: Select $k$–1 unique tuple from $SOT$ that are closest to $T$
   (a) Insert them into cluster $NEW$
   (b) Generalize $NEW$ into $gNEW$.
2. Step 2: For each cluster $C_k$ which covers $T$
   (a) Calculate the ILoss,
   (b) Choose a cluster with less ILoss
   (c) Call the $Ck_{best}$ cluster.
   (d) If $Ck_{best}$ exists and $Ck_{best}$ produces less ILoss than $gNEW$ then
         i Publish $T$ with $Ck_{best}$ generalization;
        ii Update $round_{time}$ estimation;
       iii $Synchronized(Ck_{best})$
           { Update $Ck_{bestpublishtime}$ }
        iv Do in $SOT$ for every tuple $t$
              if $(current_{time} - arrival_{time} + estimated\ round_{time}) < Te$ then
                 $Synchronized(S)$
                 { Insert $t$ as the first element of $S$; }
              else
                 Suppression and publication of $t$;
              end if
           end for
       else
         i Publication of $NEW$ with $gNEW$;
        ii Update of $round_{time}$ estimation;
       iii $Synchronized(SKC_t)$
           { Insert $gNEW$ into $SKC$ and set its time of publication; }
        iv Do in $(SOT - Set_{new})$ for every tuple $t$
              if $(current_{time} - arrival_{time} + estimated\ round_{time}) < Te$ then
                 $Synchronized(S)$
                 {
                    Insert $t$ as the first element of $S$;
                 }
              else
                 Suppression and publication of $t$;
              end if
           end for
   end if
```

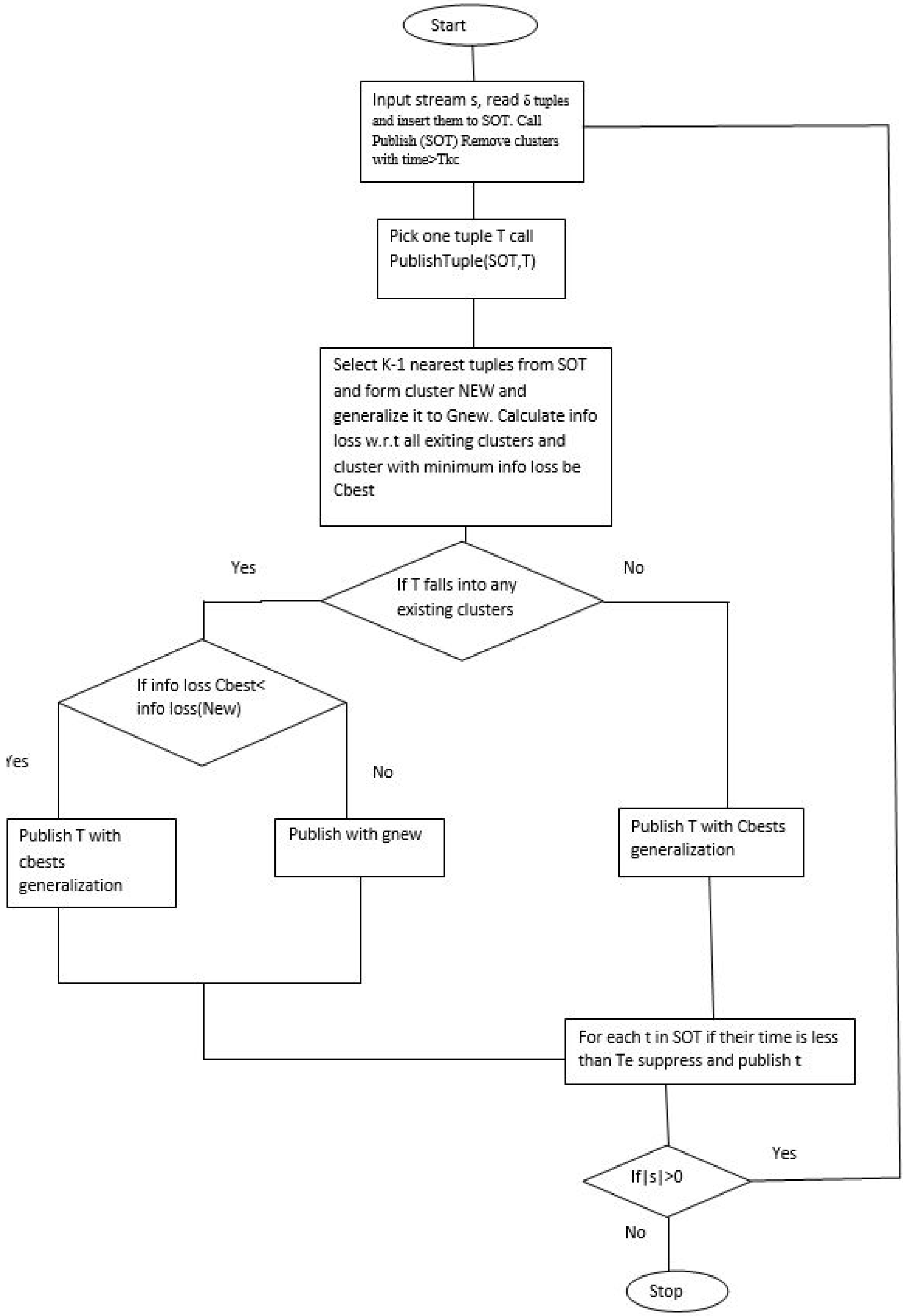
Start
Input stream s, read δ tuples and insert them to SOT. Call Publish (SOT) Remove clusters with time>Tkc
Pick one tuple T call PublishTuple(SOT,T)
Select K-1 nearest tuples from SOT and form cluster NEW and generalize it to Gnew. Calculate info loss w.r.t all exiting clusters and cluster with minimum info loss be Cbest
Yes
If T falls into any existing clusters
No
If info loss Cbest< info loss(New)
Yes
No
Publish T with cbests generalization
Publish with gnew
Publish T with Cbests generalization
For each t in SOT if their time is less than Te suppress and publish t
Yes
If|s|>0
No
Stop

Figure 5. Flowchart for stream data (Algorithms 2, 3 and 4)

than or equal to a threshold value of l then the proposed ILD model adds a different sensitive attribute tuple from the nearest equivalence class to a current equivalence class to achieve required threshold l-diversity. The proposed ILD model decreases the probability of attribute disclosure as compared to l diversity.

