

Privacy preserving of Healthcare Data in Internet of Medical Things



By

Bakhtawar Mudassar
(Registration No: 00000329266)

Department of Information Security

Military College of Signals

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2024)

Privacy preserving of Healthcare Data in Internet of Medical Things



By

Bakhtawar Mudassar

(Registration No: 00000329266)

A thesis submitted to the National University of Science and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Masters in

Information Security

Supervisor: Assoc. Prof.Dr. Shahzaib Tahir

Co-Supervisor: Asst. Prof. Dr. Fawad

Military College of Signals

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2024)


THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by Ns Bakhtawar Mudassar, Registration No. 00000329266, of Military College of Signals has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

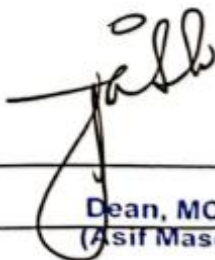
Signature:  _____

Name of Supervisor: Dr. Shahzaib Tahir

Date: _____


Signature (HOD): _____
HoD
Information Security
Military College of Sigs

Date: _____

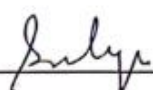
Signature (Dean/Principal)  _____
Date: 22/10/24 _____
Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY
MASTER THESIS WORK


We hereby recommend that the dissertation prepared under our supervision by **NS Bakhtawar Mudassar, MSIS-19 Course** Regn No **00000329266** Titled: "**Privacy preserving of Healthcare Data in Internet of Medical Things.**" be accepted in partial fulfillment of the requirements for the award of **MS Information Security** degree.

Examination Committee Members


1. Name: **Dr. Sadiqah Arshad**

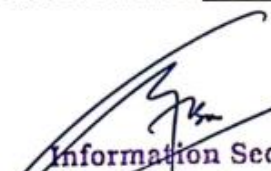
Signature: 

2. Name: **Lecturer Anum Hassan**

Signature: 

Supervisor's Name: **Dr. Shahzaib Tahir**

Signature: 



HoD
Information Security
Military College of Sigs
Head of Department

Date: _____

_____ Date

COUNTERSIGNED

Date: 22/10/24


Brig
Dean, MCS (NUST)
(Asif Masood, Phd)
Dean

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "Privacy preserving of Healthcare Data in Internet of Medical Things," was conducted by Bakhtawar Mudassar under the supervision of Dr. Shahzaib Tahir. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Military College of Signals, National University of Science & Technology Information Security Department in partial fulfillment of the requirements for the degree of Master of Science in Field of Information Security Department of information security National University of Sciences and Technology, Islamabad.

Student Name: Bakhtawar Mudassar


Signature: 

Examination Committee:

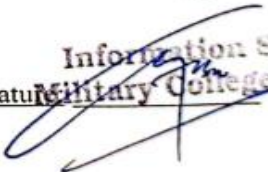
a) External Examiner 1: Name Asst Prof Dr Sadiqa Arshad. (MCS) Signature: _____

b) External Examiner 2: Name Lec Anum Hasan.(MCS) Signature  _____

Name of Supervisor: Dr. Shahzaib Tahir


Signature: 

Name of Dean / HOD. Dr Muhammad Faisal Amjad

Signature: 
 Information Security
Military College of Signals

AUTHOR'S DECLARATION

I Bakhtawar Mudassar hereby state that my MS thesis titled Privacy preserving of Healthcare Data in Internet of Medical Things, is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world. At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.


Student Signature:  _____
Name: Bakhtawar Mudassar
Date: _____

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled **Privacy preserving of Healthcare Data in Internet of Medical Things**.is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/ revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: 
Name: Bakhtawar Mudassar
Date: _____

DEDICATION

“In the name of Allah, the most Beneficent, the most Merciful”

This research work is dedicated

to

MY PARENTS, TEACHERS AND HUSBAND

for their love, endless support, and encouragement

ACKNOWLEDGEMENTS

I am profoundly grateful to Allah Almighty for granting me the strength and perseverance to complete this thesis despite the many challenges and obstacles I encountered. All praises are for HIM and HIM alone.

I would like to express my sincere gratitude to my Thesis Supervisor, Asst Prof Dr. Fawad Khan, and my Co-Supervisor, Asst Prof Dr. Shahzaib Tahir. Their guidance, encouragement, and invaluable support have been instrumental in achieving the aims of my research. I am also thankful to the GEC members for their constructive feedback and supervision throughout this journey.

Special thanks are due to my father, Mudassar Ali, and my mother, Naila Mudassar, for their unwavering care, love, and support during moments of stress and excitement. Their belief in me has been a constant source of motivation.

I would also like to extend my heartfelt appreciation to my husband, M. Ashraf, whose patience, understanding, and encouragement have been vital to the completion of this thesis.

Lastly, I wish to acknowledge all those who have contributed to my study and supported me along the way, even though I may not have mentioned their names individually. Your assistance and encouragement have been greatly appreciated.

Thank you all for your contributions and support.

ABSTRACT

Protecting patient privacy in the era of digital health records is a major challenge while allowing healthcare data to be useful. The study of differential privacy as a strong privacy-preserving method for medical data is examined in this thesis. Differential privacy is a mathematical framework that prevents sensitive information from being disclosed while preserving data utility for analysis. It does this by introducing controlled noise to the data. Thus, the main objective of this thesis is privacy preserving of healthcare data in internet of things by using differential privacy. This research aims to propose a secure, privacy-preserving scheme to ensure maximum privacy of an individual by also maintaining its utility and allowing to perform queries based on sensitive attributes under differential privacy. This mechanism guarantees the individual's privacy by consuming minimum computation and communication costs. We have designed a basic framework that tries to achieves differential privacy guarantee and evaluate the results regarding the level of privacy can be achieved in electronic healthcare data. For this purpose, we have practically implemented differential privacy on two different publicly available datasets such as Breast Cancer Prediction Dataset and the Nursing Home COVID-19 Dataset. By applying differential privacy mechanisms to these datasets, it evaluates the balance between privacy and data utility, demonstrating the effectiveness of differential privacy in real-world healthcare scenarios. Additionally, we have conducted time comparison by performing multiple complex queries on these datasets to analyze the computational overhead introduced by differential privacy. The outcomes demonstrate that, despite a slight increase in query processing time, it remains within reasonable bounds, ensuring the practicality of differential privacy for real-time applications. A significant part of this study involves the selection of the privacy parameter ϵ , which determines the degree of privacy protection. Moreover, we have examined the impact of varying ϵ values on both the privacy and utility of the data. Our experiments demonstrate that a lower ϵ value enhances privacy at the cost of reduced data utility, whereas a higher ϵ value offers better utility but with less privacy protection. This trade-off analysis provides crucial insights into optimizing ϵ for different healthcare data use cases.

The findings of this thesis contribute to the increasing corpus of information on privacy-preserving data analysis in the healthcare industry by providing useful suggestions and insights for using differentiated privacy in various healthcare data scenarios. This work underscores the importance of adopting advanced privacy-preserving techniques to foster trust and compliance in healthcare data sharing and analytics.

Keywords: Differential Privacy, Healthcare data, Data sharing, User Privacy, Data Utility.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	VII
ABSTRACT	VIII
LIST OF FIGURES	XI
LIST OF TABLES	XII
LIST OF ABBREVIATIONS	XIII
INTRODUCTION	1
1. Overview	1
1.1. Motivation.....	2
1.2. Research Objectives	4
1.3. Contribution.....	4
1.4. Thesis Outline.....	5
LITERATURE REVIEW	7
PRELIMINARIES OF PRIVACY PRESERVATION	19
3. Privacy Preservation Models.....	19
3.1. Anonymization	19
3.2. K-Anonymity	20
3.3. I-Diversity	20
3.4. T-Closeness.....	21
3.5. Cryptographic techniques	21
3.6. Multidimensional Sensitivity Based Anonymization	22
3.7. Data Distribution technique	22
DIFFERENTIAL PRIVACY	25
4.1. Differential Privacy	25
4.1.1. Definition.....	25
4.1.2. Sensitivity of queries:	26
4.2. The Privacy Budget	26
4.2.1. Sequential Composition:.....	26
4.2.2. Parallel Composition:	27
4.3. Mechanisms of Differential Privacy.....	27
4.3.1. Laplace Mechanism:	27
4.3.2. Gaussian Mechanism:	29
4.3.3. Exponential Mechanism:	30
4.4. Methods to Implement Differential Privacy.....	30

4.4.1. Local Differential Privacy:	31
4.4.2. Global Differential Privacy:	31
4.5. Differentially Private Data Release	32
4.5.1. Interactive Data Release	33
4.5.2. Non-Interactive Data Release	33
4.6. Selection of Privacy Parameter ϵ	34
EXPERIMENTAL STUDY OF DIFFERENTIAL PRIVACY ON EHRs	37
5.1. Experimental DP Framework in Healthcare	37
5.2. Algorithm Details	38
5.2.1. Algorithm for Laplace Mechanism.....	38
5.2.2. Algorithm for Gaussian Mechanism.....	39
5.3. Datasets Description.....	39
5.3.1. COVID-19 Home Nursing Dataset.....	39
5.3.2. Breast Cancer Prediction Dataset	39
5.4. Experimental Results on Breast Cancer Prediction Dataset.....	40
5.4.1. Varying Privacy Budget using Breast Cancer Prediction Dataset.....	42
5.4.2. Time complexity Analysis with Breast Cancer Prediction Dataset.....	44
5.4.3. Comparison Analysis of Laplace vs Gaussian Mechanism.....	46
5.5. Experimental Results on COVID-19 Home Nursing Dataset	47
5.5.1. Varying Privacy Budget using COVID-19 Home Nursing Dataset	53
5.5.2. Time complexity Analysis with Nursing Home COVID-19 Dataset	54
CONCLUSION AND FUTURE WORK	58
6. Conclusion.....	58
6.1. Future Work.....	58
BIBLIOGRAPHY	60

LIST OF FIGURES

Figure 1 Laplace Distribution	28
Figure 2 Laplace vs Global Differential Privacy	32
Figure 3: Methods of Data Release.....	34
Figure 4 Architecture of the system.....	37
Figure 5 Breast Cancer Prediction attributes	40
Figure 6 Actual count without DP on Breast Cancer Prediction Dataset	41
Figure 7 Count with DP on Breast Cancer Prediction Dataset	41
Figure 8 Results Comparisons using Breast Cancer Prediction Dataset.....	42
Figure 9 Varying Epsilon values on Breast Cancer Prediction Dataset.....	43
Figure 10 Analysis of privacy parameter using Breast Cancer Prediction Dataset	43
Figure 11 Queries with Time Comparison using Breast Cancer Prediction Dataset	44
Figure 12 Time Comparison Query 1 on Breast Cancer Prediction Dataset	45
Figure 13 Time Comparison Query 2 on Breast Cancer Prediction Dataset	45
Figure 14 Time Comparison Query 3 on Breast Cancer Prediction Dataset	45
Figure 15 Laplace Mechanism.....	46
Figure 16 Gaussian Mechanism.....	47
Figure 17 Query 1 Result without DP on COVID-19 Home Nursing Dataset.....	48
Figure 18 Query 1 Result with DP on COVID-19 Home Nursing Dataset	48
Figure 19 Query 2 Result without DP on COVID-19 Home Nursing Dataset.....	49
Figure 20 Query 2 Result with DP on COVID-19 Home Nursing Dataset	49
Figure 21 Query 3 Result without DP on COVID-19 Home Nursing Dataset.....	50
Figure 22 Query 3 Result with DP on COVID-19 Home Nursing Dataset	50
Figure 23 Query 4 Result without DP on COVID-19 Home Nursing Dataset.....	51
Figure 24 Query 4 Result with DP on COVID-19 Home Nursing Dataset	51
Figure 25 Results Comparison using COVID-19 Home Nursing Dataset	52
Figure 26 Varying Epsilon values on Nursing Home COVID-19 Dataset.....	53
Figure 27 Analysis of privacy parameter using Nursing Home COVID-19 Dataset	54
Figure 28 Queries with Time Comparison using Nursing Home COVID-19 Dataset	55
Figure 29 Time Comparison Query 1 on Nursing Home COVID-19 Dataset	56
Figure 30 Time Comparison Query 2 on Nursing Home COVID-19 Dataset	56
Figure 31 Time Comparison Query 3 on Nursing Home COVID-19 Dataset	57

LIST OF TABLES

Table 1 Study of Existing Privacy Preservation Mechanisms in Healthcare.....	18
Table 2 Comparison of Privacy Preservation Techniques.....	24
Table 4 Comparing Results using Breast Cancer Prediction Dataset.....	42
Table 5 Comparing Results using COVID-19 Home Nursing Dataset.....	52

LIST OF ABBREVIATIONS

EHRs	Electronic Health Records
FedAvg	Federated Averaging
XACML	Extensible Access Control Markup Language
ABAC	Attribute-Based Access Control
ERT	Extremely Randomized Trees
TA	Trusted Authority
OEECDH	Optimized Encryption-Based Elliptical Curve Diffie-Hellman
ECC	Elliptical Curve Cryptography
CNN	Convolutional Neural Networks
SE	Searchable Encryption
PLS	Physical Layer Security
HLSTM	Hierarchical Long-Term Memory
MES	Modular Encryption Standard
MCC	Mobile Cloud Computing
MGP	Mosaic Gradient Perturbation
HSPs	Health Service Providers
PET	Privacy-Preserving Equality Test
MAC	Message Authentication Code
GDP	Geometric Data Perturbation
HDFS	Hadoop Distributed File System
SMC	Secure Multi-Party Computation
PHI	Patient Health Information
PPSS	Privacy-Preserving Stage Selection

PPSU	Privacy-Preserving Stage Update
ε	Privacy Budget
b	Scale Parameter
Δf	Sensitivity of the function
LDP	Local Differential Privacy
GDP	Global Differential Privacy
$L(\phi, D)$	Loss Function
sPL	Privacy Loss
$U(\phi, O)$	Utility Measure
ΔQ	Global sensitivity
Y_i	SampledNoise from Laplace distribution
R	Randomized Algorithm
O_1 and O_2	Neighboring Datasets
$P(O)$	Perturbed Output
$N(\sigma^2)$	Sampled Noise from Gaussian distribution
z_i	Actual Values
\hat{z}_i	Predicted Values

INTRODUCTION

1. Overview

The healthcare sector has changed dramatically in recent years, due to depending more and more on big data to improve patient care, enhance or improve operational effectiveness, and forward medical researches. Even if it's a good thing, the quick rise of electronic healthcare records but it presents serious obstacles to protecting patient privacy. With the increasing integration of electronic healthcare records and other forms of health data into the healthcare ecosystem, safeguarding patient's sensitive or personal data from breaches and unauthorized access has taken on paramount importance.

One of the main challenges for the protection of electronic healthcare record is the inherent contradiction between data accuracy and privacy. In order to facilitate progress in epidemiological researches, advance public health initiatives and health information needs to be readily available and functional. Storing and sharing this data publicly even in anonymized forms, it can still be violated patient's privacy by disclosing such information.

Differential privacy has emerged as a strong and provable privacy guarantee model to address this paradox by safeguarding data privacy and preserving the analytical usefulness of datasets. Differential privacy preserves individual personal identifiable information by proving mathematically that output distribution of a query remains independent of whether an individual record is present or not in the datasets. By adding calibrated noise to the query output or data, the balance between the data privacy and data accuracy can be established without compromising the overall insights that can be derived from the information.

The application of differential privacy in healthcare sector is particularly appealing as healthcare sector requires accurate and comprehensive data for research and industrial

purposes. In healthcare organizations trusting environment can be built among researchers, patients, doctors and other participants by implementing differential privacy to ensures the protection of data shared publicly or with any other third party for research purposes, policy making and collaborative initiatives.

Despite of its potential, implementation of differential privacy also faces challenges in healthcare industry. These include the potential impact on the accuracy of clinical and research findings, the challenges of precisely calibrating noise to preserve data utility, and the need for robust legal and ethical frameworks to oversee its deployment. To solve these concerns, a multidisciplinary approach is required that considers moral dilemmas, robust policy development, and other advancements.

