International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228312

Privacy Preserving Parallel Distributed Data Stream Anonymization

Brinit Trivedi[1], Sheshang Degadwala[2], Dhairya Vyas[3]
[1]Research Student, Department of Computer Engineering, Sigma Institute of Engineering, Vadodara, Gujarat, India
[2]Associate Professor, Department of Computer Engineering, Sigma Institute of Engineering, Vadodara, Gujarat, India
[4]Managing Director, Shree Drashti Infotech LLP, Vadodara, Gujarat, India

Article Info
Volume 8, Issue 3
Page Number 53-66

Publication Issue :
May-June-2022

Article History
Accepted: 01 May 2022
Published: 08 May 2022

ABSTRACT

Sustainable stream processing algorithms have gained popularity in recent years. Flow control is a way of searching and modifying real-time data streams. Missing values are ubiquitous in real-world data streams, making data stream privacy challenging to safeguard. On the other hand, most privacy preservation methods need not take absent values into account when developed. They can anonymize data in certain study, however this results in data loss. This research proposes a unique parallel distributed approach for protecting privacy while using incomplete data streams. This method uses a production computational system to continually anonymize data streams, using clustering to construct each tuple. It clusters data in partial and complete forms using variable and array dimensions as similarity metrics. In order to prevent values and outliers' pollution, a generalization approach that is based on more than matches is used. The experiments used real data to compare current systems with varied settings. This research will cover several anonymization mechanisms and their advantages. There are also drawbacks. Finally, we will explore the future of continuous data anonymization research.

Keywords — Data Stream, Anonymization, K-Anonymity, Generalization, L-Diversity, T-Closeness

I. INTRODUCTION

Recently, "personal data privacy has risen to become a prominent concern in the field of data security research. PPD is founded on the principle of data modification in order to facilitate the use of data mining techniques without jeopardising the quantity of sensitive data available for analysis. However, unwanted susceptibility of sensitive information may arise throughout the data collection, publication, and communication (i.e., delivery of data mining conclusions) phases of the data mining process."

Data mining "is a technique for obtaining highly sensitive information. The processing power of

intelligent algorithms, on the other hand, puts vital and secret data stored in massive and dispersed data storage at danger. Large volumes of information data, such as criminal records, purchase histories, credit and healthcare histories, and driving records, may now be obtained and analysed in real time. This information is critical in a variety of fields, including scientific research, law enforcement, and national security. Anonymity is a term used to describe the right to maintain control over one's personal data."

Privacy "preservation difficulties are able to resolve this issue by safeguarding identifying and preventing sensitive information leakage while simultaneously delivering legitimate data for public usage. These tactics are designed to expose data in as much detail as possible while keeping the data's identity hidden. The greater the amount of information included within this publicly available data, the greater the likelihood of data breaches, which might expose protected persons and sensitive information. To put it another way, it will be more difficult to ensure that individuals cannot be recognised and that sensitive information is not revealed."

II. RELATED WORKS

In [1] Lu Yang et. all in this approach, a slide-window-based computational framework is offered to continuously send encrypted data streams, wherein every tuple may be created with clustering either anonymized cluster, depending on the situation. In order to cluster data across partial and complete datasets, they employ variable and group dimensions as similarity criteria. This enables for the creation of clusters with little leakage. They provide a generalization strategy based on the possibility of a match for generalizing partial data in order to reduce the number of missing values. Experiments on real-world datasets have also shown that proposed approach is capable of effectively anonymizing incomplete data streams while maintaining their relevance.

In [2] Sabrina De Capitani di Vimercati et. All of them offered a method for allowing a distributed anonymization procedure over big datasets of sensor data. Their method uses an indefinite number of workers inside the Spark framework to anonymize big datasets (which may not fit in main memory). They show how to parallelize the anonymization procedure by splitting the dataset properly. Their experiments indicate that the suggested method is scalable and does not degrade the anonymized dataset's quality.

In [3] Pelin Canbay and colleagues They've all made advantage of anonymization, which combines the characteristics of incognito and clustering in one package. Anonymity indicates that only registered persons are able to identify records that are associated with a certain individual. Anonymity is the most often used method in privacy protection systems. When it comes to identity exposure, the aim is to protect datasets from being compromised by an adversary who connects to a certain kind of data point and obtains sensitive information about the person connected. An investigation on the difference in variance amongst basic and clustered data was conducted inside this system in terms of anonymization.

They have utilized encryption with the characteristics of data security in [4] to protect their information. Mohamed Nassar and colleagues, they go through encryption algorithm and demonstrate how to efficiently create Paillier's addition homomorphic encryption in the process. This document contains an examination of the encryption mechanism (Partial homomorphic encryption and fully homomorphic encryption). Partially key cryptographic systems are simpler to use since they only enable one kind of decryption to occur. They are far more practicable and may be utilized for a number of functions, such as secure vote and collision resistance, as well as for other applications. FHE (Fully Homomorphic Encryption) is somewhat of a cryptography for both

addition and multiplication to be performed simultaneously. With FHE, it is possible to develop computers that can be operated on encryption inputs to generate encrypted outputs, which is a powerful tool for security professionals.

