SRTTU

Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

Presenting a Model of Data Anonymization in Big Data in the Context of In-Memory Processing Framework

E. Shamsinejad [1], *T. Banirostam* [1,*], *M. M. Pedram* [2], *A. M. Rahmani* [3]

[1]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

[2]*Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran.*

[3]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*

Article Info

Article History:
Received 19 April 2023
Reviewed 18 June 2023
Revised 05 July 2023
Accepted 15 August 2023

Keywords:
Big Data
Anonymity
Confidentiality
Data disclosure
Privacy

*Corresponding Author's Email Address:
h.banirostam.eng@iauctb.ac.ir

Abstract

Background and Objectives: Nowadays, with the rapid growth of social networks extracting valuable information from voluminous sources of social networks, alongside privacy protection and preventing the disclosure of unique data, is among the most challenging objects. In this paper, a model for maintaining privacy in big data is presented.

Methods: The proposed model is implemented with Spark in-memory tool in big data in four steps. The first step is to enter the raw data from HDFS to RDDs. The second step is to determine m clusters and cluster heads. The third step is to parallelly put the produced tuples in separate RDDs. the fourth step is to release the anonymized clusters. The suggested model is based on a K-means clustering algorithm and is located in the Spark framework. also, the proposed model uses the capacities of RDD and Mlib components. Determining the optimized cluster heads in each tuple's content, considering data type, and using the formula of the suggested solution, leads to the release of data in the optimized cluster with the lowest rate of data loss and identity disclosure.

Results: Using Spark framework Factors and Optimized Clusters in the K-means Algorithm in the proposed model, the algorithm implementation time in different megabyte intervals relies on multiple expiration time and purposeful elimination of clusters, data loss rates based on two-level clustering. According to the results of the simulations, while the volume of data increases, the rate of data loss decreases compared to FADS and FAST clustering algorithms, which is due to the increase of records in the proposed model. with the formula presented in the proposed model, how to determine the multiple selected attributes is reduced. According to the presented results and 2-anonomity, the value of the cost factor at k=9 will be at its lowest value of 0.20.

Conclusion: The proposed model provides the right balance for high-speed process execution, minimizing data loss and minimal data disclosure. Also, the mentioned model presents a parallel algorithm for increasing the efficiency in anonymizing data streams and, simultaneously, decreasing the information loss rate.

Introduction

Because sensitive data are distributed among different computational resources, in big data, unauthorized access to centralized data structures will be easily provided. The expansion of the distributed computing infrastructure, as well as the extent of mobile devices, has

Doi: 10.22061/jecei.2023.9737.651

raised concerns about the processing and sharing of personal and sensitive data of users [1]-[4]. In this framework, various mechanisms such as encryption, access control, audit and similar cases have been considered for maintaining data confidentiality [5]-[8].

The data stream is one of the most important big data types, which exploring them reveals hidden patterns and provides valuable information to different sciences [9]-[12]. Along with these benefits, because of the aggregation of data from various sources and exploring these data, the issue of privacy of individuals and maintaining corporate secrets are particularly regarded important. To solve this issue, various research has been done [13]. Because of their weaknesses, makes their use in big data streams impossible or suboptimal. For preventing disclosure of personal data, unique personal identifiers such as identification numbers, insurance numbers, and other distinguishing attributes are deleted before release [14]-[17]. However, after deleting the identifiers, in some cases, attackers reach personal data through public databases [18]. In order to solve this problem, a lot of research has been done to maintain the anonymity of individuals with the least changes in the dataset. In this context, methods such as k-anonymity have been proposed [19]-[21].

K-anonymity represents the anonymity by putting the tuples in K clusters. In some applications, a huge volume of data is delivered by the system in the form of a data stream that needs real time anonymization. So, anonymization of such data types through the existing algorithms is among the difficult problems. As the result, representing methods for anonymization of big data streams is inevitable [27]-[29].

In the following, the literature related to privacy protection methods in data anonymization, types of attacks and advantages and disadvantages of anonymization techniques, challenges of big data anonymization algorithms, data anonymization as well as parameters, features, methods, and algorithms of anonymization will be. Ten related works that commonly use K-anonymity for data anonymization and privacy protection for big data dissemination will be reviewed and the parameters used in the related works will be compared. And finally, the proposed model and simulations will be described and conclusions and future works will be presented.

Subject Literature

Through the data collection phase, the data publisher, who is responsible for the online anonymization of data before public release or mining, receives data streams from various sources [30], [31]. Typically, in data publishing methods, the privacy of any tuple t is considered as (1): [33].

t (Explicit Identifier, Quasi Identifier (QI), Sensitive Attributes, Non-Sensitive Attributes) (1)

The data may be streamed into the system in the form of data streams or the tables which have been stored earlier, and anonymization will be done on this kind of data. In fact, anonymity is an approach that tries to hide the identity of individuals or values of sensitive attributes from others [34].

The attacker's sum of information from public databases, her/his background knowledge, and the new anonymized database release should be less than the new database release [35]. However, it is clear that background knowledge leads to data disclosure to some extent.

Types of Privacy Methods in Data Anonymization

Privacy is one of the most important issues users confront when releasing data, especially when the data are private; such as, user's identity, location, disease background, etc.

As a result, user privacy researchers have represented various methods to protect privacy; among which, K-anonymity has been extensive.

This is true to the extent that the mentioned method is used in all environments, like centralized or distributed ones, and in services, such as centralized and distributed data mining or location-based services.

Privacy protecting methods are divided into four groups Fig. 1. Through these privacy protecting methods, all the identifiers should be deleted before release, and pseudo-identifiers, sensitive data, and non-sensitive data should be released after various anonymization operations.

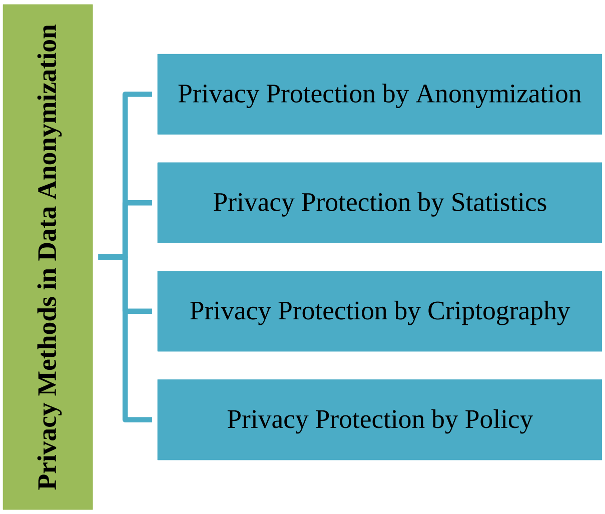

Privacy Methods in Data Anonymization
Privacy Protection by Anonymization
Privacy Protection by Statistics
Privacy Protection by Criptography
Privacy Protection by Policy

Fig. 1: Types of privacy protecting methods [59], [61], [62].

Types of Attacks

The types of attacks are divided into three categories according to Fig. 2.

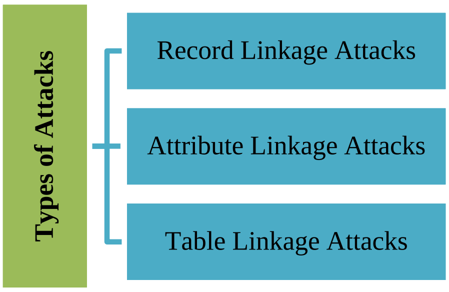


Fig. 2: Types of attacks [59].

Record Linkage Attacks

In a linkage attack record, a small number of records are distinguished based on quasi-identifier values. These numbers of records make up a group. If the quasi identifier related to the victim is mapped to this group, the attacker can identify his victim with a high probability according to his background knowledge. To deal with these types of attacks, k-anonymity was the first model offered [22]. Other models presented to contrast the record linkage attack are (x, y)-anonymity and multi-relational k-anonymity [65], [66]. These models contrast a linkage attack record by hiding the victim's report in a group with the same QI; However, if most of the reports placed in a group with the same QI have the same value for the sensitive attributes, but without accurately identifying the victim's report, the sensitive amount attributes (e.g. the type of disease) can be got. This mode is placed in the category of linkage attributes attacks [24].

Attribute Linkage Attacks

In attribute linkage attacks, the attacker may not be able to accurately determine the victim's tuple, but by mapping the victim to a group of tuple-QI with the same QI and the same amount in sensitive attributes, it can get the sensitive amount attributes of the victim with a high probability. The main idea for solving this problem is to eliminate the relationship between quasi-ID and sensitive adjective values. To solve this problem, the L-Diversity method is provided in [25]. In this method, in each group of QIs, the values of sensitive attributes should get at least l different values. In this model, if the value l is considered being k=l, k-anonymity is also guaranteed.

Other models presented to contrast attributes linkage attack are (x, y)-Privacy and (a, k)-anonymity [26] models that largely act like previous methods. If sensitive attributes are not properly distributed in the data set, the introduced models cannot contrast with the attribute linkage attack.

Suppose, in a data set, 95% of people have colds and 5% have AIDS. Now, if they have 50% AIDS and 50% cold in a QI group, a Diversity-2 condition is established. Here, the attacker can be informed of a particular person's AIDS with a 50% confidence. In the initial case, an attacker can guess a particular person's illness with a 5% confidence. T-Closeness method was presented to solve this problem in [26]. In this method, in each group of QIs, data must have a distribution close to the original data.

Table Linkage Attacks

In the attacks of the record linkage and the link of attributes, it is assumed that the attacker is aware of the existence of the victim in the published table. While sometimes the existence or absence of a person in the table can disclose sensitive information. For example, when a hospital publishes a table for AIDS patients, the knowledge of the existence or absence of a person on the table can be equal to exposing a sensitive attribute. In order to contrast this attack, a δ-present method was presented [25]. In this method, the probability of a person's presence in the published table must be limited between the two $\delta = (\delta_{min}, \delta_{max})$. This model implicitly deals with record linkage attacks and linkage attributes. In the attachment, the aggregate table compares the influential parameters of methods and algorithms. It may not imagine an end for this conflict.

Data Anonymization Techniques

In this section, a number of data anonymization techniques will be divided as shown in Fig. 3, also Table 1 will present the advantages and disadvantages of each of the introduced techniques.