Table 3 represents an example of an anonymizing dataset of healthcare, which has a sensitive attribute is the disease. In the healthcare dataset having two equivalence classes, tuple 1–5 represents one equivalence class, and tuple 6–10 represents another equivalence class in the same dataset. Equivalence class 1 represents two-diversity, and equivalence class 2 represents three-diversity in the sensitive attribute. After applying the proposed ILD model in healthcare dataset in Table 4, to maintain the required threshold of 3-diversity in each equivalence class, the proposed ILD model adds a different sensitive attribute from the nearest equivalence class to a current equivalence class. Thus, the proposed ILD model increases diversity which also increases the privacy level.

S. No.	Non-Sensitive Attributes			Sensitive Attribute
	Zip Code	Age	Nationality	Disease
1	130**	< 30	*	Cancer
2	130**	< 30	*	Cancer
3	130**	< 30	*	Corona
4	130**	< 30	*	Cancer
5	130**	< 30	*	Cancer
6	130**	3*	*	Heart Disease
7	130**	3*	*	Heart Disease
8	130**	3*	*	Cancer
9	130**	3*	*	Cancer
10	130**	3*	*	Corona

Table 3. l-diversity before ILD model

S. No.	Non-Sensitive Attributes			Sensitive Attribute
	Zip Code	Age	Nationality	Disease
1	130**	< 30	*	Cancer
2	130**	< 30	*	Cancer
3	130**	< 30	*	Corona
4	130**	< 30	*	Heart Disease
5	130**	< 30	*	Cancer
6	130**	3*	*	Heart Disease
7	130**	3*	*	Heart Disease
8	130**	3*	*	Cancer
9	130**	3*	*	Cancer
10	130**	3*	*	Corona

Table 4. l-diversity after ILD model

3 DATASET USED

For the experimental purpose, this work used three datasets: the poker dataset, adult dataset and MPSEC dataset.

3.1 Poker Dataset

The poker dataset [37], collected from UCI, has 11 numerical attributes and millions of instances. Here the first ten predictive attributes are used as quasi-identifiers and t class variable is used as the sensitive attribute.

3.2 Adult Dataset

The adult dataset [38], collected from UCI, has 14 numerical and categorical attributes and 48 842 instances. This dataset is widely used for the privacy-preserving purpose. Here the sensitive attribute in the dataset is age (numerical) and profession (categorical).

3.3 MPSEC Dataset

This is for the first time when the proposed methodology has been implemented on a newly collected dataset from Madhya Pradesh State Election Commission (MPSEC), Bhopal, India. It is state voter and candidate dataset. It consists of 34 attributes. After pre-processing, 12 useful attributes were extracted, specifically age, district code, candidate name, gender, category, mobile no., candidate designation, ward no., votes, marital status, auto-id, and occupation. In the proposed work, "Occupation" is considered a sensitive attribute. The dataset has a candidate name and mobile number as EI attributes and the rest of the attributes are QI. We find the interesting patterns from MPSEC datasets, if combined with the demographic data, the percentage of people who were eligible and voted in the elections can be calculated. The correlation and dependency between the caste of the voters and the winning candidate can also be found. The co-dependency between the female candidates and their occupations might be detected. The percentage of female candidates amongst total candidates can be calculated. If combined with Aadhar card data, the voting pattern between the reserved category and unreserved category voters can be found. The percentage of different age groups standing for election can be detected, whether it has more of the young candidates or older candidates. Table 5 represents the number of tuples and the corresponding data size of MPSEC dataset for experiment purpose.

4 RESULTS AND DISCUSSION

The platform used for the deployment is HP Z840 workstation. It consists of 64-bit dual-core processors and 8 GB of RAM. Apache storm combination with Python is

MPSEC Dataset	
Number of Tuples	**Size**
35 000	4 MB
100 000	10.5 MB
1 million	104.2 MB
10 million	1.01 GB
100 million	10.4 GB

Table 5. MPSEC dataset

used to implement the proposed algorithms in the multi-node environment. The multi-node environment created by using 5 workstations. Each workstation has 40 cores. In our experiments, 40 cores are used for the name node, and 160 cores are used for worker nodes for implementing the proposed algorithms. The parameters used for comparison are completeness, running time, and information loss. Existing methods – FADS, FAST, MRA, and SKA – are implemented in the same environment. The proposed algorithms have been applied to MPSEC dataset, adult dataset [37], and poker dataset [38]. The proposed algorithms can also be applied to any other datasets which require the privacy mechanism.

4.1 Completeness

Completeness describes whether the data is fully anonymized or not. In the asymmetric algorithm, all data is generalized in an asymmetric way so that IAKA achieves 100 percentage completeness. The value of k achieved is 1 523. In ISKA, symmetric grouping value from k is changed. If (new $k <$ gen. k) $<$ 100 percentage completeness, data is only generalized, not anonymized, and it can be easily predictable, if (new $k >=$ gen. k). Here also, ISKA tries to achieve 100 percentage completeness, the value of k achieved is 1 283. The proposed model gives a better result with large dataset having higher k value. The higher k value guaranteed the strongest privacy.

4.2 Running Time

The running time complexity of IAKA and ISKA is described as follows, both the algorithms having mainly three functions. The first function is distance function, which is used for calculating the distance between two tuples, for finding the best nearest tuples for anonymization purpose. It is used by symmetric and asymmetric intervals for ISKA and IAKA algorithms, respectively. Distance function loop variable is incremented by a constant amount of time for both algorithms, which represents the time complexity of the distance variable being $O(n)$. The second function is the information loss function, and it is used to find the best optimal cluster, which is having minimum information loss. Similar to distance function, the information loss function is incremented by a constant amount of time. The time complexity of the information loss function is $O(n)$. The third function checks the expiration time of tuple, whether the time since tuple arrived is less than that of proactive heuristic and generalizes them within those time limits. This function takes a constant amount of time. So the overall time complexity of the proposed algorithms IAKA and ISKA is $O(n)$. After IAKA and ISKA algorithms, the proposed ILD algorithm maintains the required threshold of diversity in each equivalence class. In the proposed ILD model, diversity function loop variable is incremented by a constant amount of time, which represents the time complexity of the diversity function being $O(n)$. So the overall time complexity of the proposed ILD algorithm is $O(n)$. The time complexity of the proposed IKA and ILD algorithms found to be $O(n)$ where n = number of tuples of a given attribute. The proposed algorithms work on a lesser number of iterations: only three iterations in the case of both the algorithms of IKA and four iterations in the case of ILD algorithm. The comparison of time complexity of the proposed algorithms with the existing methods is shown in Table 6. The proposed work and the existing methods FADS, FAST, MRA and SKA have been implemented in the same experimental environment. In the experiment the anonymity degree k varied from 10 to 640 and the diversity level l is set to 6 in this proposed work.

