Data Sensitivity Preservation-Securing Value Using Varied Differential Privacy Method (SP-SV Method)

Supriya G Purohit, Dr Veeragangadhara Swamy
Research Scholar, Dept. of Computer Science and Engineering[1]
Professor, Department of Computer Science and Engineering[2]
GM Institute of Technology, Davanagere, Karnataka, India[1, 2]
Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[1, 2]

***Abstract*—Numerous governmental entities, including hospitals and the Bureau of Statistics, as well as other functional units, have shown great interest in personalized privacy. Numerous models and techniques for data posting have been put forward, the majority of which concentrated on a single sensitive property. A few scholarly articles highlighted the need to protect the privacy of data which includes many sensitive qualities. Utilizing current techniques like the sanctity of privacy in data gets decreased if many sensitive values are published while maintaining k-anonymity and l-diversity simultaneously. Furthermore, customization hasn't been investigated in this context. We describe a publishing strategy in this research that handles customization when publishing material that has many sensitive features for analysis. The model makes use of a slicing strategy that is reinforced by fuzzy approaches for numerical sensitive characteristics based on variety, generalization of categorical sensitive attributes, and probabilistic anonymization of quasi-identifiers using differential privacy. We limit the confidence that an adversary may draw about a sensitive value in a publicly available data collection to the level of understanding as an inference drawn from known information. Both artificial datasets based on real-life healthcare data were used in the trials. The outcomes guarantee that the data value is maintained while securing individual's privacy**.

***Keywords*—Big data; privacy preservation; security; data publish; data privacy**

I. INTRODUCTION

One of the largest technological breakthroughs in the near time, cloud computing, has developed quickly. These days, the market for cloud computing is well-established, leading many big businesses to construct effective cloud infrastructures. Cloud computing and data analytics go hand in hand, and the degree of data analytics available on cloud platforms is growing day by day.

Innovative advances in e-commerce, healthcare, and other fields are made possible by new approaches and platforms for big data analytics, which also open up a plethora of beneficial opportunities for companies [31]. But as fresh concerns about privacy are on the raise, the act of gathering and organizing data presents noteworthy privacy hazards, making information "security" and "privacy" a matter of concern. A trustworthy privacy model must be in place while processing to guard against external assaults and stop data leaks. Preventing potential security issues is necessary even during the storage phase. Several strategies are in practice to protect big data privacy. The primary methods in use may be categorized into three [1]: noise-based methods, data encryption, and anonymization techniques [12]-[14]. The first two aid in concealing the private details, but they can't guarantee privacy due to the prevalence of re-identification procedures [2]. Nonetheless, carrying out data analytics statistically employing noise-based methods is more beneficial and successful [3].

We have examined and used the noise-based privacy algorithm known as "Differential Privacy" on the Hadoop data analytics platform in this work. It claims to address the drawbacks of privacy solutions based on anonymization and encryption [17]-[19]. We refer to this privacy platform as "Data Sensitivity Preservation-Securing Value using Varied Differential Privacy Method (SP-SV Method)". It provides a solution in situations when the hazards related to privacy and the expenses for providing the required privacy of data are rising. A mathematical concept known as "differential privacy" describes the loss of one's privacy and measures the extent to which a particular privacy strategy, such as random noise insertion, would be effective in sustaining the privacy of specific information inside a dataset [4].

The noise or disturbance which has to be introduced to the attributes to gain the appropriate level of privacy depends on the security settings based on sensitivity of attributes. The degree to which the dataset's privacy-preserved outputs may be discriminated statistically is measured by the privacy-based approach [5]. As Hadoop can handle large-scale computing challenges, it is utilized for models of parallel data processing [6]. Still, there are shortcomings with this platform's privacy-related features. Actually, the privacy of a dataset is based upon if it is encrypted or anonymized [30]. The platform's privacy and security haven't changed in a while and many of Hadoop's sections have been developed independently [7] making privacy preservation more challenging.

The privacy preservation methods mostly concentrate on providing security and privacy to the data [22]. The data usability section is mostly slipping off the focus. The new "SP-SV Method" ensures the security of sensitive personal data in a dataset for data analysis through the implementation of a varied Differential Privacy algorithm in the Hadoop Map Reduce platform focusing on data usability. This ensures privacy with data usability for analytics. The SP-SV Method is an adaptable method since it doesn't need any extra knowledge to compute on the datasets providing secure and useful data.