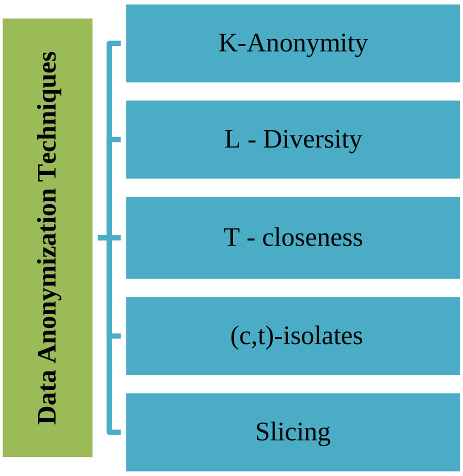


Fig. 3: Data anonymization techniques [35]-[38].

Table 1: Advantages and disadvantages of different data anonymization techniques

Anonymity Techniques	Advantage	Disadvantage
K-anonymity [23], [25]	- Simplicity in implementation - High scalability - High speed - Lower risk percentage for re-identification if K is large.	- Inefficiency against the previous knowledge of the intruder - No work against communication between data - High processing time - Inefficiency if the data is available as a query - Inefficiency in high data diversity
L–diversity [35], [36]	- Shrink and summarize data. Sensitive identifiers with equal numbers in the set. The information is repeated. - Scalability	- To be dependent on the range of changes of sensitive indicators (L-variability requires L.L is a different value for indicators.) - To be vulnerable to hacker background knowledge - In data grouping, their semantic relationship is not considered.
T-closeness [35], [36]	- Prevents skewness (sensitive diagnoses using large differences in group distribution and overall distribution).	- Computational complexity - Loss of relationship between different identifiers - low speed - Lack of scalability - Inefficiency in data diversity
(c,t)-isolation [37]	- Prevent hackers from isolating records - Scalability of performance in high data diversity	- High computational volume - Not to consider the semantic relationship between attributes - low speed
Slicing [38]	- To be suitable for high volume data - High data productivity because nothing is removed from the data.	- Due to the permutation process, it may disappear relationship between attributes. - Not to use data utility

Challenges of Anonymity in Big Data

The three main characteristics (volume, diversity and speed) provide many challenges once working with this type of data. Big data anonymization is not excluded from this rule, and in big data, in order to use any of the techniques of anonymity potential limitations and challenges must be considered. The challenges of anonymity in big data are divided into seven parts in Fig. 4.

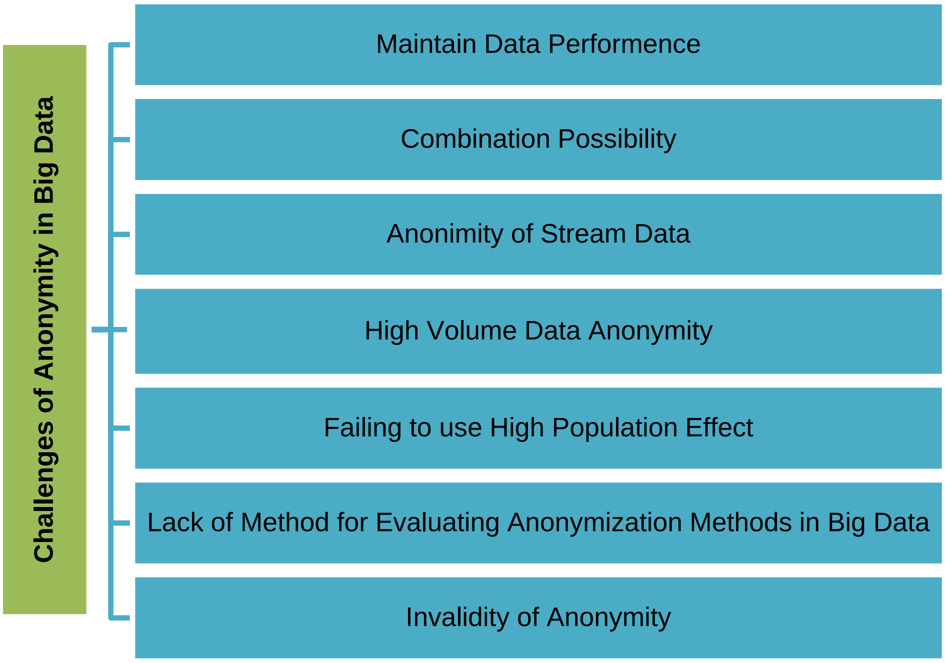

Challenges of Anonymity in Big Data
Maintain Data Performence
Combination Possibility
Anonimity of Stream Data
High Volume Data Anonymity
Failing to use High Population Effect
Lack of Method for Evaluating Anonymization Methods in Big Data
Invalidity of Anonymity

Fig. 4: Challenges of anonymity in big data [38]-[40], [63].

HDFS

Hadoop; is an open source system for distributed storage and scalable data processing. Hadoop provides a distributed file system called as Hadoop Distributed File System (HDFS) and MapReduce programming paradigm. HDFS provides to keep several copies of data and stores these copies on several nodes of cluster. It is a reliable, efficient and cost-effective system for storing large amounts of data. MapReduce provides a model for processing large amounts of data for distributed and parallel programming. The MapReduce operation basically consists of the Map and Reduce functions [18].

RDD

A new abstraction called resilient distributed datasets (RDDs) that enables efficient data reuse in a broad range of applications. RDDs are fault-tolerant, parallel data structures that let users explicitly persist intermediate results in memory, control their partitioning to optimize data placement, and manipulate them using a rich set of operators [17].

SPARK

Hadoop and Spark are two fundamental big data technologies. Hadoop provides processing data on disk while Spark process data on memory. Spark runs 100 times faster than Hadoop. This difference plays an important role for some projects requiring short response time. Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching and optimized query execution for fast queries against data of any size. Simply put, Spark is a fast and general engine for large-scale data processing. The fast part means that it’s faster than previous approaches to work with Big Data like classical MapReduce. The secret for being faster is that Spark runs on memory (RAM), and that makes the processing much faster than on disk drives. Spark stores the data in the RAM of servers which allows quick access and in turn accelerates the speed of analytics [17], [18].

Anonymity Operators

Basically, datasets do not meet privacy requirements without making changes before publication. For privacy, a sequence of anonymity operators such as generalization, suppression, permutation, anatomization and perturbation are required to apply to the dataset. In the Fig. 5 shows each of the anonymous.

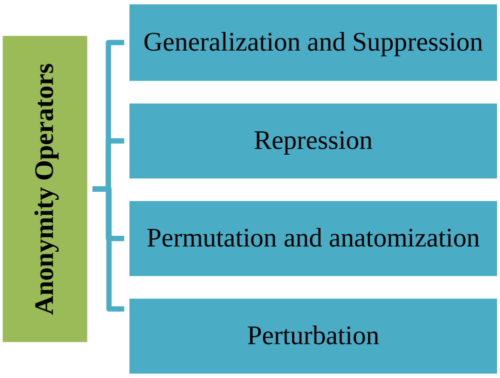


Fig. 5: Anonymity Operators [31], [33].

Anonymization Methods

Depending on the types of attack models and anonymity operators, there are various methods in data anonymization, which will be introduced in Fig. 6 and described below.

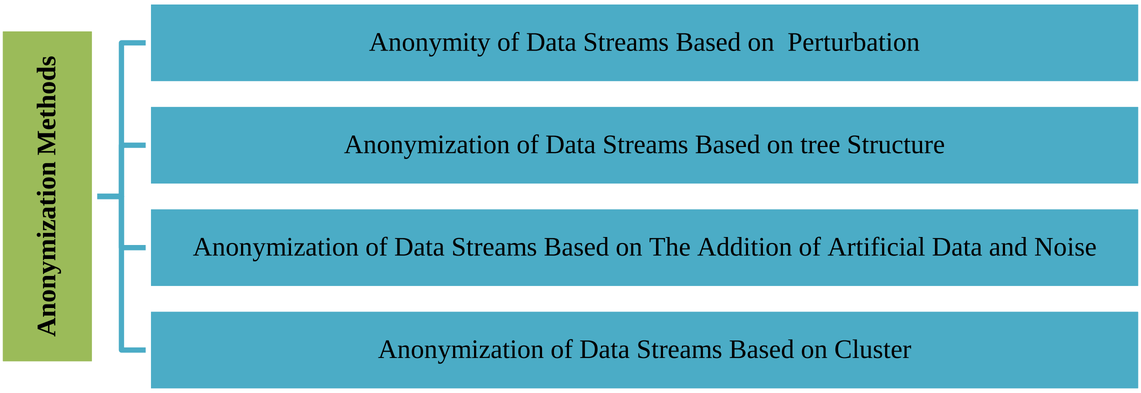

Anonymization Methods
Anonymity of Data Streams Based on Perturbation
Anonymization of Data Streams Based on tree Structure
Anonymization of Data Streams Based on The Addition of Artificial Data and Noise
Anonymization of Data Streams Based on Cluster

Fig. 6: Anonymization Methods [41]-[47], [60], [64].

Anonymity of Data Streams Based on Perturbation

In these algorithms, data is extracted and combined with a random noise from a statistical distribution.

The two main categories of this approach are examined below [48].

Additive Perturbation

In Additive Perturbation method, a private data set is considered as (2).

$$D = d1,\ \mathrm{d2},\ \ldots,\mathrm{dn} \tag{2}$$

For each $di \in D$, random noise ri which selected from known statistical distributions such as uniform distribution and Gaussian distribution is added to the data. At last, D' dataset would be available for data miners in the form of (3) [49]-[50].

$$D' = d_1 + r_1,\ \mathrm{d}_2 + r_2, \ldots,\ d_n + r_n \tag{3}$$

Data miners use $di + ri$ as a Maximal Expectation Algorithm to obtain the value di. This method of randomization is used for many data mining applications such as classification and Association Rule Mining.

Multiplicative Perturbation

One of the alternative methods proposed the Additive Perturbation method is the multiplicative Perturbation method [51]-[53]. Two common strategies in this method are derived from statistics.

In the first method, all components of D(d_i) are multiplied by a random number derived from a Gaussian distribution (usually considered one) and variance σ^2.

