

Big data anonymization using Spark for enhanced privacy protection

Abdelmadjid Guessoum Graba^{1,2}, Adil Toumouh²

¹Evolutionary Engineering and Distributed Information Systems Laboratory, Djillali Liabes University, Sidi-Bel-Abbes, Algeria

²Communication Networks, Architectures and Multimedia Laboratory, Djillali Liabes University, Sidi-Bel-Abbes, Algeria

Article Info

Article history:

Received Jan 1, 2024

Revised Mar 20, 2024

Accepted Apr 1, 2024

Keywords:

Big data privacy

Data anonymization

Parallel computing

Privacy-preserving algorithms

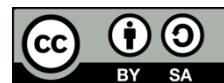
Resilient distributed datasets

Spark distributed computing

ABSTRACT

This article introduces an advanced solution for anonymizing large-scale sensitive data, addressing the limitations of traditional approaches when applied to vast datasets. By leveraging the Spark distributed computing framework, we propose a method that parallelizes the data anonymization process, enhancing efficiency and scalability. Utilizing Spark's resilient distributed datasets (RDD), the approach integrates two primary operations, *Map_RDD* and *ReduceByKey_RDD*, to execute the anonymization tasks. Our comprehensive experimental evaluation demonstrates our solution's effectiveness and improved performance in preserving data privacy while balancing data utility and confidentiality. A significant contribution of our study is the development of a wide array of solutions for data owners, particularly notable for a 500 MB dataset at an anonymity level of $K=100$, where our methodology produces 832 unique solutions. This study also opens avenues for future research in applying different privacy models within the Spark ecosystem, such as l-diversity and t-closeness.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Abdelmadjid Guessoum Graba

Evolutionary Engineering and Distributed Information Systems Laboratory, Djillali Liabes University

Sidi-Bel-Abbes, Algeria

Email: abdelmadjid.graba@dl.univ-sba.dz

1. INTRODUCTION

The emergence of big data, generated from mobile devices, sensor networks, social networks, and the internet of things (IoT), has increased data volumes and the potential for personal information leakage. As diverse sources contain identifiable information, there is a heightened risk to privacy. Given the increased risk to privacy from diverse sources containing identifiable information, it is critical to prevent personal data leaks by anonymizing sensitive information before publication.

In recent years, significant research has been conducted on preserving privacy during data publication to protect individuals while modifying data as little as possible [1], [2]. Different methods are used to transform the original data, depending on the selected privacy model, such as k-anonymity [3], l-diversity [4], and t-closeness [5]. K-anonymity is the most commonly used privacy model, which protects privacy by transforming data so that each set of quasi-identifier values appears at least k times in the table to be published [6], [7]. Additionally, l-diversity is a data de-identification method that allows sensitive information to be distributed in a dataset while ensuring k-anonymity [4]. t-closeness is a technique proposed to address the weaknesses of k-anonymity and l-diversity, which aims to avoid the clustering of sensitive information with similar values by adjusting the distribution of sensitive data around a specific value [5]. Today, several de-identification technologies prevent personal information leaks when publishing big data.

With the growing demand for publishing big data across various fields, the development of support systems has generated increasing interest [8]. In this context, our article presents an algorithm aimed at anonymizing large and confidential data to ensure secure publication. The algorithm employs a lattice where each node represents a potential generalization of the original table. The algorithm aims to identify the lowest level of the generalization lattice containing one or more solutions and all their generalizations. To aid in selecting the optimal solution from the stored nodes, we provide a metric of information loss (IL) to the data publisher. To accelerate the anonymization process on Big Data platforms, we have employed the distributed execution framework Spark.

2. PRELIMINARY NOTIONS

This section introduces the terminology, data model, and metric employed in this article to measure information loss. The terminology clarifies specific concepts for understanding the methods discussed, while the data model outlines the organization and handling of data critical for the analysis. The information loss metric is crucial for evaluating the impact of data anonymization techniques on data utility.

2.1. The k-anonymity model

Regarding confidentiality, the data publisher has a table that includes identifiers, quasi-identifiers, and sensitive and non-sensitive attributes. An original table T can be represented as T (*identifier, quasi-identifier, sensitive attribute, non-sensitive attribute*). In order to ensure a good understanding of these terms within the scope of this document, it is crucial to review their definitions.

- Definition 1 (Identifier): an identifier is specific information, like a first name, last name, or social security number, used to recognize or distinguish an individual. Identifiers should be excluded from published data.
- Definition 2 (Quasi-identifier): a quasi-identifier is an attribute that can indirectly disclose personal information when combined with external data sources. For instance, {Sex, zip code, date of birth} can uniquely identify individuals in large datasets. Like this set, Quasi-identifiers must be anonymized to protect privacy, often using the k-anonymity model.
- Definition 3 (Sensitive attribute): a sensitive attribute contains data that individuals tend to keep confidential or not disclose, such as medical or salary information.
- Definition 4 (non-sensitive attribute): a non-sensitive attribute is neither an explicit identifier nor a quasi-identifier and does not directly identify a person or reveal sensitive information. The k-anonymity model, proposed as the first privacy protection model, transforms quasi-identifier values into more general ones, forming equivalence classes. Each class comprises data with the same quasi-identifier value. According to the principle of k-anonymity, defined in [9]:
- Definition 5 (k-anonymity): a database table T with quasi-identifier attributes Q satisfies the k-anonymity criterion if every distinct tuple in the projection of T onto Q appears at least k times.

