Implementation of Big Data Privacy Preservation Technique for Electronic Health Records in Multivendor Environment

Ganesh Dagadu Puri, D. Haritha
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,
Vaddeswaram, AP, India

***Abstract*—Various diagnostic health data formats and standards include both structured and unstructured data. Sensitive information contained in such metadata requires the development of specific approaches that can combine methods and techniques that can extract and reconcile the information hidden in such data. However, when this data needs to be processed and used for other reasons, there are still many obstacles and concerns to overcome. Modern approaches based on machine learning including big data analytics, assist in the information refinement process for later use of clinical evidence. These strategies consist of transforming various data into standard formats in specific scenarios. In fact, in order to conform to these rules, only de-identified diagnostic and personal data may be handled for secondary analysis, especially when information is distributed or transferred across institutions. This paper proposes big data privacy preservation techniques using various privacy functions. This research focused on secure data distribution as well as security access control to revoke the malicious activity or similarity attacks from end-user. The various privacy preservation techniques such as data anonymization, generalization, random permutation, k-anonymity, bucketization, l-diversity with slicing approach have been proposed during the data distribution. The efficiency of system has been evaluated in Hadoop distributed file system (HDFS) with numerous experiments. The results obtained from different experiments show that the computation should be changed when changing k-anonymity and l-diversity. As a result, the proposed system offers greater efficiency in Hadoop environments by reducing execution time by 15% to 18% and provides a higher level of access control security than other security algorithms.**

***Keywords*—Privacy preservation; data privacy; data distribution; anonymization; slicing; privacy attacks; HDFS**

I. INTRODUCTION

The widespread access to virtual health documents and assets controlled through Health Information Systems (HISs) enables to strengthen scientific studies, support care centers through medical education. It also assists various groups for fitness methods through management and governance (e.g., scientific audits for high-satisfactory development and care coordination) [1], [2]. Furthermore, health data is often processed and linked with different data sources, along with facts from scientific trials, having allowed for the harvesting of extra insights which might be beneficial. It is now no longer most effective for scientific practices, however additionally for designing and improving facts fitness structures and contributing to greater efficient policymaking [3]. While the benefits of secondary use of studies and nursing data makes vital for boosting the high-satisfactory of treatment. There are nevertheless several uncertainties regarding how this data is accessible, through whom, and under what conditions. Data sharing (in particular while it includes private and sensitive attributes and is made accessible to the third party enterprise or geographical region in which it's miles engendered) can bring about a lack of privacy for individuals, along with users and health professionals, further to the requirement to gain earlier than taking part in research [4]. As a result, data safety and affected person privacy are vital demanding situations to address.

Another problem in processing health data for secondary applications is informational discrepancies and diversity in clinical records and data. This is related to the lack of uniform data representation. Hospital Information System (HIS) handles both structured and unstructured documents. Although there are still many unresolved issues that may hinder the use of hybrid materials, especially disjointed health data, they are made possible by leveraging and efficiently integrating structured and semi-structured health information. There is a distinct position arising from their complementary use. In a single, structured way while considering data responsiveness, privacy can hamper, and ethical concerns raise [5]. Digital evidence should be transformed into a unified, standardized, and codified representation using modern methods such as big data analysis [6]. Proper use of this data requires the use of appropriate anonymization procedures. This paper presents a novel integrated architecture for collecting clinical data from heterogeneous sources and transforming them into formats useful for clinical secondary use while adhering to the above requirements. Proper distribution of attribute weightage is also important [7].

The main contributions of this paper are:

- Protect the distribution of large heterogeneous datasets with a novel approach to maintaining privacy in Hadoop environments.
- Defeat various network and database attacks such as SQL injection, collusion attacks and similarity attacks.

The rest of Section II describes state-of-the-art systems that demonstrate previous work by various authors. The Section III

describes the research methods used for the proposed system, including a detailed description of the proposed architecture and execution flow. Section IV provides a description of the algorithm and determines how to handle large data distribution tasks. This section describes data protection against internal attacks and data security algorithms against external attacks. Section V focuses on the results and detailed discussion. Extensive experimental analysis and obtained results are defined in tabular and graphical formats. Finally, Section VI describes the conclusions of the proposed system and future issues.