The IAKA and ISKA are more efficient than the existing algorithms (FADS, FAST, MRA, SKA). The disadvantage of the FADS is that the algorithm does not take a look at the remaining time of tuples that are kept within the buffer in each round and are outputted once they are probably taken into consideration to have expired. The critical weakness of FADS and FAST is that they are not able to handle a larger dataset of 10M size. The major drawbacks of MRA are multiple iterations and file management, and as the number of iterations increases the performance decreases. In addition, the time complexity of the MRA is very high, i.e. $O(n^2)$. The time complexity of the SKA algorithm is also high, i.e. $O(n \log n)$. Lack of diversity is the major drawback of SKA, which causes attribute discloser. To overcome the FADS weakness the proposed algorithms check the expiration time of tuple, whether the time since tuple arrived is less than that of proactive heuristic and generalize them within that time limits. The running time has improved due to the proposed IKA and ILD algorithms which take only fewer iterations and execute on the multi-node environment of big data. The proposed improved algorithms are efficient to handle large database and proposed ILD model maintains at least 6 diversity in each equivalence class what overcomes attribute discloser. Comparing both algorithms of IKA, IAKA is more efficient as it is taking less running time as compared to ISKA. The running time declines with the increasing number of tuples or records, mostly because fewer iterations are required to satisfy privacy requirements. Tables 7, 8, 9 show the running time of MRA, SKA, IAKA and ISKA on 1M, 10M and 100M dataset with respect to different k values, respectively. In Table 7, when the value of k is 10 on MPSEC 1M dataset then the running time of IAKA and ISKA is 335 and 328 seconds and when the value of k is 640 the running time of IAKA and ISKA is 305 and 302.3 seconds,

S. No.	Algorithms	Time Complexity	Remark
1	FADS	$O(s)$, which is linear to the stream size s also the space complexity is $O(c)$, which is constrained by a constant c.	FADS algorithm does not check the remaining time of tuples and FADS is not parallel and cannot handle a larger dataset of 10M size.
2	FAST	$O(n \log n)$	Large dataset results in high information loss and cannot handle a larger dataset of 10M size.
3	MRA (Map Reduce-based Anonymization)	$O(n^2)$	Multiple iterations and file management are the major drawbacks of this technique and as the number of iterations increases the performance degrades.
4	SKA (Scalable k-Anonymization)	$O(n \log n)$	Lack of diversity and high information loss are the major drawbacks of this work.
5	**Proposed IKA and ILD algorithms**	$O(n)$	In the proposed algorithms complexity decreases due to lesser number of iterations.

Table 6. Comparison of time complexity of proposed algorithms with competing or existing methods

respectively. In Table 8, when the value of k is 10 on MPSEC 10M dataset then the running time of IAKA and ISKA is 2 070.2 and 1 989.3 seconds and when the value of k is 640 the running time of IAKA and ISKA is 1 641.5 and 1 555.8 seconds, respectively. ISKA performs best in running time and both IAKA and ISKA algorithms are outperformed as compared to the existing MRA and SKA algorithms. When the value of k is higher, i.e. higher privacy, then running time goes down in all the algorithms due to low computational cost required, similarly to Tables 7, 8, and 9, it is representing the running time on MPSEC 100M dataset, respectively. This work also finds an interesting pattern that the running time values of on MPSEC 10M and 100M dataset are not increasing proportionally as compared to Table 7, these running time values are much smaller by using our proposed algorithms. So our proposed algorithms take less running time as compared to the existing algorithms for larger datasets. Table 10 shows the running time of FADS, FAST, IAKA and ISKA on 1M dataset in which IAKA repeatedly performs best.

Value of	Running Time in Seconds			
k	MRA	SKA	IAKA	ISKA
10	1 200	675	335	328
20	1 189	541	330.2	327.5
40	1 089	500	326.3	324.3
80	1 000	472	321.2	320
160	920	421	317.3	316.2
320	880	400	310	308.2
640	812	398	305	302.3

Table 7. Running time comparison of MRA, SKA, IAKA, and ISKA on MPSEC 1M dataset in seconds

Value of	Running Time in Seconds			
k	MRA	SKA	IAKA	ISKA
10	8 000	4 010	2 070.2	1 989.3
20	7 800	3 900	2 000.2	1 965.1
40	7 100	3 508	1 905.3	1 845.8
80	6 600	3 280	1 857.6	1 780.3
160	6 200	3 121	1 779.8	1 697.4
320	5 807	2 872	1 701	1 623.2
640	5 410	2 710	1 641.5	1 555.8

Table 8. Running time comparison of MRA, SKA, IAKA, and ISKA on MPSEC 10M dataset in seconds

4.2.1 Comparison of IAKA and ISKA Algorithms with Existing Batch Data Anonymization Algorithms MRA and SKA

The average running times on 1M, 10M, and 100M datasets are depicted in Figures 6, 7, and 8, respectively. As can be seen in the figures, both IAKA and ISKA have smaller running time than that of MRA and SKA. And SKA has smaller running time than that of MRA [35, 36] because SKA performs the task in less number

Value of	Running Time in Seconds $* 10$			
k	MRA	SKA	IAKA	ISKA
10	7 800	3 840	1 980.2	1 909
20	7 100	3 509	1 920.5	1 875.8
40	5 200	2 690	1 797.2	1 745
80	4 100	2 150	1 791.8	1 680.6
160	4 400	2 198	1 699.2	1 588.4
320	4 120	2 098	1 600.8	1 505.6
640	3 900	1 850	1 541.8	1 432.2

Table 9. Running time comparison of MRA, SKA, IAKA, and ISKA on MPSEC 100M dataset in seconds $* 10$

Value of k	Running Time in Seconds			
	FADS	FAST	IAKA	ISKA
10	610	422.3	335	328
20	608.5	419.5	330.2	327.5
40	609	417	326.3	324.3
80	607	415	321.2	320
160	608	411.3	317.3	316.2
320	607.5	410	310	308.2
640	607	409.3	305	302.3

Table 10. Running time comparison of FADS, FAST, IAKA, and ISKA on MPSEC 1M dataset in seconds

of iterations than that of MRA. In the case of ISKA, there will be no overhead for checking the distance between the tuples as it is concerned more with equal-sized cluster, but on the nearness degree of tuples in the dataset. On the other hand, IAKA is related more with nearness of data than with the equality of cluster sizes. This requires to calculate the distance between the tuple of interest and the generalized cluster to decide whether the tuple can be inserted in that cluster or not which results in more running time of IAKA and relatively less running time of ISKA.