K-anonymity ensures that at least $k-1$ other records in the same equivalence class cannot be distinguished from a particular record [3], [6]. For instance, evaluating the 2-anonymity of Table 1 with 'Age,' 'Gender,' and 'Zip code' as quasi-identifier attributes, we find that each combination of values (equivalence classes) appears at least twice, meeting the criterion. Conversely, Table 2 fails the 2-anonymity criterion, as some equivalence classes contain only one record (the last in the table).

Table 1. Table that satisfies 2-anonymity

Age	Gender	Zip code	Disease
30-39	Male	1415*	Fever
30-39	Male	1415*	Asthma
60-69	Female	1417*	Back Pain
60-69	Female	1417*	Heart Attack
60-69	Female	1417*	Diabetes

Table 2. Table that does not satisfy 2-anonymity

Age	Gender	Zip code	Disease
30-39	Male	1415*	Fever
30-39	Male	1415*	Asthma
60-69	Female	1417*	Back Pain
60-69	Female	1417*	Heart Attack
70-79	Female	1417*	Diabetes

2.2. Generalization techniques

The goal of generalization is to strengthen k-anonymity. This technique involves partitioning data into groups based on their quasi-identifier value and then modifying the quasi-identifier values in each group to make them less specific. As a result, each individual can be confused with k-1 other individuals in the published table. There are two approaches to achieving k-anonymity: global generalization and local generalization [10], [11].

Global generalization is a data generalization technique that uses a generalization lattice for the original data. This lattice represents the possible combinations of predefined taxonomic trees for all quasi-identifier attributes. Each taxonomic tree is a hierarchical tree where the higher the level, the higher the level of attribute generalization, and the more data accuracy is lost. For example, Figure 1 shows the taxonomic trees of the Age, Gender, and Postal Code attributes and their generalization level (GL). In addition, Figure 2 illustrates the lattice associated with these same attributes. Each node of the lattice corresponds to a possible generalization of the original table, and the lower the level, the closer the generalized data is to the original data, while the higher the level, the greater the distortion of the original data. Therefore, the appropriate level of generalization must be selected based on the required level of anonymity. Global generalization is predicated on the following generalization property, as elucidated in Definition 6.

- Definition 6 (generalization property): Let E and F be two sets of attributes of a table T such that the attribute set of F is more general than E (E is more specific than F). If T is k-anonymous with respect to E , then T is also k-anonymous with respect to F . For example, if $\langle A1, G0, ZC2 \rangle$ satisfies k-anonymity, and if $\langle A1, G1, ZC2 \rangle$ and $\langle A2, G0, ZC2 \rangle$ are more general than $\langle A1, G0, ZC2 \rangle$, then $\langle A1, G1, ZC2 \rangle$ and $\langle A2, G0, ZC2 \rangle$ also satisfy k-anonymity. Conversely, if T is not k-anonymous with respect to F , then T is not k-anonymous with respect to E . For example, if $\langle A1, G0, ZC2 \rangle$ does not satisfy k-anonymity, and if $\langle A0, G0, ZC2 \rangle$ and $\langle A1, G0, ZC1 \rangle$ are more specific than $\langle A1, G0, ZC2 \rangle$, then $\langle A0, G0, ZC2 \rangle$ and s also do not satisfy k-anonymity.

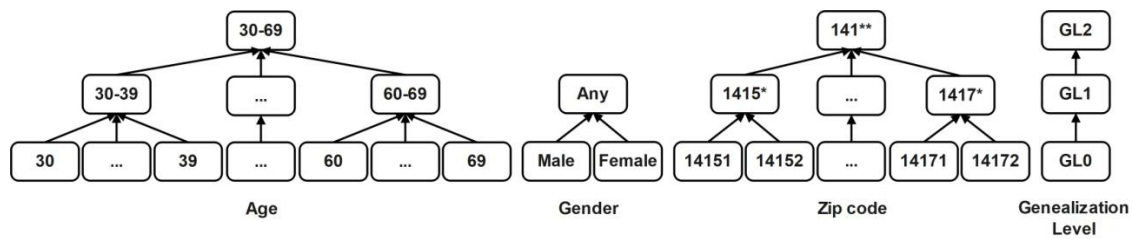


Figure 1. Taxonomic trees of the attributes Age, Gender, and Zip Code

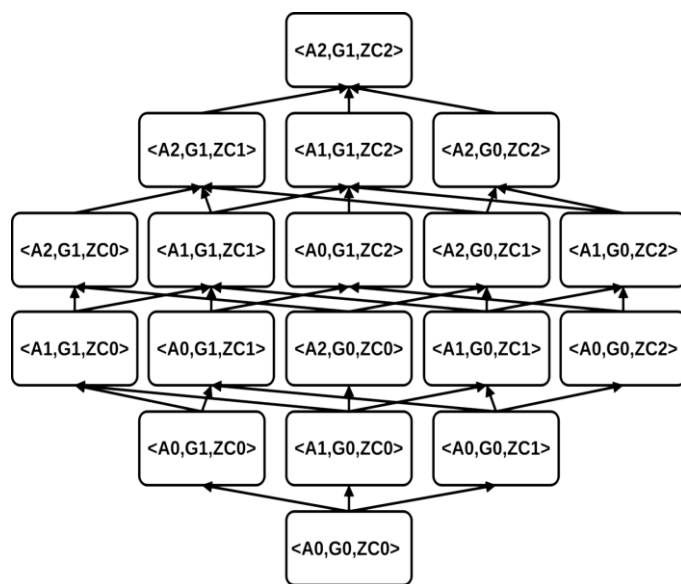


Figure 2. Generalization lattice of the three attributes Age, Gender, and Zip Code

On the other hand, local generalization employs clustering techniques to group records, forming equivalence classes containing at least k records [7]. Here, data confidentiality is maintained by substituting original values with cluster centroids. The level of anonymization under local generalization may vary based on user-defined cluster characteristics. However, the process of preserving confidentiality leads to a loss of information. It is essential to minimize this loss to ensure that relevant information can still be extracted from the published data. The following section will explore the appropriate information loss metric for this objective.