In the second method, the D dataset is first converted with a natural logarithm function, so that the converted components are $z_i = \ln(d_i)$. Then, a random new r_i is then added to each of the converted components, which is extracted from a multivariate equation of zero and µ Gaussian. This Gaussian distribution is considered with mean $\sigma^2 = c\Sigma z$. In this relation, $0 < c < 1$ and Σz are equal to the covariance of the converted components, that means z_i. The data that are published for data miners can be in the form of (4).

$$D' = \exp(z1+r1),\ \exp(z2+r2),\ \exp\ (zn+rn) \tag{4}$$

Anonymization of Data Streams Based on Tree Structure

Another category of anonymity algorithm for data streams is algorithms based on tree structures. In this context, algorithms such as SKY, SWAF and KIDS [32] have almost similar structure.

Anonymization of Data Streams Based on The Addition of Artificial Data and Noise

Despite introducing methods, in this method in order to anonymize the data stream, quasi-identifier attributes remain unchanged. In this method, data privacy is maintained by adding artificially generated data to the original data [44], [67].

Framework of Zero-Delay Anonymization Method

The main goal of this method is the real time construction of an L-variety data stream out of the main data stream. This method guarantees that the probability of guessing sensitive attributes related to a given person, in the data stream, is less than 1/1 [45].

Anonymization of Data Streams Based on Fuzzy Method

A method for protecting the privacy of the data stream is represented based on fuzzy logic [46]. In this method, the values of sensitive attributes in the data stream are converted into fuzzy values and added, in the form of a column, to the structure of records related to the same data stream.

Anonymization of Data Streams Based on Cluster

At the approach of cluster-based data anonymization, each cluster is placed in a cluster in a way that each cluster has at least tuples k. Then these tuples are published by using cluster generalization. According to Fig. 7, cluster-based anonymity algorithms will be introduced.

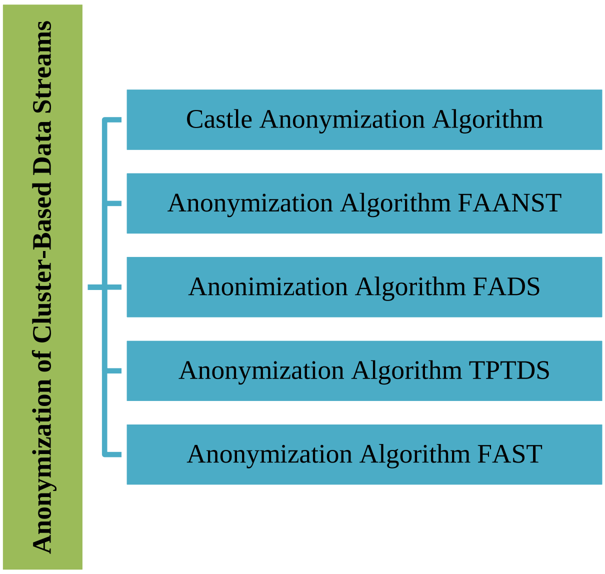

Anonymization of Cluster-Based Data Streams
Castle Anonymization Algorithm
Anonymization Algorithm FAANST
Anonimization Algorithm FADS
Anonymization Algorithm TPTDS
Anonymization Algorithm FAST

Fig. 7: Cluster-based anonymity algorithms [43]-[45].

In the following, in Table 2, the various parameters of anonymization methods and anonymization algorithms with characteristics such as data loss rate, time order, the types of data which have been subjected to anonymization approach are all examined. Furthermore, additional data production, the usability and desirability of big data technology, and the rate of response delay algorithms to access the desired data have been investigated.

Table 2: Different parameters of anonymization methods and anonymization algorithms

Algorithm	Attributes	Data loss rate	Temporal complexity	Data Type	Additional Data production	Appropriate For Big data	Delay rate
Perturbation-based	**Additive Perturbation**	Very high	O(S)	Numerical	Yes	Almost	Low
	Multiplicative Perturbation	Very high	O(S)	Numerical	Yes	Almost	Low
Tree structure-based	**SWAF**	Very high	O(S^2 logS)	Numerical and deductive	No	Inappropriate	High
Artificial data-based and noise	**DF & Fuzzy**	Very high	O|S^2|	Numerical and deductive	Yes	Inappropriate	Almost low
Social Network Graphs –based [28]	**k-anonymity [28]**	Low	O((V+E)logk)	Numerical and deductive	No	Almost	High
	k-candidate [28]	Low	O((V+E)logk)	Numerical and deductive	No	Almost	High
	k-degree [28]	Low	O((V+E)logk)	Numerical and deductive	No	Almost	High
Cluster-based	**CASTEL**	Medium	O|S^2|	Numerical	No	Inappropriate	High
	FAANEST	Medium	O|S^2|	Numerical and deductive	No	Inappropriate	High
	FADS	Medium	O(S)	Numerical and deductive	No	Inappropriate	High
	TPTDS	Low	O(S)	Numerical and deductive	No	Appropriate	Not important
	FAST	Low	O(S)	Numerical and deductive	No	Appropriate	Low

Related Works

In the following, 10 related works that have been recently reviewed, will be introduced. In these works, K-anonymity method is generally used to anonymize data and maintain data confidentiality for publication. Most of these researchers have implemented their model and architecture, implementation in big data set on Adult dataset.

This dataset contains 48842 samples, 14 attributes, numerical and non-numerical data types and also 6465 missing values [50], [58].

Table 3 indicates the various parameters of the algorithms presented in the related tasks, such as data loss rate, strengths, weaknesses, time order, data types that have been anonymized and etc., are examined.

N. Victor et al. in [7].

The release of the result of a request is accompanied by noise, so the attacker might not be able of capturing information with 100% assurance. This model can protect privacy even when the attacker has background information. Even if the person does not have the correct information to publish, it has no effect on the anonymization algorithm, which is compatible with interactive and non-interactive requests.

M.Kiabod et al. in [51].

They developed an algorithm that deals with attacks in which a user's privacy is compromised by having some knowledge of a person's neighbor and friends. This algorithm tries to anonymize the data by using two techniques of generalizing data and adding additional edges to the input graph, as well as using some meta-heuristic methods. In short, this algorithm by scrolling all the input graph heads tries to expose the neighbors for the components and then isomorphic in pairs from the perspective of neighboring groups. Two important defects in this algorithm are that, firstly, a specific generalization technique has been used, which is only applicable in specific environments and specimens and does not have relative generalization. Second, during the implementation of the algorithm to decide whether to use generalization or to add a new edge to the graph. The allocation of priority between these two actions has been used to estimate the cost of each of these two practices, which has ambiguity and seems to have a random mode.

W. Zheng et al. in [12].

Greedy algorithm is used for the anonymization of vertices' degrees, whose input and output are the graph G with n vertices and G' with anonymized k-degrees respectively. This algorithm is always capable of producing a graph of k-degrees of anonymity. The problem with this algorithm is its applicability to only one type of attack on privacy, i.e. type degree, which happens rarely compared to other types. The five methods above are investigated about the anonymization of social networks' graphs. The focus of the current article is not on this issue, but the related algorithms are discussed due to the thematic proximity.

J. Tekli et al. in [38].

Besides K-anonymity, it pays attention to the L-variety of the data. This model is based on anonymization using information clustering. Alongside receiving new data and comparing it to the existing clusters, the new data is embedded in one of the clusters if possible. In the case that the new data is not suitable for any of the clusters, the data is embedded in the most suitable cluster with the lowest rate of loss after applying the enlargement process to the clusters. The process of enlargement of a cluster is handled by extending the intervals of the respected cluster's variables. In this method, the time period, since receiving the data until releasing the data through a cluster, should not be more than a defined value σ. Regarding the threshold, it will be taken into account whether the data of unreleased clusters have reached their threshold. If such data is found, the respected cluster is released. This cluster can be immediately released when the number of data in it is greater than or equal to k; otherwise, a strategy is used to combine the cluster with the closest adjacent cluster to produce a cluster with a quantity greater than or equal to k. The writers, also, include the L-variety anonymization technique in their model so the security level increases. In this model, in a general sense, because of lack of buffering, the data are embedded in one of the existing clusters as soon as being received.

A. Otgonbayar et al. in [52].

At first, the proposed model focused on numerical data to represent a model for rapid anonymization of data streams, but it also supported non-numerical data then. Despite the represented model in CASTLE which processes and clusters the data immediately after receiving, a processing window is defined in the proposed method. Three main variables are K, MU, and DELTA which respectively refer to the anonymity variable in K-anonymity, the considered size of the processing window, and the defined threshold for information loss in each cluster. The first phase of clustering performs when the quantity of the received data in the processing window reaches MU. Some information may possibly remain in the window through this stage, not being embedded in any cluster. New data is receivable after some places in the processing window are emptied. After clustering the information, only the clusters containing data with a quantity greater than or equal to k and showing a loss rate lower than DELTA are accepted. Finally, since K-means algorithm is not usable for non-numerical attributes, clustering algorithms based on Medoid are proposed. One of the problems with this model is that no threshold is considered for preserving data in the processing window; this leads the data to remain longer in the window. This issue hinders the immediate processing of data which is among the principal necessities of data streaming**.**

J. Wang et al, in [32].

The proposed model used Encryption method to maintain confidentiality in big data. In this model, the data are first clustered using rule-based methods. Rule-based methods can be used for large volumes of data, so they can be effective for using big data. This model uses the public key asymmetric encryption method to control data access. In this model, three levels of security are considered.

- The first level, the main work of encrypting raw data is done. For this purpose, the RSA encryption method is used, which is an asymmetric encryption method. Then, the signature of the main database administrator is added to the encrypted database.
- Second level, after confirming the signature added to the database, each of the middle users ensures the accuracy of the information. They then access only part of the customer information needed for data mining and perform the desired operations. Finally, with the help of rule-based methods, information clustering operations are performed.
- The third level, which is known as the general layer, allows all users to access the extracted rules in the second level, but the original data is kept secret from users.

As mentioned, in this model, an attempt has been made to maintain the confidentiality of users' information by using encryption methods. Due to computational overhead and the need for real-time processing, encryption method is not proper for this volume of data.

B. B. Mehta et al, in [11].

Tries to provide a model for maintaining the confidentiality of big data. This model consists of three main components as follows: information anonymity component, update component, anonymous information management component.