In [5] Mohammad-Reza Zare-Mirakabad et al., they employed K-anonymization with the characteristics of time series, encryption techniques, and N-grams, among other things. K-anonymization may be used to anonymize a wide range of data types, including census information, social network linkage information, gene expression data, including medical data records, to name just a few examples. In this study, the authors propose a novel private information framework for disseminating N-gram time series data over the internet.

Tsubasa"Takahashi and colleagues [6] make use of Complex data, which comprises both single-valued and set-valued features, allows us to correlate attribute values and evaluate correlations by correlating attribute values and examining correlations. Additionally, they proposed a top-down itemset recoding approach for difficult data, which converts item sets into generalized itemset in order to ensure k-anonymity, in addition to the traditional top-down recoding method. Anonymization is applied equally to complex data that contains both single-valued and set-valued attributes."

They "combine the characteristics of t-closeness with those of k-anonymity and l-diversity in their work. [7] Ninghui Li and colleagues Because of k-anonymity, it is not necessary to reveal one's identity. To address this problem, L-diversity requires that each sensitive attribute in each intermediate node have at least two well-represented values, which is known as "well-represented values." They propose a novel privacy notion known as t-closeness, which stipulates that the distribution of a sensitive attribute in any equivalence class must be close to the distribution of the sensitive attribute in the overall table in order for the attribute to be considered private (i.e., the distance between the two distributions should be no more than a threshold t). They use the Planetary Mover Dissimilarity measure to determine whether or not we are t-close enough."

In [8] Ahmed Ali Mubark and colleagues have employed semantic anonymization to protect their identities by using the l-diversity characteristic. Using domain-based semantic criteria, they demonstrate how to preserve categorical data in order to prevent similarity attacks on the data set in question.

The K-anonymity approach, as described by R. Mahesh et al. in [9], protects individuals' privacy against identity revelation attacks alone. This solution, however, is vulnerable to an attribute disclosure attack. The k-anonymity limit is reached using the l-diversity technique. However, in certain cases, it may not provide enough protection against identity and attribute disclosure attacks. The t-closeness approach is more efficient than k-anonymity and l-diversity. However, Computation is more complicated than other suggested approaches. The authors provide a novel strategy for protecting people' sensitive information against attribute and identity disclosure attacks in this study. The suggested technique achieves complete privacy protection by generalising quasi-identifiers and establishing range values.

Rashmi B. Ghate et al., [10] It is required to provide personal information when enrolling on a social networking site; some of this information is sensitive and must be kept private. Anonymization is used to protect a user's privacy on a social networking site. Individuals' personal information is either masked or removed from the dataset in the anonymization process, resulting in anonymous data. When a dataset is released, it is critical to protect data against unintended exposure and to strike a balance between the information's use and privacy. The proposed work provides an anonymized view of a data collection as well as the outcome of a single pass k-means Anonymization algorithm implementation. The

dataset is anonymized using generalization and suppression techniques.

The effectiveness of privacy-preserving data mining algorithms is judged in terms of performance, data utility, amount of ambiguity or resistance to data mining algorithms, and so on in [11] M. B. Malik et al. However, there is no privacy-preserving method that beats all others on every criterion. Rather, one algorithm could outperform another on a single criterion. As a result, the purpose of this work is to explain the current state of privacy-preserving data mining tools and approaches while also suggesting potential future research areas.

M.-J. Choi et al. address the dilemma of maintaining mining accuracy as well as privacy while releasing sensitive time-series data in [12] M.-J. Choi et al. People with heart problems, for example, do not want their ECG time-series to be shared, but they do accept the mining of certain accurate patterns from their time-series. They present the corresponding assumptions and needs based on this discovery. They demonstrate that only randomization approaches meet all assumptions, but even then, they fall short of the criteria. As a result, we look at randomization-based solutions that meet all of the assumptions and criteria. They leverage the noise averaging effect of piecewise aggregate approximation (PAA) to do this, which may help to solve the issue of eliminating distance ordering in randomly disturbed time-series. They initially offer two naïve strategies based on the noise averaging effect, which employ random data perturbation in publishing time-series while using the PAA distance in calculating distances. In terms of uncertainty and distance orders, however, there is a compromise between these two methods.

According to [13] X. Xiao et al., we may be providing inadequate protection to one group of individuals while imposing excessive privacy controls on another. We provide a novel generalization framework based on the idea of individualized anonymity as a result of this. Their method uses the smallest amount of generalization possible to meet everyone's needs, retaining the most information possible from the microdata. They conduct a thorough theoretical investigation that yields useful insight into the behavior of different solutions. Their mathematical study, in particular, identifies the situations in which earlier work fails to safeguard privacy and proves the superiority of the offered alternatives. Extensive experiments back up the theoretical conclusions.