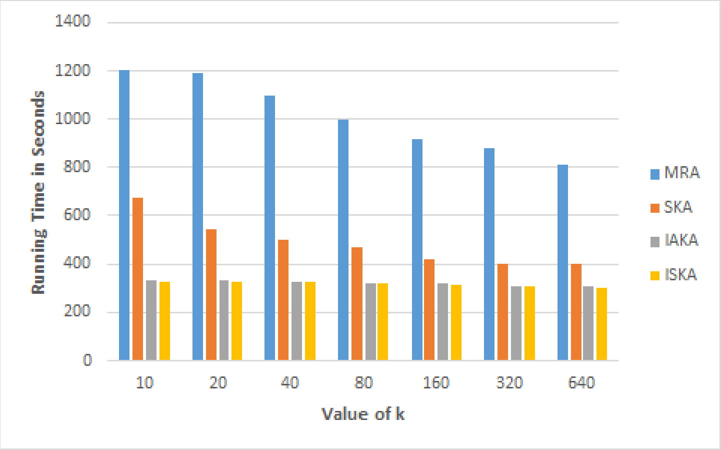

Figure 6. Running time of MRA, SKA, IAKA, and ISKA on MPSEC 1M dataset

4.2.2 Comparison of IAKA and ISKA Algorithms with Existing Stream Data Anonymization Algorithms FADS and FAST

The average running time on 1M synthetically generated MPSEC stream dataset are depicted in Figure 9. As can be seen in this figure, both IAKA and ISKA

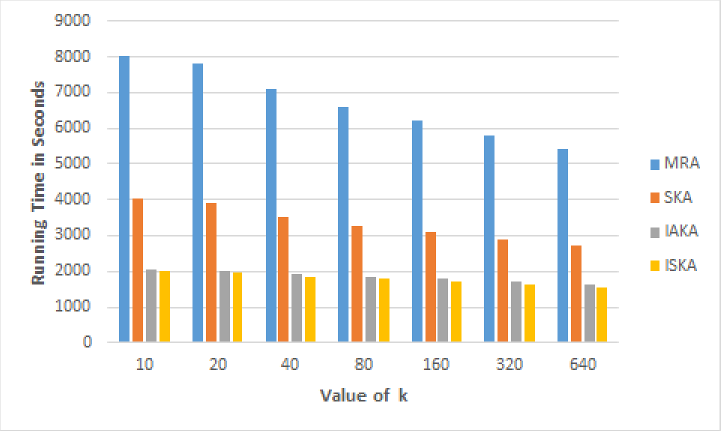

Figure 7. Running time of MRA, SKA, IAKA, and ISKA on MPSEC 10M dataset

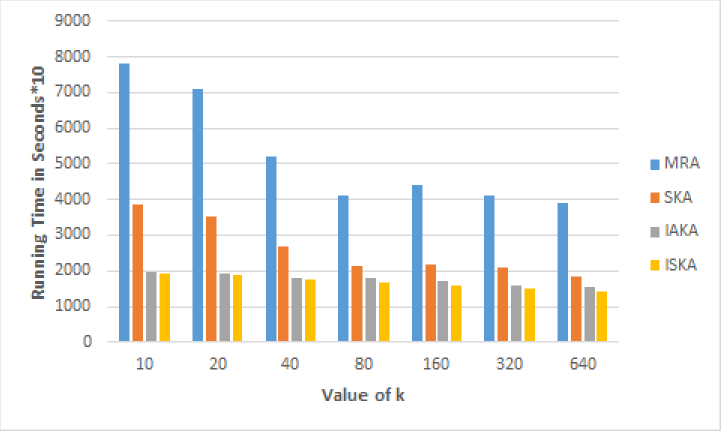

Figure 8. Running time of MRA, SKA, IAKA, and ISKA on MPSEC 100M dataset

have smaller running time than that of FADS and FAST. And FAST has smaller running time than that of FADS [31, 32]. FAST have smaller running time than that of FADS as its implementation uses the concept of multithreading. IAKA has larger running time than ISKA and the reasons are the same as mentioned in the Section 4.2.1.

4.3 Information Loss

Information loss is a term that is shown in Equation (1). In this equation, $lower_{ij}$ and $upper_{ij}$ represent lower and upper bound of attribute j in tuple i after general-

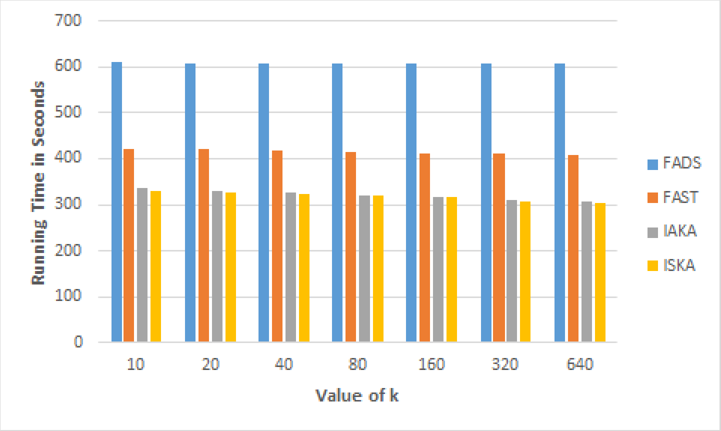

Figure 9. Running time of FADS, FAST, IAKA, and ISKA on MPSEC 1M dataset

ization, respectively, $\min_j$ and $\max_j$ represent the minimum and maximum values, respectively, taken by attribute j over all records.

$$I = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|upper_{ij} - lower_{ij}|}{n.m|Max_j - Min_j|}. \tag{1}$$

In the experiment, the anonymity degree k varied from 10 to 640, and the diversity level l is set to 6 in this proposed work. The reported values (except for the information loss) are averaged over two runs. Tables 11, 12, 13 show the information loss of MRA, SKA, IAKA and ISKA on 1M, 10M and 100M dataset with respect to different k values, respectively. In Table 11, when the value of k is 10 on MPSEC 1M dataset then the information loss of IAKA and ISKA is 18 and 21 percentage, and when the value of k is 640 the information loss of IAKA and ISKA is 27.39 and 28.97 percent, respectively. In the case of existing algorithms MRA and SKA in the same table, when the value of k is 10 on MPSEC 1M dataset then the information loss is 33.4 and 29.84 percent, and when the value of k is 640 the information loss is 54.40 and 44.87 percent, respectively. The proposed algorithms show better performance regarding the information loss as compared to the existing methods. In Table 12, when the value of k is 10 on MPSEC 10M dataset then the information loss of IAKA and ISKA is 15.28 and 17.89 percent, and when the value of k is 640 the information loss of IAKA and ISKA is 24.18 and 25.07 percent, respectively. In the case of existing algorithms MRA and SKA in the same table, when the value of k is 10 on MPSEC 10M dataset then the information loss is 31.2 and 25.9 percent, and when the value of k is 640 the information loss is 48.80 and 41.08 percent, respectively. As compared to Table 11, data size is increasing 1M to 10M, information loss is decreasing and proposed algorithms IAKA and ISKA show incredible performance, as compared to the existing algorithms.