Anonymity operations are performed on the anonymity component of information. In this regard, the generalized method is used for anonymity of information and thus, each data is mapped to an appropriate generalized level. The update component is designed with the input of new information as well as in order to map them to the appropriate levels. After entering the information, each data is mapped to the most appropriate level of anonymity. In the meantime, with the arrival of new information, it may be necessary to make updates at the anonymity level of the database. In this case, the entire database is mapped to a new level. The anonymous information management component is responsible for maintaining the anonymous information in order to avoid the cost of recalculating the anonymous information. It can be seen that in this method, all dimensions of big data are not considered. For example, considering the "diversity" dimension in big data, this method does not provide any mechanism for assigning an appropriate level of anonymity depending on the type of input data. In this method, it should be noted that if any updates are needed, this change will be applied to the entire database. In addition to the high computational cost, this action also increases execution time, which is considered as a barrier to real-time processing. Considering the dimensions of big data, it has been tried to maintain the confidentiality of information to minimize the amount of information loss. For this purpose, an attempt is made to place the relevant data in a subgroup by dividing the data into appropriate subgroups. Due to the arrival of new information, if anonymity needs to be updated, changes need to be applied only to a portion of the database. By considering the time limit in determining the appropriate level of anonymity, the ground for real-time processing is provided.

J. Andrew et al, in [57].

The suggested model is actually an architecture based on (K, L)-anonymity. The data input source includes personal identifications from health sector, details of personal identifications, and details of individual's bank account. Some data may be received for analysis so confidentiality protection is essential before release. In the suggested architecture, first, pre-processing is done to distinguish between textual data and numerical data and classify them. The suggested architecture is designed to deal with numerical data and classifying. In the next step, anonymization techniques are carried out for the generalization of the table, through which a heuristic algorithm is used. The output of the generalized table is used as input for another confidentiality model. Then, by adding Laplacian noise, confidentiality violation is more limited. Generalization algorithm and heuristic suppression are performed based on pseudo-identifiers. First, pseudo-identifiers and sensitive identifiers are chosen according to the coefficient of their effect on confidentiality. The following criteria are considered for model evaluation: Distortion, Prec, NCP, and RMSE. The diagram comparing Distortion and Prec shows that Prec increases while K does, but after k=50 the value of Prec will be fixed. NCP evaluation results show different values of Distortion and Prec for confidentiality. The results show that the proposed method resists any type of attack.

P. Jain et al, in [67].

They represented the improved algorithms of K-Anonymization and L-Diversity for confidentiality protection in big data. They believed that these two approaches do not show hopeful results in voluminous datasets. The main issue with the current anonymization algorithms is their high rate of data loss and the huge time they need to be executed. To overcome this issue, they suggest the new models of Improved K-Anonymization (IKA) and Improved L-Diversity (ILD). IKA, through a symmetric algorithm and an asymmetric anonymization algorithm, takes K. After data anonymization using IKA, ILD is used to increase privacy. ILD makes the data more various and, consequently, increases privacy. The implementation framework for the suggested model is Apache Storm. This paper also compares the suggested model to the current anonymization algorithms like FADS, FAST, MRA, SKA. The results of implementation show that, IKA and ILD have improved significantly considering the rate of data loss and execution time.

A. Raj et al, in [66].

They presented the data anonymization algorithm with K-anonymity technique using Map-Reduce processing on a cloud base. Analyzing the data with traditional systems might be exhausting while the data volume increases. However, Map-Reduce framework is efficient and synchronizable in huge volumes of data. They presented the generalization technique for anonymization through two phases of Map and Reduce using Top Down Specification mode. Their Map-Reduce approach consists of five stages: 1-Assigning a value by Map processor to the input key K1 and sending all the data related to the mentioned key to the processor, 2-Executing each user's Map only once for each K1 and producing the gathered key values, 3- Determining the value of K2 based on the produced Map through Reduce, 4- Executing Reduce only once for each K2 produced by Map and 5- Producing the final output of Map-Reduce from all gathered and ordered Reduce outputs. The results of its application show that big data anonymization in Map-Reduce framework is efficient.

Table 3: Comparison of previous parameters and algorithms

Attributes	Researchers	Data loss rate	Strengths	Weaknesses	Chronological order	Data type	Additional data generation	Suitable for big data	Delay rate
Based on artificial data and noise	N. Victor et al, (2016) [7]	Medium	Can protect privacy even when the attacker has background information	---	---	Numerical and deductive	No	Almost	Almost low
Based on social network graphs	M.Kiabod et al, (2019) [51]	Low	---	– Using a special generalization technique – Ability to run in special environments	O((V+E)log k)	Numerical and deductive	No		High
	W. Zheng et al, (2018) [12]	Low	---	The ability to execute a special attack	O((V+E)log k	Numerical and educative	No	Almost	Almost low
Cluster based	J. Tekli et al, (2018) [39]	Medium	– Definition of the threshold for non-interruption of data release – No use of buffer – Higher security	---	---	Numerical and deductive	No	Appropriate	Almost low
	A.Otgonbayar et al, (2018) [52]	Medium	---	– No threshold is considered for preserving data in the processing window – Data remains in the window for a long time – Not suitable for real-time processing	$O\|S^2\|$	Numerical	No	Appropriate	High
	J.Wang et al, (2018) [32]	Low	Suitable for big data	– Computational overhead – Not suitable for real-time processing	$O\|S^2\|$	Numerical and deduct	No	Appropriate	High
	B. B. Mehta et al, (2018) [11]	Low	– Minimizing the loss rate – Being resistant to any type of attack	– Update problem – Increases execution time – Not suitable for real-time processing	$O\|S^2\|$	Numerical and eductive	No	Appropriate	Almost low
	J. Andrew et al, (2020) [57]	Low	– Suitable for big data. – focus on the attributes of pseudo-identifiers. – omitting the need for previous determination of K parameter. – being needless of the awareness about the duplicated values in the columns of dataset. – being needless of adding fake data to the tables for reaching a certain K threshold. – assuring of that the highest possible value of K threshold is considered.	---	---	Numerical and deductive	No	Appropriate	High
	P.Jain et al, (2020) [67]	low	– Suitable for big data – Significant improvement in runtime and rate of data loss	---	O(n)	Numerical and deductive	No	Appropriate	Almost low
	A.Raj et al, (2019) [66]	medium	– Significant improvement in runtime and rate of data loss	---	---	Numerical and deductiv	No	Almost	Almost low
	Our Proposed model	low	– Definition of the threshold for non-interruption of data release – No use of buffer – Higher security	---	O((V+E)logk	Numerical and deductive	No	Appropriate	Almost low

Presenting the Proposed Model

To put light on the topic, the main and basic concepts used in the proposed model are represented in summary:

Methods of Determining the Optimized Quantity of Clusters

Methods of determining the optimized quantity of clusters are divided into two groups Fig. 8.

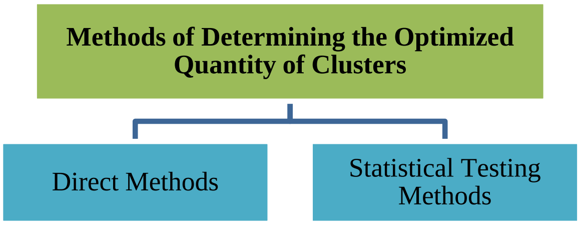

Methods of Determining the Optimized Quantity of Clusters
Direct Methods
Statistical Testing Methods

Fig. 8: Methods of determining the optimized quantity of clusters.

Direct Methods

This method seeks optimizing a particular scale like Cluster Sum of Square (WSS) or Average Silhouette. Among these methods are elbow and methods based on silhouette scale.

Statistical Testing Methods

This method seeks synchronizing the observations with a null hypothesis of a statistical test. Gap Statistics is among these methods.

The optimized quantity of clusters in K-means clustering algorithm is calculated in R programming language by the codes below (Fig. 9).

Algorithm 1

```
# Elbow method
fviz_nbclust(df, kmeans, method= "wss") + geom_vline(xintercept = 4, linetype = 2) +  labs(subtitle = "Elbow method")
# Silhouette method
fviz_nbclust(df, kmeans, method = "silhouette") +labs(subtitle = "Silhouette method")
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot = 500 for yor analysis.
# Use verbose = FALSE to hide computing progression
set.seed(123)
fviz_nbclust(df, kmeans, nstart = 25, method ="gap_start", nboot = 50)+labs(subtitle = "Gap statistic method")
```

Fig. 9: Elbow method.

Optimal Tuple of Clusters

Most previous algorithms have used cluster generalization to send clusters. In this way, a cluster is mapped to a tuple by placing its attributes in the range obtained by an equation [54]- [56]. One of the disadvantages of this method was the high data loss rate, especially in wide-range data. In this method, we define the generalization function G, as G: PowerSet (Tuple) → Tuple with brief modifications. In this definition, Tuple represents the set of all possible tuples. PowerSet also represents all the clusters that can be defined in the S stream. The definition of the G function in (5) is fully defined.

$$G(c)=gt \quad \text{and} \quad \forall\, t \in c\ .\forall\, q \in QI\ .t \cdot q \subseteq gt \cdot q \tag{5}$$

In this regard, QI specifies a set of identifiers. The meaning of this equation is a tuple out of a cluster is the subset of the same cluster.

Also, (6) will specify how to determine the selected tuple attributes of each cluster.

$$\begin{cases} [\text{mean}(q_1 \ldots q_k) \pm \sigma] & \text{if } q \text{ are numerical attribute} \\ \text{Ancestor}(q_1 \ldots q_k) & \text{if } q \text{ are categorical attribute} \\ \text{mod}\ (q_1 \ldots q_k) & \text{if } q \text{ are other attribute} \end{cases} \tag{6}$$

In the first case, if the data type is numerical, primarily the mean and variance of the data is calculated and the interval obtained from addition and subtraction will be considered for publication. If the data type is a tree structure, the top order is selected for the existing data. If the data are different from these two cases, the data with the most repetition is considered (if the number of attributes is equal, it selects one at random). For example, considering cluster C as follows and Fig. 10, the optimal tuple of C is calculated as follows:

C={<"ali.ahmadi" , male , Academic , 42> , <"maryam.mahmoudi" , female , non_Academic , 38> , <"amin.davoodi" , male , non_Academic , 50>}

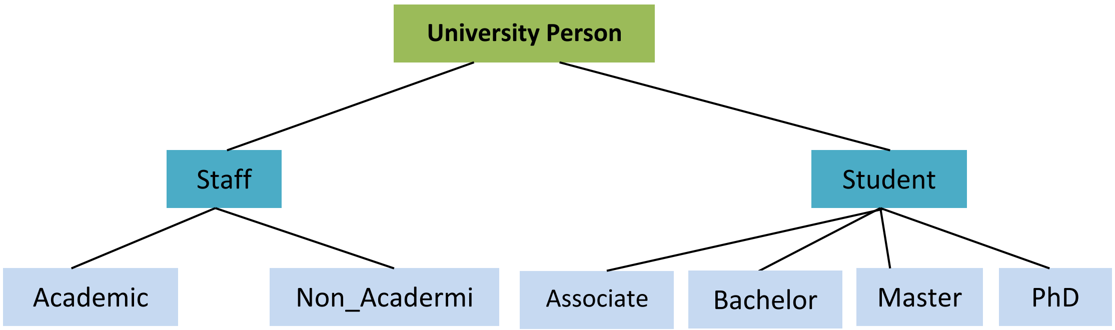


Fig. 10: University hierarchy.