C. C. Aggarwal et al. explore randomization, k-anonymization, and distributed privacy-preserving data mining in their paper [14]. They also go through situations when data mining programmers' output has to be cleaned for privacy reasons. They talk about the technological and theoretical limitations of privacy preservation across large data collections.

They all handle the difficulty of providing person-specific data while maintaining the anonymity of the persons to whom the data refers in [15] Pierangela Samarati et al. The strategy is founded on the concept of k-anonymity. If efforts to connect overtly identifiable information to its contents ambiguously map the information to at least k individuals, the table gives k-anonymity. They demonstrate how generalization and suppression approaches may be used to produce k-anonymity. They establish the idea of minimum generalizations, which describes the ability of the release process to distort data only as much as is required to ensure k-anonymity. They show how to pick between various minimum generalizations using different preference procedures. Finally, they provide a method as well as experimental findings obtained when the approach was used to generate real-world medical data releases. They also measure the accuracy and completeness of the findings for various values of k to report on the quality of the provided data.

Freny Presswala et al., [16] The database security research group and the administration statistical organizations have had a long-term objective of protecting sensitive data from unwanted access. As a

result, the security problem has lately become a substantially more essential area of study in data mining. As a result, privacy-preserving data mining has received a lot of attention recently. Data anonymization is a kind of data purification whose expectation is security insurance, and it is one of the ways of privacy-preserving data mining. It's a technique for encrypting or expelling identifying data from data sets, with the purpose of keeping the people the data depicts anonymous. They have evaluated numerous data anonymization strategies and presented a comparative study of the same in this work.

III. METHODOLOGY

Many strategies have been developed for extracting information from data that is protected from disclosure. The concepts of K-Anonymity, T-Closeness, L-Diversity, and Advanced Encryption standard are all discussed in detail in this chapter.

- Anonymity: "Anonymity is the most often used approach in privacy protection systems. When it comes to identity exposure, the aim is to protect datasets from being compromised by an adversary who connects to that kind of data item and obtains sensitive information about the person connected. Individuals working in the project proposed using the k-anonymity strategy to reduce the risk of being identified [1]. An extra set of anonymization constraints for the original data are implemented when k-anonymity is used to safeguard sensitive information from disclosure. A number of novel anonymity algorithms have been developed, including the following:"

- The "anonymization procedure was used. In the case of K-Anonymity [1], there are two aspects: generalization and suppression. The first procedure, generalisation, is turning attribute values into a variety in order to make specification more straightforward and concise. For the purpose of limiting the possibilities of recognition, the birth certificate may be standardized to a number, such as the year of birth. In the second phase of suppression, all of the values associated with the property are completely erased. It goes without saying that such tactics reduce the risk of being identified when utilising publicly available information, but they also reduce the accuracy of systems relying on changed data[13]. Specific identifiers (I) [1] are relevant information that specifically and explicitly identifies the record investor and is customarily immediately removed from the full disclosure data, such as a specific word, personal details, or cell phone number. Clear and unambiguous notations (I) [1] are data that clearly and explicitly designates the record owner and thus is normally deactivated from the disclosed data. When it comes to quasi-identifiers (QIDs),[1] information like as a person's date of birth, gender, and Postal address are examples of information that might potentially identify their record owner and that is often updated in the published data. Strictly confidential data characteristics (S)[1] are data attributes that include sensitive data from the data owner, such as income or illness, that should really be kept confidential. K-Means [1] is a widely used clustering method that is both straightforward and easy. It categorises a group of facts into a K value that has been specified. A collection of randomly selected initial cluster centres serves as the starting point for the clustering operation, which continues to redistribute data items in the dataset to cluster centres based on the distance between cluster centres and the data item throughout. This technique is continued until a condition is fulfilled, at which point the process ends. The K-means approach for clustering was used to assess the change in an anonymized dataset based on the change in cluster numbers [1] in our experiments."

- L-diversity: "It's possible that the idea of k-anonymity, as well as the variety of anonymization methods accessible, will make this paradigm

especially appealing to data suppliers. Despite this, it has been shown that this strategy is vulnerable to a number of attacks, especially when the attacker has access to background information on the target. l-diversity is a new and enlarged paradigm for protecting one's personal information. The authors of the research pointed out that anonymity has k-weaknesses in two attack models: the homogeneity assault and the background-knowledge attack. As a result of this assault, all values for a sensitive attribute included inside an equivalence class are the same as one another. Therefore, even if the data is k-anonymous, the sensitive attribute value for each record in a group of size k may be predicted with 100 percent accuracy. This is true whether the data is structured or unstructured. Attack on the Background Information: In this approach, the adversary may take advantage of a relationship between one or more quasi-identifier characteristics and the sensitive attribute, as well as publicly available information about the target, to rule out particular values for the sensitive attribute from consideration. Consider the following scenario: If a young human's QI is associated to an ordered pair in which all principles of the responsive attribute "disease" are either Arthritis, Alzheimer's disease, or Flu, the target's sensitive information is almost certainly "Flu," because the first two values are highly unlikely to occur in a young person. It is l-diverse if a collection of records belonging to almost the same Linearly Separable q* has at least l "well-represented" options for the Delicate Attribute S. This is defined as follows: L-diversity is defined as the presence of an Equivalence Class q* in every row of a given table T, and the table is said to have l-diversity. The l-diversity principle ensures that each block of records (similarity class) has l "well-represented" values, but it does not specify what "well-represented" means in this context." [12].