II. Literature Survey

This section describes various state-of-the-art systems used by previous researchers. According to [8], the newly proposed Secured Map Reduce (SMR) layer introduces a security and privacy layer between HDFS and the MR (Map Reduce) layer, and this approach is known as SMR model. A major value in this work is to facilitate data exchange for knowledge mining. This architecture ensures privacy and security for data consumers, addresses privacy scaling issues, and maintains a trade-off between privacy and utility. SMR models significantly reduce runtime and information loss compared to traditional techniques, and minimizes CPU and memory consumption. According to the work proposed in [9][10], current PPDM strategies have been exhaustively investigated and classified based on data modification methods. This is the researcher's main contribution and will help researchers in the field to fully understand the PPDM. In addition, they compared and considered the advantages and limitations of various PPDM approaches. The vast increase in customer data retention has spawned a new field of research known as privacy-preserving data mining (PPDM). The fundamental challenge of PPDM is to modify data using specific techniques and create powerful data mining models of the modified data while meeting specified privacy requirements and ensuring that information is available for intended data analysis activities. Current review studies aim to leverage data mining jobs without compromising the security of people's sensitive information, especially at the record level.

In the research work proposed in [11], the authors provide a well-designed taxonomy that allows systematic and rigorous classification of this difficult research subject. Recently, the term “big data” has become popular. The proliferation of social networks, the Internet of Things (IoT), and the outsourcing of cloud computing have created an incredible amount, velocity, and variety of data. According to [12], authors proposed Mondrian-based k-anonymity method. A deep neural network (DNN)-based architecture is presented to protect the privacy of high-dimensional data. Experimental results show that the proposed method reduces data information loss while preserving privacy. Many companies actively or passively collect data from consumers. It also collects personal data from various databases.

This data includes personally identifiable information (PII) that can be used to identify an individual. Data analysts and researchers have paid much attention to protecting privacy in the explosion of data for big data and cloud computing. Numerous data anonymization strategies and DNN privacy models have been thoroughly researched.

In research activities [13], data analysts and academics have paid much attention to privacy-aware data distribution for big data and cloud computing. Various data anonymization approaches and DNN models have been thoroughly researched to protect privacy. A public identity-based PDP protocol for secure data storage helps to protect the privacy of many users. This approach allows TPA to correctly assess the integrity of group-shared data. According to [14], it is a privacy-preserving cloud-based mobile multimedia data exchange system with attribute names and values for each attribute, and only attribute names are visible in access policies. However, the attribute values are included in the cipher text. Encryption has two phases online and offline. Data owners can prepare the intermediate cipher text components in an offline step. After receiving a specific access policy and multimedia data encryption request, the data owner can quickly create the final legal cryptogram in an online phase. Most of the processing costs for verification testing and decryption are offloaded to cloud servers using a decryption outsourcing approach. According to the safety case, PPCMM is adaptively safe in the standard model.

In research work [15], three strategies are used to ensure data confidentiality and integrity. It describes homomorphic encryption, order-preserving encryption schemes, and attribute-based encryption. These strategies are best used in the cloud to ensure privacy. It is also ideal for big data to maintain efficiency and scalability for huge datasets for decision making. Big data is a vast accumulation of enormous data sets that cannot be analyzed by ordinary computing techniques. Big data is therefore a vast amount of rapidly changing data with mixed data types. According to [16], IoT deployments in many industries meet the privacy challenges faced by IoT in resource-constrained devices. It provides an opportunity to address some of the uses of blockchain in various fields and IoT privacy concerns. Based on the utilization of blockchain in IoT, authors proposed various research studies. This study aims to review current research on blockchain applications in IoT for privacy protection. After reviewing current solutions, it was determined that blockchain is the most effective way to avoid identity disclosure, surveillance, and tracking in IoT.

According to [17], a new privacy utility approach uses lightweight elliptic curve cryptography (ECC) to protect privacy and particle swarm optimization (PSO) clustering to maintain utility. PSO is used to cluster datasets and ECC is used to ensure confidentiality of clustered datasets. The proposed method is tested on medical datasets and compared to other methods based on various performance criteria such as clustering accuracy, F-measure, data usefulness, and privacy metrics. According to [18], the various impacts of privacy laws on forensic investigations of embedded devices, the role of anti-forensics, and proposals for embedded forensic investigation guidelines and initiatives to address privacy concerns.