II. LITERATURE SURVEY

Cynthia Dwork et al., [15] laid out the mathematical foundations of differential privacy, which is a mathematical approach to safeguard individual privacy in big data analysis. It involves adding noise to data to protect individual information from being exposed. Using differential privacy concept, data can be allowed for analysis yet protecting the privacy of data.

Cynthia Dwork et al. [16], the paper discusses practical considerations and challenges in implementing differential privacy in real-world scenarios. It provides insights into deploying differential privacy techniques effectively.

HybrEx [20] is specifically a paradigm for cloud computing anonymity, security, and confidentiality that is intended for hybrid clouds. HybrEx has separated its data into sensitive and non-sensitive categories. Sensitive data is stored in a private cloud, while non-sensitive data is sent to public clouds. One of HybridEx's shortcomings is that it cannot manage generated values in Map stages in clouds that are private or public.

Machanavajjhala et al. [23], used Differential privacy nonetheless, to produce artificial datasets for statistical examination of patterns of commute in mapping applications. Handling datasets with broad domains was a problem because, despite the sparseness of the data, noise permeated the whole domain. The domain size was reduced by the use of exogenous data and procedures; nonetheless, the distribution of travel lengths was only appropriate for research involving very short journeys.

R. Agrawal et al. [24], reasoned that it is hard to estimate user privacy correctly when randomization is used since it disrupts the personal data of the user. They made an effort to respond to the query, "Is it still possible to construct sufficiently accurate predictive models with a large number of users who do this perturbation?"

S. R. Ganta et al. [25], subsequent research has shown that such criteria fall short of protecting an individual's privacy. The objective of the secure multi-party computing technique is to create a data mining model spanning many databases without disclosing the specific entries in each database.

Building a centralized warehouse may not be possible because of privacy concerns;

M. Kantarcioglu et al. [26], addressed issues with calculating association rules in such a setting. They assumed that all sites with homogenous databases had the same schema. On the other hand, every website has data on various entities. They intended to create universally applicable association guidelines while also restricting the amount of information that could be disclosed about individual sites.

Two general techniques for privacy-preserving data mining have been proposed: safe multi-party computing and randomization. While safe multi-party computing seeks to develop a method for mining data across many databases without disclosing specific information, randomization concentrates on protecting individual privacy. A platform for expanding data analysis called Privacy Integrated Queries (PINQ).

F. D. McSherry [27], performed calculations on private information while providing total privacy guarantees for each and every record in the underlying data sets. To prevent noise from affecting the computation's intermediate findings, PINQ employs a request/reply paradigm and stores the results on a reliable data server that is supplied by a system that is distributed.

I. Roy et al. [28], Airavat algorithm enforces restrictions on access and applies differential privacy to safeguard data. As far as safe computing and information privacy in MapReduce systems are concerned, this is the first technology that offers almost a required solution. To prevent unauthorized mappers from leaking information outside the group as well as granting mappers access to its contents and network, Airavat employs a required control scheme.

The task while adhering to the same underlying principles as Airavat [28], SP-SV Method has added additional functionality, such as a combiner, and refrained from altering the core source code of Hadoop. SP-SV Method is a privacy-protecting data analysis tool. It delivers the promised anonymity by combining Hadoop MapReduce with the differential anonymity approach to aggregate attributes from the datasets being used without revealing any specifics about individual data objects.

As defined by Cynthia Dwork, "The outcome of any analysis is essentially equally likely, independent of whether any individuals join or refrain from joining the dataset," When the computation output for every single input is independent of the input's existence, we designate a calculation on a set of data as being highly private in the input data set.

Take note that we have focused on Hadoop MapReduce security as well as privacy in the SP-SV Method. This indicates that, although we prevent some of the ways intruders may acquire information through information disclosure (using insecure Reducers), while maintaining the privacy of the individual, we nonetheless accept the intrusive party's certainty regarding the existence or nonexistence of any information in the MapReduce outcome.

III. METHODOLOGY

A. Working on the Hadoop MapReduce Platform

Hadoop's MapReduce is an open-source initiative and a popular data processing framework that works well for many different kinds of workloads, such as log evaluation, analyzing social networks, searches, and clustering We chose to use the Hadoop platform since a lot of companies, including Amazon, Facebook, Yahoo, including the New York Times [8], have successfully adopted it to run their applications on clusters. MapReduce is the primary tool in Hadoop's toolkit.