G(c)=<"Explicit identifier" , mod(male, female, male), Ancestor[Academic, non_Academic], mean(42,38,50)±5]> = <*, male, staff, [38,48]>

k-anonymity Cluster

If a cluster C consists of an S stream and has more members than K, it is called a k-anonymity cluster.

Division of Clusters

Clusters whose number has reached K * k are divided into k parts by new K-means algorithm with new branches and the number of members is considered zero. K is the number of members to maintain anonymity and k is the coefficient considered in cluster division. Dividing the cluster allows new tuples to be selected in subsequent rounds, so that the values in the appropriate clusters are close together. As a result, the rate of data loss will be reduced and the result of data output analysis will be increased.

Targeted Removal of Clusters

In order not to exceed a certain number of clusters stored in the algorithm, we use (7) to determine the clusters that should be deleted. After the number of clusters exceeds the specified limit during the execution of the algorithm steps, we use the mentioned equation to remove the cluster that has the least number of members and the oldest reference time (LRU).

$$\mathrm{C_{del}} = (1 - \alpha)\mathrm{n} + \alpha\mathrm{t} \qquad (7)$$

In this equation, n is the number of tuples in the cluster, t is the last time a tuple refers to this cluster, and α is the coefficient that controls the weight of the two components. Clusters with the lowest values in this regard have a higher priority elimination.

Tuple Expiration Time

Real time response is among the data stream's most significant necessities which should be considered when designing anonymization systems. A tuple may remain in the system after several rounds of execution of the algorithm and will not be published and then, will be published after allowed time. This causes the system to be out of the real-time response mode and significantly increases the cost criterion. To solve this problem in the proposed algorithm, an Expiration Time (ET) parameter is considered, which indicates the maximum tolerable latency in the system. There is also a simple, innovative function called Estimated Round Time to prevent tuple publication being released after the allowed time. This function maintains the estimated time to run the next round of the algorithm and is updated in each round of running the algorithm. Next, for the remaining tuples in the system, (8) is checked and if it is correct, the tuples return to the corresponding cluster. Otherwise, the tuples should be published immediately with the cluster representative.

$$\text{(Current\_time- Arrival\_Time)+ EstimatedRoundTime} < \text{Expiration\_time} \qquad (8)$$

Distance of Two Tuples

Suppose cluster C is to be formed of a set of data (TSet) so that |T_Set| ≥K and the amount of data loss are minimized. The closer the tuples in a data set are, the less information is lost. The distance between two tuples $t1$ and $t2$ is determined by (9).

$$\text{distance (t1,t2)= w × distance(t1.QI , t2.QI)} \qquad (9)$$

In (9), w is a vector of weight n×1, and t1. QI is a vector for tuple pseudo-identifiers t1. In fact, the weight vector of an array contains classified data that are used to determine the distance of this type of data. The distance between (t1. QI, t2. QI) is determined by the equation $distance\ (QI.\ QI,\ QI.\ QI)$ = $[d1,\ldots,\ dn]$. In this equation di is calculated from (10).

$$d_i = \begin{cases} \dfrac{|t_l.\text{number} - t_k.\text{number}|}{t_l.\text{number}} & \text{for numerical data} \\ \dfrac{|t_l.\text{level} - t_k.\text{level}|}{t_l.\text{number}} & \text{for categorical data} \\ 1 & \text{if } t_l.\text{atrrib} \neq t_k.\text{attrib} \end{cases} \quad (10)$$

The method of constructing the classified data tree is based on the rules of the construction of the perfect tree in the building, and the arrangement of data is based on the properties of the tree, but the arrangement of the stream of input data in the tree formation will not be effective. Due to the need for high-speed data anonymization in real environments and the inefficiency of existing algorithms in this field and the high amount of data lost during publishing, in this section a parallel algorithm to increase efficiency in anonymizing data streams and at the same time reduction in the rate of data loss is provided. The proposed model consists of four steps. The first step is to enter the raw data from the HDFS into the RDDs, the second step is determined by a function, m clusters and their headers. In the third step, the obtained tuples are placed in parallel in separate RDDs, and finally in the fourth step, the work of classifying and publishing the clusters is done. The general process of work is shown in Fig. 11 and each step is described in detail below.

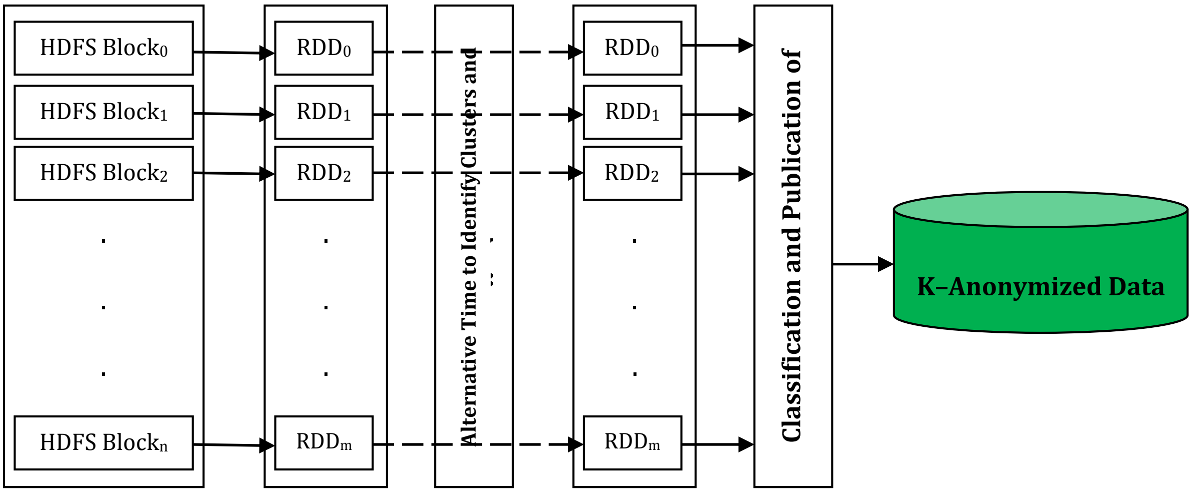


Fig. 11: The general procedure of the proposed model

Step 1. The raw data blocks in HDFS are transferred to the RDDs embedded in the system. This transfer is a type of memory transfer mapping and the number of HDFS blocks will not necessarily be equal to the number of RDDs at this stage (Fig.12).

Step 2. In this step, according to the function problem space, it is called to introduce the m point as the primary representatives of the clusters and place each in a separate cluster. This function can introduce agents randomly or by dividing the problem space. The following quasi-code describes the function of this function (Fig. 12).

Algorithm 2

```
DefinitionCluster(Dataset , m)
  for m point do
     for each attribute in dataset do
       If (attributei was numerical ) then
          Split (attributei) to m segment;
           Insert each point of segment to agenti,m;
       else
          Random select types(attributei) without placement;
          Insert selected item to agenti,m;
       End if
     End for
   Insert agenti, m into cluster Cm and add Cm in Call;
  End for
```

Fig. 12: Transfer data blocks to RDDs and introduce them points.

These clusters are then read in parallel and each placed in new RDDs. In the last step, the function that does the sorting and publishing is presented in the following quasi-code (Fig. 13).

Algorithm 3

```
Main (S ,Call, Te , setexpir)
    Create a new thread;
    while S ≠ 0 do
        for each tuple tp in setexpir do
            Update expire_timetp;
        if (expire_timetp) > Te then
            Publish tp in cluster Ci whit agenti;
            Read tuples and inseart each  into a RDDtp;
        end if
        Call function Categorize(RDDtp, Call);
    End while
```

Fig. 13: Sorting and publishing.

Here S represents the inputs, Call all available clusters, Te the maximum system latency, and setexpir the unpublished tuples. The Categorize function is also defined according to the following quasi-code (Fig. 14).

Algorithm 4

```
Categorize (RDDn,Call)
    for each agent in Call
        Calculate information loss between RDDn and  agenti;
            Insert RDDn into cluster Ci with  incures less  information loss;
    End for
  Call function PublishData(Ci,k,Nset);
  Terminate the thread;
```

Fig. 14: The categorize RDDs.

At this stage, by examining the size of the cluster, the necessary measures are taken for tuple publication. To preserve the K-anonymity property, minimum cluster size should be published to K, so if the number of cluster tuples reaches to K, we first calculate optimal tuple of clusters and publish the members in the cluster with the optimal calculated point. If the number of members of the cluster is greater than K and less than Nset, we publish the newly added tuple of the cluster with the optimum point calculated for the cluster in the previous steps.

A cluster whose tuple has become Nset, will first publish the newly added tuple with the optimal cluster point, then call Split Cluster function to divide the cluster. The following quasi-code checks the function of the publication function (Fig. 15).

Algorithm 5

```
PublishData(Ci,k,Nset)
 numi=the number of members cluster Ci;
    If (numi ≥ k ) then
        If (numi = k ) then
           Call  function ObtimumTuple(Ci);
            Publish all members of the cluster Ci whit  optimum tuple calculated;
        End if
    Else if (numi = Nset ) then
        Publish new member of cluster Ci with optimum  tuple;
    Ksplit= Nset / k    \\ for K-means factor
  Call function SplitCluster(Call,Ci,NC, Ksplit);
  End if
  Else if (k < numi < Nset ) then
    Publish new member whit optimum tuple of cluster  Ci;
       End if
    Else return
```

Fig. 15: Examining size of the cluster and call SplitCluster function.