In Table 13, when the value of k is 10 on MPSEC 100M dataset then the information loss of IAKA and ISKA is 12.8 and 14 percent, and when the value of k is 640 the information loss of IAKA and ISKA is 21.91 and 24.58 percent, respectively. In the case of existing algorithms MRA and SKA in the same table, when the value of k is 10 on MPSEC 100M dataset then the information loss is 27.12 and 19.41 percent, and when the value of k is 640 the information loss is 45.01 and 41.47 percent, respectively. It is clear that as data size increases, information loss decreases due to the large crowd effect. It is also observed that IAKA outperforms ISKA, MRA and SKA in terms of information loss. ISKA also presents remarkable improvements in terms of information loss as compared to existing methods.

In Table 14, there is a considerable difference in the information loss of our proposed IAKA and ISKA algorithms with streaming data FADS and FAST algorithm. This is because the FADS and FAST algorithms fail to take full advantage of the entire data, because of their inability to merge the values across big data chunks. Typically, a larger difference is expected if the data is split into more chunks. Another drawback of a FAST algorithm for anonymizing data is that it takes super parent node for replacement instead of the current parent node for categorical attribute to enhance time which results in high information loss. The drawback of MRA and SKA also is high information loss. Among IAKA and ISKA, IAKA has less information loss than that of ISKA because, in ISKA, the size of each cluster has to be the same and in order to satisfy this property it sometimes has to compromise over the nearness of the tuples in the cluster which is otherwise done for less information loss in the anonymized data. On the other hand, the IAKA does not impose any condition on the size of the cluster but concentrates more on the nearness of the data in the cluster that results in less information loss. Another efficiency of proposed IAKA and ISKA of IKA model is using the MDTDG technique for categorical attributes, so information loss rate decreased. The proposed algorithms also utilize the large crowd effects, i.e. the same amount of privacy applied to larger dataset. It is also able to achieve low information loss and privacy-utility trade-off.

Value of k	Information Loss in Percentage			
	MRA	SKA	IAKA	ISKA
10	33.4	29.84	18	21
20	38.57	32.08	20	22.4
40	41.72	36.12	21.32	23.6
80	44.08	38.27	23.87	24.74
160	47.40	39.9	25.18	25.75
320	51.81	41.87	26.43	27.88
640	54.40	44.87	27.39	28.97

Table 11. Information loss comparison of MRA, SKA, IAKA, and ISKA on MPSEC 1M dataset

Value of	Information Loss in Percentage			
k	MRA	SKA	IAKA	ISKA
10	31.2	25.9	15.28	17.89
20	36.01	29.8	17.87	19.72
40	39.92	34	19.08	21
80	42.02	36	20.87	22.57
160	44.08	37.01	21.84	23.27
320	46.09	39.84	22.70	24.89
640	48.80	41.08	24.18	25.07

Table 12. Information loss comparison of MRA, SKA, IAKA, and ISKA on MPSEC 10M dataset

Value of	Information Loss in Percentage			
k	MRA	SKA	IAKA	ISKA
10	27.12	19.41	12.8	14
20	32.5	24.05	13.49	16.72
40	37.61	30.58	16.19	19.29
80	40.8	32.41	18.09	21.08
160	43.21	38.48	19.18	22.39
320	44.75	40	20.04	23.68
640	45.01	41.57	21.91	24.58

Table 13. Information loss comparison of MRA, SKA, IAKA, and ISKA on MPSEC 100M dataset

4.3.1 Comparison of IAKA and ISKA Algorithms with Existing Batch Data Anonymization Algorithms MRA and SKA

The average information loss on 1M, 10M and 100M MPSEC datasets are depicted in Figures 10, 11 and 12, respectively. As can be seen in the figures, both IAKA and ISKA outperformed SKA and MRA. SKA outperformed MRA [35, 36]. The reason is that the former algorithm uses the proactive heuristic variable to maintain

Value of	Information Loss in Percentage			
k	FADS	FAST	IAKA	ISKA
10	38	31	18	21
20	42	33.4	20	22.4
40	47	35.8	21.32	23.6
80	49	39.2	23.87	24.74
160	51	43	25.18	25.75
320	57	45.5	26.43	27.88
640	65.3	51	27.39	28.97

Table 14. Information loss comparison of FADS, FAST, IAKA, and ISKA on MPSEC 1M dataset

the relativity of the data to the situation. As the size of dataset increases the information loss decreases due to large crowd effect.