To add different kinds of features to Hadoop, multiple modules have been independently created throughout time. But until recently, Hadoop's security was not a top priority for development. The security mechanism's vulnerability has therefore emerged as a major obstacle to Hadoop's progress, despite the platform's growing adoption. Over time, MapReduce and other Hadoop framework components may

have difficulties because of a dearth of a uniform security approach and many security risks involved.

The findings [29] state that Hadoop is readily recognized by hackers worldwide. All they have to do is sniff open instances to do this. We chose the Hadoop MapReduce platform and concentrated on addressing its privacy concerns since it is open-source, accessible to a large global user base, and has security flaws. On the other side, the growing importance of data analytics helped us pick this platform.

The designed SP-SV Method is a privacy-protecting data analysis tool. It combines the Differential Privacy technique with Hadoop MapReduce to aggregate characteristics from input datasets while maintaining the promised privacy by not disclosing any information about individual data items [10] [11]. In this work, we have concentrated on Hadoop MapReduce security as well as privacy using the SP-SV Method which is based on varied Differential Privacy for safeguarding the privacy of the person and yet having data of value for analysis.

B. Proposed Method

The proposed work Fig. 1 was evaluated for patient datasets for its usefulness in providing privacy while allowing for data analysis. Comparisons were made before and after applying the proposed varied differential privacy concept with the SP-SV method on the datasets.

Comprehensive approaches have been introduced to present the concept of privacy-preservation. The randomization approach makes sure that no one knows the real data, instead just random information about data sets is revealed, thereby protecting individual privacy. Specifically, Cryptographic Random Number Generators (RNGs) are specialized algorithms designed to produce random numbers with certain properties that make them suitable for cryptographic applications. These properties include unpredictability, uniform distribution, and resistance to various attacks which are aimed at predicting or manipulating the generated numbers.

C. Enforcement of Differential Privacy

In the initial stage, the Mapper code was created the procedure which comprised defining the keys and identifiers. In addition, the privacy parameters "N" and "n" were supplied, and a preconfigured Reducer was chosen. Once the code has been written, compiled, and the jar file has been produced, the next step is to specify the Differential Privacy settings for the Proposed Model. This is necessary in order for the model to generate the right level of noise.

Laplace's Differential Privacy method adds noise to the data, and it can be explained in this way.

$$f(x)+(Lap(\Delta f/epsilon)) \quad (1)$$

D. Overall Algorithms Steps

1) Input splitting: Input splits are the smaller portions of the input data that are separated. Each split is handled by a map job. Considering input attributes in the dataset as follows Patient Id, Age, Name, Gender, City, Job, Specialist, Disease, Marital Status.

2) Mapping: Every mapping task handles its input splits individually. It reads the incoming data, implements the data anonymization logic [12], and outputs a collection of key-value pairs that are intermediate.

The attributes considered for anonymization are Age, City, Gender and Job. And for every considered attribute, add the epsilon and sensitivity to maintain the privacy of the data.

Anonymization Logic: To achieve differential privacy, Laplace noise is applied to each sensitive attribute's original value.