This thesis aims to implement privacy preservation mechanism in healthcare using differential privacy, it further explores differential privacy's theoretical foundations, practical applications in healthcare data, and broader implications for privacy preservation. Through practical implementations of differential privacy in healthcare data, this research will provide insights into best practices and potential risks. It will support the development of safer and more effective data privacy preservation and data sharing technique within the healthcare sector. Through detailed study, this paper seeks to demonstrate the critical need for privacy-preserving mechanism and to push for their integration into electronic health data management.

1.1.Motivation

The increase in the digitalization of healthcare data has brought about a dramatic transformation in the collection, storing and utilization of medical data. Electronic health records (EHRs), health information systems, and wearable technologies are generating vast quantity of medical data nowadays. This offers a great chance to enhance patient outcomes, advancement in medical research, and improve the efficiency of healthcare services with the

utilization of this healthcare data. However, this digital transformation raises major concerns regarding the security and privacy of sensitive health data.

The growing number of cyberattacks and data breaches aimed at health information systems emphasizes how critical it is to deal privacy issues in the healthcare sector. In addition to putting private information at risk, these privacy violations undermine public trust in healthcare organizations, which may deter patients from sharing critical information that might help them to receive treatment. Additionally, there can be major consequences from the unauthorized disclosure of health information, including identity theft, discrimination, and psychological distress.

It has been noticed that standard methods of de-identification and anonymization are not adequate to fully protect patient privacy. Technological advances in data re-identification have demonstrated that datasets that are apparently anonymized may often be re-linked to individual users given the correct auxiliary information. This vulnerability necessitates the development of more sophisticated privacy-preserving techniques in order to provide a stronger guarantee against re-identification while maintaining the useful use of data.

Differential Privacy offers an acceptable solution to these problems. Differential privacy is a mathematically rigorous framework that ensures either the presence or absence of any individual's data will have a negligible impact on the study as a whole by protecting individual's privacy. By introducing calibrated noise into the data, differential privacy maintains the datasets utility for statistical and analytical purposes without compromising privacy. Maintaining this balancing is particularly crucial in the healthcare sector, where accurate and comprehensive data are essential for advance research purposes, public health surveillance, and clinical decision-making.

The goal of this research is to preserve data utility while also finding solutions to current privacy concerns. It aims to examine differential privacy's practical use in real-world scenarios,

its potential issues and solutions and evaluate its effectiveness. Healthcare data management relies heavily on privacy preservation, and the concepts and methods derived from this research will be essential as the healthcare industry absorbs ever-more complex technology and data sources.

1.2. Research Objectives

The thesis primary goals are:

- To achieve individual's data privacy by using differential privacy so that it doesn't reveal any private information of the user by sharing it publicly.
- To achieve the maximum security and the utility required in the dataset by varying the privacy budget.
- To perform multidimensional queries on the sensitive attributes under differential privacy.
- To develop an efficient and cost-effective scheme in terms of communication and computation overhead.
- To analyze performance comparison by varying matrices that influence the accuracy.

1.3. Contribution

The proposed Privacy Preservation Scheme in Healthcare Data will contribute in following ways:

- Used differential privacy techniques on publicly available healthcare datasets to demonstrate the practical feasibility and effectiveness of preserving patient privacy.
- Demonstrated that differential privacy can effectively balance privacy and utility, guaranteeing that converted results can still be used for insightful analysis and research.
- Examined impact of varying values of the privacy budget (epsilon) on both privacy protection and data utility.

- Conducted a comparative analysis of the Gaussian and Laplace mechanisms within the differential privacy framework. Evaluated the performance and usefulness of these mechanisms, emphasizing the situations in which each mechanism operates at its best.
- Analyzed the time complexity of applying differential privacy techniques, focusing on the computational efficiency as the parameters of user queries increase. Provided insights into the scalability of differential privacy methods, offering guidance on their practical implementation in real-world healthcare data systems.

1.4. Thesis Outline

The following chapters provide an organization and distribution of this research work:

- Chapter 1: An overview is provided, a problem statement is emphasized, the purpose for the research is explained, and the research objectives are listed. The contributions that this research made are also highlighted.
- Chapter 2: This chapter includes an overview of existing privacy preservation schemes in electronic healthcare data, followed up by pros and cons of each technique.
- Chapter 3: The early approaches for maintaining privacy before the idea of differentiated privacy emerged are covered in this chapter. It gives a general overview of the syntactic privacy models that make use of data publication mechanisms to preserve privacy.
- Chapter 4: This chapter provides a comprehensive overview of differential privacy, encompassing its definition, operational principles, mechanisms and techniques available to implement differential privacy. Additionally, it introduces a mathematical approach for determining the optimal value for privacy parameter ϵ .
- Chapter 5: This chapter introduces the framework employed to evaluate differential privacy, providing details on the algorithms used, experimental results of DP across different datasets, the impact of varying the privacy parameter, time complexity comparisons for complex queries on datasets of varying sizes, and comparative analyses

of Laplace and Gaussian mechanisms.

- Chapter 6: This chapter covers the conclusions, recommendations, and future work.

LITERATURE REVIEW

Kumar et al.[1] focus on the necessity of large datasets for training robust deep learning models in healthcare, while also acknowledging the privacy concerns and regulatory constraints that restrict data sharing in this field. To address these challenges, the authors highlight the potential of federated learning to overcome these barriers by allowing data to remain with the local party (such as a hospital), thus ensuring confidentiality and compliance with data protection regulations. The authors specifically focus on two algorithms: Federated Averaging (FedAvg) and FedProx, Using federated learning in healthcare highlights several limitations, privacy risks still exist as model updates sent to a central server could be intercepted. Communication costs are also notable due to the frequent data exchanges between clients and the server.

A hybrid strategy is presented by Joshi et al. [2] and is combined with a number of approaches to protect private patient data from breaches and unwanted access. This research methodology minimizes the impact on data utility while protecting privacy by integrating two key techniques: the FP-Growth algorithm for mining frequent patterns and anonymization processes to conceal sensitive information.

In order to solve privacy problems in healthcare big data, Suneetha et al. [3] offer a novel system that combines Apache Spark with established anonymization approaches like K-anonymization and L-diversity. A notable development in the field is the integration of these techniques with Apache Spark, which offers excellent speed and efficiency for handling massive datasets.

For the purpose of safeguarding local models in Internet of Things-based healthcare systems, Zhang et al. [4] suggested integrating homomorphic encryption with federated learning

mechanisms. The model integrates data from many medical facilities, and each participant trains local models independently, using their own data. Before the local models are aggregated, homomorphic encryption techniques are performed to safeguard the data. This stops possible adversaries from using inversion or model reconstruction attacks to deduce private information.

Seol et al. [5] thoroughly implemented attribute-based access control model to protect the electronic healthcare data (EHR) on XML based system. Sensitive data is partially encrypted by the system using XML encryption after access control. Next, it secures the data against unauthorized changes and access by utilizing XML digital signatures.

This research [6] by Abdullah et al, examined blockchain-based technology with the goal of improving the security and privacy of medical data. The approach focuses on decentralizing data storage through the use of blockchain technology, which lessens the vulnerabilities connected to centralized databases. It uses peer-to-peer (P2P) networks, where data is stored among numerous nodes. The massive volumes of data that are common in healthcare settings may make it difficult for the blockchain framework to scale effectively, which could result in longer transaction times and higher computational cost.

Aminifar, A., et al. [7] implemented machine learning approach by using Extremely Randomized Trees (ERT) that is specifically designed for distributed structured health data. This distributed ERT technique modifies traditional approach to adapt a distributed setting, ensuring that data privacy is upheld by avoiding direct data environment. Instead, data insights are derived through secure multi-party computation methods that allow entities to collaborate without exposing their private data.

The studies [8] by Charles, V., et al. used the improved ElGamal and ResNet classifier for maintaining the heart disease database privacy. The patient uses wearable devices; sensors connected with these devices will gather data and transfer it to the microprocessor and then

send it to the cloud. The upgraded ElGamal encryption technique will be used by the trusted cloud to safely protect patient data from outside threats. To accurately predict whether a patient suffering with heart disease or not, the CNN Classifier with ResNet-50 has been employed for data categorization and refining. However, key generation and encryption add to the computation cost, and its implementation depends on the trusted Authority (TA).

For securely detection of heart diseases, Rosy et al. [9] present a sophisticated cryptographic architecture that uses an Optimized Encryption-Based Elliptical Curve Diffie-Hellman (OEECDH) technique. To improve data security and privacy in cloud environments, the methodology combines the Diffie-Hellman mechanism with Elliptical Curve Cryptography (ECC) for key exchange. Sensitive data is encrypted before being transferred to the cloud using this optimized technique for creation of secure keys. This ensures that the information is kept private both during transferring and storing data. Processed and categorized encrypted data using deep Convolutional Neural Networks (CNN) in the cloud, enabling effective management of big datasets without sacrificing privacy. The same elliptical curve cryptographic techniques are used to safely decrypt the data after processing, guaranteeing that only authorized users can view the original data.

In order to guarantee privacy protection in IoT-enabled healthcare systems, Bi, H., et al. [10] investigate a deep learning-based solution. The approach protects sensitive data obtained from wearable devices by combining privacy isolation zones and deep learning techniques. Before the data is transferred to the cloud, sensitive information is recognized and segregated at the user's end. By doing this, user privacy is improved and only non-sensitive health-related data is processed further. After being segregated, the non-sensitive data is examined at the cloud by utilizing CNN, which is made to carry out secure data analytics without jeopardizing privacy. However, it is still difficult to discriminate between sensitive and non-sensitive information,

and doing so occasionally results in data distortion or the loss of crucial health-related information.

Research by Wang, K., et al. [11] outlines a novel searchable encryption (SE) scheme designed for IoT-enabled healthcare systems, focusing on forward privacy and verifiability. Searchable encryption allows encrypted data to be searched by authorized users without first decrypting it. Forward privacy ensures that updates to the dataset do not reveal any information about the contents of past search queries, thus enhancing the security of dynamically changing databases like those found in healthcare systems. The solution proposed by Wang et al. improves upon these by incorporating a trapdoor permutation function, ensuring that newly inserted records do not compromise the privacy of previously performed searches.

Furthermore Ahmed, J., et al. [12] describes a methodology that combines Federated Learning (FL) with Physical Layer Security (PLS) to enhance the privacy and efficiency in medical record. FL is employed to train local models at various nodes without sharing the unprocessed data among them. Only model parameters are shared with a central server or amongst nodes, significantly reducing the risk of exposing sensitive health data.

Another approach that Singh, P., et al. [13] describe uses cloud computing to facilitate the distribution of a Hierarchical Long-Term Memory (HLSTM) architecture among distributed Dew servers. Before the data is utilized to train the model, it is pre-processed to assure quality from IoMT devices. The complex series of events in the IoMT data flow is intended to be handled by the HLSTM architecture. In order to preserve the integrity of hierarchical data structures, it makes use of a two-layered LSTM network, in which the first layer creates a phrase vector and the second layer collects these into a document vector. Federated learning is used in the intrusion detection model, which forms the basis of the methodology. Subsets of the data are used to train local models on Dew servers, which subsequently feed into the creation of a global model.

In research by Shabbir, M., et al. [14] implemented Modular Encryption Standard (MES) for securing health data in Mobile Cloud Computing (MCC) environments. Health data is categorized and recognized according to its sensitivity before encryption. Several encryption modules are employed at different stages of the multi-layered encryption method used by the MES technique. This approach ensures that data is treated in accordance with its security classification at every stage, starting with the user's mobile device and continuing to the cloud. A comprehensive methodology to improve the security of medical sensor data in Internet of Things environments is outlined by Khan, M. A., et al. [15]. The first step in this strategy is user registration, when individuals enter their biographical and biometric information. A hash function is used to validate the user's credentials during the login procedure after registration. Throughout this procedure, the SHA-512 algorithm is used to ensure the security and integrity of user data. This framework combines two different encryption techniques. The first technique, the Substitution Caesar Cipher, is a straightforward character substitution approach that encrypts sensor data and offers a minimal amount of security. The data is encrypted again using IECC (Improved Elliptic Curve Cryptography) after the first encryption. Compared to conventional ECC, this approach is known for its strength in data protection since it uses a generated secret key that adds an extra layer of security.

Krall et al. [16] explore an innovative way to maintain privacy in predictive healthcare analytics by utilizing the Mosaic Gradient Perturbation (MGP) technology. Based on differential privacy, the concept aims to preserve model correctness while reducing the danger of model inversion attacks. The MGP method is intended to cause more of a perturbation to the gradient parts of the objective function linked to sensitive characteristics than to non-sensitive characteristics. Furthermore, the difficulties of accomplishing searchable and privacy-preserving data exchange in cloud-assisted electronic health environments were examined by Xu et al. [17]. The suggested system makes use of modern cryptographic algorithms to facilitate effective,

private data sharing and searches. The system enables health service providers (HSPs) to search encrypted PHI data using keyword range and multi-keyword searches using dynamic searchable encryption techniques. By using this technique, patient privacy is protected because it guarantees that the data is encrypted during all operations. Numerical analysis queries on encrypted data are made possible by the Privacy-Preserving Equality Test (PET) Protocol, which protects sensitive data. Message Authentication Codes (MACs) are used to eliminate out erroneous data and confirm the accuracy of PHI files.

Song, J., et al. [18] present another advanced method for secure data organization, in order to improve privacy of data collected in healthcare systems. This method guarantees that user data collection stays private and secure while providing precise data aggregation for healthcare analytics. The computation of matrix eigenvalues forms the basis of the proposed secure arrangement technique. Each user's data is placed in a secret position determined by this calculation. The process begins with an offline Trusted Authority (TA) distributing initial data to all users. Matrices are used to represent and manipulate data positions securely. Basic matrix operations such as addition, multiplication, eigenvalue, and trace calculation over a Galois Field are employed.

Another, framework proposed by Zhou, X., et al. [19] which makes use of a role-based access management mechanism where access to EMRs is provided depending on the assigned duties of medical professionals. Anonymous RBACAnony Scheme is based on a bilinear group that has two subgroups, one of which hides the patient's identification. The scheme makes sure that an attacker can't figure out who a patient is by looking at a random string. RBACAnony-F, Anonymous Strategy based on a four-subgroup bilinear group, enhancing security by hiding the patient's identity in a composite-order subgroup. EMRs are encapsulated using on-demand access policies that allow one-to-many encryption. This enables different medical staff members to access the same EMR based on their roles. Patients and their doctors can search

for EMRs without revealing their identities. The search mechanism ensures that only authorized users can access the relevant EMRs.

A technique of attribute-focused anonymization for publishing healthcare data was proposed by Onesimu, J. A., et al. [20]. The goal of the Fixed-Interval Anonymization technique is to safeguard numerical properties. To ensure generalization, the original values are substituted by computed mean values within predetermined intervals. Sorting the numerical characteristics, figuring out the interval width by comparing the highest and lowest values, and substituting the computed mean for the original values within each interval are the steps involved in the procedure. Sensitive attributes are protected using an enhanced version of the l-Diverse Slicing approach.

Zala, K., et al. [22] focuses on the integration of cryptographic and steganographic methodologies to guarantee the confidentiality and integrity of medical records that are kept on external cloud platforms. The architecture uses a data security method that consists of five steps. It employs AES-128 encryption for authentication and authorization in order to protect user credentials. For steganography, it encrypts patient EHRs using AES-128 and hides them within images using the LSB (Least Significant Bit) technique. To Access Control, it allows patients to assign access rights to their EHRs for doctors and relatives. For Data Hiding, it uses anonymization to protect sensitive EHR data from unauthorized access. Hybrid technique is further used for combining AES-128 encryption with steganography to provide double-layer security.

Zhang, M., et al. [23] introduced PPO-CPQ technique in electronic healthcare systems to preserve privacy for clinical pathway queries. Privacy-Preserving Comparison (PPC) protocol allows two parties to compare private values by converting input data into binary format and executes secure bitwise comparisons. Privacy-Preserving Clinical Comparison (PPCC) handles negative numbers and ensures accurate comparisons in the clinical context. Based on lowest

cost, the PPSS protocol determines the most suitable stage in the clinical pathway. PPSU protocol makes sure that only necessary changes are shared between servers, updating the pathway's stage while protecting privacy.