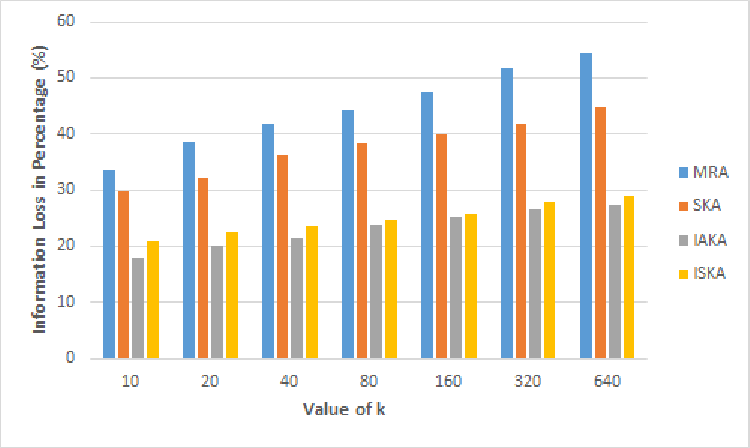

Figure 10. Information loss of MRA, SKA, IAKA, and ISKA on MPSEC 1M dataset

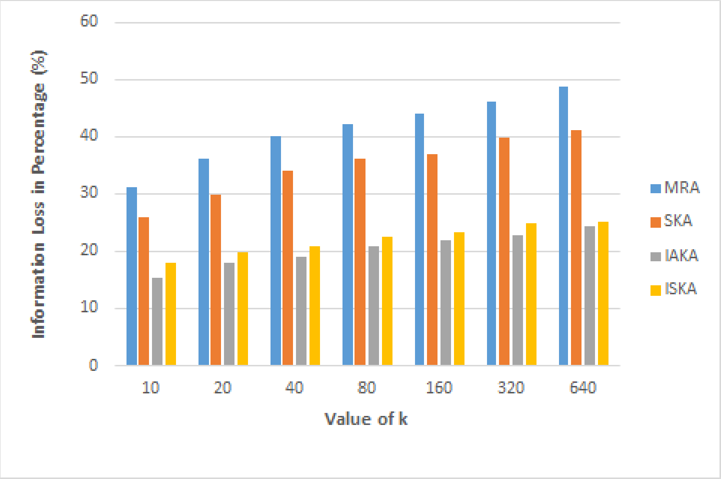

Figure 11. Information loss of MRA, SKA, IAKA, and ISKA on MPSEC 10M dataset

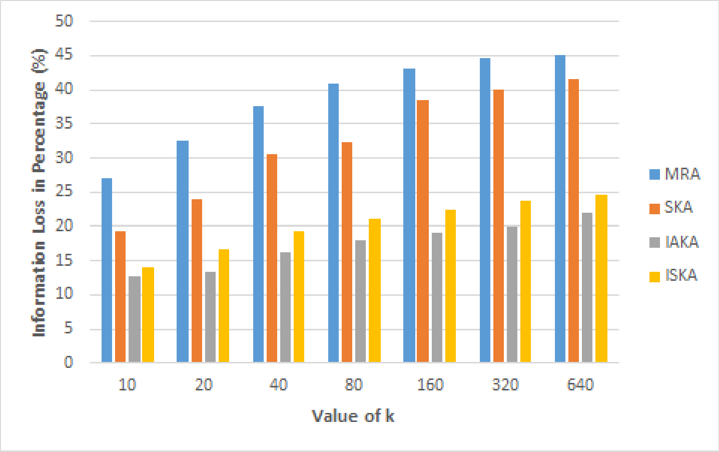

Figure 12. Information loss of MRA, SKA, IAKA, and ISKA on MPSEC 100M dataset

4.3.2 Comparison of IAKA and ISKA Algorithms with Existing Stream Data Anonymization Algorithms FADS and FAST on Synthetically Generated Stream Data

The average information loss on 1M synthetic MPSEC stream data is depicted in Figure 13. As can be seen in the figure, IAKA and ISKA perform better than FADS and FAST. FAST outperformed FADS [31, 32]. Here for stream data, we are comparing only till 1M dataset as no other previous papers have considered dataset over that size. Here also, IAKA performs better than ISKA and the reasons are the same as mentioned in Section 4.3.1.

4.4 Comparison of Proposed Algorithms Using Different Datasets

Proposed IKA and ILD algorithms have also been applied to adult dataset [37] and poker dataset [38]. Tables 15 and 16 show the comparison of proposed IAKA and ISKA algorithms used with different datasets (adult dataset, poker dataset) of 10M size. The proposed IAKA and ISKA and the existing methods, MRA and SKA, are implemented in the same experiment environment using different dataset, i.e. adult dataset and poker dataset. In this experiment, anonymity degree k is set to 80, i.e. most widely used value, and the diversity level l is set to 6. Table 15 represents the running time comparison of MRA, SKA, IAKA, and ISKA on different datasets in which the performance of ISKA is best with all three datasets. Both the proposed algorithms of IKA have an optimum time complexity, i.e. only $O(n)$, and the reason is already discussed in Section 4.2. The major drawbacks of MRA are multiple iterations and file management, and as the number of iterations increases performance decreases. In addition, the time com-

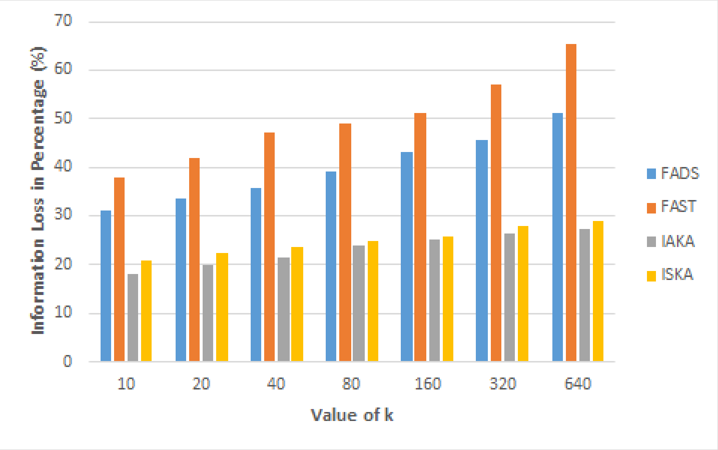

Figure 13. Information loss of FADS, FAST, IAKA, and ISKA on MPSEC 1M dataset

plexity of the MRA is very high, i.e. $O(n^2)$. The time complexity of the SKA algorithm is $O(n \log n)$. Lack of diversity and high information loss are the major drawbacks of SKA. Table 16 represents information loss comparison of MRA, SKA, IAKA, and ISKA on different datasets, in which performance of IAKA is the best with all three datasets because IAKA concentrates more on the nearness of the data in the cluster and it does not impose any condition on the size of the cluster that results in less information loss. ISKA also shows the significant reduction in information loss because both algorithms of IKA model use MDTDG technique for categorical attributes and also utilize the large crowd effects. So both algorithms of IKA are able to achieve low information loss and privacy-utility tradeoff.