Nset is the maximum number of members allowed in each cluster. To split a cluster whose tuple of members has reached a specified number, the SplitCluster function is called. This function first checks the number of existing clusters before adding the newly formed clusters to the set and adds the clusters to the set if they are less than the allowable limit, but if the number is more than the allowed limit, it first calls the targeted deletion function of the clusters to remove the member for the new categories generated from the cluster set and then adds new clusters to the set. The members of the deleted clusters, if they have enough time, put them back in the other cluster, etc. Otherwise, it publishes it with its corresponding header. The quasi-code provides the following steps (Fig. 16).

Algorithm 6

```
SplitCluster(Call,Ci,NC,Ksplit)
    Split Ci with Ksplit-Means algorithm;
    Numc = the number of members Call;
    If (numc + Ksplit ≥ NC) then
        Call   function TargetedRemove(Call, Ksplit);
    Add Ksplit new cluster to Call;
```

Fig. 16: Run SplitCluster.

Here, Nc refers to the maximum number of members per cluster.

So, since the proposed model is represented based on in-memory processing tools, its performance time is lower than previous models which were not implemented on a big data basis or used non-in-memory big data tools. Also, the rate of data loss decreases because of determining the non-random optimal cluster head in K-means algorithm; previous methods have used totally random determination.

Finally, the Spark logic of the model below is observed (Fig. 17).

Algorithm 7

```
Main
Input:    k-value, data, partition num, attr
Output:  k-value, node, k-Anonymized table
//Step 1: Create Taxonomy Tree and
        Generaliztion Lattice Tree
        Make_Taxonomy Tree (attr)
        Make_Generalization_Lattice(t-tree)
//Step 2: Make RDD and Partiton. Cache
        Make_RDD_From_HDFS(data);
        RDD_Repartition(partition num);
        Cache();
//Step 3: K-Anonymity (Map & Reduce)
        K_check = false;
        while(!k_check)
            node = next Generalization Lattice;
            result = MapReduce(node);
            k_check = Check_k_value(result);
        end while;
        Save_Output_to_HDFS(path);
```

Fig. 17: Main code.

Results and Discussion

Through the implementation of the proposed model, the dataset [58] will have various attributes, according to the Fig. 18.

```
root
 | - - CUST_ID: string  (nullable  =  true)
 | - - BALANCE: double  (nullable  =  true)
 | - - BALANCE_FREQUENCY: double  (nullable  =  true )
 | - - PURCHASES: double (nullable  =  true)
 | - - ONEOFF_PURCHASES: double (nullable  =  true)
 | - - INSTALLMENTS_PURCHASES: double (nullable  =  true)
 | - - CASH_ADVANCE: double  (nullable  =  true)
 | - - PURCHASES_FREQUENCY: double (nullable  = true)
 | - - ONEOFF_PURCHASES_FREQUENCY: double (nullable  =  true)
 | - - PURCHASES_INSTALLMENTS_FREQUENCY: double (nullable  =  true)
 | - - CASH_ADVANCE_FREQUENCY: double (nullable  =  true)
 | - - CASH_ADVANCE_TRX: integer (nullable  =  true)
 | - - PURCHASES_TRX: integer (nullable  =  true)
 | - - CREDIT_LIMUT: double (nullable  =  true)
 | - - PAYMENTS: double (nullable  =  true)
 | - - MINIMUM_PAYMENTS: double  (nullable  =  true)
 | - - PRC_FULL_PAYMENT: double  (nullable  =  true)
 | - - TENURE: integer (nullable =  true)
```

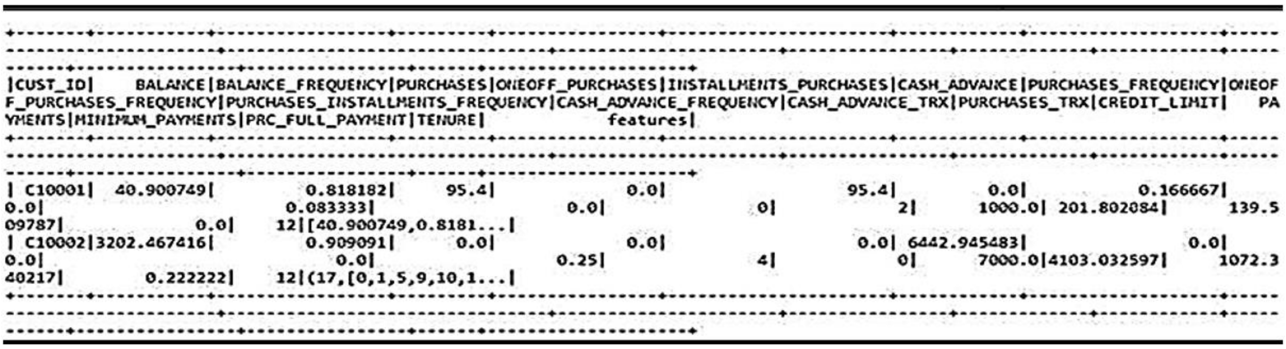
Fig. 18: Various attributes dataset [58].

First, preprocessing and standardization of the given data will be exerted according to the code below (Fig. 19).

```
from pyspark.ml.feature  import  VectorAssembler
data_customer.columns
assemble=VectorAssembler(inputCols=[
  ' BALANCE ',
  ' BALANCE_FREQUENCY ',
  ' PURCHASES ',
  ' ONEOFF_PURCHASES ',
  ' INSTALLMENTS_PURCHASES ' ,
  ' CASH_ADVANCE ',
  ' PURCHASES_FREQUENCY ',
  ' ONEOFF_PURCHASES_FREQUENCY ',
  ' PURCHASES_INSTALLMENTS_FREQUENCY ',
  ' CASH_ADVANCE_FREQUENCY ',
  ' CASH_ADVANCE_TRX ',
  ' PURCHASES_TRX ' ,
  ' CREDIT_LIMIT ',
  ' PAYMENTS ',
  ' MINIMUM_PAYMENTS ' ,
  ' PRC_FULL_PAYMENT ' ,
  ' TENURE ' ],  outputCol=' features ')
assembled_data=assemble.transform(data_customer)
assembled_data.show (2)
```

Fig. 19: Preprocessing codes.

The results of preprocessing the first two lines are illustrated in Fig. 20.



```
+--------+------------+-----------------+---------+----------------+----------------------+------------+-------------------+-----
--------------------+--------------------------------+----------------------+----------------+-------------+------------+-----
-----+--------------+----------------+------+--------------------+
|CUST_ID|     BALANCE|BALANCE_FREQUENCY|PURCHASES|ONEOFF_PURCHASES|INSTALLMENTS_PURCHASES|CASH_ADVANCE|PURCHASES_FREQUENCY|ONEOF
F_PURCHASES_FREQUENCY|PURCHASES_INSTALLMENTS_FREQUENCY|CASH_ADVANCE_FREQUENCY|CASH_ADVANCE_TRX|PURCHASES_TRX|CREDIT_LIMIT|    PA
YMENTS|MINIMUM_PAYMENTS|PRC_FULL_PAYMENT|TENURE|            features|
+--------+------------+-----------------+---------+----------------+----------------------+------------+-------------------+-----
--------------------+--------------------------------+----------------------+----------------+-------------+------------+-----
-----+--------------+----------------+------+--------------------+
| C10001|  40.900749|          0.818182|     95.4|             0.0|                  95.4|         0.0|           0.166667|
0.0|                         0.083333|                  0.0|               0|            2|      1000.0| 201.802084|      139.5
09787|            0.0|    12|[40.900749,0.8181...|
| C10002|3202.467416|          0.909091|      0.0|             0.0|                   0.0| 6442.945483|                0.0|
0.0|                              0.0|                 0.25|               4|            0|      7000.0|4103.032597|     1072.3
40217|       0.222222|    12|(17,[0,1,5,9,10,1...|
+--------+------------+-----------------+---------+----------------+----------------------+------------+-------------------+-----
--------------------+--------------------------------+----------------------+----------------+-------------+------------+-----
-----+--------------+----------------+------+--------------------+
```

Fig. 20: The results of preprocessing the first two lines.

When implementing the proposed method, k=2 has been considered; the results are illustrated in Table 4.

Table 4: Results of implementing the proposed model with k=2

Original Table			
RID	Age	Gender	Disease
1	31	M(1)	Diabetes
2	21	M(1)	Anemia
3	26	F(0)	Pneumonia
4	36	F(0)	Anemia
5	34	M(1)	Diabetes
6	25	F(0)	Pneumonia

2-Anonyized Table			
RID	**Age**	**Gender**	**Disease**
1	**30~39**	***(0~1)**	**Diabetes**
2	**20~29**	***(0~1)**	**Anemia**
3	**20~29**	***(0~1)**	**Pneumonia**
4	**30~39**	***(0~1)**	**Anemia**
5	**30~39**	***(0~1)**	**Diabetes**
6	**20~29**	***(0~1)**	**Pneumonia**

According to Fig. 21, the results of the loss criterion have reached the stability value of 0.30.

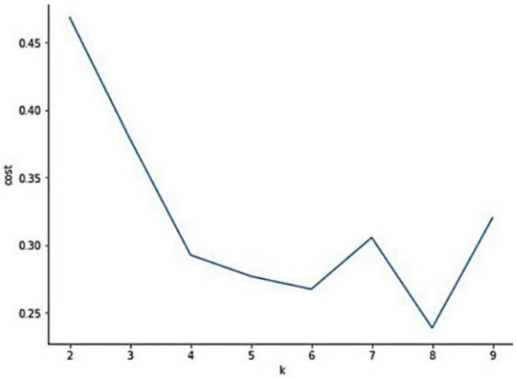


Fig. 21: Results of the loss criterion with k=9.

Finally, the rate of data loss in the suggested model with two other clustering algorithms, i.e. FADS and FAST, are shown with different values of K for 10 and 100 megabytes of data are shown in Figs. 22 and 23 respectively.