They also proposed a SMART system that enables built-in digital forensic investigations to protect privacy at every level of traditional forensic frameworks. This protects cooperation

with unincorporated owners of embedded computers in cybercrime investigations. According to Slawomir Goryczka et al. [19], it is considered an insider attack against a number of data providers that insert records and attempts to take closure through covert attacks against records inserted by other providers. In this work, the authors proposed a secure multiparty computing protocol for ensuring privacy across multiple data providers. A research paper proposed in [20] identifies privacy issues related to big data and its use. It has been suggested that processing different types of data through different channels poses different threats to user privacy. The authors of [21] emphasized a no-delay framework for the release of medical records to protect privacy. The usefulness of published data is enhanced by a late validation approach. Similarity and skewness attacks could be possible when forming sensitive value groups for publishing records. In a research paper [22], [23], the authors proposed a framework for processing streaming data and measuring term similarity within groups using standard means and addressed the similarity attack issue.

III. Proposed System Design

Fig. 1 below shows the multivendor environment and different constraints for data collection. First, it collects synthetic and real-time data from a variety of sources, including medical systems, historical data, and runtime data collection from various web applications. This data may include sensitive user information. Also, records collected; need to create a privacy view when sending data. Malicious user can predict the data of provider by using background knowledge. There is need of the study that defines some security policies on the pre-trained model for secure data distribution. The problem of database attacks and privacy violations on privacy view works like a data hiding technique should be eliminated.

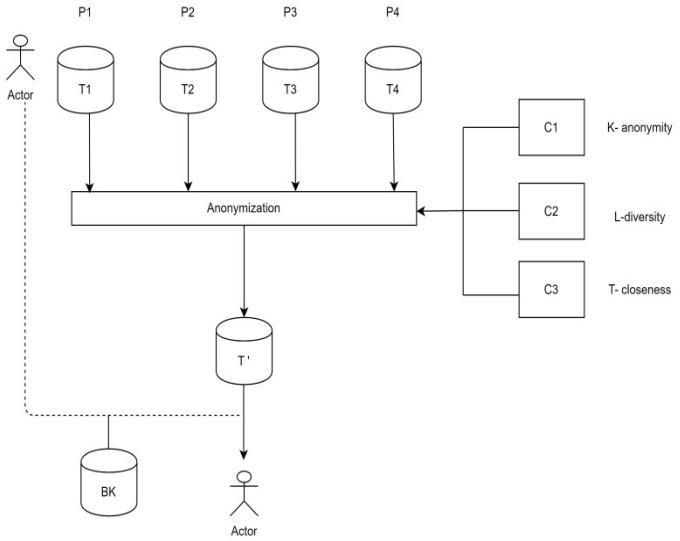

Fig. 1. Multivendor environment and different constraints for data collection

Fig. 2 shows the proposed system architecture with strategic execution. In data providers block, data collected from a variety of sources. It includes various systems such as medical systems, historical data, and runtime data collection from various web applications. As data collected is very large and considered as big data, nodes are added to process such large data. Using HDFS name nodes are added for processing of data as shown in Fig. 2. Data nodes will provide information for processing to generate privacy view and at the same time detect attack and attacker. Name nodes are keeping directory view of all files in HDFS. Data nodes are sending information to these name nodes and respond to name node in all file operation of privacy view and attack prevention system. This data may include sensitive user information. This study defines some security policies on the pre-trained model for secure data distribution. Privacy view works like a data hiding technique that eliminates the problem of database attacks and privacy violations.

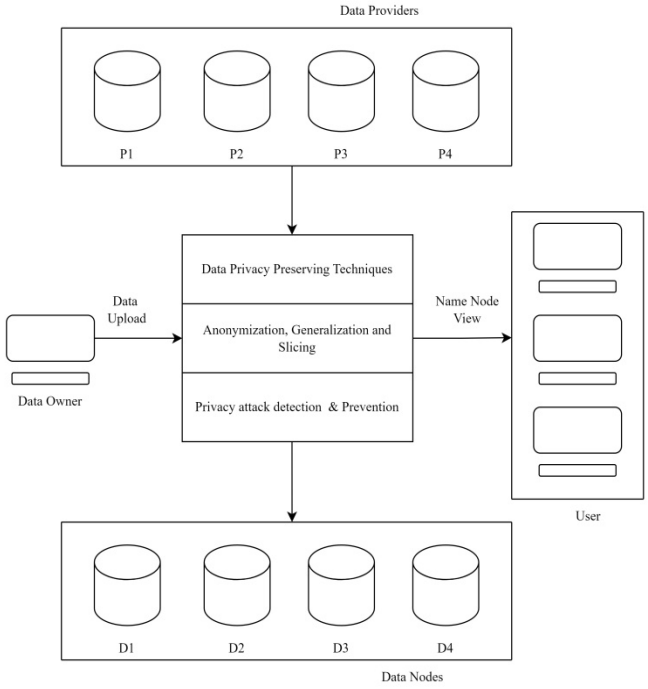

Fig. 2. System architecture for privacy preservation and privacy attack detection in distributed environment