2.3. Information loss metric

Ensuring data anonymization while minimizing information loss resulting from modification is one of the generalized data that differs from the original data, which results in information loss. The greater the information loss, the more significant the difference between the original and generalized data. Additionally, extensive generalization of attributes leads to decreased data utilization. Information loss values can be used to determine the extent to which anonymous data has been transformed compared to the original data through generalization techniques, enabling evaluation of its usefulness. Assuming that equivalent class e comprises m attributes, information loss can be defined as (1) [7].

$$IL(e) = |e| \cdot \sum_{i=1, \dots, m} \frac{H(\Lambda(U_{C_i}))}{H(\text{Tr}_{C_i})} \quad (1)$$

where $|e|$: the number of records in the equivalence class e ; $\Lambda(U_{C_i})$: the subtree rooted at the lowest common ancestor of each value in the union of categorical attributes C_1, \dots, C_n ; and $H(\text{Tr})$: the height of the taxonomy tree Tr , i.e., the distance between the root and the farthest leaf in the tree. Suppose that equation is the set of all equivalence classes in a generalized table T' . In this case, the total value of information loss for the generalized table can be calculated using the following formula [7].

$$\text{Total_IL}(T') = \sum_{e \in E_q} IL(e) \quad (2)$$

3. RELATED WORKS

In order to avoid negative consequences related to the disclosure of sensitive personal data, it is essential to preserve their confidentiality when publishing a dataset. Adequate data anonymization is necessary to accomplish this goal. This section examines the most relevant anonymization approaches to the solution we developed with Apache Spark.

The researchers explored several techniques to optimize the generalization process, including top-down specialization (TDS). This approach iterates through taxonomic trees from the root to the leaves, selecting generalizations that best preserve the classification quality while ensuring the required level of k -anonymity [12]. TDS effectively balances the trade-off between data utility and privacy by systematically evaluating each possible generalization, demonstrating its practical value in maintaining robust data protection.

An implementation of the TDS algorithm for Apache Spark was presented in [13], where the dataset is partitioned into p partitions on n Spark nodes, with n equal to p , using this approach. In this setup, the data is partitioned, and scores required for the TDS algorithm, such as privacy loss and information gain, are computed by the master node. Experimentation with this implementation on the "adult" dataset revealed that significant performance improvements can be obtained by increasing the number of Spark nodes in proportion to the dataset size, regardless of k and dataset size values. Another study in [14] proposed a multidimensional sensitivity algorithm based on Apache Spark. This algorithm uses pre-determined quasi-identifiers for anonymization and a pre-calculated k -value using linear regression. This approach considers each quasi-identifier's probability and prioritizes anonymizing attributes with higher probability. Additionally, an RDD-based method was proposed to reduce data transmission between memory and disk.

Furthermore, SparkDA, presented in [15], is a technique based on RDDs and uses critical operations FlatMapRDD and ReduceByKeyRDD to perform data anonymization in memory. In [16], a generic framework is proposed for implementing subtree-based generalization. This approach includes three phases that require the output of the previous phase as input. The first phase ensures an equitable workload distribution for each partition and avoids duplications. The second computes privacy and utility scores for each attribute, and the third verifies that the generalized dataset satisfies k -anonymity requirements.

Furthermore, Ashkouti *et al.* [17] proposes a distributed in-memory method to preserve the ℓ -diversity privacy model on Spark in three phases. Two distance functions were designed to meet the requirements of the ℓ -diversity model. Vimercati *et al.* [18] using the efficient Mondrian approach, a solution

is proposed to apply κ -anonymity and ℓ -diversity in a distributed manner on large datasets. This data partitioning limits exchanges between workers, allowing each worker to anonymize a portion of the data independently. Finally, Bazai *et al.* [19] proposes a hybrid approach for efficient and scalable multidimensional data anonymization. This approach allows for fewer RDDs and smaller partitions than existing approaches, thereby reducing re-computation, shuffle, and cache management costs.

In another field of data anonymization using differential privacy [20], techniques have been proposed to anonymize the k-means clustering algorithm on the Spark platform [21], [22]. These techniques rely on a new partitioning mechanism optimized for the dynamic allocation of data sets, allowing for fast processing. The authors describe a formal proof of confidentiality that meets the requirement of ϵ -differential privacy. In a related method, Yin and Liu [23] used the Map-Reduce model to govern the parallel distribution of k-means clustering and adopted the Laplace method to ensure differential privacy protection.

In this article, we adopt an anonymization-based approach to protect data privacy using Spark. Unlike approaches based on differential privacy, which focus mainly on theoretical foundations, anonymization offers a practical and easy-to-implement solution for businesses and organizations. Additionally, anonymization techniques can be integrated seamlessly into existing data analysis processes without disrupting analysis results or reducing data utility. Overall, anonymization protects data privacy while preserving data utility for analysis and research.