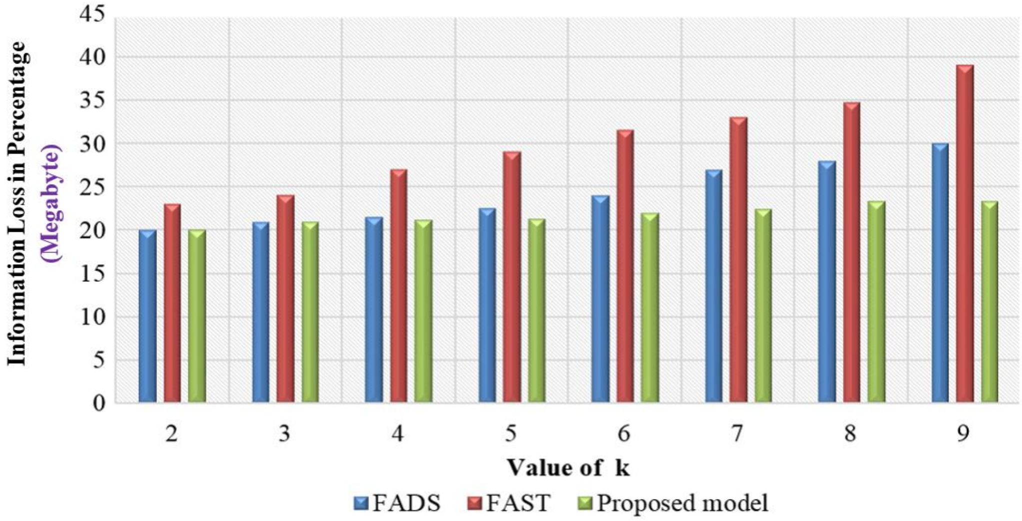


Fig. 22: The rate of data loss in FADS and FAST clustering algorithms and the suggested model with 10 megabytes of data.

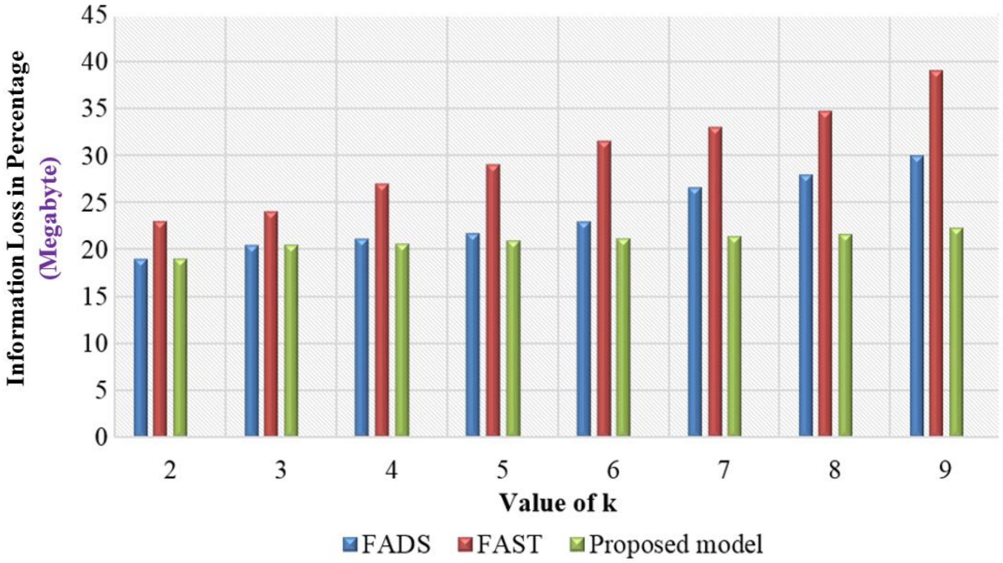


Fig. 23: The rate of data loss in FADS and FAST clustering algorithms and the suggested model with 100 megabytes of data.

Considering the results of simulation in Figs. 23 and 24 the rate of data loss in suggested model, in both cases of 10 megabytes and 100 megabytes of data, is lower than that in FAST and FADS clustering algorithms so, despite the increase of data volume, the rate of data loss is reduced because of increasing the records.

Conclusions and Future Works

Anonymization is not limited only to omitting some attributes and replacing some values with other ones; actually, it is an effort for finding a method to make an optimized relation between privacy protection and the possibility of using data while decreasing the rate of information loss. In this paper, various methods of anonymization, such as anonymization based on perturbation, based on a tree structure, based on zero-delay, based on the addition of artificial data, based on fuzzy method, based on clustering, and common algorithms are presented, accompanied by a comparison of their various parameters. The architectures and models in the related literature which have dealt with big data anonymization are investigated, and factors such as the data loss rate, amount of extra data production, suitability for big data environment, and delay time are compared. Reviewing the related literature, it is revealed that there still exists the necessity of high-speed data anonymization in real environments, the inefficiency of the current algorithms, and the high rate of data loss through release. However, the results of investigating the suggested model and solution show that clustering by in-memory processing of Spark platform provides a suitable and reasonable time for the anonymized release of big data, and the designed steps reduce the information loss to the lowest possible amount. Considering the results, while the data volume increases, the rate of data loss decreases compared to FADS and FAST clustering algorithms which are because of increasing the records in the suggested model.

Author Contributions

All the authors participated in all aspects of the preparation and writing of this article.

Acknowledgment

The authors thankfully appreciate the anonymous reviewers and the editor of JECEI for their useful comments and suggestions.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

AIDS	Acquired Immunodeficiency Syndrome
ET	Expiration Time
FAANST	Fast Anonymizing Algorithm for Numerical Streaming data
FADS	Feature anomaly detection system
HDFS	Hadoop Distributed File System
IKA	Improved K-Anonymization
ILD	Improved L-Diversity
KIDS	K-anonymization Data Stream
LRU	Least Recently Used
Mllib	Machine Learning Library
MRA	MapReduce based Anonymization
NCP	Normalized Certainty Penalty
RDD	Resilient Distributed Datasets
RMSE	Root Mean Square Error
RSA	Rivest Shamir Adleman
SKA	Scalable k-Anonymization
SKY	Stream K-anonymity
SWAF	Sliding Window Anonymization Framework
TKC	Tanzu Kubernetes Cluster
TPTDS	Two Phase Top-Down Specialization
WSS	Within Cluster Sums of Squares

References

[1] P. Zhao, H. Jiang, C. Wang, H. Huang, G. Liu, Y. Yang, "On the performance of k-anonymity against inference attacks with background information," IEEE Internet Things J., 6(1): 808-819, 2019.

[2] S. Sangeetha, G. Sudha Sadasivam, Handbook of Big Data and IOT Security, first ed., Springer, Switzerland, 2019.

[3] S. Patnaik, New Paradigm of Industry 4.0: Internet of Things, Big Data & Cyber Physical Systems, first ed., Springer, Switzerland, 2019.

[4] A. Chaudhary, Ch. Choudhary, M. Kumar Gupta, Ch. Lal, T. Badal, Microservices in Big Data Analytics, first ed., Springer, Singapore, 2019.

[5] X. Zhang, Ch. Liu, S. Nepal, Ch. Yang, J. Chen, Security, Privacy and Trust in Cloud Systems, first ed., Springer, Berlin, 2013.

[6] J. Salas, J. Domingo-Ferrer, "Some basics on privacy techniques, anonymization and their big data challenges," Math. Comput. Sci., 12: 263–274, 2018.

[7] N. Victor, D. Lopez, "Privacy models for big data: A survey," J. Big Data Intel., 3: 61-75, 2016.

[8] K-K. Raymond Choo, A. Dehghantanha, Handbook of Big Data Privacy, Springer, Switzerland, 2020.

[9] M. Al-Zobbi, S. Shahrestani, Ch. Ruan, "Improving mapreduce privacy by implementing multi-dimensional sensitivity-based anonymization", J. Big Data., 4(1): 1-23, 2017.

[10] Sh. Luan Hou, X. Kun Huang, Ch. Qun Fei, Sh. Han Zhang, Y. Yang Li, Q. Lin Sun, Ch. Qing Wang, "A survey of text summarization approaches based on deep learning," J. Comput. Sci. Technol., 36: 633-663, 2021.

[11] B. B Mehta, P. Rao U, "Toward scalable anonymization for privacy-preserving big data publishing," Adv. Intel. Syst. Comput., 2: 297-304, 2018.

[12] W. Zheng, Z. Wang, T. Lv, Y. Ma, C. Jia, "K-Anonymity algorithm based on improved clustering," ICA3PP, 11335: 462-476, 2018.

[13] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, McCauley, M. J. Franklin, S. Shenker, I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," NSDI'12, 15-28, 2012.

[14] P. Ram Mohan Rao, S. Murali Krishna, A. P. Siva Kumar, "Privacy preservation techniques in big data analytics: A survey," J. Big Data., 5: 1-12, 2018.

[15] S. Khan, Kh. Iqbal, S. Faizullah, M. Fahad, J. Ali, W. Ahmed, "Clustering based privacy preserving of big data using fuzzification and anonymization operation," Int. J. Adv. Comput. Sci. Appl. (IJACSA), 10(12): 282-289, 2019.

[16] A. Dobson, K. Roy, X. Yuan, J. Xu, "Performance Evaluation of machine learning algorithms in apache spark for intrusion detection," in Proc. International Telecommunication Networks and Applications Conference (ITNAC), 127:1-6, 2018.

[17] S. Ullah Bazai, J. Jang-Jaccard, "SparkDA: RDD-Based high-performance data anonymization technique for spark platform," in Proc. International Conference on Network and System Security, 11928: 646-662, 2019.

[18] Y. Canbay, S. Sagiroglu, "Big data anonymization with spark, in Proc. International Conference on Computer Science and Engineering, (UBMK): 833-838, 2017.

[19] M. Al-Zobbi, S. Shahrestani, Ch. Ruan, "Experimenting sensitivity-based anonymization framework in apache spark," J. Big Data., 5: 1-26, 2018.

[20] M. Mittal, V. E. Balas, L. Mohan Goyal, R.Kumar, Big Data Processing Using Spark in Cloud, first ed., Springer, Singapore, 2019.

[21] Z. He, H. Cai, "Latent-Data privacy preserving with customized data utility for social network data," IEEE Trans. Veh. Technol., 67(1): 665-673, 2018.

[22] B. Matturdi, X. Zhou, S. Li, F. Lin "Big data security and privacy: a review," China Commun., 11(14): 135-145, 2014.

[23] Z. Ouazzani, H. Bakkali, "A new technique ensuring privacy in big data: k-anonymity without prior value of the threshold k," Procedia Comput. Sci., 127: 52-59, 2018.

[24] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, Q. Ni, "A k-anonymity based schema for location privacy preservation," IEEE Trans. Sustainable Comput., 4(2): 156-167, 2019.

[25] Y. Canbay, Y. Vural, S. Sagiroglu, “Privacy Preserving Big Data,” in Proc. International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 24-29, 2018.