A. Techniques for Maintaining Privacy

Anonymization, generalization and slicing are the main methods used in the proposed model. In the anonymization, L-diversity and K- anonymity methods are used. To increase the anonymization level, top-down and bottom-up generalization is used. Slicing basically relies on splitting attributes and tuples. For attribute split (vertical split), split data as {name},{age-zip},{gender} and for tuple split (horizontal split), split as {t1,t2,t3,t4,t5,t6}. In attribute partitioning, age and zip code are quasi-identifiers (QIs), so they are highly correlated and thus partitioned together. These QIs may be known to the attacker. A tuple partitioning system should check the sensitive attribute (SA) column for L-diversity. Equivalence groups are created using K values of quasi identifiers in such a way that they indistinguishable from each other. In that equivalence class L different sensitive values will be added to make it impossible to identify value of individual from anonymized data.

B. Privacy Attack Detection and Prevention

Another important feature of this system is the detection of privacy attacks and the prevention of the use of defined algorithms. End users can perform insider, collusion, sql injection and similarity attacks by making minor changes to update the actual value and making it available to another user. Using privacy protection and fingerprint generation technology, such attacks can be easily detected and prevented efficiently

IV. Algorithm Design

A. Privacy View Generation

It is similar to one-way hash functions to generate the privacy view. The goal of the algorithm presented is to keep sensitive data secure and avoid privacy intrusions. As a result, anonymous views on miniature buckets are generated. In algorithm step 1 to step 6 are used to read the input record wise and apply generalization on quasi-identifiers of the records. While performing anonymization on entire set of quasi-identifiers, validation of that is done using K-anonymity. The records which do not satisfy the criteria of anonymization are added in bucket. Permutation is applied on records so that more records get anonymized. Step 7 to step 12 are applied for the pruning and creating final bucket after anonymization. Bucketization is used to avoid leakage of the records in case of privacy preserved view. It stores records which do not satisfy constraints after pruning and permutation methods.

Algorithm 1: Algorithm for privacy view generation

Input: Input dataset DSet, total number of data providers Dp, Constraint policy C {K_Anonimity, L_Diversity}
Output: Privacy view (NT*) with selective provider

Step 1: foreach (DSet till null)

Step 2: foreach (col in table)

foreach (row in table)

Step 3: Select quasi identifier (QiF) and set of sensitive attributes (S_Att)
Step 4: Executes generalization to classify the tuples in QiF with multiple groups
Step 5: Perform anonymization on entire set of attributes

Step 6: While **(**validate data privacy(DSet, Dp, C) = 0) do

if (DSet[i] ← DSet) validated with QiF then

add D[i] till K-anonimity

else break;

Bucket_List(i1) → DSet;

Step 7: Apply permutation on dataset (DSet[i]=(I(null-1)))

Step 8: Apply Pruning on(DSet)

Step 9: Execute step 1,2,3 on Bucket_List (i1)

Step 10: if (C != (DSet) && (Dp # 1))

Bucket(i2) → Bucket_List (i1(j))

Step 11: Show (Bucket_List (i2)!=null)

Step 12: end while

Step 13: end for

B. Algorithm 2

The top-down and bottom-up Algorithm 2 is similar to the base-up method. The main difference is in how coalition checks are performed, starting with 0-foe and working up.

Algorithm 2: Algorithm Top down and Bottom up generalization view

Input: Input dataset DSet, total number of data providers Dp, Constraint policy C {K_Anonimity, L_Diversity}
Output: Privacy view (NT*) with selective provider

Step 1 : Read data from dataset from bottom set or top set

$$data[\,] = \sum_{n=1}^{m} (\text{Row } [n])$$

Step 2: Check data count with K-anonimity and L-diversity for each block
Step 3: calculate the fitness score F_Score(DataSet[])

Step 4 : if (F_Score >= Th)

Generate best generalized view as T*

Step 5 : end loop

Step 6: return T*;

When an infringement by any foe is detected (early stop) or all m-policies are examined, the algorithm comes to a cessation. The algorithm represents the basic idea of a bottom-up speculating approach. Using K-anonimity and L-diversity for each block in the dataset, fitness score is checked to generate the best generalized view of privacy preservation. This score is checked against the threshold value for the anonymization.