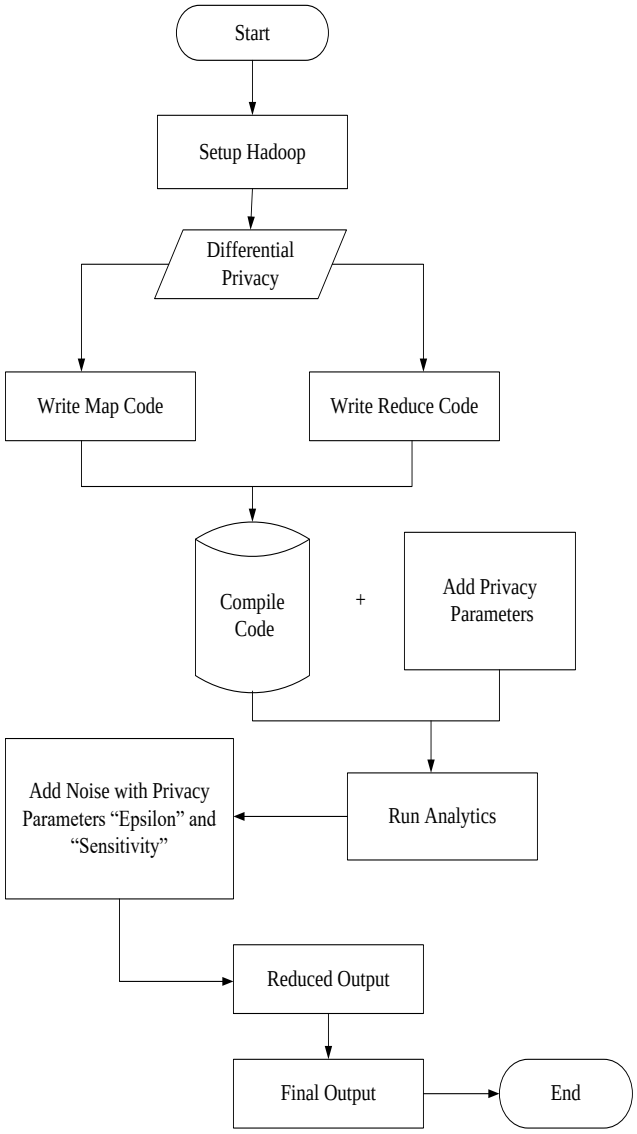

Fig. 1. Proposed flowchart.

The formula for adding Laplace noise is:

$$NoisyValue = OriginalValue + laplace(0, \frac{sensitivity}{\varepsilon})$$

Where:

- $laplace(0, \frac{sensitivity}{\varepsilon})$ represents Laplace noise with mean 0 and scale $\frac{sensitivity}{\varepsilon}$

- $sensitivity$ is the sensitivity of the attribute.

- ε is the privacy parameter.

3) Differential privacy reduction: For a given key, an ordered list of intermediate pairs of keys and values is sent to each mapper. The map function applies the differential privacy method to each key's corresponding values. Using aggregation functions or adding more noise may be necessary in this situation, based on the particular privacy needs of the considered attributes.

4) Intermediate key-value pair shuffling: The map jobs produce intermediate key-value pairs, which are then divided according to the keys and sent to the reducers. Performing this step guarantees that every value linked to the same key ends up in the same reducer.

Reduce Phase:

5) Sorting: The intermediate key-value pairs are arranged according to the keys inside each reducer.

6) Final output: A collection of key-value pairs containing anonymized data is the final output that the reducers generate. This output can be written to an external storage system or saved in HDFS.

E. Overall WorkFlow

The overall workflow of the proposed architecture is set up as in Fig. 2.

1) Setup of the job: The MapReduce job is set up with parameters for the differential privacy method (e.g., ε), as well as input and output pathways, mapper and reducer classes, input and output key-value formats, etc.

2) Job submission: The specified job is sent to the Hadoop cluster.

3) Job execution: Hadoop distributes jobs throughout the cluster nodes and coordinates the mapping process.

4) Task monitoring: We may use command-line tools or the Hadoop JobTracker interface to keep an eye on the status of your task.

When every task has been finished, the job is considered finished, and the final output including differentially private anonymized data is ready for additional processing or analysis.

This methodology guarantees the safeguarding of confidential information inside the input data, all the while for insightful analysis to be conducted on the anonymized data.

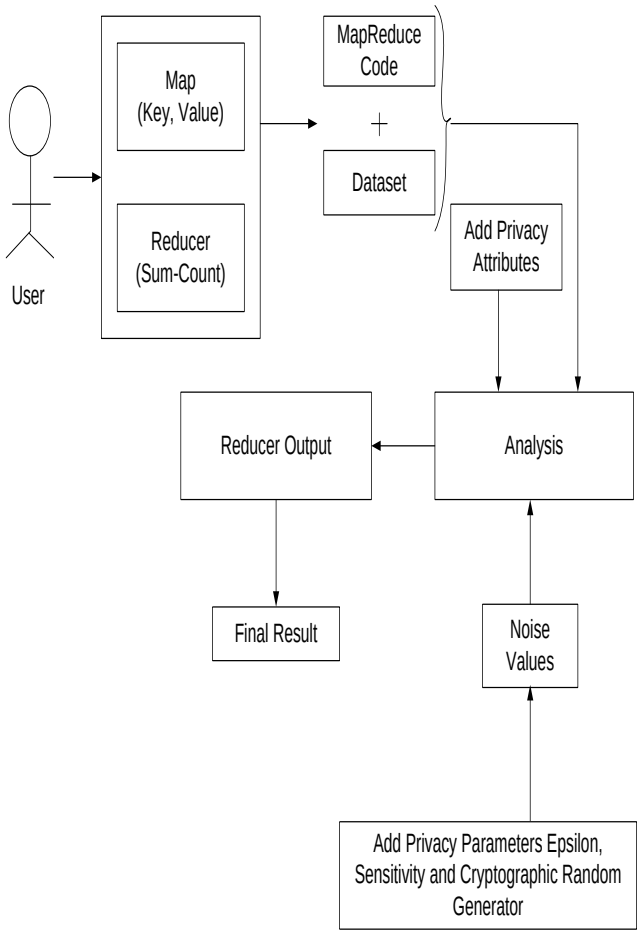

Fig. 2. Proposed architecture.