Name of Dataset	Running Time in Seconds			
	MRA	SKA	IAKA	ISKA
Adult Dataset [38]	7 135	3 415	1 975.2	1 916.3
Poker Dataset [37]	6 830	3 329	1 956.2	1 899.1
MPSEC Dataset	6 600	3 280	1 857.6	1 780.3

Table 15. Running time comparison of MRA, SKA, IAKA, and ISKA on different 10M size datasets in seconds where value of $k = 80$

5 CONCLUSION

This paper addresses the issue of high information loss and high running time of anonymization algorithms of big data. This paper proposed the IKA and ILD model and applied them to the MPSEC dataset and successfully achieved high k-value

Name of Dataset	Information Loss in Percentage			
	MRA	SKA	IAKA	ISKA
Adult Dataset [38]	47.31	40.16	21.95	23.89
Poker Dataset [37]	45.02	38.9	21.89	23.31
MPSEC Dataset	42.02	36	20.87	22.57

Table 16. Information loss comparison of MRA, SKA, IAKA, and ISKA on different 10M size datasets in percentage where value of $k = 80$

(k value 1 523 in asymmetric, k value 1 283 in symmetric) with the maintained diversity in the sensitive attribute. As shown in the experimental result, 100 percentage completeness has been achieved, that is the data was fully anonymized, and the time complexity is $O(n)$. ISKA proves to be more efficient since the running time is less when compared to IAKA and other existing algorithms FADS, FAST, MRA and SKA. The running time has improved because the proposed IKA and ILD algorithms takes fewer iterations and execute on the multi-node environment of big data. ISKA and IAKA algorithms have reduced remarkable information loss in comparison with the existing methods FADS, FAST in case of stream data. They are better than MRA and SKA in case of batch data and IAKA performs best regarding the information loss. The proposed IKA and ILD models maintain the privacy-utility trade-off. The improvement in this model is that rather than anonymizing batch data and streaming data differently with a bunch of algorithms, it is better to achieve the combined functionalities in the single algorithm. The proposed IKA and ILD algorithms can also be applied to any datasets which require privacy mechanism. These proposed algorithms are useful for healthcare, sensor networks, online flight reservation systems, marketing and other commercial companies to grow their business. As their database contains personal information, it is vulnerable to provide direct access to researchers and analysts. Since in this case the privacy of individuals is leaked, it can pose a threat and it is also illegal. The future work for this approach is directed towards creating a new model which deals with privacy issues of correlative data for big data publication purposes.

Acknowledgement

We are grateful to the Madhya Pradesh State Election Commission, India, for their enthusiastic and constant support and for providing us with the real-time big dataset needed for the research work. We also acknowledge the concern of the Madhya Pradesh Council of Science and Technology, Bhopal, India and providing us funds to carry out this research work.

REFERENCES

[1] Al-Zobbi, M.—Shahrestani, S.—Ruan, C.: Sensitivity-Based Anonymization of Big Data. 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops), Dubai, 2016, pp. 58–64, doi: 10.1109/LCN.2016.029.

[2] Marx, V.: The Big Challenges of Big Data. Nature, Vol. 498, 2013, pp. 255–256, doi: 10.1038/498255a.

[3] Sánchez, D.—Martínez, S.—Domingo-Ferrer, J.: Comment on "Unique in the Shopping Mall: On the Reidentifiability of Credit Card Metadata". Science, Vol. 351, 2016, No. 6279, pp. 1274–1276, doi: 10.1126/science.aad9295.

[4] Yu, S.—Guo, S. (Eds.): Big Data Concepts, Theories, and Applications. Springer, Cham, 2016, doi: 10.1007/978-3-319-27763-9.

[5] Xu, L.—Jiang, C.—Wang, J.—Yuan, J.—Ren, Y.: Information Security in Big Data: Privacy and Data Mining. IEEE Access, Vol. 2, 2014, pp. 1149–1176, doi: 10.1109/ACCESS.2014.2362522.

[6] Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.): Automata, Languages, and Programming (ICALP 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4052, 2006, pp. 1–12, doi: 10.1007/11787006_1.

[7] Dwork, C.—McSherry, F.—Nissim, K.—Smith, A. D.: Calibrating Noise to Sensitivity in Private Data Analysis. In: Halevi, S., Rabin, T. (Eds.): Theory of Cryptography (TCC 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3876, 2006, 2006, pp. 265–284, doi: 10.1007/11681878_14.

[8] Geng, Q.—Viswanath, P.: Optimal Noise Adding Mechanisms for Approximate Differential Privacy. IEEE Transactions Information Theory, Vol. 62, 2016, No. 2, pp. 952–969, doi: 10.1109/TIT.2015.2504972.

[9] Qu, Y.—Yu, S.—Gao, L.—Niu, J.: Big Dataset Privacy Preserving Through Sensitive Attribute-Based Grouping. 2017 IEEE International Conference on Communications (ICC), Paris, France, 2017, pp. 1–6, doi: 10.1109/ICC.2017.7997113.

[10] Tripathy, B. K.—Mitra, A.: An Algorithm to Achieve k-Anonymity and l-Diversity Anonymisation in Social Networks. 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), Sao Carlos, Brazil, 2012, pp. 126–131, doi: 10.1109/CASoN.2012.6412390.

[11] Sweeney, L.: Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems, Vol. 10, 2002, pp. 571–588, doi: 10.1142/S021848850200165X.

[12] Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 10, 2002, No. 5, pp. 557–570, doi: 10.1142/S0218488502001648.

[13] Machanavajjhala, A.—Kifer, D.—Gehrke, J.—Venkitasubramaniam, M.: L-Diversity: Privacy Beyond k-Anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 1, March 2007, No. 1, pp. 1–52, doi: 10.1145/1217299.1217302.

[14] Samet, H.: Foundations of Multidimensional and Metric Data Structures. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2005.

[15] Fung, B. C. M.—Wang, K.—Yu, P. S.: Top-Down Specialization for Information and Privacy Preservation. 21st International Conference on Data Engineering (ICDE '05), Tokyo, Japan, 2005, pp. 205–216, doi: 10.1109/ICDE.2005.143.

[16] Fung, B. C. M.—Wang, K.—Chen, R.—Yu, P. S.: Privacy-Preserving Data Publishing: A Survey of Recent Developments. ACM Computing Surveys, Vol. 42, 2010, No. 4, Art. No. 14, pp. 1–53, doi: 10.1145/1749603.1749605.

[17] Liu, X.—Xie, Q.—Wang, L.: A Personalized Extended (a,k)-Anonymity Model. 2015 Third International Conference on Advanced Cloud and Big Data (CBD), Yangzhou, China, 2015, pp. 234–240, doi: 10.1109/CBD.2015.45.

[18] Guo, L.—Guo, S.—Wu, X.: Privacy Preserving Market Basket Data Analysis. In: Kok, J. N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (Eds.): Knowledge Discovery in Databases: PKDD 2007. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4702, 2007, pp. 103–114, doi: 10.1007/978-3-540-74976-9_13.

[19] Shen, Y.—Guo, G.—Wu, D.—Fan, Y.: A Novel Algorithm of Personalized-Granular k-Anonymity. Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), 2013, Shenyang, China, pp. 1860–1866, doi: 10.1109/MEC.2013.6885357.

[20] Meyerson, A.—Williams, R.: On the Complexity of Optimal k-Anonymity. Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '04), New York, USA, 2004, pp. 223–228, doi: 10.1145/1055558.1055591.