- When an "equivalence class is t-close, the difference in between distribution of an information set in this classification and the prevalence of the attribute across all of the rows in the database is smaller than a threshold value t. t-closeness is satisfied when all of the table's clusters fulfil it. If all of a table's clusters meet t-closeness, then the record is said to fulfill t-closeness. [5]."

IV. Proposed Flow

In this section, we discuss the flow for a tentative system that will be used in the future to improve the loss of data anonymization.

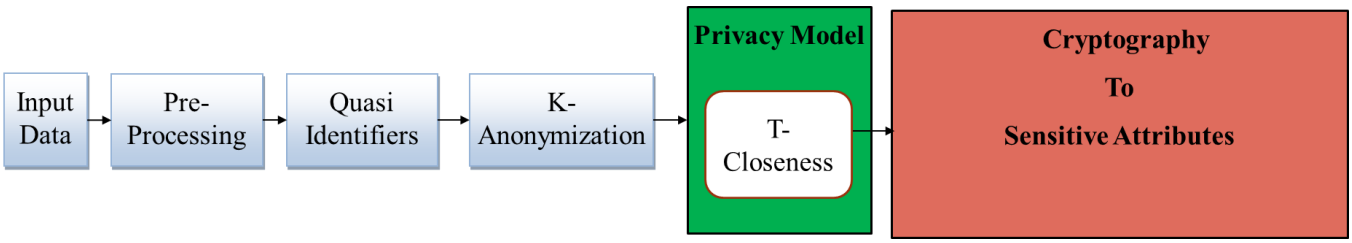

Fig. 1. Tentative Flow

Pseudo Code

Input: A set T for n records, the value k for k-anonymity

Step 1: Anonymize the original dataset with a k-anonymization algorithm

Step 2: Define i = 1 to represent number of clusters, define as an array for keeping equivalence class numbers of each loop,
Step 3: Set equivalence classes number observed by Step 1 to end of class,
Step 4: i =i +1 Find i clusters of original data with a clustering algorithm (we preferred K-means) Anonymize each cluster with the k-anonymization algorithm Calculate the sum of all equivalence classes observed by Step 4(b) and set it to Eq[i],
If Eq[i] is bigger than Eq[i-1] go to Step 4,
If Eq[i] is equal or smaller than Eq[i-1] go to Step 5,
Step 5: Return i as a satisfactory cluster number

As shown in figure 1 technology to encrypt personal data and allow k-anonymization of the encrypted data on the cloud to improve security in the anonymization of personal data. The following are some of the features of the technology that was developed:
1. Encrypted data generalisation that is secure

To anonymize data, several k-anonymization systems employ a tree structure to generalise comparable data with different values, aggregating data from a smaller group into a larger group in a hierarchy. Data from smaller regional subsets, for example, may be anonymised by combining it with data from a larger regional subset. This tree structure could not be built from encrypted data using traditional technologies since the information on the smaller subset could not be read.
2. High data security and processing speed
Encrypted data is substantially slower to process than non-encrypted data in general. The suggested encryption method allows for high-speed comparison of encrypted data while also reducing the amount of data processing needed in the encrypted state. As a consequence, the overhead increase in data processing may be maintained to a minimum to guarantee that realistic processing rates are achieved.

V. RESULTS AND ANALYSIS

```
[ ]  # we use Pandas to work with the data as it makes working with c
     import pandas as pd
```

```
# this is a list of the column names in our dataset (as the file
names = (
    'age',     'workclass', #Private, Self-emp-not-inc, Self-emp
    'fnlwgt', # "weight" of that person in the dataset (i.e. how
    'education',     'education-num',
    'marital-status',     'occupation',
    'relationship',     'race',
    'sex',     'capital-gain',
    'capital-loss',     'hours-per-week',
    'native-country',     'income',
)

# some fields are categorical and will require special treatment
categorical = set((
    'workclass',     'education',
    'marital-status',     'occupation',
    'relationship',     'sex',
    'native-country',     'race',
    'income',
))
df = pd.read_csv("../data/k-anonymity/adult.all.txt", sep=", ",
```

```
[ ]  df.head()
```