IV. ALGORITHMS

A. Algotihm 1: Map-Reduce

Input: keys are listed.

The dataset consists of the named identifiers, min-range, max-range and epsilon (ε)

Output: noisy value denoted as L is the output of dataset.

Procedure:

Step 1: The parameters k_1 … to k_n consider for mapping to get output keys.

Step 2: Compute in group by collecting every key mentioned in step1 to generate map.

Step 3: Returns Mean value if max-range is greater than min-range values.

Step 4: else, min-range should be smaller than max-range.

B. Differnetail Privacy Enfocrement Procedure

Step 1: The noise calculation is performed using the values of Epsilon and min-range/max-range.

Step 2: The result is obtained by adding the Reducer-Output and Laplacian Noise.

return Reducer-Output + Laplacian Noise.

The pseudo-code for our algorithm is displayed in Algorithm.

The f(x) function's sensitivity, represented by the symbol Δf, indicates the function's potential level of revealing and incorporates addition of noise with a scale of Δf/epsilon to maintain epsilon-differential privacy.

C. Algorithm 2: Varied Differential Privacy in Proposed Work

Input

*M*apper-Input = *F*(*X*)

Privacy Parameter such as epsilon

min-range

max-range

Output

Added Laplace Noise to Mapper Input(F(*X*))

Procedure: Requirements for Differential Privacy:

Step1: Sensitivity, Δ*F* = |*max-range* – *min-range*|

Step2: Amount of noise: *L* = *Lap*(Δ*f/epsilon*)

Step3: Apply *L* to *F*(*X*)

Return *F*(*X*)+ Laplace Noise.

The Proposed Model in our Differential Privacy code, implemented in the "Mapper" class, requires two privacy parameters: "Epsilon (epsilon)", "Sensitivity" and “Cryptographic Random Number Generators (RNGs)”. These parameters are necessary to determine the appropriate noise level to be added to the result and ensure limits on the potential disclosure of information about datasets.

1) Parameter “Epsilon”: Epsilon, is a fundamental Differential Privacy parameter that is essential to the Proposed Model. The statistics that are often produced as a consequence of calculating sensitive data that might introduce privacy issues. By calculating the degree of privacy loss brought on by a differential change in data, the parameter quantifies privacy. It is essential to acknowledge that epsilon is not an absolute measure of privacy, but rather a relative one. The degree of secrecy rises as epsilon's value falls and vice versa.

2) Parameter “Sensitivity”: Sensitivity is the variable that governs the minimum amount of noise required to be introduced into the output. It is a significant factor in the computation of DP noise is the "Sensitivity" [21]. The term "Impact" refers to the modifications that take place in the result when any input data is eliminated. The Proposed Model incorporates the addition of exponentially distributed noise by the reducers to ensure the enforcement of Differential Privacy.

3) Cryptographic Random Number Generators (RNGs) are specialized algorithms designed to produce random numbers with certain properties that make them suitable for cryptographic applications. These properties include unpredictability, uniform distribution, and resistance to various attacks aimed at predicting or manipulating the generated numbers. The key advantages we can consider as unpredictability. This means that the sequence of random numbers generated should appear statistically random, making it practically impossible for an attacker to predict the next number in the sequence, even if they have access to some of the previously generated numbers.

According to definition of sensitivity, the "Count" function has a sensitivity of 1. The count can be incremented or decremented by a maximum of 1 based on whether an item is added or deleted from the dataset.

$$\text{Max}\ (|Mmin|\ ,\ |Mmax|) = 1$$

The "Sum" function's sensitivity changes depending on the range. As an example, the sensitivity is 100 on a specified interval of integers from 0 to 100. The output will be influenced by 100 units if 100 is either added or subtracted.

The calculation of sensitivity in the proposed work necessitates the data source to explicitly state the span. Within this range, the calculation provider must provide the minimum as 0 and 100 as the maximum number.