Ref.	System model	Goals	Limitations/ Weaknesses	Privacy Preserving Techniques	Trust Model
Joshi et al. [2] 2020	<ul style="list-style-type: none"> Hybrid method using FP-Growth algorithm and anonymization 	<ul style="list-style-type: none"> Hide sensitive patient data in healthcare datasets using hybrid approaches 	<ul style="list-style-type: none"> Increased time and memory requirements for large datasets 	<ul style="list-style-type: none"> FP-Growth algorithm 	<ul style="list-style-type: none"> Anonymization and association rule hiding techniques
Suneetha et al.[3] 2020	<ul style="list-style-type: none"> Used Apache Spark for privacy preservation in healthcare big data 	<ul style="list-style-type: none"> Using K-anonymity and L-diversity for the protection of patient's data in healthcare 	<ul style="list-style-type: none"> Potential data segregation issues for transferring to HDFS 	<ul style="list-style-type: none"> K-anonymity, L-diversity 	<ul style="list-style-type: none"> Handling healthcare big data with Apache Spark for faster processing
Zhang et al. [4] 2022	<ul style="list-style-type: none"> Federated Learning along with combination of Homomorphic Encryption 	<ul style="list-style-type: none"> Ensure privacy preservation of patient's data in IoT-enabled healthcare systems 	<ul style="list-style-type: none"> Increased computation and communication overhead; Dropout clients not handled 	<ul style="list-style-type: none"> Homomorphic Encryption, Shamir Secret Sharing, Diffie-Hellman Key Agreement 	<ul style="list-style-type: none"> Honest but curious; Semi-honest participant
Seol et al.[5] 2018	<ul style="list-style-type: none"> Attribute-Based Access Control (ABAC) using XACML 	<ul style="list-style-type: none"> Providing restricted access and protect patient privacy in EHR systems 	<ul style="list-style-type: none"> Increased complexity and computational overhead due to encryption and access control mechanisms 	<ul style="list-style-type: none"> XML encryption & digital signatures 	<ul style="list-style-type: none"> Assumes semi-trusted cloud environment and authorized users for accessing EHR data

Ref.	System model	Goals	Limitations/ Weaknesses	Privacy Preserving Techniques	Trust Model
Abdullah et al. [6] 2017	<ul style="list-style-type: none"> Used MediBchain framework based on Blockchain 	<ul style="list-style-type: none"> Ensure privacy, security, and integrity of healthcare data using blockchain 	<ul style="list-style-type: none"> Increased complexity and computational overhead; requires secure key management 	<ul style="list-style-type: none"> Blockchain, Public Key Encryption (ECC) 	<ul style="list-style-type: none"> Decentralized patient-centric model
Aminifar, A., et al. [7] 2022	<ul style="list-style-type: none"> Used Distributed Extremely Randomized Trees for privacy preservation 	<ul style="list-style-type: none"> Ensure privacy-preserving machine learning for distributed health data 	<ul style="list-style-type: none"> Increased complexity and computational overhead; handling missing values 	<ul style="list-style-type: none"> Secure Multi-Party Computation (SMC), Encryption 	<ul style="list-style-type: none"> Semi-honest model; Assumes no collusion among k parties
Rosy et al. [9] 2021	<ul style="list-style-type: none"> Optimized Encryption based Elliptical Curve Diffie-Hellman (OEECDH) 	<ul style="list-style-type: none"> Ensure privacy protection for predicting heart disease using deep learning and encryption 	<ul style="list-style-type: none"> Requires secure key management; Needs efficient handling of large datasets 	<ul style="list-style-type: none"> Elliptic Curve Cryptography (ECC), Diffie-Hellman 	<ul style="list-style-type: none"> Semi-trusted cloud environment
Bi, H., et al. [10] 2021	<ul style="list-style-type: none"> Privacy Protection and Data Analytics for IoT-Enabled Healthcare using deep learning 	<ul style="list-style-type: none"> Providing privacy-preserving data analytics and secure health monitoring using IoT devices 	<ul style="list-style-type: none"> High computational requirements for deep learning algorithms 	<ul style="list-style-type: none"> Convolutional Neural Networks (CNN), Secure Multi-Party Computation (SMC) 	<ul style="list-style-type: none"> Data integrity trust on cloud service providers and wearable device manufacture; assumes secure data transmission channels

Ref.	System model	Goals	Limitations/ Weaknesses	Privacy Preserving Techniques	Trust Model
Wang, K., et al.[11] 2021	<ul style="list-style-type: none"> Used forward-privacy searchable encryption in electronic healthcare data 	<ul style="list-style-type: none"> Ensure privacy and security of healthcare data while enabling efficient search and data sharing 	<ul style="list-style-type: none"> Potential exposure of search patterns; requires efficient key management 	<ul style="list-style-type: none"> Searchable Encryption (SE), Pseudo-Random Function (PRF), Trapdoor Permutation 	<ul style="list-style-type: none"> Semi-honest adversaries; Trust in cloud service provider to follow protocol without collusion
Ahmed, J., et al.[12] 2021	<ul style="list-style-type: none"> Federated Learning (FL) combined with Physical Layer Security (PLS) in IoMT networks 	<ul style="list-style-type: none"> Enhance privacy and security in IoMT networks by using FL and PLS 	<ul style="list-style-type: none"> Increased complexity and computational overhead; Potential for localized eavesdroppers 	<ul style="list-style-type: none"> Homomorphic Encryption, PLS, Blockchain 	<ul style="list-style-type: none"> Assumes semi-trusted central server and devices in a hierarchical network
Singh, P., et al. [13] 2022	<ul style="list-style-type: none"> Dew-Cloud-Based Hierarchical Federated Learning (HFL) using Hierarchical LSTM (HLSTM) for IoMT networks 	<ul style="list-style-type: none"> Enhance data privacy, availability, and intrusion detection accuracy in IoMT networks using HFL and HLSTM 	<ul style="list-style-type: none"> Complexity in managing hierarchical models; potential latency in federated learning updates 	<ul style="list-style-type: none"> Homomorphic Encryption, Federated Learning 	<ul style="list-style-type: none"> Trust in decentralized Dew and Cloud servers; assumes secure communication channels
Shabbir, M., et al.[14] 2021	<ul style="list-style-type: none"> Modular Encryption Standard (MES) in Mobile Cloud Computing 	<ul style="list-style-type: none"> To secure health information in mobile cloud computing environments 	<ul style="list-style-type: none"> Increased complexity and computational cost; layered modeling performance 	<ul style="list-style-type: none"> Modular Encryption Standard (MES) 	<ul style="list-style-type: none"> Assumes trust in cloud service providers and mobile devices

Ref.	System model	Goals	Limitations/ Weaknesses	Privacy Preserving Techniques	Trust Model
Krall et al. [16] 2020	<ul style="list-style-type: none"> Mosaic Gradient Perturbation (MGP) in IoT-enabled healthcare systems using predictive modeling 	<ul style="list-style-type: none"> Preserving privacy and reducing the possibility of model inversion attacks with model accuracy 	<ul style="list-style-type: none"> Increased complexity in fine-tuning trade-offs; potential computational overhead in large-scale implementations 	<ul style="list-style-type: none"> Differential Privacy, Gradient Perturbation 	<ul style="list-style-type: none"> Semi-trusted entities within a decentralized framework; assumes honest-but-curious adversaries
Xu et al. [17] 2019	<ul style="list-style-type: none"> E-healthcare system with cloud assistance that includes wearables, cloud servers, IoT gateways, and health service providers (HSPs) 	<ul style="list-style-type: none"> Enable secure and efficient sharing of patient health information (PHI) using searchable encryption 	<ul style="list-style-type: none"> Performance and efficiency of the system can be affected by the quantity of files saved and retrieved, as well as the difficulty of managing massive datasets in a dynamic manner. 	<ul style="list-style-type: none"> Searchable encryption, Privacy-Preserving Equality Test (PET) protocol, Variant Bloom Filter (VBF), Message Authentication Code (MAC) 	<ul style="list-style-type: none"> Trusted Authority (TA) is fully trusted, Cloud servers are honest-but-curious, IoT gateways and health service providers (HSPs) are trusted
Onesimu, JA., et al. [20] 2022	<ul style="list-style-type: none"> Publishing healthcare data using l-diverse slicing and a fixed-interval technique for attribute-focused anonymization 	<ul style="list-style-type: none"> Privacy preservation while data releasing of EHR and provide maximum data utility 	<ul style="list-style-type: none"> Increased computational complexity with large datasets, Vulnerability to certain privacy attacks with fixed methods 	<ul style="list-style-type: none"> Enhanced l-diverse slicing for grouping attributes and fixed-interval anonymization for numerical attributes 	<ul style="list-style-type: none"> Internal data controllers are trusted, Data analysts are considered potential adversaries

Table 1 Study of Existing Privacy Preservation Mechanisms in Healthcare

PRELIMINARIES OF PRIVACY PRESERVATION

3. Privacy Preservation Models

Privacy preservation encompasses various strategies and technologies aimed at protecting individuals' personal data and information. Formerly, a lot of work have been done for privacy protection. Followings are the privacy models that have been used for privacy preservation data release publicly include anonymization, t-closeness, K-anonymity, I-diversity, and many other techniques.

3.1. Anonymization

It is a method of transforming the information that can be uniquely identified (PII) into unidentifiable form so, that it can't be to linked again with an individual without having additional information [32]. The goal is to secure the personal identification of a person while enabling public data sharing publicly. The data collector removes the particular uniquely identified information like as Name, phone number and location. But still there are challenges in data anonymization even if specific identifiers removed. Sometimes, it is possible to reidentify anonymized data by data linkage attacks, especially when combined with other datasets. Data masking techniques are used in data anonymization such as randomization that replaces identifiable data with random values and pseudonymization that substitutes identifiable information with pseudonyms or tokens that can be reversed only with a specific key or method. Techniques for anonymization must change to keep up with improvements in ways for re-identifying data.

3.2. K-Anonymity

Researchers have proposed multiple other methods for privacy preservation, to overcome the shortcomings of simple data anonymization. K-anonymity is considered as the widely used methods for protecting privacy. It ensures that individuals cannot be reidentified from anonymized datasets by making sure that every person in the record can be distinguished from at least $k - 1$ other person. [26][30]. Elements of data like age, sex and occupation that could potentially identify individuals are grouped into categories. Those individuals who have similar characteristics grouped together. Instead of recording the exact ages, age can be grouped into ranges like (30 – 35 years). Remember each group should contains at least k individuals. By organizing the data this way, it's much harder for someone to figure out who a specific person is. However, this technique is still vulnerable to the homogeneity and background knowledge attacks.

3.3. I-Diversity

To deal with the above-mentioned drawbacks, this technique emphasizes the variety of sensitive attributes (such as ethnicity or medical conditions) within each group of people who share the same quasi-identifiers (non-sensitive attributes) [24]. K-anonymity guarantees that, using quasi-identifiers, every record may be identified from at least $k - 1$ other records. It does not take into consideration how sensitive characteristics are distributed throughout these groupings. An attacker can still make inferences about individual's sensitive information, If there is no variability in the values of the sensitive characteristics within a group. The goal of this method is to prevent attackers from linking specific sensitive information to individuals based on their shared characteristics in the dataset. Similar to K-anonymity, individuals are grouped together based on identical quasi-identifiers. For example, all individuals in a group might be of the same age range, gender, and living in the same ZIP code. Within each group formed by identical quasi-identifiers, I-diversity requires that the sensitive attributes are

diverse. There should be at least ℓ different conditions of sensitive attribute. This means that no single sensitive attribute should be overly common within the group. Still even with I-diversity, datasets can be vulnerable to certain type of privacy attacks like skewness and similarity attack.

3.4. T-Closeness

It is a technique for maintaining privacy that aims to rectify the inadequacies of k-anonymity and I-diversity, particularly the vulnerabilities related to skewness and similarity attacks [26]. T-closeness guarantees that each equivalency class's sensitive attribute distribution closely resembles the dataset's general distribution of those attributes. In addition to improving data privacy, this lowers the chance of attribute exposure. The equivalency class is said to have T-closeness if there is a threshold t that distinguishes the distribution of the sensitive attribute in the equivalency class from the distribution of the attribute in the total dataset.

3.5. Cryptographic techniques

Before making the data available to the public, the data curator could encrypt it [50]. However, it is extremely difficult to encrypt vast amounts of data using standard encryption techniques, and must only be put into practice when gathering data. By using homomorphic encryption, it allows to perform calculations on encrypted data, that produces an encrypted output and the final results will be equivalent to plaintext operation after decrypting it back. Similarly, secure multiparty computation permits several parties to work together to jointly compute a function over their private inputs. Moreover, Blockchain technology used in privacy preservation of data that uses cryptographic hash functions to ensure data integrity and immutability. Cryptographic hash functions like SHA-256 convert data into a fixed-size hash, guaranteeing that tampering is readily identifiable by producing a totally distinct hash for every alteration in the input data. It provides transparency and security in data sharing and transactions. However, encryption decreases the utility of the data in addition to being difficult to execute.

3.6. Multidimensional Sensitivity Based Anonymization

It is an improved kind of anonymization that can be used to outperform more conventional anonymization methods[31]. It identifies which attributes are sensitive in the datasets. It includes both quasi-identifiers (identify individuals when combined) and direct identifier attributes. Evaluate the sensitivity of each attribute. Some attributes may be more sensitive than others, and this sensitivity can be quantified. Implement anonymization strategies to make sure the data cannot be traced back to individuals, such as generalization, suppression, or noise addition. The level of anonymization that is used can change depending on how sensitive each attribute is. Consider the interactions between multiple attributes. Even if individual attributes are anonymized, make sure the aggregation of attributes prevents re-identification. This is essential for defending against inference attacks, in which the attacker reidentifies a target using multiple attributes. It provides the enhanced privacy by considering the sensitivity of multiple attributes and their interaction. It allows for different level of anonymization based on the sensitivity of each attribute and minimizes the risk of re-identification through combinations of attributes. This technique is better suited for large scale with static data. Moreover, it is not applicable for streaming data.

3.7. Data Distribution technique

This technique involves splitting of data over multiple sites. There are two main methods for distributing data. Both strategies horizontal distribution and vertical distribution[25] decentralize data processing and storage in an effort to reduce the possibility of privacy breaches.

In horizontal distribution, a subset of dataset's records or rows stores at each site. Each subset contains the same attributes (columns) but for different individuals or entities. This technique is frequently employed, when different sites have records for different sets of individuals. For instance, medical records for various patients may be kept in multiple hospitals. By distributing

records over different sites, a site can implement its own privacy policies and controls according to their specific requirements and also there is less chance of single point failure. Only a single subset of data is compromised regardless of whether a site is compromised. Queries over the distributed data can be conducted using secure multi-party computation, which protects individual records from being revealed to unauthorized sites.

Every site in a vertical distribution holds a portion of the dataset's properties, or columns. Each subset contains different attributes but for the same set of individuals or entities. When multiple websites need to maintain various kinds of data on the same people, this approach can be helpful. For instance, financial data may be stored on one website and personally identifiable information on another. In case of breach, it reduces the risk of complete data exposure by separately storing the sensitive attributes.

Technique	Strengths	Weaknesses	Applications	Attribute Preservation	Damage to Data Utility	Complexity	Accuracy of Data Analytics Results
Anonymization	Simple, easy to implement, widely used.	Vulnerable to re-identification attacks if not done properly.	Data sharing, data publishing.	Low	Medium	Low	Medium
K-Anonymity	Reduces risk of identification, simple concept.	Does not protect against attribute disclosure, selection of k.	Healthcare data, census data.	Medium	Medium	Low	Medium
L-Diversity	Protects against homogeneity and background knowledge.	Complex to achieve with high l-values.	Data publishing, sensitive attribute protection.	High	Low	Medium	High
T-Closeness	Better protection against attribute disclosure.	More complex and computationally intensive.	Healthcare data, sensitive data publishing.	High	Low	High	High
Cryptographic Techniques	Strong protection, widely accepted, mathematically rigorous.	Computationally intensive, requires key management.	Data transmission, storage, secure computations.	High	Low	High	High
Multidimensional Sensitivity-Based Anonymization	Nuanced privacy protection, considers multiple factors.	Complex to implement, requires detailed sensitivity analysis.	Data sharing, multi-dimensional data protection.	High	Low	High	High
Differential Privacy	Provides strong privacy guarantees, resistant to many types of attacks.	Can reduce data utility, requires careful calibration of noise.	Statistical databases, privacy-preserving data analysis.	High	Medium	High	High

Table2 Comparison of Privacy Preservation Techniques

DIFFERENTIAL PRIVACY

4.1. Differential Privacy

It is a mathematical mechanism that offers robust privacy guarantee throughout data analysis and to exchange data publicly [33]. This idea was first presented by Cynthia Dwork and other associates in the early 2000s. It protects an individual's privacy by making sure that either an individual present in the dataset or not cannot impact results of any research. It helps to make guarantee that the private information about an individual is kept hidden upon aggregated data analysis. Fundamental concept of differential privacy is introducing the controlled randomness into the data analysis process [49]. Differential Privacy makes guarantee that no single data point's privacy is compromised in the output by carefully adding the noise to query results.