4. PROPOSED APPROACH WITH APACHE SPARK RDD

This section details our approach to guide data owners through the anonymization process using two key RDDs: *Map_RDD* and *ReduceByKey_RDD*. *Map_RDD* distributes tasks across nodes for parallel processing, while *ReduceByKey_RDD* aggregates the results to enhance data anonymization efficiency. Additionally, we introduce the symbols and notations in Table 3, which are crucial for understanding our methodology.

The Spark framework, specifically designed for large-scale data processing, was chosen to optimize processing time and ensure system scalability. This decision facilitated data division into several blocks, allowing for distribution across multiple cluster nodes to enhance processing efficiency. Within this framework, two key resilient distributed dataset (RDD) transformations, *Map_RDD* and *ReduceByKey_RDD*, were applied to each block. These transformations are critical for efficiently generalizing and processing data in a distributed environment, making them indispensable for handling complex data anonymization processes. By utilizing these RDD transformations, the system can parallelize tasks, thereby reducing the overall processing time while maintaining high data integrity and privacy levels.

Table 3. Symbols and notations

Symbol	Definition
PT	A private table that needs to be anonymized
K	Defines the level of K-anonymity
nodes	Contains all node(s) at level h of generalization lattice
node	A single node from nodes
qid	A quasi-identifier attribute
sa	A sensitive attribute
<i>Input_RDD</i>	Defines the RDD created by applying the textFile transformation to PT.
<i>Record(i)</i>	Defines an element of <i>Input_RDD</i> , $Record = \{qid_1, qid_2, \dots, qid_n, sa\}$
<i>qituple</i>	Contains all <i>qid</i> within a record, $qituple = \{qid_1, qid_2, \dots, qid_n\}$
<i>EqC</i>	Indicates equivalence class
<i>Generalized_RDD</i>	Contains a set of (EqC, sa)
SA	Contains a set of <i>sa</i> associated with an EqC
Result	Contains a set of (EqC, SA)
listOfkAnon	List of nodes from generalization lattice that satisfy k-anonymity

4.1. Map_RDD

The *Map_RDD* transformation, algorithm 1, begins by reading the blocks of the original table and nodes of the generalization lattice of height h. For each record in each block, the *Map_RDD* transformation replaces each *qid* attribute in the *qituple* with the value of its direct parent according to the corresponding node of the generalization lattice. Once the *qid* attributes have been generalized, each record in each block is associated with an equivalence class based on the generalized values of the *qituple* attributes. Records sharing the same generalized values for all attributes of the *qituple* are grouped into the same *EqC*. Finally, each record converted into an *EqC*, associated with its sensitive attribute (*sa*), is passed to the *ReduceByKey_RDD* function as a pair (key: *EqC*, value: *sa*).

Algorithm 1. Map_RDD

```

Input: Input_RDD, node
Output: Generalized_RDD
1 begin
2   for i in Size(Input_RDD) do
3     line = Record(i);
4     qidtuple = line.get(qidtuple);
5     sa = line.get(sa);
6     EqC = Generalization(qidtuple, node);
7     Generalized_RDD += (EqC, sa);
8   end
9   return Generalized_RDD;
10 end

```

4.2. ReduceByKey_RDD

The *ReduceByKey_RDD* transformation, Algorithm 2, collects all the values associated with an *EqC* key. In other words, it allows gathering sensitive data from the same equivalence class and grouping it into a single list. This reduction operation merges all the intermediate results associated with the same key, and only one value is returned for each key. The reduction technique simplifies the data structure by aggregating the information related to each key and reducing the amount of data that needs to be processed.

Algorithm 2. ReduceByKey_RDD

```

Input: Generalized_RDD
Output: Result = {(EqC, SA)}
1 begin
2   foreach EqC ∈ Generalized_RDD do
3     SA += sa;
4   end
5   return Result;
6 end

```

4.3. The main function

Our data anonymization solution is centered around the main function, as outlined in Algorithm 3. This function orchestrates the entire anonymization process, managing data inputs and outputs and facilitating the execution of essential functions. Among these are the two transformation functions described earlier, which are critical for effectively modifying the data while ensuring privacy.

The main function takes as input the original private table *PT*, the value *K*, and the generalization hierarchy for each *qid*. It generates the generalization lattice from the taxonomy trees (step 2), reads the original data HDFS file, and saves it to an *Input_RDD* (step 3). The Main function caches *Input_RDD* to improve performance because it is used in many subsequent operations.

Step 12 of our algorithm involves transferring all unmarked nodes located at the average height of the lattice (where the average height *h* is calculated as $h = (low + high)/2$) to the *Map_RDD* and *ReduceByKey_RDD* transformation functions. By doing so, we enable parallel processing of different generalization steps, significantly speeding up the anonymization process, especially when dealing with significant data sources. The algorithm works iteratively to find optimal nodes, starting by considering the entire lattice in the first iteration. Each node corresponds to a possible generalization of the original table.