This will be used to find the sensitivity, as it is possible to get a rough estimate of the sensitivity by calculating it within the provided range. The range declaration is determined solely by the data values in a dataset and the query. The data provider must assess their dataset and determine the specific information they need to extract from it. Based on this assessment, they can then establish the lowest and maximum values for the range.

The "Count" function requires the range to have a minimum value of 0 and a maximum value of 1. The range for the "Sum" function is from 0 to the largest value. The sensitivity will be determined and the noise will be computed by specifying the range, Δf.

$$\text{noise} \sim \text{Lap}(\Delta f/\text{epsilon})$$

During the process, the estimated noise will be added to the Mapper's output. The sensitivity of a function corresponds to the amount of information it discloses about whether or not an item is present in the input dataset.

In the Reducer Phase, which is part of a Hadoop MapReduce job. It's responsible for merging anonymized data for the same patient ID.

D. Step by Step Procedures

1) The reduce method takes four parameters:

key: A Text object representing the key.

values: An Iterator<Text> containing the values associated with the key.

output: An OutputCollector<Text, Text> used to collect the output of the reduce operation.

reporter: A Reporter object to report progress and status.

2) Creates a 'HashMap' called 'mergedData' to store the merged data. It will store key-value pairs where the key is a string (presumably an attribute of the patient) and the value is also a string (the value of that attribute).

3) Iterates over the values associated with the key.

4) This line retrieves the next value, converts it to a string, and then splits it into an array of strings using a comma (',') as the separator.

5) This starts another loop that iterates over each part of the split string array.

6) The key-value line splits each part into two strings based on a colon (':') separator.

7) These checks if the split resulted in exactly two parts (a key and a value). If so, it proceeds to the next step.

8) Merge the data by adding the key-value pair to the 'mergedData' map. 'keyValue[0]' is the key and 'keyValue[1]' is the value.

The overall purpose of this code is to merge data from multiple values associated with the same key. Each value is assumed to be a comma-separated string of key-value pairs, where each pair is separated by a colon (':') [9]. The code splits these strings and stores the key-value pairs in a map ('mergedData'). Finally, the merged data is collected as the output of the reduced operation [32].

E. Example

Consider a set of datasets with age, city, gender and job. The original dataset had the values as in Table I.

Choose the attributes for which we apply Differential Privacy based SPSV method to Secure Privacy and Safeguarding Value. The attributes Age, Gender, City, Disease are chosen as sensitive attributes. According to Cynthia Dwork, safeguarding Age, Gender and City is very critical and if could fetch the value for those attributes, identification of individual is not impossible. So we choose them along with our main sensitive attribute, Disease. Other attributes are sliced and truncated.

The Chosen attributes in Table II are encoded and then applied Hadoop's Mapper and Reducer algorithms. The varied Differential Privacy Preservation technique, the SP-SV Method is applied. The encoded and transformed data as shown in Table III is decoded to get disclosure safe data. The safe data can be utilized for analytics with almost no chance for re-identification.

The Diseases and cities are plotted on the original data sets before applying Varied DP, Fig. 3. The plot shows a significant amount of change after the application of the Varied Differential privacy method i.e. the SP-SV method in Table II data, Fig. 4.

The transformed data is useful with respect to the chosen sensitive attributes and as getting back to the original data is difficult and nearly impossible with the usage of Epsilon, Sensitivity factor, Crypto Random Generator, the individual data is safeguarded.

TABLE I. ORIGINAL DATASET

Patient Id	Name	Age	Gender	City	Job	Specialist	Disease	Marital Status
PId-900	Aaditya	45	Male	Belgaum	Engineer	Cardiologist	Headache	Unmarried
PId-901	Rashmi	27	Female	Davanagere	Designer	Gynecologist	Uterine Fibroid	Married
PId-902	Tejasvi	63	Male	Ballari	Painter	Oncologist	Prostate Cancer	Married
PId-903	Lakshmi	35	Female	Belgaum	Architect	Specialist	Heart Problem	Married

TABLE II. SENSITIVE DATA

Age	Gender	City	Disease
45	Male	Belgaum	Headache
27	Female	Davanagere	Uterine Fibroid
63	Male	Ballari	Prostate Cancer
35	Female	Belgaum	Heart Problem

TABLE III. TRANSFORMED ENCODED DATASET

Patient Id	Name	Age	Gender	City	Job	Specialist	Disease	Marital Status
PId-900	Aaditya	67	Male	Ballari	Engineer	Cardiologist	Headache	Unmarried
PId-901	Rashmi	71	Female	Belgaum	Designer	Gynecologist	Cancer	Married
PId-902	Tejasvi	34	Male	Davanagere	Painter	Oncologist	Headache	Married
PId-903	Lakshmi	35	Female	Belgaum	Architect	Specialist	Uterine Fibroid	Married

V. RESULTS

The varied Differential Privacy Technique, SP-SV Method has transformed the original datasets and as the Epsilon values are difficult to guess and makes it almost impossible with Sensitivity factor in the equation, being generated by Crypto Random Number Generator.