4.1.1. Definition: A randomized algorithm R is (ϵ, δ) -differentially private for any two adjacent datasets O_1 and O_2 for all subsets Q of the output space of R .

$$\Pr[R(O_1) \in Q] \leq e^\epsilon \Pr[R(O_2) \in Q] + \delta$$

Neighboring datasets O_1 and O_2 are means to be adjacent datasets that are different by no more than one element. Here, a positive privacy parameter called ϵ epsilon is used to evaluate the loss of privacy. Smaller value of epsilon indicates the stronger privacy. While δ is a positive parameter, which is usually close to zero, permits a minor relaxation of the strict privacy guarantee. It is pure differential privacy, if the value of $\delta = 0$, then we obtain a stricter definition of ϵ -differential privacy.

$$\frac{\Pr[R(O_1) \in Q]}{\Pr[R(O_2) \in Q]} \leq e^\epsilon$$

4.1.2. Sensitivity of queries:

Sensitivity is a crucial aspect in the application of differential privacy. The greatest difference that can exist between the output of a function R for any pair of neighboring inputs O_1 and O_2 , is the sensitivity of the function. It measures the highest possible alteration to a query's response that could come from either including or eliminating a single person's data. Queries with high sensitivity require more noise to ensure privacy, whereas queries with low sensitivity require less noise.

$$\Delta = \max_{O_1, O_2} \|R(O_1) - R(O_2)\|$$

4.2. The Privacy Budget

The level of privacy guarantee in a mechanism is managed by privacy budget, refer as ϵ [48]. It indicates a limit on how much of information that can be derived from a computation's output on an individual. Choosing lesser value of epsilon provides more privacy but may result in less accurate results. Conversely, higher ϵ values yield outcomes that are more precise but with less privacy protections.

4.2.1. Sequential Composition:

Sequential composition describes the situation in which several operations or searches are carried out consecutively on the same dataset. Each query introduces a certain amount of privacy loss which is measured by its privacy parameter. If the queries are run sequentially on the same dataset, then overall loss in privacy accumulates for each query. Suppose there are the n number of operations $P_1, P_2, P_3, \dots, P_n$ performed each with an ϵ -differential privacy parameter on a dataset consecutively then total privacy loss is equal to:

$$Privacy\ Loss_{Total} = \sum_{i=1}^n \epsilon_i$$

In sequential composition, it overall leads to cumulative privacy loss as addition of each query reduces the overall privacy guarantee. To achieve desired level of accuracy, the total privacy budget must be carefully managed. If large number of queries are performed or if the individual ϵ values are too high, the overall privacy guarantees can be significantly weakened.

4.2.2. Parallel Composition:

On the other hand, parallel composition describes the scenarios in which multiple queries and operations performed consecutively or independently on disjoint datasets. In parallel composition, each query or operation executed independently in terms of its privacy loss. Suppose there are the n number of operations $P_1, P_2, P_3, \dots, P_n$ performed, each applied to separate datasets with ensuring ϵ -differential privacy individually then the entire privacy remains essentially same as for a single query.

$$\text{Privacy Loss}_{Total} = \max(\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n)$$

In this case, the total privacy guarantee is determined by the highest ϵ value among the queries. This methodology guarantees privacy is maintained across each individual queries, assuming no association exist between datasets used in each query. In parallel composition, management of privacy budget becomes more simplified while handling multiple operations or queries that can be executed simultaneously.

4.3. Mechanisms of Differential Privacy

4.3.1. Laplace Mechanism:

It is used in differential privacy to add controlled amount of noise to the output of computations [47]. Laplace Mechanism can be applied to achieve differential privacy for making sure that either presence of an individual or not doesn't will not significantly alter the result of a calculation or analysis. The amount of noise added in computation's output is evaluated from

the Laplace distribution by the Laplace mechanism [35]. The likelihood density function of the Laplace distribution, which is employed in this mechanism for differential privacy, is represented by this expression.

$$f(u) = \frac{\varepsilon}{2b} \exp\left(\frac{-\varepsilon|u|}{b}\right)$$

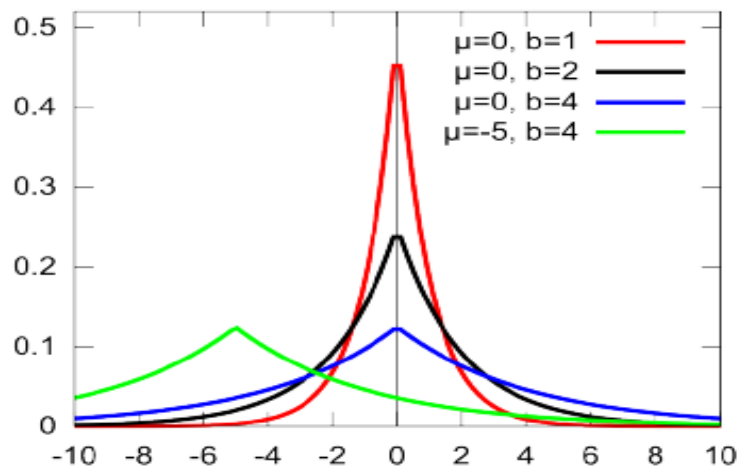


Figure 1 Laplace Distribution

Privacy Budget ε is a privacy parameter that responsible for privacy to obtain differential privacy. Lesser value of ε , provides a higher privacy protection. Here b is a scale parameter to determines the spread of distribution ($b > 0$). Larger the value of b , increase in the amount of added noise more widely, that leading to more fluctuation in final results. $|u|$ defines the absolute value of u , to ensures the Laplace distribution is symmetric around its mean. The value of $|u|$ often calculated as $|u| = \frac{b \Delta f}{\varepsilon}$

Here, privacy and utility both are trade-offs larger the amount of $|u|$ provides stronger privacy but it reduces the utility of the output and vice versa.

The change in the output of a function when it applies to two adjacent datasets, these are neighboring datasets that varied in by the presence and absence of single individual's record is known as sensitivity of function Δf .

$$\Delta f = \max_{d_1, d_2: |d_1 \Delta d_2| = 1} \| f(d_1) - f(d_2) \|_1$$

Definition: Given a function $f: O \rightarrow \mathbb{R}$ that operates on a dataset O , the Laplace mechanism perturbs the output of $f(O)$ to ensure ϵ -differential privacy. The perturbed output $P(O)$ is defined as:

$$P(O) = f(O) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right)$$

4.3.2. Gaussian Mechanism:

It is also an alternative way of Laplace mechanism to inject noise into the results of a function for ensuring privacy while preserving data utility in differential privacy. Because of its bell-shaped distribution, the Gaussian mechanism smooths out noise and is frequently chosen when there is a need for more precise control over noise distribution or when sensitivity σ is high.

Definition: Given a function $f: O \rightarrow \mathbb{R}$ that operates on a dataset O , the Gaussian mechanism perturbs the output of $f(O)$ to achieve ϵ -differential privacy. The perturbed output $P(O)$ is defined as:

$$P(O) = f(O) + N(\sigma^2)$$

Here $f(O)$ is the exact result of the function f on dataset O . $N(\sigma^2)$ represents noise that is evaluated through Gaussian distribution. Σ refers a parameter that evaluated from sensitivity of the function f . It measures the change in output of $f(O)$ can change when one element of O is changed. where s is the sensitivity of the function and \log represents the natural logarithm.

$$\sigma^2 = \frac{2s^2 \log\left(\frac{1.25}{\delta}\right)}{\epsilon^2}$$

The privacy parameter that regulates the quantity of noise generated is epsilon. This mechanism balances privacy and utility by controlling the ϵ and σ parameters. Larger ϵ and smaller σ

provides weaker privacy guarantees as it adds less noise while provides higher utility. In the same way smaller ϵ and larger σ adds more noise that strengthening privacy but potentially reducing utility.

4.3.3. Exponential Mechanism:

It is well known that not all query functions are able to return numerical values in their output. A more general approach to handling and responding to qualitative queries was put out by McSherry and Talwar [28]. So, this mechanism deals with non-quantitative e queries. Exponential function formally defined as below:

Definition of Exponential Mechanism:

Given a set N of acceptable outputs, and a utility function $u : N \times O \rightarrow \mathbb{R}$ Which quantifies the desirability of every outcome $n \in N$ given the dataset O . This Mechanism [37] selects an output y probabilistically to ensure ϵ -differential privacy.

$$P(O) = \Pr[n | O] \propto \exp\left(\frac{\epsilon u(n, O)}{2\Delta u}\right)$$

Here $u(n, O)$ is the utility of output n given dataset O . Δu is the maximum sensitivity which measures how much the utility function $u(n, O)$ can change when one element of O is changed. It determines the scale of possible changes in utility across datasets. Similarly, amount of noise added dependent on the privacy parameter epsilon.

4.4. Methods to Implement Differential Privacy

Both local and global DP approaches adhere to the core principle of differential privacy by ensuring that an individual's data remains protected as shown in Figure2 [24]. The selection

between local and global differential privacy depends on the specific application, the level of trust in the central server, and the desired privacy guarantees.

4.4.1. Local Differential Privacy:

In LDP data contributor is responsible for adding noise in data before sharing it with central aggregator and data collector. So, it doesn't require any trusted party. In Local DP [45] noise introduces to the individual data points. Suppose each user has a sensitive bit of information $b_i \in \{0,1\}$. Each user perturbs their data locally using a randomized response mechanism with probability $\frac{1+\epsilon}{2}$, report b_i or with probability $\frac{1-\epsilon}{2}$, report $1 - b_i$.

It ensures that the privacy of each individual's preserved before aggregation or analysis occurs. In LDP, noise addition occurs at the individual level. The main advantage of Local DP, it does not require to trust data aggregator, as it is unaware of the real values. But problem is that every user will have to introduce noise in personal information that will overall increase the total amount of added noise. But this problem can be mitigated by using the high values of epsilon (ϵ).

4.4.2. Global Differential Privacy:

In GDP it generated noise to the final results of query by the central aggregator before sharing it with any third party [46]. In this model, each user will share their actual data with a central aggregator without adding noise. To add noise to the entire dataset, the central aggregator will use a differential privacy method. Global differential privacy make sure either an individual present in the dataset or not does not alter probability distribution in the final output.

Consider a function f that calculates a sum over a dataset O : $f(O) = \sum_{i=1}^n x_i$. The Laplace mechanism allows noise taken from the LaPlace distribution to be added in order to achieve global DP.

$$P(O) = f(O) + Lap\left(\frac{\Delta f}{\epsilon}\right)$$

As the central aggregator has access to real dataset so, it requires to trust data collector. This model's primary benefit lies in the fact that low values of epsilon (ϵ) can yield useful results without requiring a significant amount of noise. But before sharing the data, it must requires the trust of users on data collector. If in case the data aggregator gets compromised, the data can be leaked and it increases the risk of privacy failure.

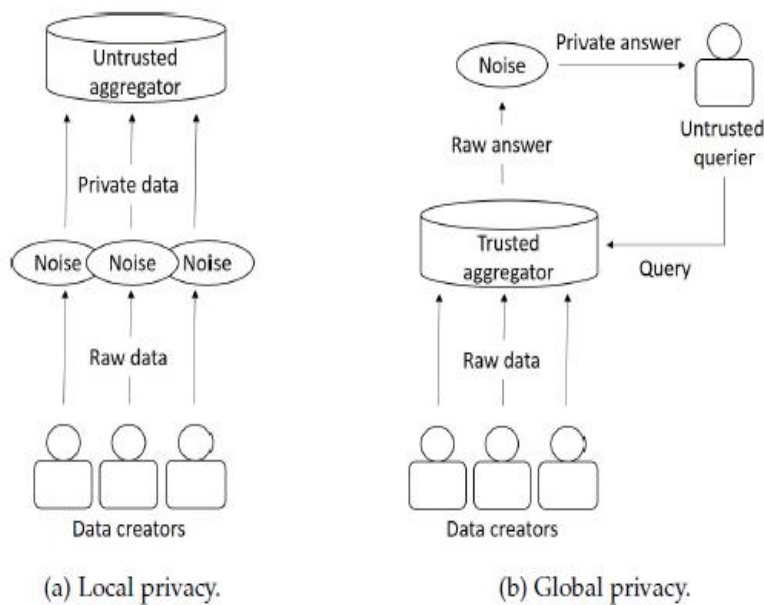


Figure 2 Laplace vs Global Differential Privacy

4.5. Differentially Private Data Release

Data release refers to the process of making data accessible for use or analysis by maintaining the user's data privacy [44]. The objective is to minimize the possibility of disclosing private information about any individual in the dataset while yet providing accurate and useful information. Depending on the sequence of answers for query set, there are two different arrangements i.e. interactive data release and non-interactive(differentiated in Figure3) [24] can be utilized to release sensitive data while preserving privacy.

4.5.1. Interactive Data Release

Interactive data release occurs when a user or data analyst engages with a system or mechanism to query the data while maintaining privacy. First the user submits a query to the system or mechanism that holds the private dataset. Then the system applies a differential privacy mechanism to the query in order to add noise or perturbation to the results to prevent exact data reconstruction while ensuring statistical accuracy. After this system returns a differentially private response to the query. The main aim of this response is to maintain privacy while offering useful statistical information. Iteratively, the user may submit more than one query. The response of each query depends on the differential privacy guarantees that are applied to that particular query while taking the cumulative privacy budget into account.

4.5.2. Non-Interactive Data Release

Non-interactive data release [43] involves the pre-calculating differentially private aggregates or summaries of a dataset without requiring user input is known as non-interactive data release. The goal is to release useful statistical information by reducing the need for real-time interaction of user. Before releasing the dataset, differential privacy mechanisms are applied to aggregate statistics or summaries. Then differential privacy mechanisms add noise or perturbation to these pre-computed summaries to ensure privacy. After that perturbed statistics or query results are made available to users or analysts. It reduces the risk of privacy breaches associated with dynamic interaction as the users access the pre-computed results without interacting directly with the private dataset but it may limit flexibility.

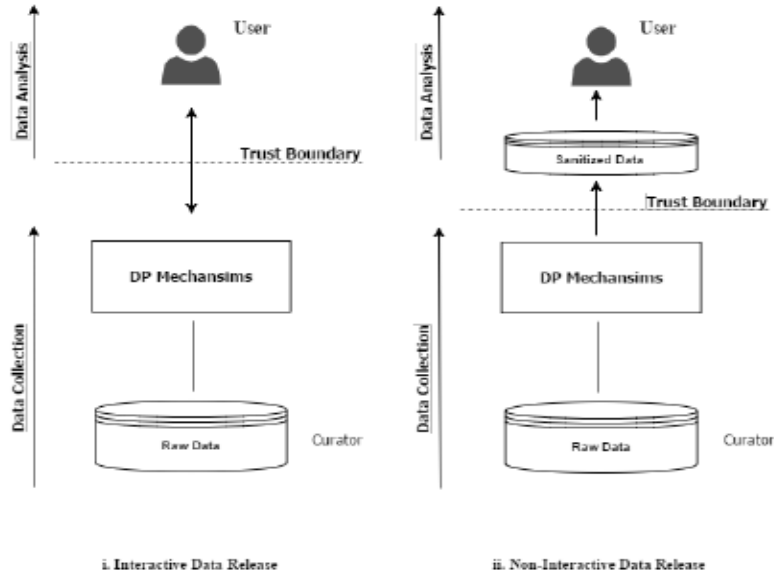


Figure 3: Methods of Data Release

4.6. Selection of Privacy Parameter ϵ

Setting the value for epsilon is a challenging task for implementing differential privacy effectively in any application [35] [42]. Desirable balance between privacy and utility can be controlled by adjusting the value of epsilon typically values range from 0.01 to 1 are for strong privacy, but higher values might be used depending on its application or context.