V. Results and Discussions

The proposed system is implemented using Java 1.8 and NetBeans 8.0 in an open source Hadoop 2.0 environment. The Intel 2.7 GHz hardware setup is done with 12 GB of RAM. The Hadoop setup is done with a name node and two data nodes using a MapReduce process. According to the problem description, the results obtained are demonstrated using a standalone machine and a Hadoop system. The input dataset size is 101850 instances for both experiments, with and without Hadoop environment. Anonymized views are generated using a definition of C constraints, including K-anonymity and L-diversity. Execution time is measured in milliseconds. Publishing the records using constraints is tedious task if the input data is large. In many cases data need to be preprocessed before giving to the privacy preservation system. This increased data with the need of preprocessing and formatting cannot be executed timely by existing infrastructure and framework.

TABLE I. EXECUTION TIME FOR TRADITIONAL MACHINE WITH VARIOUS L AND K VALUES

Measures	Slicing	Bucket	Final Bucket
L=8 and K=10	11586	1109	184
L=9 and K=12	11776	1316	187
L=10 and K=15	13330	1541	182
L=11 and K=13	11891	1257	188
L=12 and K=15	12950	1550	388

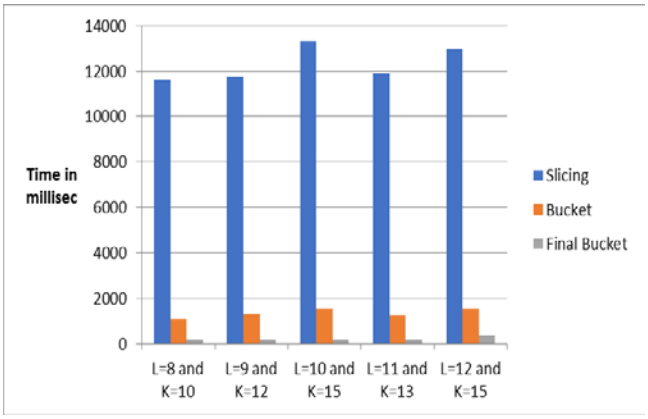

Fig. 3. Execution time for traditional machine with various L and K values

Table 1 and Fig. 3 show the time required to generate T* views as slicing, bucket generation and final bucket generation. This experiment was conducted on a typical configuration of a single machine. As a result, the execution time of all three processes increased even as the values of L and K increased.

TABLE II. EXECUTION TIME FOR HDFS WITH VARIOUS L AND K VALUES

Measures	Slicing	Bucket	Final Bucket
L=8 and K=10	9827	225	222
L=9 and K=12	7178	262	220
L=10 and K=15	8773	271	186
L=11 and K=13	9354	244	223

After the privacy view generation, few records do not satisfy the constraints set by L-diversity and K-anonymity. If these records are dropped in the system, there is significant loss of information. In this proposed system, these records are stored in the bucket. Bucket is storage area where we can apply the privacy view generation constraints again. Few records may not satisfy the constraints in this stage also. Again dropped records are stored in the final bucket and applied with L-diversity and K-anonymity criteria. For different values of K and L execution time is varying. This execution time is measured in milliseconds.

Fig. 4 shows the time required to generate T* views using different K and L values mentioned in Table 2. Graph shows the execution time in the Hadoop environment. Execution time on Hadoop is about 15-18% faster than on a standalone machine with a similar dataset. Increasing the L and K value will increase the execution time subsequently on traditional machine. On HDFS also there is increase in execution time with the increase in these values. Time for privacy view generation, bucket formation and final bucket formation is considered in the execution time.