	age	workclass	fnlwgt	education	education-num	marita
0	39	State-gov	77516	Bachelors	13	Neve
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-c
2	38	Private	215646	HS-grad	9	
3	53	Private	234721	11th	7	Married-c
4	28	Private	338409	Bachelors	13	Married-c

```
def get_spans(df, partition, scale=None):
    spans = {}
    for column in df.columns:
        if column in categorical:
            span = len(df[column][partition].unique())
        else:
            span = df[column][partition].max()-df[column][partition].min()
        if scale is not None:
            span = span/scale[column]
        spans[column] = span
    return spans
```

```
[ ]  full_spans = get_spans(df, df.index)
     full_spans
```

```
{'age': 73,
 'workclass': 9,
 'fnlwgt': 1478115,
 'education': 16,
 'education-num': 15,
 'marital-status': 7,
 'occupation': 15,
 'relationship': 6,
 'race': 5,
 'sex': 2,
 'capital-gain': 99999,
 'capital-loss': 4356,
 'hours-per-week': 98,
 'native-country': 42,
 'income': 2}
```

Fig. 3. Get Span Information

```
[ ]  def split(df, partition, column):
         """
         :param        df: The dataframe to split
         :param partition: The partition to split
         :param    column: The column along which to split
         :        returns: A tuple containing a split of the original partition
         """
         dfp = df[column][partition]
         if column in categorical:
             values = dfp.unique()
             lv = set(values[:len(values)//2])
             rv = set(values[len(values)//2:])
             return dfp.index[dfp.isin(lv)], dfp.index[dfp.isin(rv)]
         else:
             median = dfp.median()
             dfl = dfp.index[dfp < median]
             dfr = dfp.index[dfp >= median]
             return (dfl, dfr)
```

Fig. 4. Implement a split function

It takes a dataframe, partition and column and returns two partitions that split the given partition such that all rows with values of the column column below the median are in one partition and all rows with values above or equal to the median are in the other.

```
[ ]  # we apply our partitioning method to two columns of our dataset, using "income" as the sensitive attribute
     feature_columns = ['age', 'education-num']
     sensitive_column = 'income'
     finished_partitions = partition_dataset(df, feature_columns, sensitive_column, full_spans, is_k_anonymous)
```

```
[ ]  # we get the number of partitions that were created
     len(finished_partitions)
```

Fig. 5 Data Column Selection

```
[ ]  def is_k_anonymous(df, partition, sensitive_column, k=3):
         """
         :param               df: The dataframe on which to check the partition.
         :param        partition: The partition of the dataframe to check.
         :param sensitive_column: The name of the sensitive column
         :param                k: The desired k
         :returns               : True if the partition is valid according to our k-anonymity criteria,
         """
         if len(partition) < k:
             return False
         return True

     def partition_dataset(df, feature_columns, sensitive_column, scale, is_valid):
         """
         :param               df: The dataframe to be partitioned.
         :param  feature_columns: A list of column names along which to partition the dataset.
         :param sensitive_column: The name of the sensitive column (to be passed on to the `is_valid` f
         :param            scale: The column spans as generated before.
         :param         is_valid: A function that takes a dataframe and a partition and returns True if
         :returns               : A list of valid partitions that cover the entire dataframe.
         """
         finished_partitions = []
         partitions = [df.index]
         while partitions:
             partition = partitions.pop(0)
             spans = get_spans(df[feature_columns], partition, scale)
             for column, span in sorted(spans.items(), key=lambda x:-x[1]):
                 lp, rp = split(df, partition, column)
                 if not is_valid(df, lp, sensitive_column) or not is_valid(df, rp, sensitive_column):
                     continue
                 partitions.extend((lp, rp))
                 break
             else:
                 finished_partitions.append(partition)
         return finished_partitions
```

```
[ ]  # we sort the resulting dataframe using the feature c
     dfn.sort_values(feature_columns+[sensitive_column])
```

	age	count	education-num	income
469	17.000000	3	3.000000	<=50k
615	17.000000	5	4.000000	<=50k
110	17.000000	36	5.000000	<=50k
111	17.000000	198	6.000000	<=50k
0	17.000000	334	7.200599	<=50k
120	17.000000	14	9.000000	<=50k
43	17.000000	5	10.000000	<=50k
616	18.000000	6	4.000000	<=50k
329	18.000000	10	5.000000	<=50k
121	18.000000	249	9.000000	<=50k
44	18.000000	189	10.000000	<=50k
1	18.227876	451	7.283186	<=50k
2	18.227876	1	7.283186	>50k
470	18.375000	8	3.000000	<=50k
211	18.645833	96	6.000000	<=50k
471	19.000000	12	4.000000	<=50k