Loss Function (L):

The loss function $L(\phi, D)$ for a model with parameters ϕ on dataset O . For example, the mean square error (MSE) is commonly used as the loss function in linear regression:

$$L(\phi, O) = \frac{1}{n} \sum_{i=1}^n (z_i - \hat{z}_i)^2$$

Where z_i are the actual values, and \hat{z}_i are the predicted values

Privacy Loss (PL):

The privacy loss $PL(\epsilon)$ quantifies the risk of information leakage as ϵ changes. Generally, lower ϵ implies higher privacy.

$$PL(\epsilon) = \frac{K}{\epsilon}$$

where K is a constant representing the baseline privacy risk when $\epsilon=1$.

Utility Measure (U):

The utility measure $U(\phi, O)$ evaluates the model's performance or effectiveness on dataset O , typically measured by metrics such as accuracy or predictive performance.

$$U(\phi, O) = \frac{1}{L(\phi, O)}$$

To achieve an optimal balance, we need to minimize the combined cost function $F(\epsilon)$, which considers both the privacy loss and the loss function (inversely related to utility).

$$F(\epsilon) = \alpha \cdot PL(\epsilon) + \beta \cdot L(\phi, O)$$

where α and β are weighting factors that balance the importance of privacy and utility.

Combined Cost Function:

Substituting $PL(\epsilon)$ and $L(\phi_\epsilon, O)$ into the cost function:

$$F(\epsilon) = \alpha \cdot \frac{K}{\epsilon} + \beta \cdot L(\phi_\epsilon, O)$$

Selecting Optimal ϵ :

To find the optimal ϵ , we will calculate the derivative of $F(\epsilon)$:

$$\frac{dF(\varepsilon)}{d\varepsilon} = -\alpha \cdot \frac{K}{\varepsilon^2} + \beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon} = 0$$

Solving for ε :

$$-\alpha \cdot \frac{K}{\varepsilon^2} + \beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon} = 0$$

$$\alpha \cdot \frac{K}{\varepsilon^2} = \beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon}$$

$$\varepsilon^2 = \frac{\alpha \cdot K}{\beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon}}$$

$$\varepsilon = \sqrt{\frac{\alpha \cdot K}{\beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon}}}$$

This provides a formula for selecting ε depending on constant K , weighting factors α & β , and the sensitivity of the loss function to ε .

Optimal value of ε can be derived as:

$$\varepsilon = \sqrt{\frac{\alpha \cdot K}{\beta \cdot \frac{\partial L(\phi_{\varepsilon}, O)}{\partial \varepsilon}}}$$

By using this formula, one can select ε in a way that manages both privacy (represented with α and K) and accuracy (represented with β and the sensitivity of the loss function). A lower value of ε provides stronger privacy guarantees. For selecting lower value of ε it requires increasing the value of α which increases the emphasis on minimizing privacy loss and decreasing the value of β that reduces the emphasis on preserving the utility or accuracy. By appropriately choosing α and β , one can control the emphasis on privacy versus utility, ensuring an optimal balance tailored to specific application requirements.

EXPERIMENTAL STUDY OF DIFFERENTIAL PRIVACY ON EHRs

Differential Privacy can be practically implemented by using multiple possible mechanism. In this thesis, we have used two different primary mechanism for the implementation of differential privacy in healthcare data. These methods are the Laplace and Gaussian mechanisms, algorithm of used mechanisms have described below in section 5.2.1 and 5.2.2

5.1. Experimental DP Framework in Healthcare

In this proposed scenario, underlying architecture can be used for practical implementation of DP in electronic healthcare data as demonstrated in the below Figure 4. In this model, Global differential privacy implemented on sensitive data to achieve privacy.

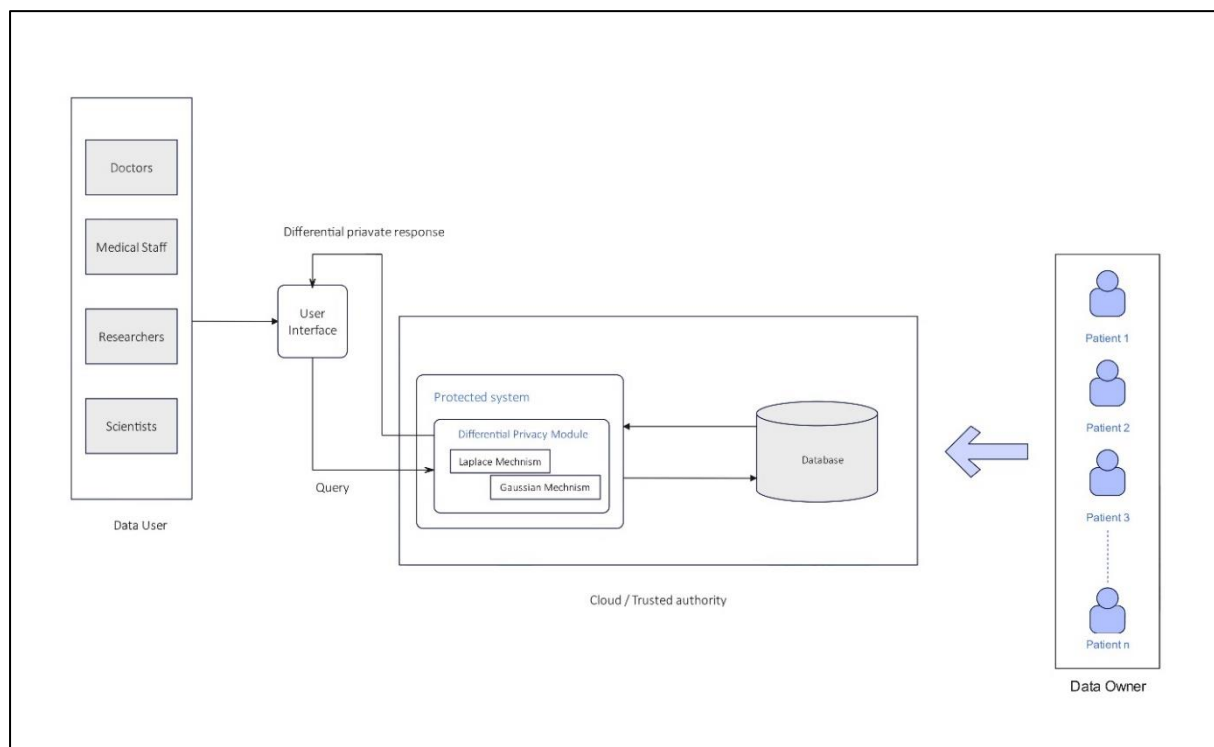


Figure 4 Architecture of the system

In this part of framework, users or data analysts will connect with the database by using user interface. The user will request the desired data in the form of queries and achieve the differentially private results. The protected system will receive queries that had made by data analyst or involved user and then it will pull out the unprocessed information from the stored database. After this it generates noise in the final outcome using DP in accordance with each query's global sensitivity. To achieve experimental results, python is selected as programming language on the basis of processing large datasets within the minimum time period and having ability to deal with computational tasks. Moreover, to handle large datasets PyDP, Pandas, Numpy and matplotlib libraries are used. PyDP is differential privacy project from Google, in which all the computation methods use Laplace noise only.

5.2. Algorithm Details

In this thesis two different primary methods such as Laplace and Gaussian mechanisms used for purpose of showing how to implement differential privacy. Further both algorithms described below that have applied in implementing DP.

5.2.1. Algorithm for Laplace Mechanism

```
1. function LAPLACE(O, Q, ε)
2. ΔQ = GS(Q) // Calculate global sensitivity
3. Y = [0] * k // Initialize noise array of size k
4. for a in range(k):
   noise[a] = Lap(ΔQ / ε) // Calculating sampled noise
5. end for
6. return Q(O) + noise[a] // Add noise to the actual count
7. end function
```

5.2.2. Algorithm for Gaussian Mechanism

```
1. function GAUSSIAN_MECHANISM(D, Q,  $\epsilon$ ,  $\delta$ , lower_limit, upper_limit)
2. filtered_data = filter(D, lower_limit, upper_limit) // Filter the dataset with given query
3. actual_count = count(filtered_data)// Compute the actual count
4.  $\sigma = \text{sqrt}(2 * \ln(1.25 / \delta)) / \epsilon$ // Calculate the standard deviation for Gaussian noise
5. noise = sample_normal(0,  $\sigma$ )// Generate Gaussian noise
6. noisy_count = actual_count + noise// Add noise to the actual count
7. return noisy_count
8. end function
```

5.3. Datasets Description

5.3.1. COVID-19 Home Nursing Dataset

Another dataset “COVID-19HomeNursing Data” has used to perform experiment by applying differential privacy in electronic healthcare data. This dataset also publicly available on data.cms.gov and Kaggle [41] websites. It consists of around 510,000 records with 39 number of attributes.

5.3.2. Breast Cancer Prediction Dataset

In this study, Breast Cancer Prediction dataset used that is publicly available on Kaggle [40]. The following dataset contains information of 20,000 digital and 20,000 film-screen mammograms collected against women with age group between 60-89 years for breast cancer prediction. It has almost 30,000 instances (patient record) with 13 attributes that are as follows:

Attribute	Type	Values
Age_At_The_Time_Of_Mammography	Numerical	≥ 53 , < 90
Radiologists_Assessment	String	Benign findings, highly suggestive of malignancy, Needs additional imaging, Negative, probably benign, Suspicious abnormality
Is_Binary_Indicator_Of_Cancer_Diagnosis	Boolean	True, False
Comparison_Mammogram_From_Mammography	String	Yes, no, Missing
Patients_BI_RADS_Breast_Density	String	Almost entirely fatty, extremely dense, heterogeneously dense, Scattered fibroglandular densities
Family_History_Of_Breast_Cancer	String	Yes, no, Missing
Current_Use_Of_Hormone_Therapy	String	Yes, no, Missing
Binary_Indicator	String	Yes, no, Missing
History_Of_Breast_Biopsy	String	Yes, no, Missing
Is_Film_Or_Digital_Mammogram	Boolean	True, False
Cancer_Type	String	Invasive cancer, ductal carcinoma in situ, No cancer diagnosis
Body_Mass_Index	Float	
Patients_Study_ID	Numerical	1 - 36715

Figure 5 Breast Cancer Prediction attributes

5.4. Experimental Results on Breast Cancer Prediction Dataset

In this implementation, it presents the comparison difference in the actual count and differential private outcome. First imported CSV file of breast cancer prediction dataset in IPython (Jupyter Notebook). After this performed multiple queries on this data to extract count for patients between different age groups during mammography with having true value of history of breast biopsy. In Figure 5 it shows the actual count for number of patients with different age groups without apply differential privacy.


```

In [8]: def typical_count_above(column_name1, column_name2, lower_limit, upper_limit):
        return Patient_Data[(Patient_Data[column_name1] > lower_limit) & (Patient_Data[column_name1] < upper_limit) & (Patient_Data[column_name2] == 'True')]

number_over_threshold = typical_count_above('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 60, 70)
P1 = print(f"Number of Patients with age between 60 and 70: {number_over_threshold}")
number_over_threshold = typical_count_above('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 70, 80)
P2 = print(f"Number of Patients with age between 70 and 80: {number_over_threshold}")
number_over_threshold = typical_count_above('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 80, 90)
P3 = print(f"Number of Patients with age between 80 and 90: {number_over_threshold}")
number_over_threshold = typical_count_above('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 90, 100)
P4 = print(f"Number of Patients with age between 90 and 100: {number_over_threshold}")

Number of Patients with age between 60 and 70: 5088
Number of Patients with age between 70 and 80: 3057
Number of Patients with age between 80 and 90: 915
Number of Patients with age between 90 and 100: 0

```

Figure 6 Actual count without DP on Breast Cancer Prediction Dataset

After getting true values Figure 6 shows the count for patients between different age groups during mammography with having true value of history of breast biopsy with applying differential privacy. Here differential privacy implemented through PyDP that uses Laplace mechanism. Experiment performed by selecting different values for Epsilon, here value for epsilon is $\epsilon = 0.2$.

```

In [9]: def private_count_between(column_name1, column_name2, privacy_budget, lower_limit, upper_limit):
        x = Count(privacy_budget, dtype='int')
        return x.quick_result(list(Patient_Data[(Patient_Data[column_name1] > lower_limit) & (Patient_Data[column_name1] < upper_limit) & (Patient_Data[column_name2] == 'True')]))

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 0.2, 60, 70)
print(f"PRIVATE: Number of Patients with age between 60 and 70: {private_number_between_threshold}")
private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 0.2, 70, 80)
print(f"PRIVATE: Number of Patients with age between 70 and 80: {private_number_between_threshold}")
private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 0.2, 80, 90)
print(f"PRIVATE: Number of Patients with age between 80 and 90: {private_number_between_threshold}")
private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 0.2, 90, 100)
print(f"PRIVATE: Number of Patients with age between 90 and 100: {private_number_between_threshold}")

PRIVATE: Number of Patients with age between 60 and 70: 5081
PRIVATE: Number of Patients with age between 70 and 80: 3035
PRIVATE: Number of Patients with age between 80 and 90: 911
PRIVATE: Number of Patients with age between 90 and 100: 3

```

Figure 7 Count with DP on Breast Cancer Prediction Dataset

Table 4 shows the comparison between actual value results and differentially private results. As it can be seen that noise introduced in actual count to make data private while maintaining data’s utility and accuracy. So, this data can be used by data analyst for research purposes.

Actual Results	DP Results
5088	5081
3057	3035
915	911
0	3

Table 4 Comparing Results using Breast Cancer Prediction Dataset

For comparison we plotted a graph between true values and differentially private values by setting the $\epsilon = 0.2$. Y-axis shows the count for patients between different age groups during mammography with having true value of history of breast biopsy and X-axis shows the patient's age group.

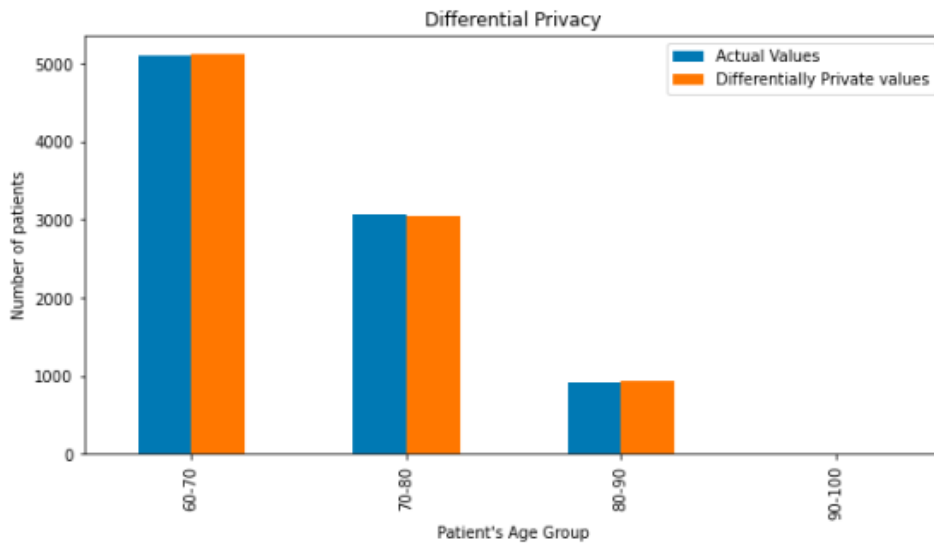


Figure 8 Results Comparisons using Breast Cancer Prediction Dataset

5.4.1. Varying Privacy Budget using Breast Cancer Prediction Dataset

Experiment performed for different values of epsilon to examine the protection level provided by DP mechanism with identical attributes but setting different values for privacy parameter. Here results evaluated by selecting different values for epsilon (0.02, 0.01, 0.2, 0.4, 0.6 & 0.8).