[26] A. Kayem, C. T. Vester, Ch. Meinel, "Automated k-anonymization and l-diversity for shared data privacy," in Proc. International Conference on Database and Expert Systems Applications (DEXA), 9827: 105-120, 2016.

[27] J. Shish Patel, S. Priyanka, "Online analytical processing for business intelligence in big data," J. Big Data, 8(6): 501-518, 2020.

[28] K. R. Macwan, S. J. Patel, "k-NMF anonymization in social network data publishing," Secur. Comput. Syst. Networks Comput., 61(4): 601–613, 2018.

[29] A. Reiza, M. A. Armengol de la Hoz, M. S. Garcíaa, "Big data analysis and machine learning in intensive care units," Med. Intensiva, 43(7): 416-426, 2019.

[30] J. Novotny, P. A. Bilokon, A. Galiotos, F. Deleze, Machine Learning and Big Data with kdb+/q, first ed., Wiley, London, 2020.

[31] M. Bowles, Machine Learning with Spark and Python, Second ed., John Wiley & Sons., Indianapolis, 2020.

[32] J. Wang., Zh. Cai, Y. Li, D. Yang, L. Li, H. Gao, "Protecting query privacy with differentially private k-anonymityin location-based services," Pers. Ubiquitous Comput., 22: 453–469, 2018.

[33] L. Arbuckle, Kh. El Emam, Building an Anonymization Pipeline, first ed., O'Reilly Media, California, 2020.

[34] S. Ram Prasad Reddy, K. V.S.V.N. Raju, V. Valli Kumari, "Personalized privacy preserving incremental data dissemination through optimal generalization," J. Eng. Appl. Sci., 13(11): 4205–4216, 2018.

[35] J. Domingo-Ferrer, "Big data anonymization requirements vs privacy models," in Proc. International Conference on E-Business and Telecommunication Networks (ICETE), 2: 305-312, 2018.

[36] S. A Abdelhameed, Sh. M Moussa, M. E Khalifa, "Restricted sensitive attributes-based sequential anonymization (RSA-SA) approach for privacy-preserving data stream publishing," Knowledge-Based Syst., 164: 1-20, 2019.

[37] Y. Canbay, A. Kalyoncu, M. Ercimen, A. Dogan, S. Sagiroglu, "A clustering based anonymization model for big data," in Proc. International Conference on Computer Science and Engineering (UBMK): 720-725, 2019.

[38] J. Tekli, B. Al Bouna, Y. Bou Issa, M. Kamradt, R. Haraty, "(k, l)-clustering for transactional data streams anonymization," International Conference on Information Security Practice and Experience (ISPEC), 11125: 544-556, 2018.

[39] P. Jain, M. Gyanchandani, N. Khare, "Improved k-anonymity privacy-preserving algorithm using madhya pradesh state election commission big data," Commun. Security, Stud. Comput. Intel., 771: 1-10, 2019.

[40] K. Guo, Q. Zhang, "Fast clustering-based anonymization approaches with time constraints for data streams," J. Software, 24: 1852-1867, 2014.

[41] Y. Wang, Zh. Chi, X. Tong, L. Li, "A differentially k-anonymity-based location privacy-preserving for mobile crowdsourcing systems," Procedia Comput. Sci., 129: 28-34, 2018.

[42] C. Eyupoglu, M. Aydin, A. Zaim, A. Sertbas, “An Efficient big data anonymization algorithm based on chaos and perturbation techniques," Entropy, 20(5): 1-18, 2018.

[43] A. Nezarat, Kh. Yavari, "A distributed method based on mondrian algorithm for big data anonymization," in Proc. International Congress on High-Performance Computing and Big Data Analysis (HPC), 891: 84–97, 2019.

[44] H. Silva, T. Basso, R. Moraes, D. Elia, S. Fior, "A re-identification risk-based anonymization framework for data analytics platforms," in Proc. European Dependable Computing Conference (EDCC): 101-106, 2018.

[45] K. Abouelmehdi, A. Beni-Hessane, H. Khaloufi, "Big healthcare data: Preserving security and privacy," J. Big Data, 5: 1-18, 2018.

[46] J. Domingo-Ferrer, J. Soria-Comas, "Anonymization in the Time of Big Data," International Conference on Privacy in Statistical Databases (PSD), 9867: 57–68, 2016.

[47] P. Ghavami, Big Data Analytics Methods: Analytics Techniques in Data Mining, Deep Learning and Natural Language Processing, second ed., De Gruyter, Berlin, 2020.

[48] M. Z. Zgurovsky, Y. P. Zaychenko, Big Data: Conceptual Analysis and Applications, first ed., Springer Nature, Switzerland, 2020.

[49] D. Kumar Mishra, X. She Yang, A. Unal, Data Science and Big Data Analytics: ACM-WIR 2018 (Lecture Notes on Data Engineering and Communications Technologies, 16), first ed., Springer, Singapore, 2019.

[50] Rexa.info at the University of Massachusetts Amherst {Datasets Adult}.

[51] M. Kiabod, M. N. Dehkordi, B. Barekatain, “TSRAM: A Time-Saving k-degree Anonymization Method in Social Network,” Expert Syst. Appl., 125: 378-396, 2019.

[52] A. Otgonbayar, Z. Pervez, K. Dahal, S. Eager, "K-VARP: k-anonymity for varied data streams via partitioning," Inf. Sci., 467: 238-255, 2018.

[53] G. Kaur, S. Agrawal, "Differential privacy framework: impact of quasi-identifiers on anonymization," in Proc. 2nd International

Conference on Communication, Computing and Networking, 46: 35–42, 2018.

[54] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, "Protecting query privacy with differentially private k-anonymity in location-based services," Pers. Ubiquitous Comput., 22: 453–469, 2018.

[55] Ch. N. Yang, Sh. L. Peng, L. C. Jain, Security with Intelligent Computing and Big-data Services, first ed., Springer Switzerland, 2020.

[56] L. Oneto, N. Navarin, A. Sperduti, D. Anguita, Recent Advances in Big Data and Deep Learning. Springer, Genova, 2020.

[57] J. Andrew, J. Karthikeyan, Privacy-Preserving Big Data Publication: (K, L) Anonymity, Advances in Intelligent Systems and Computing (AISC), 67: 77–88, 2020.

[58] Rexa.info at the University of Massachusetts Amherst {Datasets Bank and Marketing}.

[59] T. Banirostam, H. Banirostam, M. M. Pedram, A. M. Rahamni, "A review of fraud detection algorithms for electronic payment card transactions," J. Adv. Comput. Eng. Technol., 7(3): 157-166, 2021.

[60] H. Banirostam, E. Shamsinezhad T. Banirostam, "Functional control of users by biometric behavior features in cloud computing," in Proc. International Conference on Intelligent Systems, Modelling and Simulation: 94-98, 2013.

[61] H. Banirostam, A. Hedayati, A. Khadem Zadeh, E. Shamsinezhad, "A trust based approach for increasing security in cloud computing infrastructure," in Proc. UKSim-International Conference on Computer Modeling and Simulation: 717-721, 2013.

[62] H. Banirostam, A. R. Hedayati, A. Khadem Zadeh, "Using virtualization technique to increase security and reduce energy consumption in cloud computing," Int. J. Res. Comput. Sci., 4(2): 25-30, 2014.

[63] E. Shamsinezhad, A. Shahbahrami, A. Hedayati, A. Khadem Zadeh, H. Banirostam, "Presentation methods for task migration in cloud computing by combination of Yu router and post-copy," Int. J. Comput. Sci. Issues (IJCSI), 10(4): 98-102, 2013.

[64] T. Banirostam, E. Shamsinejad, M. M. Pedram, A. M. Rahamni, "A review of anonymity algorithms in big data," J. Adv. Comput. Eng. Technol., 7(3): 187-196, 2021.

[65] Z. El Ouazzani, H. El Bakkali, "A new technique ensuring privacy in big data: k-anonymity without prior value of the threshold k," in Proc. 1th International Conference On Intelligent Computing in Data Sciences, 127: 52-59, 2018.

[66] A. Raj, R. G L D'Souza, "Big data anonymization in cloud using k-anonymity algorithm using map reduce framework," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., 5(1): 50-56, 2019.

[67] P. Jain, M. Gyanchandani, N. Khare, " Improved k-anonymize and l-diverse approach for privacy preserving big data publishing using MPSEC dataset," Comput. Inf., 39(3): 537–567, 2020.

Biographies

Elham Shamsinejad is a Ph.D. student in the Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran. Her research interests include Machine learning, Deep learning, Big Data, Data Analytics and Python Programming.

- Email: e.shamsinejad.eng@iauctb.ac.ir
- ORCID: 0009-0001-4941-8921
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Touraj Banirostam is an Assistant Professor in the Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran. His research interests include Cognitive Science Engineering, Artificial Intelligence, Learning, Self-Management Systems.

- Email: h.banirostam.eng@iauctb.ac.ir
- ORCID: 0000-0002-3477-9046
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://ctb.iau.ir/faculty/t-banirostam-comp/en

Mir Mohsen Pedram received the B.Sc. degree in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, 1990, and the M.Sc. and Ph.D. degrees in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 1994 and 2003, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Kharazmi University. His main areas of research are intelligent systems, machine learning, data mining, and cognitive science.

- Email: pedram@khu.ac.ir
- ORCID: 0000-0002-0674-4428
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://eng.khu.ac.ir/cv/318/en

Amir Masoud Rahmani is currently working as a Professor for Islamic Azad University, science and research branch, Tehran. He is the author/co-author of more than 220 publications in technical journals and conferences. His research interests are in the areas of distributed systems, wireless sensor networks, Internet of Things and evolutionary computing. Address: Amir Masoud Rahmani, Computer Engineering dept, Islamic Azad University, Science and Research branch, Hesarak, Ashrafi Esfahani, Poonak Square, Tehran, IRAN.

- Email: rahmani@srbiau.ac.ir
- ORCID: 0000-0001-8641-6119
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://eng.khu.ac.ir/cv/318/en

How to cite this paper:
E. Shamsinejad, T. Banirostam, M. M. Pedram, A. M. Rahmani, "Presenting a model of data anonymization in big data in the context of in-memory processing framework," J. Electr. Comput. Eng. Innovations, 12(1): 79-98, 2024.

DOI: 10.22061/jecei.2023.9737.651

URL: https://jecei.sru.ac.ir/article_1927.html