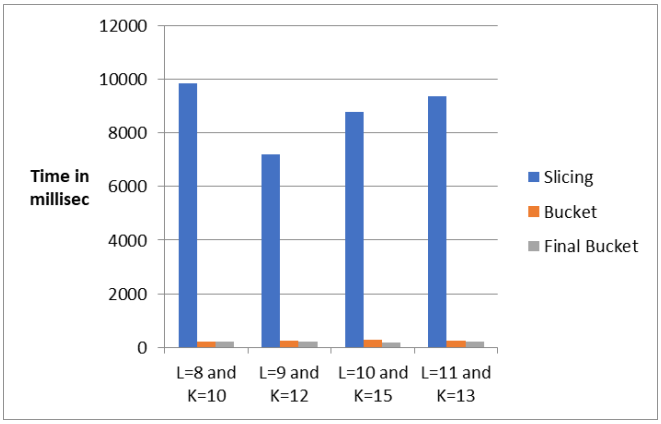

Fig. 4. Execution time for HDFS with various L and K values

TABLE III. EXECUTION TIME FOR TRADITIONAL MACHINE WITH CONSTANT (L=8)

Measures	Slicing	Bucket	Final Bucket
L=8 and K=10	11586	1109	184
L=8 and K=11	12994	299	174
L=8 and K=12	13110	1180	180
L=8 and K=13	11123	1229	196

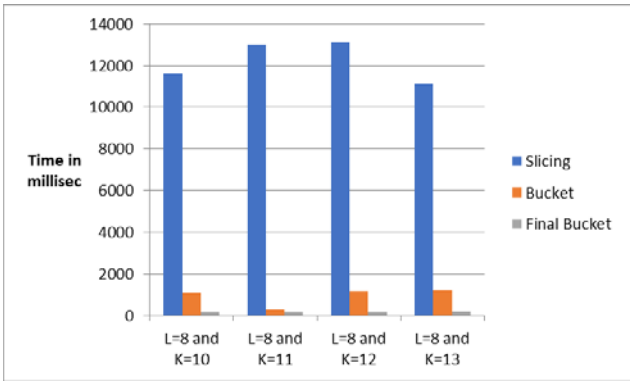

Fig. 5. Execution time for traditional machine with constant (L=8)

Table 3 and Table 4 are enlisting the values for constant L value. Experiment is carried out using constant value 8. In this experiment, value of L-diversity is kept constant and varying k-anonymity values to illustrate the time required generating anonymized views. In this experiment, there is some variation in run time when both values are changed. As only K value is changed, more number of records will appear and available for creation of L-diverse sensitive group. So with increase in k values execution time is reduced.

TABLE IV. EXECUTION TIME FOR HDFS WITH CONSTANT (L=8)

Measures	Slicing	Bucket	Final Bucket
L=8 and K=10	8735	263	208
L=8 and K=11	10362	325	214
L=8 and K=12	9861	281	227
L=8 and K=13	7675	238	213

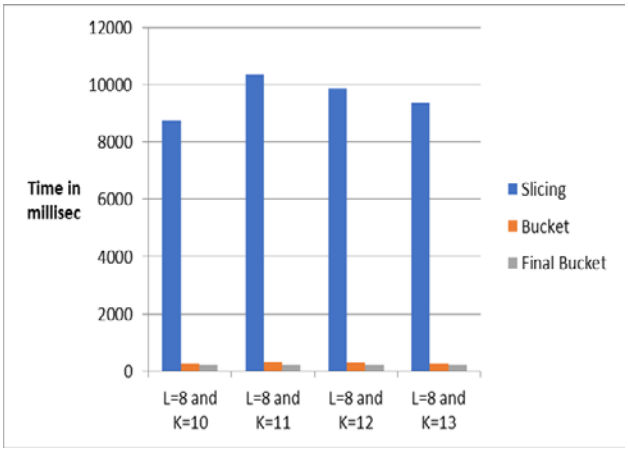

Fig. 6. Execution time for HDFS with constant L-diversity

The Fig. 6 describes the similar execution according to Fig. 5 in Hadoop environment. Almost 24% execution time is reduced using HDFS based a parallel execution.

TABLE V. EXECUTION TIME FOR TRADITIONAL MACHINE WITH CONSTANT (K=15)

Measures	Slicing	Bucket	Final Bucket
L=10 and K=15	13330	1541	182
L=11 and K=15	9500	1534	174
L=12 and K=15	8184	1540	188
L=13 and K=15	10424	1601	192