Fig.6 K-Anonymity Function

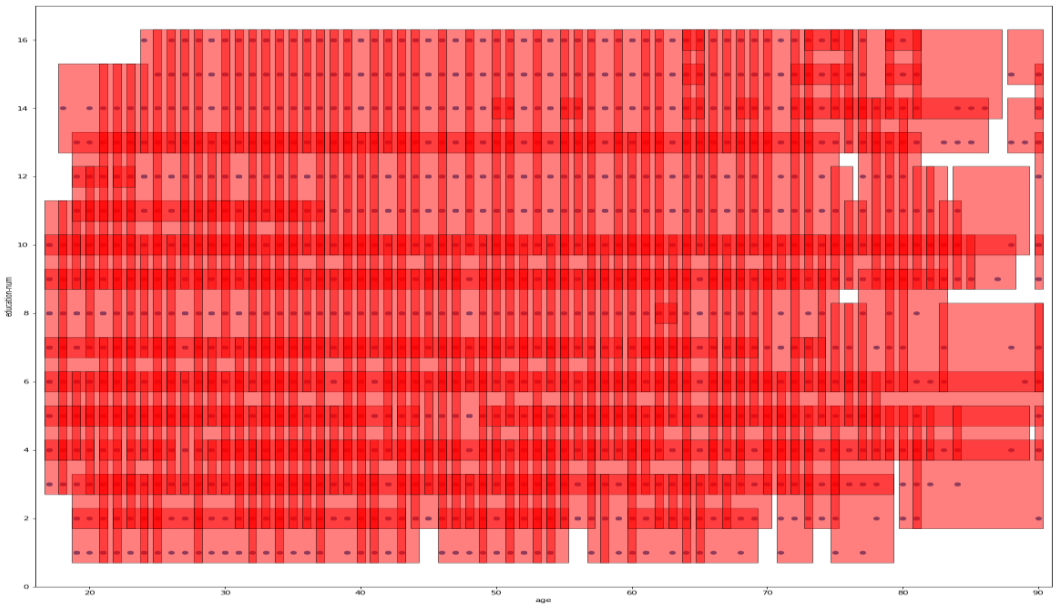

Fig.7 Plot K-Anonymity

```
[ ]  def diversity(df, partition, column):
         return len(df[column][partition].unique())

     def is_l_diverse(df, partition, sensitive_column, l=2):
         """
         :param               df: The dataframe for which to chec
         :param        partition: The partition of the dataframe
         :param sensitive_column: The name of the sensitive colum
         :param                l: The minimum required diversity
         """
         return diversity(df, partition, sensitive_column) >= l
```

```
[ ]  # Let's see how l-diversity improves the anonymity of our dataset
     dfl.sort_values([column_x, column_y, sensitive_column])
```

	age	count	education-num	income
0	17.706107	785	7.248092	<=50k
1	17.706107	1	7.248092	>50k
114	18.341463	40	3.365854	<=50k
115	18.341463	1	3.365854	>50k
4	19.320276	1301	10.000000	<=50k
5	19.320276	1	10.000000	>50k
2	20.080607	1707	9.000000	<=50k
3	20.080607	5	9.000000	>50k
122	20.364583	95	7.333333	<=50k
123	20.364583	1	7.333333	>50k
244	20.500000	17	13.166667	<=50k
245	20.500000	1	13.166667	>50k
10	21.000000	568	10.000000	<=50k
11	21.000000	2	10.000000	>50k
116	21.142857	62	3.063492	<=50k
117	21.142857	1	3.063492	>50k