The output disease count matches with the original count but has modified with age, gender and city. This result is helpful in generating useful data for analysis yet keeping the individual's identity very safe.

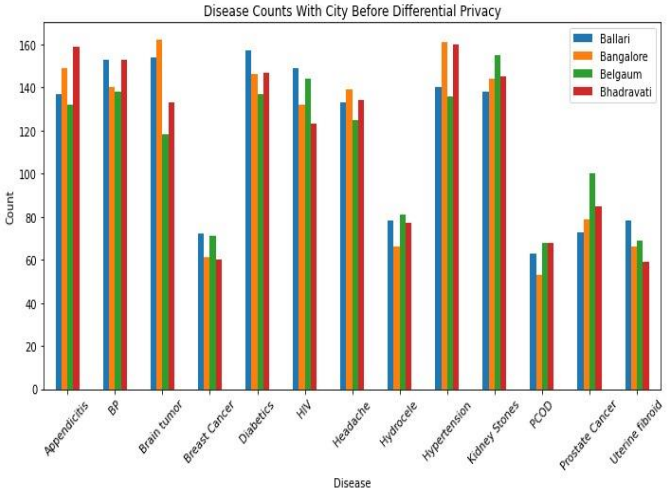

Fig. 3. Count of disease based on city before Differential Privacy (DP).

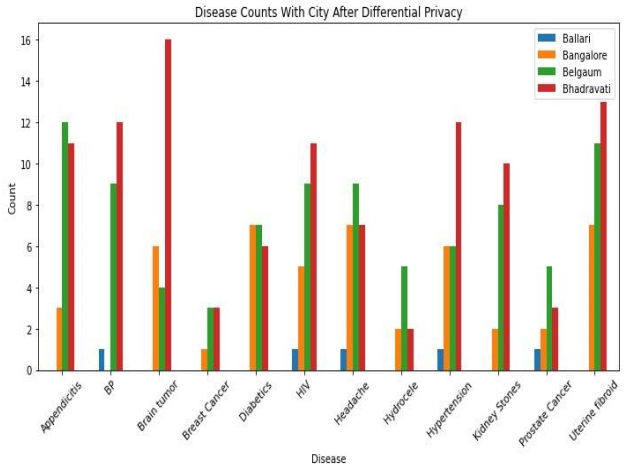

Fig. 4. Count of disease based on city after Differential Privacy (DP).

VI. CONCLUSION

In this work, we have put forth a MapReduce-based varied computation module that preserves privacy of personal information and ensures the utility of the data. SP-SV Method ensures that the computed output for every given input is independent of its presence or absence in the data by implementing the Differential Privacy based SP-SV method using Hadoop MapReduce. This privacy-preserving module ensures privacy preservation by determining the appropriate noise levels to maintain the trade-off between privacy and final output accuracy.

SP-SV Method restricts the calculations and stops data leaks that go beyond the terms of the data provider. Although Airavat served as an inspiration for this model, the Apache Hadoop code itself was left unaltered unlike Airavat, and SP-SV Method's source code was built entirely from scratch. We were ultimately unable to compare the efficiency of SP-SV Method with Airavat since we were unable to obtain source code of Airavat. However, in line with the fundamental principles of the Differential Privacy Method, an individual is not identified specifically when a specific piece of data is added or removed from the database which satisfies the need for privacy of individual and also provides useful data for analytics.

REFERENCES

[1] K. M. P. Shrivastva, M. Rizvi, and S. Singh, "Big data privacy based on differential privacy a hope for big data," in 2014 International Conference on Computational Intelligence and Communication Networks. IEEE, 2014, pp. 776–781.

[2] D. D. Hirsch, "The glass house effect: Big data, the new oil, and the power of analogy," Me. L. Rev., vol. 66, p. 373, 2013.

[3] C. Dwork et al., "Calibrating noise to sensitivity in private data analysis," Journal of Privacy and Confidentiality, vol. 7, no. 3, pp. 17–51, 2016.