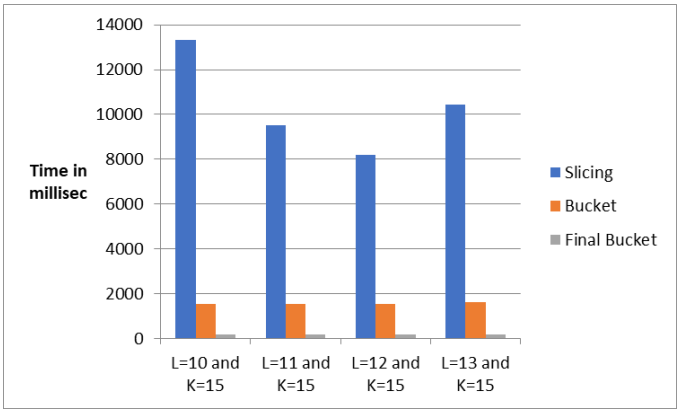

Fig. 7. Execution time for traditional machine with constant k-anonymity

Execution time for traditional machine with constant (K=15) is listed out in Table 5. Fig. 7 shows the generation of T* views with constant K-anonymity and different L-diversity using a standalone machine. The time required to create the final anonymized view of slices and buckets is reduced, even if the K and L values change. As multiple providers are providing electronic health records with disease as sensitive attribute, insider attack is possible. M-privacy algorithm takes care of collusion attack which takes place in multivendor environment [19]. But M-privacy algorithm cannot work on large scale data. In this research slicing and bucketization techniques are applied on big data using hadoop distributed file system.

TABLE VI. EXECUTION TIME FOR HDFS WITH CONSTANT K AND DIFFERENT L VALUES

Measures	Slicing	Bucket	Final Bucket
L=10 and K=15	9109	260	461
L=11 and K=15	6555	269	232
L=12 and K=15	7947	272	225
L=13 and K=15	7755	242	219

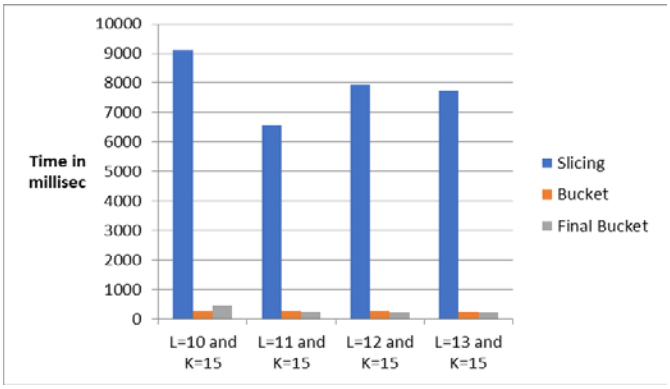

Fig. 8. Execution time for HDFS with constant k and different L values

Table 6 and Fig. 8 above show the generation of privacy views with different values of L-diversity and constant K-anonymity in the Hadoop framework. Slightly changing L-diversity by the k constant does not make much difference. In many cases, the input data cannot satisfy the constraints of C, which can lead to longer times.

VI. CONCLUSION

In this paper, various privacy-preserving techniques for large-scale health datasets in distributed environments are implemented. Various privacy techniques such as data anonymization, generalization, random permutation, slicing and fingerprinting are used to protect and eliminate privacy attacks. This system provides maximum security in HDFS-based distributed environments and standalone systems. This approach is effective when dealing with real-time data containing sensitive information. Experiments are evaluated on the entire execution using synthetic and real-time healthcare datasets. The system provides 100% privacy with privacy-preserving technology while maintaining the highest accuracy in privacy-based data delivery. Implementation of various machine learning techniques for distributed dynamic data security is a future challenge for the system.

REFERENCES

[1] J. Walonoski et al., “Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic

health care record," J. Am. Med. Informatics Assoc., vol. 25, no. 3, pp. 230–238, 2018, doi: 10.1093/jamia/ocx079.

[2] M. Asfaw, K. Yitbarek, and J. Gustav, "Emnet : a system for privacy-preserving statistical computing on distributed health data," Ep.Liu.Se, no. June 2015, 2015, [Online]. Available: http://www.ep.liu.se/ecp/115/006/ecp15115006.pdf.

[3] E. Hutchings, M. Loomes, P. Butow, and F. M. Boyle, "A systematic literature review of attitudes towards secondary use and sharing of health administrative and clinical trial data: a focus on consent," Syst. Rev., vol. 10, no. 1, 2021, doi: 10.1186/s13643-021-01663-z.

[4] F. Earls and S. Cook, "INTEGRATED ADDENDUM TO ICH E6(R1): GUIDELINE FOR GOOD CLINICAL PRACTICE," Child Psychiatry Hum. Dev., vol. 13, no. 4, 1983.