[21] Han, J.—Yu, H.—Yu, J.—Cen, T.: A Complete (α,k)-Anonymity Model for Sensitive Values Individuation Preservation. 2008 International Symposium on Electronic Commerce and Security, Guangzhou City, China, 2008, pp. 318–323, doi: 10.1109/ISECS.2008.92.

[22] Sei, Y.—Okumura, H.—Takenouchi, T.—Ohsuga, A.: Anonymization of Sensitive Quasi-Identifiers for l-Diversity and t-Closeness. IEEE Transactions on Dependable and Secure Computing, Vol. 16, 2019, No. 4, pp. 580–593, doi: 10.1109/TDSC.2017.2698472.

[23] Xiao, X.—Tao, Y.: Anatomy: Simple and Effective Privacy Preservation. Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06), Seoul, Korea, September 2006, pp. 139–150.

[24] Jin, X.—Zhang, M.—Zhang, N.—Das, G.: Versatile Publishing for Privacy Preservation. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10), 2010, pp. 353–362, doi: 10.1145/1835804.1835851.

[25] Tran, Q.—Sato, H.: A Solution for Privacy Protection in MapReduce. 2012 IEEE 36th Annual Computer Software and Applications Conference, Izmir, Turkey, 2012, pp. 515–520, doi: 10.1109/COMPSAC.2012.70.

[26] Jain, P.—Gyanchandani, M.—Khare, N.: Improved k-Anonymity Privacy-Preserving Algorithm Using Madhya Pradesh State Election Commission Big Data. In: Krishna, A., Srikantaiah, K., Naveena, C. (Eds.): Integrated Intelligent Computing, Communication, and Security. Springer, Singapore, Studies in Computational Intelligence, Vol. 771, 2019, pp. 1–10, doi: 10.1007/978-981-10-8797-4_1.

[27] Koli, A.—Shinde, S.: Parallel Decision Tree with Map Reduce Model for Big Data Analytics. International Conference on Trends in Electronics and Informatics (ICEI 2017), Tirunelveli, India, 2017, pp. 735–739, doi: 10.1109/ICOEI.2017.8300800.

[28] Mohamed, N.—Fung, B. C. M.—Hung, P. C. K.—Lee, C. K.: Centralized and Distributed Anonymization for High-Dimensional Healthcare Data. ACM Transactions on Knowledge Discovery from Data, Vol. 4, 2010, No. 4, Art. No. 18, doi: 10.1145/1857947.1857950.

[29] Jain, P.—Gyanchandani, M.—Khare, N.: Data Privacy for Big Data Publishing Using Newly Enhanced PASS Data Mining Mechanism. Book Chapter. In: Thomas, C. (Ed.): Data Mining. Intech Open Publisher, 2018, doi: 10.5772/intechopen.77033.

[30] Zakerzadeh, H.—Osborn, S. L.: FAANST: Fast Anonymizing Algorithm for Numerical Streaming DaTa. In: Garcia-Alfaro, J., Navarro-Arribas, G., Cavalli, A., Leneutre, J. (Eds.): Data Privacy Management and Autonomous Spontaneous Security (DPM 2010, SETOP 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6514, 2011, pp. 36–50, doi: 10.1007/978-3-642-19348-4_4.

[31] Guo, K.—Zhang, Q.: Fast Clustering-Based Anonymization Approaches with Time Constraints for Data Streams. Knowledge-Based Systems, Vol. 46, 2013, pp. 95–108, doi: 10.1016/j.knosys.2013.03.007.

[32] Mohammadian, E.—Noferesti, M.—Jalili, R.: FAST: Fast Anonymization of Big Data Streams. Proceedings of the 2014 International Conference on Big Data Science and Computing (BigDataScience '14), 2014, Art. No. 23, pp. 1–8, doi: 10.1145/2640087.2644149.

[33] Yadav, G. S.—Ojha, A.: A Fast and Efficient Data Hiding Scheme in Binary Images. 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Piraeus, Greece, 2012, pp. 79–84, doi: 10.1109/IIH-MSP.2012.25.

[34] LeFevre, K.—DeWitt, D. J.—Ramakrishnan, R.: Mondrian Multidimensional k-Anonymity. 22nd International Conference on Data Engineering (ICDE '06), Atlanta, GA, USA, IEEE, 2006, pp. 1–11, doi: 10.1109/ICDE.2006.101.

[35] Zakerzadeh, H.—Aggarwal, C. C.—Barker, K.: Privacy-Preserving Big Data Publishing. Proceedings of 27th International Conference on Scientific and Statistical Database Management (SSDBM '15), ACM, 2015, Art. No. 26, 11 pp., doi: 10.1145/2791347.2791380.

[36] Mehta, B. B.—Rao, U. P.: Privacy Preserving Big Data Publishing: A Scalable k-Anonymization Approach Using MapReduce. IET Software, Vol. 11, 2017, No. 5, pp. 271–276, doi: 10.1049/iet-sen.2016.0264.

[37] CATTRAL, R.—OPPACHER, F.: Poker Hand Data Set. UCI Repository of Machine Learning Databases, University of California, School of Information and Computer Science, Irvine, CA, 2007.

[38] KOHAVI, R.—BECKER, B.: Adult Data Set. UCI Repository of Machine Learning Databases, University of California, School of Information and Computer Science, Irvine, CA, 1996.

Priyank JAIN is working as Ph.D. Research Scholar. He has more than eight years of experience as Assistant Professor and in the research field. He has experience from Indian Institute of Management, Ahmedabad, India (IIM A) in the research field. His Ph.D. is in the big data privacy area. His educational qualification is M.Tech. and B. E. in information technology. His areas of specialization are big data, big data privacy and security, data mining, privacy-preserving, and information retrieval. He has publications in various international conferences, international journals and national conferences. He is a member of HIMSS.

Manasi GYANCHANDANI is working as Assistant Professor in MANIT Bhopal. She has more than 20 years of experience. She obtained her Ph.D. in computer science and engineering. Her area of specialization is in big data, big data privacy and security, data mining, privacy-preserving, artificial intelligence, expert system, neural networks, intrusion detection and information retrieval. She has publications in 8 international conferences, 15 international journals and 8 national conferences. She has her Life Time Membership of ISTE.

Nilay KHARE is working as Professor in MANIT Bhopal. He has more than 21 years of experience. He obtained his Ph.D. in computer science and engineering. His areas of specialization are big data, big data privacy and security, wireless networks, theoretical computer science. His publications are in 54 international and national conferences, and international journals. He has his Life Time Membership of ISTE.