At step 13, for each node at the same level in the generalization lattice (i.e., nodes located at height *h*), we check if the number of sensitive attributes for each obtained equivalence class is greater than or equal to *K*. If the number of sensitive attributes for at least one equivalence class is less than *K*, the node and all its direct and indirect specializations are removed from the lattice because they do not satisfy *k*-anonymity (Definition 6). On the other hand, if the number of sensitive attributes for all equivalence classes is greater than or equal to *K*, the node and all its direct and indirect generalizations are marked because they also satisfy *k*-anonymity (Definition 6). If, at this height, at least one node satisfies *k*-anonymity, the lower half of the lattice ($high = h$) is defined as a new search area (step 22). However, if no node satisfies *k*-anonymity, the algorithm searches for acceptable nodes in the upper half of the lattice ($low = h + 1$) in the next iteration. The goal is to find the lowest level in the lattice where one or more solutions and all their generalizations are located (step 30). The algorithm stops when $h = 0$ and returns the list of stored nodes to the user. Once the list of stored nodes is available, the data publisher can choose the best solutions based on the information loss metric *IL*, which we calculate for each retained node.

Algorithm 3. Main

```

Input: PT, K, Taxonomy Trees (one for each qid)
Output: listOfkAnon
1 begin
2   Construct_Generalization_Lattice(Taxonomy Trees);

```

```

3  Input_RDD = textFile(PT).cache;
4  low =0; high = height_of_lattice;
5  sol = ∅;
6  while low < high do
7    h = (low + high)/2;
8    nodes = {node | height(node) = h};
9    k_check = false;
10   foreach node ∈ nodes do
11     if node.isMarked() == false then
12       Result_RDD = Input_RDD .Map_RDD(Input_RDD,node)
13       ReduceByKey_RDD(Generalized_RDD);
14       if Check_k_node(Result_RDD) == true then
15         k_check = true;
16         node.markAllGeneralizations(true);
17       else
18         nodes.remove(node);
19         Lattice.remove(node.getAllSpecializations());
20     end
21   end
22   if k_check == true then
23     Solution = nodes;
24     high = h;
25   else
26     low = h+1;
27   end
28 end
29 foreach nod ∈ Solution do
30   listOfkAnon.add(node.getAllGeneralizations());
31 end
32 return listOfkAnon;
33 end
    
```

5. METHOD

Figure 3 provides a detailed visual of our Spark-based anonymization method, depicting the workflow from raw data to anonymized output. Leveraging Apache Spark and RDDs, our approach ensures privacy in large-scale datasets. It showcases the execution cycle, beginning with data input and progressing through sequential and parallel processing steps via *Map_RDD* transformations for generalizing quasi-identifiers and *ReduceByKey_RDD* for aggregating them into equivalence classes. This process upholds dataset confidentiality, meeting k-anonymity standards.

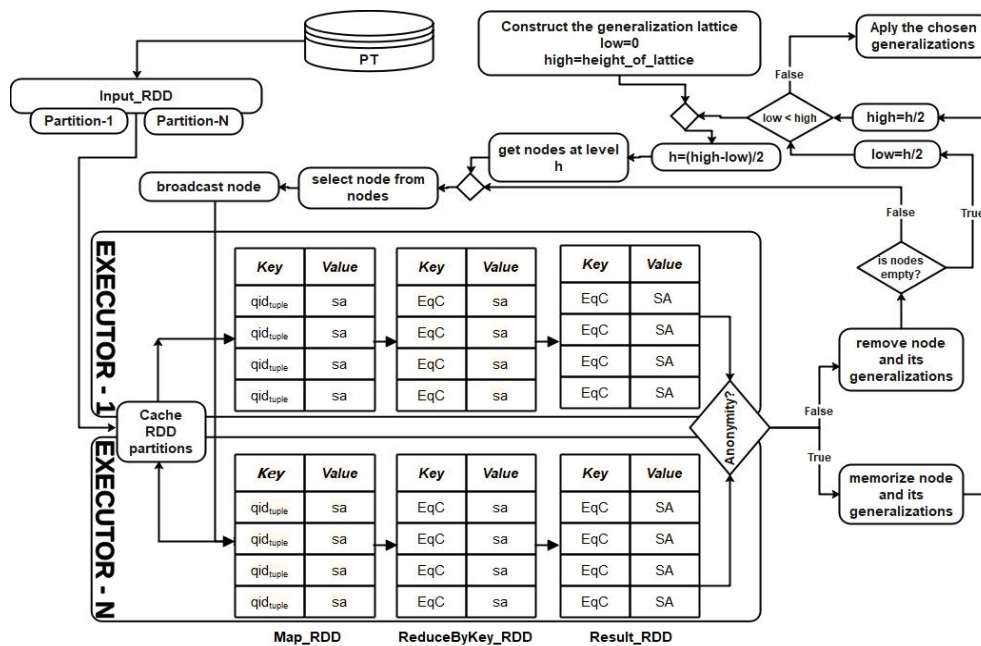


Figure 3. Workflow of the proposed solution

5.1. Data preparation

In our study, we leverage the foundational work on taxonomic trees established by Zhang *et al.* [24], incorporating their structured approach as a cornerstone in our analysis. We apply this methodology to the Adult dataset [25], a widely recognized data privacy benchmark that serves as the empirical basis for our investigation. This integration of established taxonomic structures with robust datasets allows us to evaluate the effectiveness of our anonymization processes thoroughly.

Table 4 elucidates the hierarchy and diversity of quasi-identifiers extracted from these taxonomic trees, offering a nuanced perspective on the complexity of the data. These quasi-identifiers range from 'Workclass' to 'Native_country', each assigned a designated generalization level (GL) that reflects the degree of abstraction applied to preserve individual privacy while retaining the utility of the data for analysis.