It can be seen in figure 8 that by decreasing ϵ value, it added more noise and vice versa.

```

x = Count(privacy_budget, dtype='int')
return x.quick_result(list(Patient_Data[(Patient_Data[column_name1] > lower_limit) & (Patient_Data[column_name1] < upper_lim

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P2 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.02: {private_number_between_threshold}")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P2 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.01: {private_number_between_threshold}")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P2 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.2: {private_number_between_threshold}")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P3 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.4: {private_number_between_threshold}")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P5 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.6: {private_number_between_threshold}")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Ty
P5 = print(f"PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.8: {private_number_between_threshold}")

```

Number of Patients with age between 60 and 70: **36**
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.02: 143
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.01: 59
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.2: 33
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.4: 38
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.6: 36
PRIVATE: Number of Patients with age between 60 and 70 with Privacy Budget 0.8: 38

Figure 9 Varying Epsilon values on Breast Cancer Prediction Dataset

For further demonstration, a graph plotted between privacy parameter epsilon and results for queries to compare the exact results and data with introduced noise. In Figure 9 it can be seen that the actual count for the number of patients between age 60 to 70 at the time of Mammography with having true value of history of breast biopsy is 36. After decreasing value for the privacy parameter epsilon by applying DP, the more noise added in actual value data.

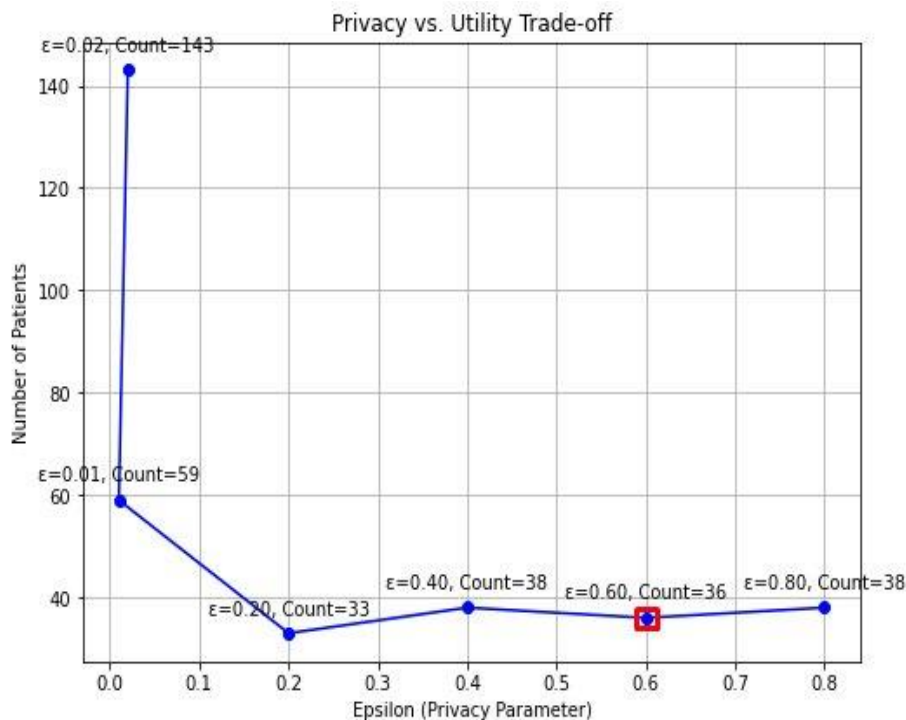


Figure 10 Analysis of privacy parameter using Breast Cancer Prediction Dataset

5.4.2. Time complexity Analysis with Breast Cancer Prediction Dataset

The execution time of queries will somewhat rise by increasing the conditions in the query. The first query filters the data to include only patients whose age at the time of the mammograph is between 60 and 70 years. It involves a simple range filter on one attribute. Second query adds another condition to the previous query by checking if the patient has a history of breast biopsy. It involves filtering based on two attributes. Next query is the most complex, combining multiple conditions across several attributes, including logical operations and comparisons. The time increases from 0.01544 seconds to 0.01999 seconds and then to 0.03899 seconds by increasing conditions. Slight rise in execution time observed with each additional attribute but it is not drastic, suggests that the filtering operations scale reasonably well by applying differential privacy with increasing the number of conditions. The time complexity in practice suggests that the operations are manageable within the given execution times.

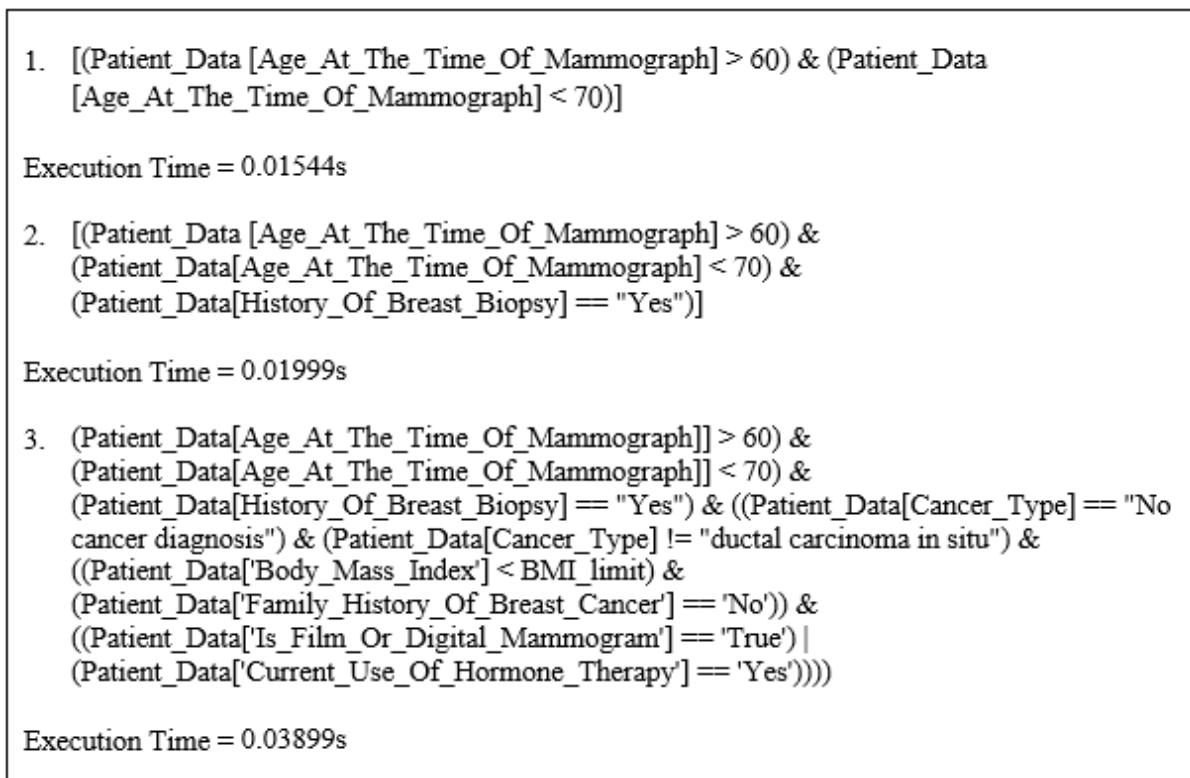


Figure 11 Queries with Time Comparison using Breast Cancer Prediction Dataset

```

start_time = time.time()

private_count_between(column_name1, privacy_budget, lower_limit, upper_limit):
    x = Count(privacy_budget, dtype='int')
    return x.quick_result(list(Patient_Data[(Patient_Data[column_name1] > lower_limit) & (Patient_Data[column_name1] < upper_limit)]

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 0.2, 60, 70)
print(f"PRIVATE: Number of Patients with age between 60 and 70: {private_number_between_threshold}")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Number of Patients with age between 60 and 70: 19313
 Execution time: 0.01544046401977539 seconds

Figure 12 Time Comparison Query 1 on Breast Cancer Prediction Dataset

```

private_count_between(column_name1, column_name2, column_name3, privacy_budget, lower_limit, upper_limit):
    x = Count(privacy_budget, dtype='int')
    return x.quick_result(list(Patient_Data[(Patient_Data[column_name1] > lower_limit) & (Patient_Data[column_name1] < upper_limit) & (Patient_Data[column_name2] == "Yes")

private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Type', 0.2, 60, 70)
print(f"PRIVATE: Number of Patients with age between 60 and 70: {private_number_between_threshold}")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Number of Patients with age between 60 and 70: 5087
 Execution time: 0.019996166229248047 seconds

Figure 13 Time Comparison Query 2 on Breast Cancer Prediction Dataset

```

def private_count_between(column_name1, column_name2, column_name3, privacy_budget, lower_limit, upper_limit, BMI_limit):
    x = Count(privacy_budget, dtype='int')
    Patient_Data['Body_Mass_Index'] = pd.to_numeric(Patient_Data['Body_Mass_Index'], errors='coerce')

    condition = (
        (Patient_Data[column_name1] > lower_limit) &
        (Patient_Data[column_name1] < upper_limit) &
        (Patient_Data[column_name2] == "Yes") &
        (
            (Patient_Data[column_name3] == "No cancer diagnosis") &
            (Patient_Data[column_name3] != "ductal carcinoma in situ") &
            (
                (Patient_Data['Body_Mass_Index'] < BMI_limit) &
                (Patient_Data['Family_History_Of_Breast_Cancer'] == 'No')
            ) &
            (
                (Patient_Data['Is_Film_Or_Digital_Mammogram'] == 'True') |
                (Patient_Data['Current_Use_Of_Hormone_Therapy'] == 'Yes')
            )
        )
    )

    return x.quick_result(list(Patient_Data[condition][column_name1]))

privacy_budget = 0.2
private_number_between_threshold = private_count_between('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Type', 0.2, 60, 70, 25)
print(f"PRIVATE: Number of Patients with age between 60 and 70: {private_number_between_threshold}")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Number of Patients with age between 60 and 70: 42
 Execution time: 0.03899812698364258 seconds

Figure 14 Time Comparison Query 3 on Breast Cancer Prediction Dataset

5.4.3. Comparison Analysis of Laplace vs Gaussian Mechanism

Implementation of Laplace mechanism using PYDP adds noise sampled from Laplace distribution based on privacy budget to provide results with provable privacy guarantees under differential privacy. To generate noise for Gaussian Mechanism it uses Gaussian distribution based on privacy budget which also provides differentially private results but typically used for scenarios where smoothness and sensitivity are key considerations. Laplace mechanism [38] generally efficient due to the simplicity of sampling from a Laplace distribution. While Gaussian Mechanism [39] slightly more computationally intensive due to the nature of sampling from a Gaussian distribution, which involves more complex calculations. Laplace mechanism often provides better accuracy for discrete counting queries. In conclusion, both Laplace and Gaussian mechanisms offer differential privacy solutions with different trade-offs in accuracy, implementation ease, and computational complexity. The choice of selection between them relies on particular needs and conditions of the differential privacy application and queries nature being performed on the datasets.

```
import pandas as pd
import time
from pydp.algorithms.laplacian import Count

start_time = time.time()
# Load dataset
Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\Data_Cancer\\data.csv')

# Define function for Laplace mechanism using PyDP
def laplace_mechanism(column_name1, column_name2, privacy_budget, lower_limit, upper_limit):
    x = Count(privacy_budget, dtype='int')
    return x.quick_result(list(Patient_Data[(Patient_Data[column_name1] > lower_limit) &
                                           (Patient_Data[column_name1] < upper_limit) &
                                           (Patient_Data[column_name2] == "Yes")][column_name1]))

# Example usage
privacy_budget = 0.2
lower_limit = 60
upper_limit = 70

result_laplace = laplace_mechanism('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', privacy_budget, lower_limit, up
print(f"PRIVATE (Laplace): Number of Patients with age between {lower_limit} and {upper_limit}: {result_laplace}")
end_time = time.time()

print(f"Execution time (Laplace): {end_time - start_time} seconds")
```

PRIVATE (Laplace): Number of Patients with age between 60 and 70: 5086
Execution time (Laplace): 0.18149495124816895 seconds

Figure 15 Laplace Mechanism

```

import time

start_time = time.time()
# Load dataset
Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\Data_Cancer\\data.csv')

# Define function for Gaussian mechanism
def gaussian_mechanism(column_name1, column_name2, column_name3, privacy_budget, lower_limit, upper_limit):
    # Filter data based on conditions
    filtered_data = Patient_Data[(Patient_Data[column_name1] > lower_limit) &
                                (Patient_Data[column_name1] < upper_limit) &
                                (Patient_Data[column_name2] == "Yes")][column_name3]

    # Compute the actual count
    actual_count = len(filtered_data)

    # Compute the amount of noise to add
    std_dev = np.sqrt(2 * np.log(1.25 / privacy_budget)) # Standard deviation for Gaussian noise
    noise = np.random.normal(loc=0, scale=std_dev) # Sample noise from a Gaussian distribution

    # Compute the noisy count
    noisy_count = actual_count + noise

    return noisy_count

# Example usage
privacy_budget = 0.2
lower_limit = 60
upper_limit = 70

result_gaussian = gaussian_mechanism('Age_At_The_Time_Of_Mammography', 'History_Of_Breast_Biopsy', 'Cancer_Type', privacy_budget)
print(f"PRIVATE (Gaussian): Number of Patients with age between {lower_limit} and {upper_limit}: {result_gaussian}")
end_time = time.time()

print(f"Execution time (Gaussian): {end_time - start_time} seconds")

```

PRIVATE (Gaussian): Number of Patients with age between 60 and 70: 5084.744629113658
Execution time (Gaussian): 0.23212838172912598 seconds

Figure 16 Gaussian Mechanism

5.5. Experimental Results on COVID-19 Home Nursing Dataset

To implement differential privacy, we performed different queries on this another dataset to show the comparison of real outcome and differential private outcome of queries. First query shows the overall count of beds that are in use in facilities of city “**RUSSELLVILLE**” where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6. In Figure 15, it shows the actual count for this query without implementing differential privacy.

```

import pandas as pd

# Load your dataset
Covid_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

# Function to count the total number of occupied beds in facilities located in a specific city,
# where Staff Weekly Confirmed COVID-19 is 0 and Residents Weekly Confirmed COVID-19 is less than a threshold
def count_total_occupied_beds_in_city_with_conditions(column_name, city):
    filtered_data = Covid_Data[
        (Covid_Data['Provider City'] == city) &
        (Covid_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Covid_Data['Residents Weekly Confirmed COVID-19'] < 6) # Corrected threshold
    ]
    return filtered_data[column_name].sum()

# Define parameters
city_name = 'RUSSELLVILLE'

# Get the count
total_occupied_beds_count = count_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name)
print(f"Total number of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 12491.0")

```

Total number of occupied beds in facilities in RUSSELLVILLE with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 12491.0

Figure 17 Query 1 Result without DP on COVID-19 Home Nursing Dataset

After getting the actual results, next Figure 16 shows the overall count of beds that are in use in facilities of city “**RUSSELLVILLE**” where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6 with implementing differential privacy. Here differential privacy implemented through PyDP using Laplace mechanism with selected epsilon value is $\epsilon = 0.2$.

```

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

def private_total_occupied_beds_in_city_with_conditions(column_name, city, epsilon, lower_bound, upper_bound):
    y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)
    Patient_Data['Staff Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Staff Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Residents Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Residents Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

    condition = (
        (Patient_Data['Provider City'] == city) &
        (Patient_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Patient_Data['Residents Weekly Confirmed COVID-19'] < 6)
    )

    # Filter data based on the conditions and drop NaN values
    filtered_data = Patient_Data[condition][column_name].dropna()

    # Convert filtered data to a List of integers
    data_list = filtered_data.astype(int).tolist()
    #print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

    # Apply differential privacy sum
    private_sum = y.quick_result(data_list)

    return private_sum

# Parameters
epsilon = 0.2
lower_bound = 5
upper_bound = 92
city_name = 'RUSSELLVILLE'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 12508")

```

C:\Users\ebryx\AppData\Local\Temp\ipykernel_19500\1963621815.py:5: DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or set low_memory=False.
Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')
PRIVATE: Total sum of occupied beds in facilities in RUSSELLVILLE with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 12508

Figure 18 Query1 Result with DP on COVID-19 Home Nursing Dataset

Second query in Figure 17 and Figure 18 shows the overall count of beds that are in use in facilities of city “ABILENE” where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6 without and with implementing Differential Privacy.