Fig.8 L-Diversity

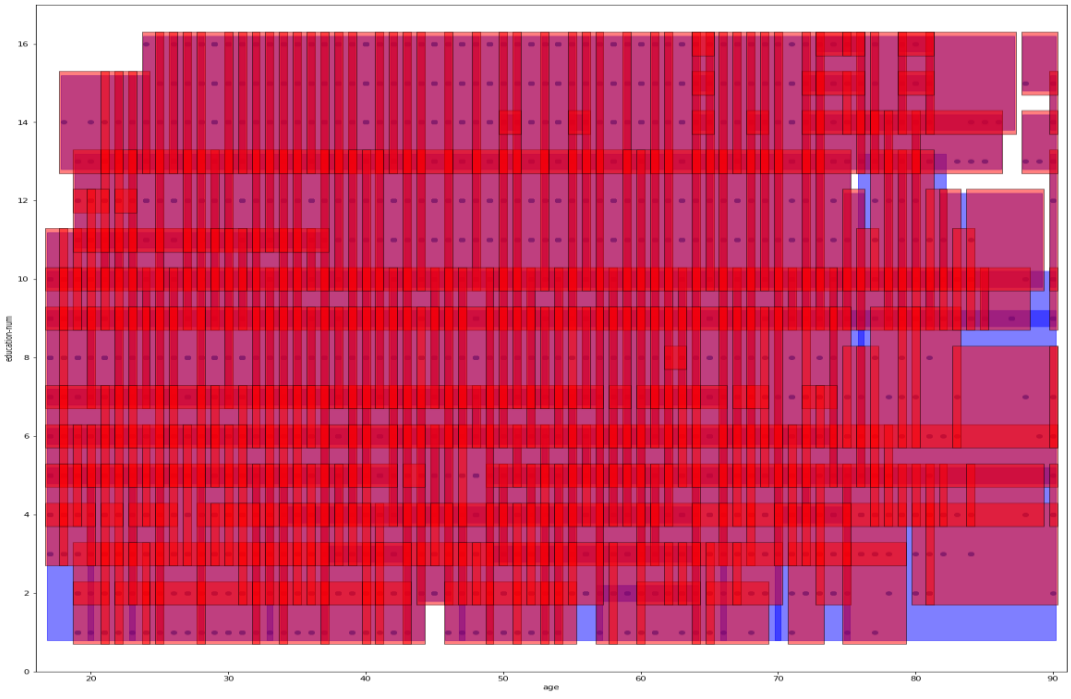

Fig.9 Plot L-Diversity

```
[ ]  def t_closeness(df, partition, column, global_freqs):
         total_count = float(len(partition))
         d_max = None
         group_counts = df.loc[partition].groupby(column)[column].agg('count')
         for value, count in group_counts.to_dict().items():
             p = count/total_count
             d = abs(p-global_freqs[value])
             if d_max is None or d > d_max:
                 d_max = d
         return d_max


     def is_t_close(df, partition, sensitive_column, global_freqs, p=0.2):
         """
         :param               df: The dataframe for which to check l-diversity
         :param        partition: The partition of the dataframe on which to check l-diversity
         :param sensitive_column: The name of the sensitive column
         :param     global_freqs: The global frequencies of the sensitive attribute values
         :param                p: The maximum allowed Kolmogorov-Smirnov distance
         """
         if not sensitive_column in categorical:
             raise ValueError("this method only works for categorical values")
         return t_closeness(df, partition, sensitive_column, global_freqs) <= p
```

```
[ ]  # Let's see how t-closeness fares
     dft.sort_values([column_x, column_y, sensitive_column])
```

	age	count	education-num	income
12	24.543287	738	11.476788	<=50k
13	24.543287	59	11.476788	>50k
2	25.747108	5617	10.000000	<=50k
3	25.747108	520	10.000000	>50k
0	26.697666	10248	8.124394	<=50k
1	26.697666	677	8.124394	>50k
26	29.000000	112	11.367647	<=50k
27	29.000000	24	11.367647	>50k
4	29.434809	3385	13.299485	<=50k
5	29.434809	1470	13.299485	>50k
28	30.487395	198	11.432773	<=50k
29	30.487395	40	11.432773	>50k
46	32.000000	90	11.378151	<=50k
47	32.000000	29	11.378151	>50k
48	33.000000	102	11.421875	<=50k
49	33.000000	26	11.421875	>50k

Fig.10 t-closeness

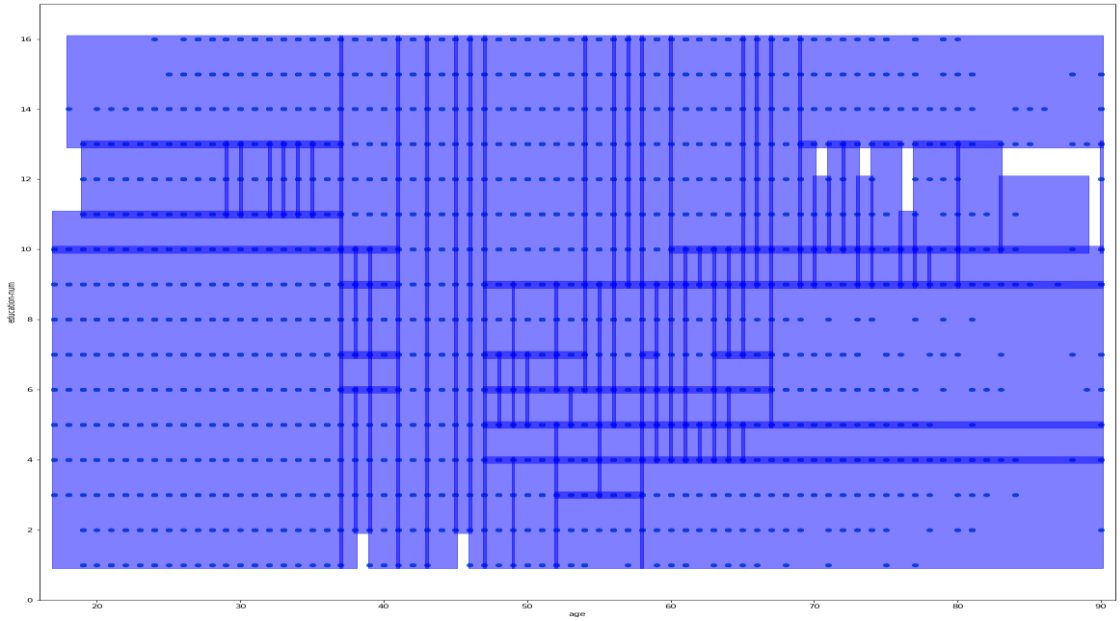

Fig.11 Plot of t-closeness

V. CONCLUSION

"To preserve sensitive data, privacy is a vital problem. People are particularly anxious about sensitive data that they do not like to expose. The deployment of an anonymization approach decreases data loss and increases privacy protection. T-closeness, which stipulates that a sensitive attribute's distribution in any equivalence class should be near to the attribute's distribution in the whole database (i.e., the gap between the two distributions should be no greater than a threshold t) (i.e., the distance between the two distributions should be no more than a threshold t). As a consequence, the totally homomorphic encryption approach is utilised to make the system more secure. The system will be more secure, confidential, and encrypted as a consequence of it."