[4] C. Dwork, "Differential privacy: A survey of results," in International conference on theory and applications of models of computation. Springer, 2008, pp. 1–19.

[5] M. Yang et al., "Personalized privacy preserving collaborative filtering," in International Conference on Green, Pervasive, and Cloud Computing. Springer, 2017, pp. 371–385.

[6] G. S. Bhathal and A. Singh, "Big data computing with distributed computing frameworks," in Innovations in Electronics and Communication Engineering. Springer, 2019, pp. 467–477.

[7] G. Bhathal and A. Singh, "Big data: Hadoop framework vulnerabilities, security issues and attacks," Array, vol. 1-2, p. 100002, 07 2019.

[8] Which companies are using hadoop for big data analytics? [Online]. Available: https://kognitio.com/big-data/companies-using-hadoop-big-data-analytics/

[9] Supriya G Purohit, Veeragangadhara Swamy "Enhancing data publishing privacy: split-and-mould, an algorithm for equivalent specification", Indonesian Journal of Electrical Engineering and Computer ScienceVol.33, No.2, February2024, pp. 1273~1282ISSN: 2502-4752, DOI: 10.11591/ijeecs. v33.i2. pp1273-1282

[10] S. Desai et al., "Improving encryption performance using mapreduce," 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 pp. 1350–1355.

[11] R. R. Parmar et al., "Large-scale encryption in the hadoop environment: Challenges and solutions," IEEE Access, vol. 5, pp. 7156–7163, 2017.

[12] P. Goswami and S. Madan, "Privacy preserving data publishing and data anonymization approaches: A review," in 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017, pp. 139–142.

[13] N. Victor, D. Lopez, and J. H. Abawajy, "Privacy models for big data: a survey," International Journal of Big Data Intelligence, vol. 3, no. 1, pp. 61–75, 2016.

[14] What is mapreduce. [Online]. Available: https://www.talend.com/resources/ what-is-mapreduce/

[15] C. Dwork et al., "The algorithmic foundations of differential privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014.

[16] C. Dwork et al., "Differential privacy in practice: Expose your epsilons!", Journal of Privacy ,2019.

[17] A. Wood et al., "Differential privacy: A primer for a non-technical audience," Vand. J. Ent. & Tech. L., vol. 21, p. 209, 2018.

[18] O. O'Malley et al., "Hadoop security design," Yahoo, Inc., Tech. Rep, 2009.

[19] D. Das et al., "Adding security to apache hadoop." hortonworks report," 2011.

[20] S. Y. Ko, K. Jeon, and R. Morales, "The hybrex model for confidentiality and privacy in cloud computing." HotCloud, vol. 11, pp. 8–8, 2011.

[21] K. Ashoka and B. Poornima, "Stipulation-based anonymization with sensitivity flags for privacy preserving data publishing," in Advances in Intelligent Systems and Computing, vol. 707, 2019, pp. 445–454.

[22] K. Ashoka and B. Poornima "A survey of latest developments in privacy preserving data publishing," 2014, doi:10.15693/ijaist/2014.v3i12.1423.

[23] K. Ashoka and B. Poornima "A survey of latest developments in privacy preserving data publishing," 2014, doi:10.15693/ijaist/2014.v3i12.1423.

[24] A. Machanavajjhala et al., "Privacy: Theory meets practice on the map," in 2008 IEEE 24th international conference on data engineering. IEEE, 2008, pp. 277–286.

[25] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 439–450.

[26] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 265–273.

[27] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," IEEE transactions on knowledge and data engineering, vol. 16, no. 9, pp. 1026–1037, 2004.

[28] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacypreserving data analysis," in Proceedings of the 2009 ACM SIGMOD International Conference.

[29] I. Roy et al., "Airavat: Security and privacy for mapreduce." in NSDI, vol. 10, 2010, pp. 297–312.

[30] R. Millman. (2017) Thousands of hadoop clusters still not being secured against attacks. [Online]. Available: https://www.scmagazineuk.com/thousands-hadoop-clusters-not-secured-against-attacks/article/1475302

[31] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," Knowledge and information systems, vol. 28, no. 1, pp. 47–77, 2011.

[32] K. Ashoka and B. Poornima, "Enhanced utility in preserving privacy for multiple heterogeneous sensitive attributes using correlation and personal sensitivity flags," 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, vol. 2017-Janua, pp. 970–976, 2017, doi: 10.1109/ICACCI.2017.8125967.