```

import pandas as pd

# Load your dataset
Covid_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

# Function to count the total number of occupied beds in facilities located in a specific city,
# where Staff Weekly Confirmed COVID-19 is 0 and Residents Weekly Confirmed COVID-19 is less than a threshold
def count_total_occupied_beds_in_city_with_conditions(column_name, city):
    filtered_data = Covid_Data[
        (Covid_Data['Provider City'] == city) &
        (Covid_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Covid_Data['Residents Weekly Confirmed COVID-19'] < 6) # Corrected threshold
    ]
    return filtered_data[column_name].sum()

# Define parameters
city_name = 'ABILENE'

# Get the count
total_occupied_beds_count = count_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name)
print(f"Total number of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 23857.0

```

Figure 19 Query 2 Result without DP on COVID-19 Home Nursing Dataset

```

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

def private_total_occupied_beds_in_city_with_conditions(column_name, city, epsilon, lower_bound, upper_bound):
    y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)
    Patient_Data['Staff Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Staff Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Residents Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Residents Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

    condition = (
        (Patient_Data['Provider City'] == city) &
        (Patient_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Patient_Data['Residents Weekly Confirmed COVID-19'] < 6)
    )

    # Filter data based on the conditions and drop NaN values
    filtered_data = Patient_Data[condition][column_name].dropna()

    # Convert filtered data to a List of integers
    data_list = filtered_data.astype(int).tolist()
    #print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

    # Apply differential privacy sum
    private_sum = y.quick_result(data_list)

    return private_sum

# Parameters
epsilon = 0.2
lower_bound = 5
upper_bound = 92
city_name = 'ABILENE'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 23570

```

C:\Users\ebryx\AppData\Local\Temp\ipykernel_16952\1754046723.py:5: DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or set low_memory=False.

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

PRIVATE: Total sum of occupied beds in facilities in ABILENE with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 23570

Figure 20 Query 2 Result with DP on COVID-19 Home Nursing Dataset

Third query in Figure 19 and Figure 20 represents the overall count of beds that are in use in facilities of city “YORK” where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6 without and with implementing Differential Privacy.

```

: import pandas as pd

# Load your dataset
Covid_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

# Function to count the total number of occupied beds in facilities located in a specific city,
# where Staff Weekly Confirmed COVID-19 is 0 and Residents Weekly Confirmed COVID-19 is less than a threshold
def count_total_occupied_beds_in_city_with_conditions(column_name, city):
    filtered_data = Covid_Data[
        (Covid_Data['Provider City'] == city) &
        (Covid_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Covid_Data['Residents Weekly Confirmed COVID-19'] < 6) # Corrected threshold
    ]
    return filtered_data[column_name].sum()

# Define parameters
city_name = 'YORK'

# Get the count
total_occupied_beds_count = count_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name)
print(f"Total number of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 41712.0

```

Figure 21 Query 3 Result without DP on COVID-19 Home Nursing Dataset

```

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

def private_total_occupied_beds_in_city_with_conditions(column_name, city, epsilon, lower_bound, upper_bound):
    y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)
    Patient_Data['Staff Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Staff Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Residents Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Residents Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

    condition = (
        (Patient_Data['Provider City'] == city) &
        (Patient_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Patient_Data['Residents Weekly Confirmed COVID-19'] < 6)
    )

    # Filter data based on the conditions and drop NaN values
    filtered_data = Patient_Data[condition][column_name].dropna()

    # Convert filtered data to a List of integers
    data_list = filtered_data.astype(int).tolist()
    #print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

    # Apply differential privacy sum
    private_sum = y.quick_result(data_list)

    return private_sum

# Parameters
epsilon = 0.2
lower_bound = 39
upper_bound = 364
city_name = 'YORK'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 40800

```

C:\Users\ebryx\AppData\Local\Temp\ipykernel_16952\1674008330.py:5: DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or set low_memory=False.

```

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

PRIVATE: Total sum of occupied beds in facilities in YORK with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 40800

```

Figure 22 Query 3 Result with DP on COVID-19 Home Nursing Dataset

Fourth query in Figure 21 and Figure 22 shows overall count of beds that are in use in facilities of city “WYNNEWOOD” where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6 without and with Differential Privacy.

```

import pandas as pd

# Load your dataset
Covid_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

# Function to count the total number of occupied beds in facilities located in a specific city,
# where Staff Weekly Confirmed COVID-19 is 0 and Residents Weekly Confirmed COVID-19 is less than a threshold
def count_total_occupied_beds_in_city_with_conditions(column_name, city):
    filtered_data = Covid_Data[
        (Covid_Data['Provider City'] == city) &
        (Covid_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Covid_Data['Residents Weekly Confirmed COVID-19'] < 6) # Corrected threshold
    ]
    return filtered_data[column_name].sum()

# Define parameters
city_name = 'WYNNEWOOD'

# Get the count
total_occupied_beds_count = count_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name)
print(f"Total number of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 5130.0")

```

Figure 23 Query 4 Result without DP on COVID-19 Home Nursing Dataset

```

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

def private_total_occupied_beds_in_city_with_conditions(column_name, city, epsilon, lower_bound, upper_bound):
    y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)
    Patient_Data['Staff Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Staff Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Residents Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Residents Weekly Confirmed COVID-19'], errors='coerce')
    Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

    condition = (
        (Patient_Data['Provider City'] == city) &
        (Patient_Data['Staff Weekly Confirmed COVID-19'] == 0) &
        (Patient_Data['Residents Weekly Confirmed COVID-19'] < 6)
    )

    # Filter data based on the conditions and drop NaN values
    filtered_data = Patient_Data[condition][column_name].dropna()

    # Convert filtered data to a list of integers
    data_list = filtered_data.astype(int).tolist()
    #print(f"Filtered data list ({len(data_list)} records): {data_list}") # Debug print

    # Apply differential privacy sum
    private_sum = y.quick_result(data_list)

    return private_sum

# Parameters
epsilon = 0.2
lower_bound = 152
upper_bound = 171
city_name = 'WYNNEWOOD'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 5254")

```

C:\Users\ebryx\AppData\Local\Temp\ipykernel_16952\275000161.py:5: DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or set low_memory=False.

Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\faclevel_2023\\faclevel_2023.csv')

PRIVATE: Total sum of occupied beds in facilities in WYNNEWOOD with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 5254

Figure 24 Query 4 Result with DP on COVID-19 Home Nursing Dataset

Table 5 shows the comparison between actual value results and differentially private results on multiple queries. It can be noticed that noise added in the actual outcome of queries while maintaining data utility and data accuracy.

City	Overall occupied beds	Overall occupied bedswith DP
RUSSELLVILLE	12,491	12,508
ABILENE	23,857	23,570
YORK	41,712	40,800
WYNNEWOOD	5,130	5,254

Table 5 Comparing Results using COVID-19 Home Nursing Dataset

For comparison we again plotted a graph between true values and differentially private values by setting the $\epsilon = 0.2$. Y-axis shows the count for number of occupied beds where COVID – 19 Confirmed Weekly Staff is zero and COVID – 19 Confirmed Weekly Residents are less than 6 in different cities while X-axis represents the statistics for different cities.

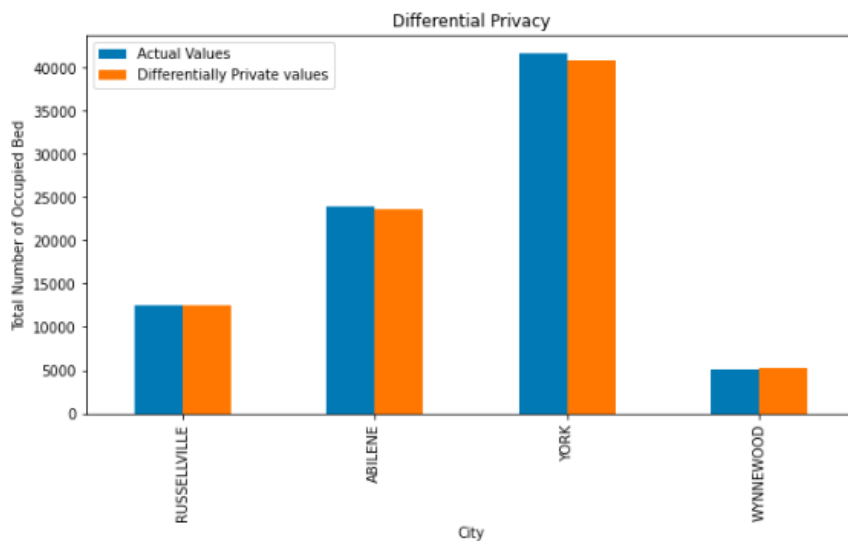


Figure 25 Results Comparison using COVID-19 Home Nursing Dataset

5.5.1. Varying Privacy Budget using COVID-19HomeNursing Dataset

In order to examine how noise affects the same query, we ran an experiment where we varied the value of epsilon between 0.8, 0.6, 0.4, 0.2, 0.01, and 0.02. In given results, we can notice that by decreasing the epsilon value, the amount of added noise also increases. Which means smaller the value of epsilon, greater the privacy required and the more noise is added. A compromise between privacy and utility exists. Adding more noise increases the privacy but it also reduces data utility. In differential privacy, parameter epsilon (ϵ) is used to control this trade-off between privacy and accuracy.

```
# Parameters
lower_bound = 152
upper_bound = 171
city_name = 'WYNNEWOOD'

total_occupied_beds_count = count_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name)
print(f"Total number of occupied beds in facilities of {city_name}: {total_occupied_beds_count}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.02, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.02: {private_occupied_beds_sum}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.01, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.01: {private_occupied_beds_sum}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.2, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.2: {private_occupied_beds_sum}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.4, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.4: {private_occupied_beds_sum}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.6, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.6: {private_occupied_beds_sum}")

private_occupied_beds_sum = total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, 0.8, lower_bound, upper_bound)
print(f"PRIVATE: Total sum of occupied beds in facilities of {city_name} with Privacy Budget 0.8: {private_occupied_beds_sum}")

C:\Users\ebryx\AppData\Local\Temp\ipykernel_16952\701267627.py:5: DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or set low_memory=False.
Patient_Data = pd.read_csv('C:\Users\ebryx\Downloads\faclevel_2023\faclevel_2023.csv')

Total number of occupied beds in facilities of WYNNEWOOD: 5130.0
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.02: 11021
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.01: 5852
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.2: 5086
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.4: 5174
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.6: 4841
PRIVATE: Total sum of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.8: 5203
```

Figure 26 Varying Epsilon values on Nursing Home COVID-19 Dataset

For further demonstration a graph plotted for different values of Epsilon (ϵ) in Figure 25. When value for epsilon is large then differential privacy added less noise which typically provides higher accuracy and utility as data remains closer to its true value. It helps data analyst to make informed decision for maintaining privacy level while also considering the usability and

reliability of the data.

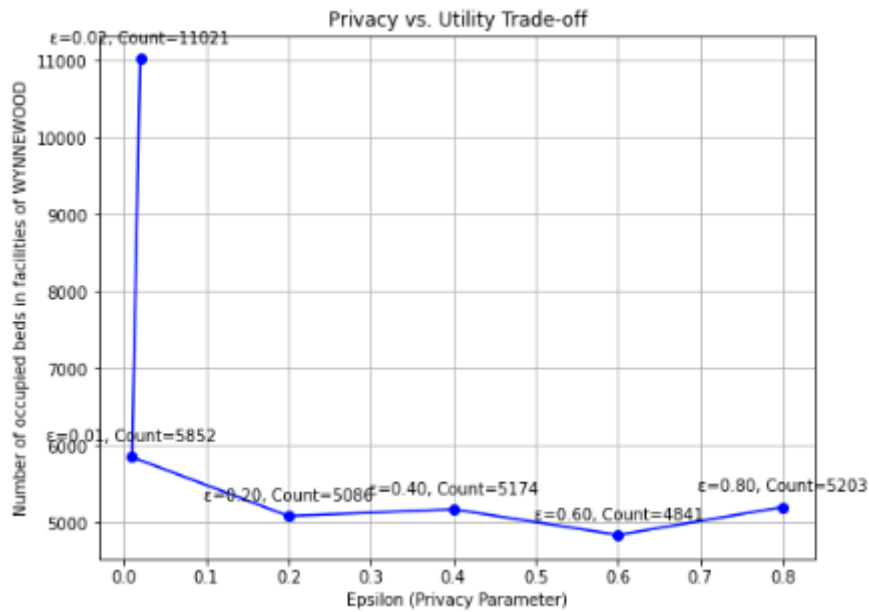


Figure 27 Analysis of privacy parameter using Nursing Home COVID-19 Dataset

5.5.2. Time complexity Analysis with Nursing Home COVID-19 Dataset

In differential privacy, the time complexity primarily relates to the computational cost of executing queries on potentially large datasets while ensuring privacy guarantees. First query involves filtering the dataset based on a single condition while second and third query involves complex filtering conditions including logical AND and OR operations across multiple columns. It can be noticed that by increasing the number of conditions in queries it will also increase the execution time for queries. The time increases from 0.06563 seconds to 0.35566 seconds and then to 7.34279 seconds due to increasing conditions. Slightly rise in execution time is typically incremental with each additional condition. However, the actual increase can also vary depending on the specific dataset characteristics (size, distribution, etc.) and the efficiency of the data processing system.

For previous dataset “Breast Cancer Prediction” with around 30,000 records and 13 attributes, it can be noticed that even with more complex queries, the execution times remain relatively low compared to larger datasets. For another dataset “Nursing Home COVID-19” with around

510,000 records and 39 attributes, execution times are slightly higher for larger dataset due to the sheer volume of data being processed.

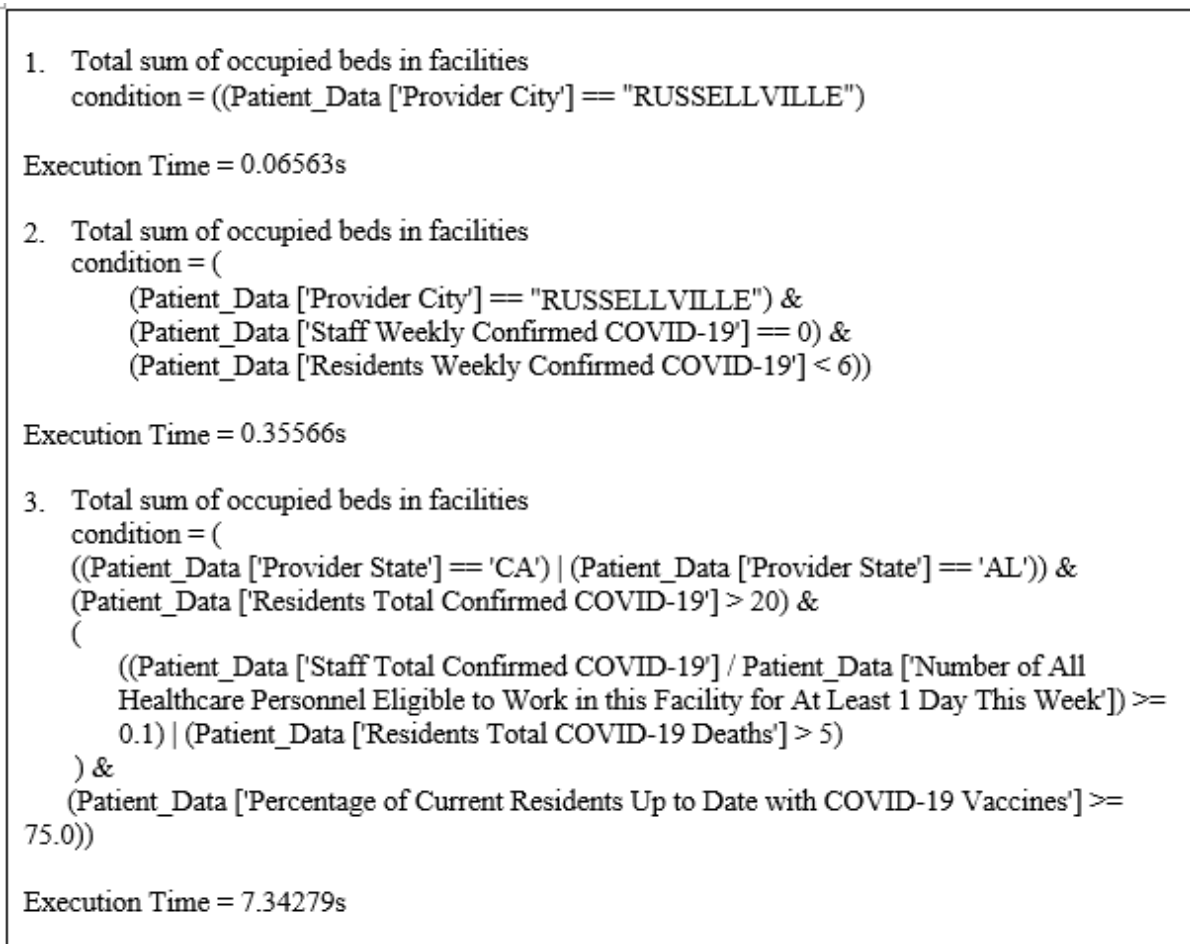


Figure 28 Queries with Time Comparison using Nursing Home COVID-19 Dataset