Based on the foregoing examination of the many sorts of algorithms, I've come to the conclusion that each algorithm for each approach has its own set of criteria and previous data knowledge.

VI. References

[1] L. Yang, X. Chen, Y. Luo, X. Lan, and W. Wang, "IDEA: A utility-enhanced approach to incomplete data stream anonymization," Tsinghua Sci. Technol., vol. 27, no. 1, pp. 127–140, 2022, doi: 10.26599/TST.2020.9010031.

[2] S. De Capitani Di Vimercati et al., "Artifact: Scalable Distributed Data Anonymization," 2021 IEEE Int. Conf. Pervasive Comput. Commun. Work. other Affil. Events, PerCom Work. 2021, pp. 450–451, 2021, doi: 10.1109/PerComWorkshops51409.2021.9431059.

[3] Pelin Canbay and Hayri Sever, "The Effect of Clustering on Data Privacy" 2015 IEEE International Conference on. IEEE 2015.

[4] Mohamed Nassar, Abdelkarim Erradi, Qutaibah M. Malluhi, "Paillier's Encryption: Implementation and Cloud Applications" KINDI Center for Computing Research Qatar University Doha, Qatar.

[5] Mohammad-Reza Zare-Mirakabad, Fatemeh Kaveh-Yazdy, Mohammad Tahmasebi, "Privacy Peservation by k-anonymizing Ngrams of Time Series" Yazd University, Iran, Dalian University of Technology, Dalian.

[6] Tsubasa Takahashi , Koji Sobataka , Takao Takenouchi , Yuki Toyoda , Takuya Mori and Takahide Kohroy "Top-Down Itemset Recording for Releasing Private Complex Data" Cloud System Research Laboratories, NEC Corporation, Kawasaki, Kanagawa Japan, Jichi Medical University Hospital, Shimotsuke, Tochigi Japan. IEEE 2013.

[7] Ninghui Li, Tiancheng Li, Suresh Venkatasubramanian, "T-Closeness: Privacy Beyond k-Anonymity and l-Diversity" Department of Computer Science, Purdue University, AT&T Labs – Research. IEEE 2007.

[8] Nirav. U.Patel, Vaishali.R.Patel, "Anonymization of Social Networks for Reducing Communication Complexity and Information Loss by Sequential Cluatering", 2015.

[9] Mahesh, R., A New Method for Preserving Privacy in Data Publishing Against Attribute and Identity Disclosure Risk (2013). International Journal on Cryptography and Information Security (IJCIS), Vol.3, No. 2, June 2013, Available at SSRN: https://ssrn.com/abstract=3685781

[10] R. B. Ghate and R. Ingle, "Clustering based Anonymization for privacy preservation," in Pervasive Computing (ICPC), 2015 International Conference on, 2015.

[11] M. B. Malik, M. A. Ghazi, and R. Ali, "Privacy preserving data mining techniques: Current scenario and future prospects,"inProc.3rdInt.Conf. Comput. Commun. Technol. (ICCCT), Nov. 2012, pp. 26–32.

[12] M.-J. Choi, H.-S. Kim and Y.-S. Moon. "Publishing time-series data under preservation of privacy and distance orders". International Journal of Innovative Computing, Information and Control (IJICIC), Vol. 8, pp. 3619-3638, 2012.

[13] X. Xiao and Y. Tao. Personalized privacy preservation. In Proceedings of ACM Conference on Management of Data (SIGMOD'06), pages 229–240, June 2006.

[14] C. C. Aggarwal and S. Y. Philip, A general survey of privacy-preserving data mining models and algorithms: Springer, 2008. 21. Olga Gkountouna, A Survey on Privacy Preservation Methods, June -2011.

[15] Pierangela Samarati and Latanya Sweeney , Protecting Privacy When Disclosing Information: K-Anonymity and its Enforcement through Generlization and Suppression.

[16] Freny Presswala, Amit Thakkar and Nirav Bhatt, Survey on Anonymizati on in Privacy Preserving Data Mining, International Journal of Innovative and Emerging Research in Engineering (IJIERE), 2015.

Cite this article as :

Brinit Trivedi, Sheshang Degadwala, Dhairya Vyas, "Privacy Preserving Parallel Distributed Data Stream Anonymization", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 3, pp. 53-66, May-June 2022. Available at doi : https://doi.org/10.32628/CSEIT228312
Journal URL : https://ijsrcseit.com/CSEIT228312