[5] M. Tayefi et al., "Challenges and opportunities beyond structured data in analysis of electronic health records," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 13, no. 6. 2021, doi: 10.1002/wics.1549.

[6] S. Silvestri, A. Esposito, F. Gargiulo, M. Sicuranza, M. Ciampi, and G. De Pietro, "A big data architecture for the extraction and analysis of EHR data," 2019, doi: 10.1109/SERVICES.2019.00082.

[7] G. D. Puri and D. Haritha, "Improving Privacy Preservation Approach for Healthcare Data using Frequency Distribution of Delicate Information," vol. 13, no. 9, pp. 82–90, 2022. (DOI) : 10.14569/IJACSA.2022.0130910

[8] P. Jain, M. Gyanchandani, and N. Khare, "Enhanced Secured Map Reduce layer for Big Data privacy and security," J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0193-4.

[9] M. Binjubeir, A. A. Ahmed, M. A. Bin Ismail, A. S. Sadiq, and M. Khurram Khan, "Comprehensive survey on big data privacy protection," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2962368.

[10] P. GaneshD, P. Dinesh D, and W. Manoj A., "RAID 5 Installation on Linux and Creating File System," Int. J. Comput. Appl., vol. 85, no. 5, pp. 43–46, 2014, doi: 10.5120/14841-3107.

[11] H. Y. Tran and J. Hu, "Privacy-preserving big data analytics a comprehensive survey," J. Parallel Distrib. Comput., vol. 134, 2019, doi: 10.1016/j.jpdc.2019.08.007.

[12] J. Andrew, J. Karthikeyan, and J. Jebastin, "Privacy Preserving Big Data Publication on Cloud Using Mondrian Anonymization Techniques and Deep Neural Networks," 2019, doi: 10.1109/ICACCS.2019.8728384.

[13] H. Yan and W. Gui, "Efficient Identity-Based Public Integrity Auditing of Shared Data in Cloud Storage with User Privacy Preserving," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3066497.

[14] Q. Li, Y. Tian, Y. Zhang, L. Shen, and J. Guo, "Efficient Privacy-Preserving Access Control of Mobile Multimedia Data in Cloud Computing," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2939299.

[15] H. Shekhawat, S. Sharma, and R. Koli, "Privacy-preserving techniques for big data analysis in cloud," 2019, doi: 10.1109/ICACCP.2019.8882922.

[16] Z. Iftikhar et al., "Privacy preservation in resource-constrained iot devices using blockchain—a survey," Electronics (Switzerland), vol. 10, no. 14. 2021, doi: 10.3390/electronics10141732.

[17] N. Yuvaraj, R. Arshath Raja, and N. V. Kousik, "Privacy Preservation Between Privacy and Utility Using ECC-based PSO Algorithm," in Advances in Intelligent Systems and Computing, 2021, vol. 1172, doi: 10.1007/978-981-15-5566-4_51.

[18] J. Pathak, S. Sankaran, and K. Achuthan, "A SMART Goal-based Framework for Privacy Preserving Embedded Forensic Investigations," 2019, doi: 10.1109/ISED48680.2019.9096232.

[19] S. Goryczka, L. Xiong, and B. C. M. Fung, "M-Privacy for Collaborative Data Publishing," IEEE Trans. Knowl. Data Eng., vol. 26, no. 10, 2014, doi: 10.1109/TKDE.2013.18.

[20] G. D. Puri and D. Haritha, "Survey big data analytics, applications and privacy concerns," Indian J. Sci. Technol., vol. 9, no. 17, 2016, doi: 10.17485/ijst/2016/v9i17/93028.

[21] S. Kim, M. K. Sung, and Y. D. Chung, "A framework to preserve the privacy of electronic health data streams," J. Biomed. Inform., vol. 50, pp. 95–106, 2014, doi: 10.1016/j.jbi.2014.03.015.

[22] G. D. Puri and D. Haritha, "Framework to avoid similarity attack in big streaming data," Int. J. Electr. Comput. Eng., vol. 8, no. 5, 2018, doi: 10.11591/ijece.v8i5.pp.2920-2925.

[23] G. D. Puri and D. Haritha, "A novel method for privacy preservation of health data stream," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 4, pp. 4959–4963, 2020, doi: 10.30534/ijatcse/2020/110942020.