Table 4. Adult dataset

qid	GL	Distinct value
Workclass	4	7
Education	4	16
Marital_status	3	7
Occupation	2	14
Relationship	2	6
Race	2	5
Gender	1	2
Native_country	3	41

In order to assess the scalability of our model, we augmented the Adult dataset by a factor of 5, 25, 50, and 100, resulting in datasets of 25, 125, 250, and 500 MB, respectively. To accomplish this, we developed a program that inserted rows with random values drawn from the unique values list for each column of the original dataset. We selected eight categorical attributes, such as Workclass, Education, Marital status, Occupation, Relationship, Race, Gender, and Native country, as quasi-identifiers.

5.2. Experimental environment

The experiments were conducted on a 20-vCPU cluster computer (10 CPUs x 2 vCPUs per CPU) with 64 GB of RAM. To enhance the computational environment for our experiments, we employed Docker technology, leveraging its containerization capabilities to create a highly flexible and controlled setup. We installed Apache Spark 2.1 using Docker images, ensuring a standardized and replicable environment across all tests. This approach allowed us to systematically vary the number of Docker nodes in our experiments, with configurations set at 1, 2, 4, 8, and 16 nodes. Such scalability was crucial for assessing our algorithm's performance under different computational loads. Each Docker container was meticulously allocated with 1 GB of RAM and one virtual CPU, clocked at 3.6 GHz, to simulate a distributed computing environment that closely mimics real-world data processing scenarios.

To rigorously evaluate the algorithm's efficiency, three critical parameters were taken into account: the number of worker nodes and partitions, which determine the algorithm's parallel processing capability; the dataset size, which tests the algorithm's ability to handle varying volumes of data; and the degree of anonymity K , which assesses the algorithm's effectiveness in preserving privacy. The primary metric for assessing efficiency was the execution time, meticulously measured in milliseconds. This comprehensive approach provided insights into the algorithm's performance under different configurations. It highlighted its scalability and efficiency in processing and anonymizing data, offering a nuanced understanding of its practical applicability in real-world scenarios.

6. RESULTS AND DISCUSSION

In delving into the privacy challenges within big data, our study advances beyond traditional analyses focused on node-to-partition ratios. We investigate the scalability of our solution itself, presenting a singular, comprehensive approach that benefits data stewards directly. Our research meticulously assesses how node-to-partition configurations impact scalability and introduces innovative measures to enhance data privacy. This approach addresses the scalability concerns head-on and offers a broad spectrum of solutions specifically designed for data custodians, enabling them to manage and secure large datasets more effectively. By doing so, our work extends the operational capabilities available to those in charge of data, providing a singular, scalable strategy for optimizing data processing and prioritizing privacy in distributed computing environments.

Our efficiency analysis, detailed in Figures 4 to 6, conducted with datasets of 25, 125, and 250 MB, meticulously examines the impact of partition counts on processing efficiency within the scope of data anonymization. By exploring configurations using 1, 2, 4, and 8 nodes with varying degrees of data anonymity (k-values of 10, 50, and 100), we highlight a crucial insight: optimal system performance requires a strategic balance between the number of partitions and computational nodes. Achieving this balance maximizes computational resource utilization, mitigating inefficiencies stemming from resource underutilization or system overload and enhancing the efficiency of data anonymization processes.

Delving further into system scalability with a 500 MB data sample, as depicted in Figure 7, our experiments reveal the system's adeptness at managing escalating data volumes. The methodology underscores the system's robustness and flexibility, notably by aligning each partition with a singular worker node and progressively augmenting both. This approach significantly reduces execution times for k-values of 10, 50, and 100, showcasing the critical role of scalability in effective data anonymization practices. These results, achieved with a fixed number of nodes and partitions, are primarily due to the consistent number of iterations for different k-values.

A significant contribution of our study is the development of a wide array of solutions designed for data custodians, especially notable for a 500 MB dataset at an anonymity level of $K=100$, where our methodology produces 832 unique solutions. This abundance of options marks a considerable improvement over traditional methods, showcasing our strategy's effectiveness in offering precise solutions that meet the complex demands of data privacy and processing efficiency.