```

start_time = time.time()

# Load your dataset
#Patient_Data = pd.read_csv('C:\\Users\\ebryx\\Downloads\\facLevel_2023\\faclevel_2023.csv')

def private_total_occupied_beds_in_city_with_conditions(column_name, city, epsilon, lower_bound, upper_bound):
    y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)

    condition = (
        (Patient_Data['Provider City'] == city))

    # Filter data based on the conditions and drop NaN values
    filtered_data = Patient_Data[condition][column_name].dropna()

    # Convert filtered data to a List of integers
    data_list = filtered_data.astype(int).tolist()
    #print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

    # Apply differential privacy sum
    private_sum = y.quick_result(data_list)

    return private_sum

# Parameters
epsilon = 0.2
lower_bound = 5
upper_bound = 92
city_name = 'RUSSELLVILLE'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name}: {private_occupied_beds_sum}")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Total sum of occupied beds in facilities in RUSSELLVILLE: 15927
Execution time: 0.06563591957092285 seconds

Figure 29 Time Comparison Query 1 on Nursing Home COVID-19 Dataset

```

y = BoundedSum(epsilon, lower_bound=lower_bound, upper_bound=upper_bound)
Patient_Data['Staff Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Staff Weekly Confirmed COVID-19'], errors='coerce')
Patient_Data['Residents Weekly Confirmed COVID-19'] = pd.to_numeric(Patient_Data['Residents Weekly Confirmed COVID-19'], errors='coerce')
Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

condition = (
    (Patient_Data['Provider City'] == city) &
    (Patient_Data['Staff Weekly Confirmed COVID-19'] == 0) &
    (Patient_Data['Residents Weekly Confirmed COVID-19'] < 6)
)

# Filter data based on the conditions and drop NaN values
filtered_data = Patient_Data[condition][column_name].dropna()

# Convert filtered data to a List of integers
data_list = filtered_data.astype(int).tolist()
#print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

# Apply differential privacy sum
private_sum = y.quick_result(data_list)

return private_sum

# Parameters
epsilon = 0.2
lower_bound = 5
upper_bound = 92
city_name = 'RUSSELLVILLE'

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', city_name, epsilon)
print(f"PRIVATE: Total sum of occupied beds in facilities in {city_name} with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 13512")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Total sum of occupied beds in facilities in RUSSELLVILLE with Staff Weekly Confirmed COVID-19 = 0 and Residents Weekly Confirmed COVID-19 less than 6: 13512
Execution time: 0.3556632995605469 seconds

Figure 30 Time Comparison Query 2 on Nursing Home COVID-19 Dataset


```

Patient_Data['Percentage of Current Residents Up to Date with COVID-19 Vaccines'] = pd.to_numeric(Patient_Data['Percentage of
Patient_Data['Total Number of Occupied Beds'] = pd.to_numeric(Patient_Data['Total Number of Occupied Beds'], errors='coerce')

condition = (
    ((Patient_Data['Provider State'] == 'CA') | (Patient_Data['Provider State'] == 'AL')) &
    (Patient_Data['Residents Total Confirmed COVID-19'] > 20) &
    (
        ((Patient_Data['Staff Total Confirmed COVID-19'] / Patient_Data['Number of All Healthcare Personnel Eligible to Work
        (Patient_Data['Residents Total COVID-19 Deaths'] > 5)
    ) &
    (Patient_Data['Percentage of Current Residents Up to Date with COVID-19 Vaccines'] >= 75.0)
)

# Filter data based on the conditions and drop NaN values
filtered_data = Patient_Data[condition][column_name].dropna()

# Convert filtered data to a List of integers
data_list = filtered_data.astype(int).tolist()
#print(f"Filtered data List ({len(data_list)} records): {data_list}") # Debug print

# Apply differential privacy sum
private_sum = y.quick_result(data_list)

return private_sum

# Example usage:
epsilon = 0.2
lower_bound = 5
upper_bound = 92

# Compute the private total occupied beds sum
private_occupied_beds_sum = private_total_occupied_beds_in_city_with_conditions('Total Number of Occupied Beds', epsilon, lower_b
print(f"PRIVATE: Total sum of occupied beds in facilities: {private_occupied_beds_sum}")

end_time = time.time()
print(f"Execution time: {end_time - start_time} seconds")

```

PRIVATE: Total sum of occupied beds in facilities: 979025
Execution time: 7.342799425125122 seconds

Figure 31 Time Comparison Query 3 on Nursing Home COVID-19 Dataset

CONCLUSION AND FUTURE WORK

6. Conclusion

This chapter presented thorough summary of the major points of the thesis as well as possible future directions for research. This study proposed a differentiated privacy-based method for protecting healthcare data on the Internet of Medical Things. Initially, this thesis examined conventional approaches that were employed in the electronic healthcare data privacy process prior to the application of differential privacy. Then it had performed in-depth analysis of differential privacy and its core characteristics. The practical implementation showcased promising experimental results, demonstrating the application of differential privacy mechanisms across multiple queries. Variations in privacy parameter i.e. Privacy budget were analyzed to illustrate their impact on preserving privacy while maintaining data utility. Comparative analyses involving Laplace and Gaussian mechanisms were conducted, by analyzing both schemes in meeting privacy and security requirements with minimal computational overhead. Furthermore, the thesis carried out a thorough examination of time complexity through application of differential privacy to complex queries on datasets of various sizes.

6.1. Future Work

Even DP mechanism is sufficient effective to provide the necessary protection and privacy in the data but they are not always adaptable enough to use in every real-world situation, that could make it more difficult to achieve the required levels of security and usability. As a result, it would be ideal to examine and customize other mechanisms as well in the future. Applying data dependency differential privacy to actual datasets which exhibit natural reliance among

individuals might expose a weak assumption in data dependency differential privacy. In such cases inference attacks can exist under differential privacy mechanism. Thus, future research should take into account to create a better mechanism that enhances the current approach.

BIBLIOGRAPHY

- [1] Kumar, Bagesh, et al. "Medical Dataset Preparation and Privacy Preservation for Improving the Healthcare Facilities Using Federated Learning Approach." 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM). IEEE, 2023.
- [2] Joshi, Apoorva, and Pratima Gautam. "An Implementation of Hybrid method towards the Privacy of HealthCare Record." 2nd International Conference on Data, Engineering and Applications (IDEA). IEEE, 2020.
- [3] Suneetha, V., Salini Suresh, and ViswaJhananie. "A novel framework using apache spark for privacy preservation of healthcare big data." 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 2020.
- [4] Zhang, Li, et al. "Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system." IEEE Transactions on Network Science and Engineering (2022).
- [5] Seol, Kwangsoo, et al. "Privacy-preserving attribute-based access control model for XML-based electronic health record system." IEEE Access 6 (2018): 9114-9128.
- [6] Al Omar, Abdullah, et al. "Medibchain: A blockchain based privacy preserving platform for healthcare data." Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2017 International Workshops, Guangzhou, China, December 12-15, 2017, Proceedings 10. Springer International Publishing, 2017.
- [7] Aminifar, Amin, et al. "Extremely randomized trees with privacy preservation for distributed structured health data." IEEE Access 10 (2022): 6010-6027.
- [8] Charles, V. Benhar, D. Surendran, and A. SureshKumar. "Heart disease data based privacy preservation using enhanced ElGamal and ResNet classifier." Biomedical Signal Processing and Control 71 (2022): 103185.
- [9] Rosy, J. Vimal, and S. Britto Ramesh Kumar. "Optimized encryption based elliptical curve Diffie-Hellman approach for secure heart disease prediction." International Journal of Advanced Technology and Engineering Exploration 8.83 (2021): 1367.
- [10] Bi, Hongliang, Jiajia Liu, and Nei Kato. "Deep learning-based privacy preservation and data analytics for IoT enabled healthcare." IEEE Transactions on Industrial Informatics 18.7 (2021): 4798-4807.
- [11] K. Wang, C. -M. Chen, Z. Tie, M. Shojafar, S. Kumar and S. Kumari, "Forward Privacy Preservation in IoT-Enabled Healthcare Systems," in IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 1991-1999, March 2022, doi: 10.1109/TII.2021.3064691.
- [12] Ahmed, Junaid, et al. "On the physical layer security of federated learning based IoMT networks." IEEE Journal of Biomedical and Health Informatics 27.2 (2022): 691-697.

- [13] P. Singh, G. S. Gaba, A. Kaur, M. Hedabou and A. Gurtov, "Dew-Cloud-Based Hierarchical Federated Learning for Intrusion Detection in IoMT," in *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 722-731, Feb. 2023, doi: 10.1109/JBHI.2022.3186250.
- [14] M. Shabbir et al., "Enhancing Security of Health Information Using Modular Encryption Standard in Mobile Cloud Computing," in *IEEE Access*, vol. 9, pp. 8820-8834, 2021, doi: 10.1109/ACCESS.2021.3049564.
- [15] M. A. Khan, M. T. Quasim, N. S. Alghamdi and M. Y. Khan, "A Secure Framework for Authentication and Encryption Using Improved ECC for IoT-Based Medical Sensor Data," in *IEEE Access*, vol. 8, pp. 52018-52027, 2020, doi: 10.1109/ACCESS.2020.2980739.
- [16] Krall, Alexander, Daniel Finke, and Hui Yang. "Mosaic privacy-preserving mechanisms for healthcare analytics." *IEEE Journal of Biomedical and Health Informatics* 25.6 (2020): 2184-2192.
- [17] Xu, Chang, et al. "Achieving searchable and privacy-preserving data sharing for cloud-assisted E-healthcare system." *IEEE Internet of Things Journal* 6.5 (2019): 8345-8356.
- [18] Song, Jingcheng, et al. "A new secure arrangement for privacy-preserving data collection." *Computer Standards & Interfaces* 80 (2022): 103582.
- [19] Zhou, Xingguang, et al. "Privacy preservation for outsourced medical data with flexible access control." *IEEE Access* 6 (2018): 14827-14841.
- [20] Onesimu, J. Andrew, et al. "Privacy preserving attribute-focused anonymization scheme for healthcare data publishing." *IEEE Access* 10 (2022): 86979-86997.
- [21] Reddy, Vulapula Sridhar, and Barige Thirumala Rao. "A Combined Clustering and Geometric Data Perturbation Approach for Enriching Privacy Preservation of Healthcare Data in Hybrid Clouds." *International Journal of Intelligent Engineering & Systems* 11.1 (2018).
- [22] Zala, Kirtirajsinh, et al. "PRMS: design and development of patients' E-healthcare records management system for privacy preservation in third party cloud platforms." *IEEE Access* 10 (2022): 85777-85791.
- [23] Zhang, Mingwu, Yu Chen, and Willy Susilo. "PPO-CPQ: a privacy-preserving optimization of clinical pathway query for e-healthcare systems." *IEEE Internet of Things Journal* 7.10 (2020): 10660-10672.
- [24] <http://www.diva-portal.org/smash/get/diva2:1113852/FULLTEXT01.pdf>
- [25] Ram Mohan Rao, P., S. Murali Krishna, and A. P. Siva Kumar. "Privacy preservation techniques in big data analytics: a survey." *Journal of Big Data* 5.1 (2018): 33.
- [26] Zhu, Tianqing & Li, Gang & Zhou, Wanlei & Yu, Philip. (2017). *Differential Privacy and Applications*. 10.1007/978-3-319-62004-6.

- [27]Mir, Darakhshan J.. Differential privacy: an exploration of the privacy-utility landscape. Retrieved from <https://doi.org/doi:10.7282/T3FT8J2Z>
- [28] Frank McSherry and Kunal Talwar. "Mechanism Design via Differential Privacy". In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science. FOCS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 94–103. ISBN: 0-7695-3010-9.
- [29] Nguyen, T. T. (2019). Differential privacy for survival analysis and user data collection. Doctoral thesis, Nanyang Technological University, Singapore.
- [30]Kar, Tonny Shekha. "A Study on Privacy Preserving Data Publishing with Differential Privacy." (2017).
- [31]Al-Zobbi, Mohammed, SeyedShahrestani, and Chun Ruan. "A Multidimensional Sensitivity-Based Anonymization Method of Big Data." *Networks of the Future*. Chapman and Hall/CRC, 2017. 415-430.
- [32]Majeed, Abdul, Safiullah Khan, and SeongOun Hwang. "Toward privacy preservation using clustering based anonymization: recent advances and future research outlook." *IEEE Access* 10 (2022): 53066-53097.
- [33] Hassan, MuneebUl, Mubashir Husain Rehmani, and Jinjun Chen. "Differential privacy techniques for cyber physical systems: A survey." *IEEE Communications Surveys & Tutorials* 22.1 (2019): 746-789.
- [34] Holohan, Naoise, et al. "The bounded laplace mechanism in differential privacy." *arXiv preprint arXiv:1808.10410* (2018).
- [35] Garfinkel, Simson L., John M. Abowd, and Sarah Powazek. "Issues encountered deploying differential privacy." *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. 2018.
- [36] OpenMined. PyDP. Available online: <https://github.com/OpenMined/PyDP> (accessed on 20 June 2024).
- [37] Dong, Jinshuo, David Durfee, and Ryan Rogers. "Optimal differential privacy composition for exponential mechanisms." International Conference on Machine Learning. PMLR, 2020.
- [38] Huang, Wen, et al. "Improving laplace mechanism of differential privacy by personalized sampling." *2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom)*. IEEE, 2020.
- [39] Balle, Borja, and Yu-Xiang Wang. "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising." *International Conference on Machine Learning*. PMLR, 2018.

- [40] Hermessi, H. Breast Cancer Screening Data Set. Available online: <https://www.kaggle.com/datasets/haithemhermessi/breast-cancer-screening-data-set> (accessed on 20 June 2024).
- [41] Kennedy, C. Nursing Home COVID-19 Data. Available online: <https://www.kaggle.com/datasets/corykennedy/nursing-home-covid19-data> (accessed on 20 June 2024).
- [42] Hsu, Justin, et al. "Differential privacy: An economic method for choosing epsilon." *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE, 2014.
- [43] Inan, Ali, Mehmet Emre Gursoy, and YucelSaygin. "Sensitivity analysis for non-interactive differential privacy: Bounds and efficient algorithms." *IEEE Transactions on Dependable and Secure Computing* 17.1 (2017): 194-207.
- [44] Mohammed, Noman, et al. "Differentially private data release for data mining." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011.
- [45] Yang, Mengmeng, et al. "Local differential privacy and its applications: A comprehensive survey." *Computer Standards & Interfaces* (2023): 103827.
- [46] Wang, Huazheng, et al. "Global and local differential privacy for collaborative bandits." *Proceedings of the 14th ACM Conference on Recommender Systems*. 2020.
- [47] Phan, NhatHai, et al. "Adaptive laplace mechanism: Differential privacy preservation in deep learning." *2017 IEEE international conference on data mining (ICDM)*. IEEE, 2017.
- [48] Dandekar, Ashish, DebabrotaBasu, and Stéphane Bressan. "Differential privacy at risk: Bridging randomness and privacy budget." *arXiv preprint arXiv:2003.00973* (2020).
- [49] Sun, Zongkun, et al. "Differential privacy for data and model publishing of medical data." *Ieee Access* 7 (2019): 152103-152114.
- [50] Kaaniche, Nesrine, and Maryline Laurent. "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms." *Computer Communications* 111 (2017): 120-141.