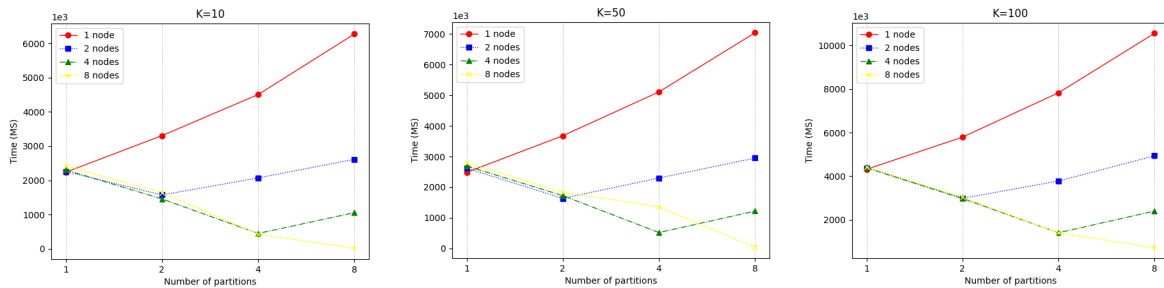


Figure 4. Efficiency for 25 MB dataset

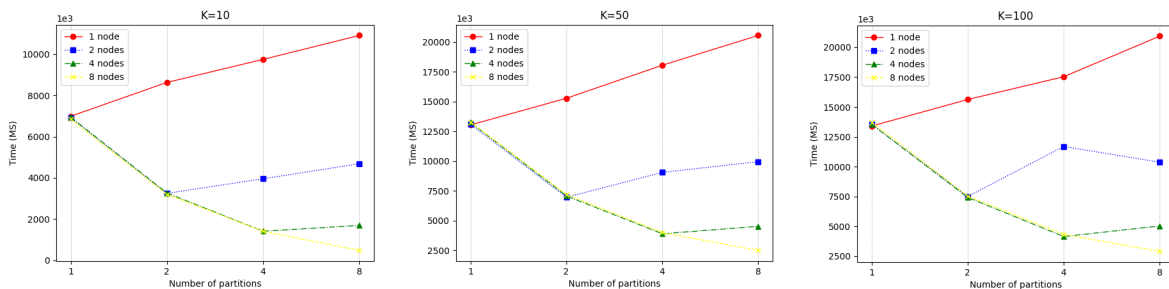


Figure 5. Efficiency for 125 MB dataset

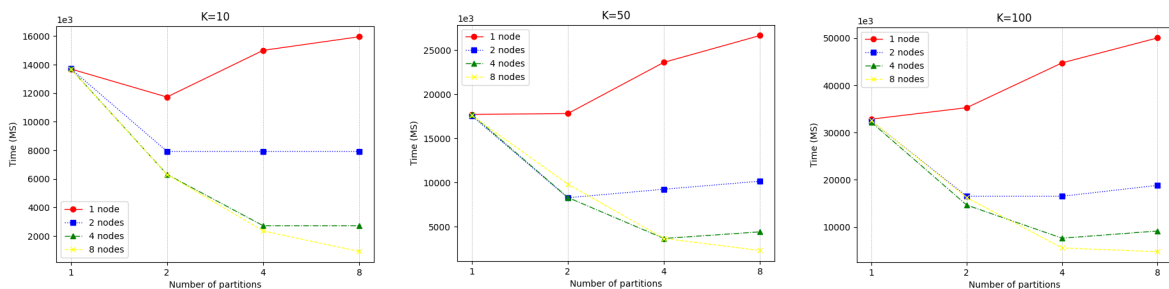


Figure 6. Efficiency for 250 MB dataset

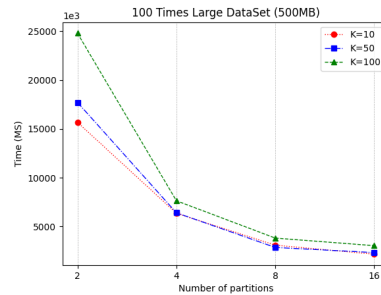


Figure 7. Scalability on 500 MB dataset

Acknowledging the limitations of our study, such as potential network overhead and varied performance in different execution environments, we recognize that the observed longer execution times compared to prior research may be attributed to increased bus communication traffic with the expansion of dataset sizes and core counts. Spark documentation underscores that local execution is primarily for testing purposes, with specific optimizations accessible only in cluster environments. Hence, we advocate for future research to focus on the practical application and performance of our anonymization solutions in diverse computational settings, particularly emphasizing deployments in real cluster environments to leverage performance enhancements fully and assess scalability and efficiency in enhancing privacy comprehensively.

Thus, our research transcends conventional node-to-partition ratio analyses by scrutinizing our solution's scalability and offering a unified approach that directly benefits data stewards. We meticulously evaluate how configurations impact scalability and devise innovative strategies to bolster data privacy. This methodology not only confronts scalability issues but also provides a comprehensive suite of solutions for data owners, enhancing their ability to secure large datasets effectively. Our findings reveal key insights into achieving optimal system performance through a strategic balance of partitions and nodes, significantly improving data anonymization efficiency.

7. CONCLUSION

This article describes a system that enables data owners to safely publish large amounts of data using the k-anonymization technique to ensure personal data privacy. We have developed a proposal that relies on two RDD transformations, namely *Map_RDD* and *ReduceByKey_RDD*. These transformations offer several advantages, including more efficient partition management, memory usage optimization through a cache to store frequently referenced values, and improved iteration processes. The latter point is crucial for our algorithm, which relies on intensive iterations.

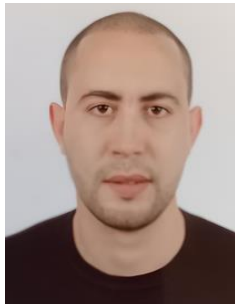
In our upcoming research, we aim to enhance our system by expanding its capability to support various privacy models, such as l-diversity and t-closeness, in addition to the currently supported k-anonymity. This development will enhance the security and protection of personal data while improving their utility. Furthermore, we plan to accelerate the data anonymization process in Apache Spark by leveraging the processing power of GPUs. We intend to use Spark-Rapids, an open-source library that optimizes Spark performance by utilizing GPUs to achieve this. This approach will allow us to process larger data volumes and obtain results faster than Spark alone.





REFERENCES

- [1] J. Kim, K. Jung, H. Lee, S. Kim, J. W. Kim, and Y. D. Chung, "Models for privacy-preserving data publishing: a survey," *Journal of KIISE*, vol. 44, no. 2, pp. 195–207, Feb. 2017, doi: 10.5626/JOK.2017.44.2.195.
- [2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing," *ACM Computing Surveys*, vol. 42, no. 4, pp. 1–53, Jun. 2010, doi: 10.1145/1749603.1749605.
- [3] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, Oct. 2002, doi: 10.1142/S0218488502001648.
- [4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian, "L-diversity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, Mar. 2007, doi: 10.1145/1217299.1217302.
- [5] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, Apr. 2007, pp. 106–115, doi: 10.1109/ICDE.2007.367856.
- [6] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, Jun. 2005, pp. 49–60, doi: 10.1145/1066157.1066164.
- [7] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," in *international conference on database systems for advanced applications*, 2007, pp. 188–200.
- [8] C. Dai, G. Ghinita, E. Bertino, J.-W. Byun, and N. Li, "TIAMAT," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1618–1621, Aug. 2009, doi: 10.14778/1687553.1687607.





- [9] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," SRI International, 1998.
- [10] S. Kim, H. Lee, and Y. D. Chung, "Privacy-preserving data cube for electronic medical records: an experimental evaluation," *International Journal of Medical Informatics*, vol. 97, pp. 33–42, Jan. 2017, doi: 10.1016/j.ijmedinf.2016.09.008.
- [11] D.-H. Kim and J. W. Kim, "A study on performing join queries over k-anonymous tables," *Journal of The Korea Society of Computer and Information*, vol. 22, no. 7, pp. 55–62, 2017.
- [12] B. C. M. Fung, Ke Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *21st International Conference on Data Engineering (ICDE'05)*, pp. 205–216, doi: 10.1109/ICDE.2005.143.
- [13] U. Sopaoglu and O. Abul, "A top-down k-anonymization implementation for Apache Spark," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 4513–4521, doi: 10.1109/BigData.2017.8258492.
- [14] M. Al-Zobbi, S. Shahrestani, and C. Ruan, "Experimenting sensitivity-based anonymization framework in Apache Spark," *Journal of Big Data*, vol. 5, no. 1, Dec. 2018, doi: 10.1186/s40537-018-0149-0.
- [15] S. U. Bazai and J. Jang-Jaccard, "In-memory data anonymization using scalable and high performance RDD design," *Electronics*, vol. 9, no. 10, Oct. 2020, doi: 10.3390/electronics9101732.
- [16] S. U. Bazai, J. Jang-Jaccard, and H. Alavizadeh, "Scalable, high-performance, and generalized subtree data anonymization approach for Apache Spark," *Electronics*, vol. 10, no. 5, Mar. 2021, doi: 10.3390/electronics10050589.
- [17] F. Ashkouti, K. Khamforoosh, A. Sheikahmadi, and H. Khamfroush, "DHkmeans- ℓ diversity: distributed hierarchical K-means for satisfaction of the ℓ -diversity privacy model using Apache Spark," *The Journal of Supercomputing*, vol. 78, no. 2, pp. 2616–2650, Feb. 2022, doi: 10.1007/s11227-021-03958-3.
- [18] S. De Capitani di Vimercati *et al.*, "Scalable distributed data anonymization for large datasets," *IEEE Transactions on Big Data*, vol. 9, no. 3, pp. 818–831, Jun. 2023, doi: 10.1109/TBDDATA.2022.3207521.
- [19] S. U. Bazai, J. Jang-Jaccard, and H. Alavizadeh, "A novel hybrid approach for multi-dimensional data anonymization for Apache Spark," *ACM Transactions on Privacy and Security*, vol. 25, no. 1, pp. 1–25, Feb. 2022, doi: 10.1145/3484945.
- [20] C. Dwork, *Differential privacy*. in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006*, Venice, Italy, Proceedings, Part II 33, pp. 1–12, 2006, doi: 10.1007/117870062006.
- [21] Z.-Q. Gao and L.-J. Zhang, "DPHKMS: an efficient hybrid clustering preserving differential privacy in Spark," in *Advances in Internetworking, Data & Web Technologies: The 5th International Conference on Emerging Internetworking, Data & Web Technologies (EIDWT-2017)*, 2018, pp. 367–377, doi: 10.1007/978-3-319-59463-7_37.
- [22] Z. Gao, Y. Sun, X. Cui, Y. Wang, Y. Duan, and X. A. Wang, "Privacy-preserving hybrid k-means," *International Journal of Data Warehousing and Mining*, vol. 14, no. 2, pp. 1–17, Apr. 2018, doi: 10.4018/IJDWM.2018040101.
- [23] S. Yin and J. Liu, "A k-means approach for map-reduce model and social network privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1215–1221, 2016.
- [24] X. Zhang, L. T. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 363–373, Feb. 2014, doi: 10.1109/TPDS.2013.48.
- [25] A. Asuncion and D. Newman, *UCI machine learning repository*. Irvine, CA, USA, 2007.

BIOGRAPHIES OF AUTHORS



Abdelmadjid Guessoum Graba     received the engineering degree in computer science from the University of Sidi Bel Abbes, Algeria, in 2006, and the master's degree in computer science in 2010 from the Department of Computer Science, University of Sidi Bel Abbes, Algeria. Currently, he is a lecturer in the Telecommunications Department at the Faculty of Electrical Engineering, University of Sidi Bel Abbes, Algeria. His research interests include data mining, big data, artificial intelligence techniques, and networks. He can be contacted at email: abdelmadjid.graba@dl.univ-sba.dz.



Adil Toumouh     received an engineering degree in computer science, a master's degree in computer science, and a doctorate degree from the Department of Computer Science at the University of Sidi Bel Abbes, Algeria. Currently, he is a lecturer in the Computer Science Department at the Faculty of Exact Sciences, University of Sidi Bel Abbes, Algeria. His research interests include artificial intelligence, knowledge engineering, text mining, and natural language processing (NLP). He can be contacted toumouh@gmail.com.