future internet

MDPI

Article

Privacy-Preserving Data Analytics in Internet of Medical Things

Bakhtawar Mudassar [1], Shahzaib Tahir [1], Fawad Khan [2,*], Syed Aziz Shah [2], Syed Ikram Shah [3] and Qammer Hussain Abbasi [4]

[1] Department of Information Security, College of Signals, National University of Sciences and Technology, H12, Islamabad 44000, Pakistan; bakhtimudassar@gmail.com (B.M.); shahzaib.tahir@mcs.edu.pk (S.T.)
[2] Research Centre for Intelligent Healthcare, Coventry University, Coventry CV1 5RW, UK; syed.shah@coventry.ac.uk
[3] College of Electrical and Mechanical Engineering, National University of Sciences and Technology, H12, Islamabad 44000, Pakistan; syed.shah15@ce.ceme.edu.pk
[4] School of Engineering, James Watt Building (South), University of Glasgow, Glasgow G12 8QQ, UK; qammer.abbasi@glasgow.ac.uk
* Correspondence: fawad.khan2@coventry.ac.uk

Abstract: 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 research. Protecting patient privacy in the era of digital health records is a major challenge, as there could be a chance of privacy leakage during the process of collecting patient data. To overcome this issue, we propose a secure, privacy-preserving scheme for healthcare data to ensure maximum privacy of an individual while also maintaining their utility and allowing for the performance of queries based on sensitive attributes under differential privacy. We implemented differential privacy on two publicly available healthcare datasets, the Breast Cancer Prediction Dataset and the Nursing Home COVID-19 Dataset. Moreover, we examined the impact of varying privacy parameter (ε) values on both the privacy and utility of the data. A significant part of this study involved the selection of ε, which determines the degree of privacy protection. We also conducted a computational time comparison by performing multiple complex queries on these datasets to analyse 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.

Keywords: differential privacy; healthcare data; data sharing; user privacy; data utility

check for updates

Citation: Mudassar, B.; Tahir, S.; Khan, F.; Shah, S.A.; Shah, S.I.; Abbasi, Q.H. Privacy-Preserving Data Analytics in Internet of Medical Things. *Future Internet* **2024**, *16*, 407. https://doi.org/10.3390/fi16110407

Academic Editors: Kuo-Yu Tsai and Chung-Wei Kuo

Received: 5 September 2024
Revised: 15 October 2024
Accepted: 1 November 2024
Published: 5 November 2024

1. Introduction

With the increasing integration of electronic healthcare records and other forms of health data into the healthcare ecosystem, safeguarding patients' sensitive or personal data from breaches and unauthorised 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 [1]. In order to facilitate progress in epidemiological research and advance public health initiatives, health information needs to be readily available and functional. Storing and sharing these data publicly, even in anonymized forms, patient privacy can still be violated by disclosing such information [2].

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 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 [3]. Differential privacy preserves individual personal identifiable information by proving mathematically that the 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, a balance between 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 the healthcare sector is particularly appealing, as the sector requires accurate and comprehensive data for research and industrial purposes. In healthcare organizations, a 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 third party for research purposes, policy making, and collaborative initiatives.

Despite its potential, the implementation of differential privacy also faces challenges in the healthcare industry [4]. 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 [5]. To solve these concerns, a multidisciplinary approach is required that considers moral dilemmas, robust policy development, and other advancements. In contrast to other differential privacy implementations, we made an attempt to maintain patient data privacy in the field of electronic healthcare data by answering the complicated, complex queries under differential privacy. Additionally, we evaluate our approach against other differential privacy frameworks by showing how well it performs across a range of privacy parameter values, highlighting its robustness and adaptability in preserving data utility while guaranteeing strong privacy protections. We perform a detailed time complexity analysis, showing how effective our approach is at handling complex queries in reasonable amounts of computational time. Next, we implement our method on datasets of different sizes to evaluate its scalability. Our results indicate that our approach performs consistently at various scales, effectively handling both small and large datasets.

The proposed privacy-preserving scheme for healthcare data will contribute in the following ways:

1. By employing differential privacy techniques on publicly available healthcare datasets to demonstrate the practical feasibility and effectiveness of preserving patient privacy;
2. By demonstrating that differential privacy can effectively balance privacy and utility, guaranteeing that converted results can still be used for insightful analysis and research;
3. By examining the impact of varying values of the privacy budget (epsilon) on both privacy protection and data utility;
4. By conducting a comparative analysis of the Gaussian and Laplace mechanisms within the differential privacy framework and evaluating the performance and usefulness of these mechanisms, emphasising the situations in which each mechanism operates at its best;
5. By analysing the time complexity of applying differential privacy techniques, focusing on computational efficiency as the parameters of user queries increase, and providing insights into the scalability of differential privacy methods, offering guidance on their practical implementation in real-world healthcare data systems.

The rest of this paper is organized as follows. Section 2 describes the theoretical background of early privacy preservation techniques. A review of related work is presented in Section 3, while Section 4 provides a comprehensive overview of the proposed method for privacy preservation using differential privacy. Experimental results are reported and discussed in Section 5. Concluding remarks and directions for future work are presented in Section 6.

2. Background

Privacy preservation encompasses various strategies and technologies aimed at protecting individuals' personal data and information. The following privacy models have been used for privacy preservation when releasing data publicly: anonymization, t-closeness, K-anonymity, and I-diversity, among many others. These methods are summarized in Table 1.

Table 1. Comparison of privacy preservation techniques.

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

2.1. Anonymization

Anonymization is a method of transforming information that can be uniquely identified (PII) into an unidentifiable form so that it cannot be to re-linked to an individual without additional information [6]. The goal is to secure the personal identification of a person while enabling the public sharing of data. The data collector removes the particular uniquely identified information, such as name, phone number, or location. However, there are still challenges in data anonymization, even if specific identifiers are removed.

Sometimes, it is possible to re-identify anonymized data by data linkage attacks, especially when combined with other datasets. Data masking techniques are used in data anonymization, such as randomization, which replaces identifiable data with random values, and pseudonymization, which 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 of re-identifying data.

2.2. K-Anonymity

Researchers have proposed multiple other methods for privacy preservation to overcome the shortcomings of simple data anonymization. K-anonymity is considered a widely used method 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 [7,8]. Elements of data like age, sex, and occupation that could potentially identify individuals are grouped into categories. Individuals who have similar characteristics are grouped together. Instead of recording exact ages, age can be grouped into ranges like (30–35 years). Each group should contain at least k individuals. By organizing the data this way, it is much harder for someone to figure out who a specific person is. However, this technique is still vulnerable to homogeneity and background knowledge attacks.

2.3. I-Diversity

To deal with the above-mentioned drawbacks, this technique emphasises 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) [9]. K-anonymity guarantees that, using quasi-identifiers, every record can be 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 an 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 and gender and living in the same ZIP code. Within each group formed by identical quasi-identifiers, I-diversity requires that the sensitive attributes be diverse. There should be at least ℓ different conditions of sensitive attributes. 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 types of privacy attacks, like skewness and similarity attacks.

2.4. T-Closeness

T-closeness is a technique for maintaining privacy that aims to rectify the inadequacies of k-anonymity and I-diversity, particularly vulnerabilities related to skewness and similarity attacks [7]. 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.

2.5. Cryptographic Techniques

Before making data available to the public, the data curator could encrypt them [10]. However, it is extremely difficult to encrypt vast amounts of data using standard encryption techniques, and such methods must only be put into practice when gathering data. Homomorphic encryption allows calculations to be performed on encrypted data, producing an encrypted output with final results equivalent to a plaintext operation after re-encryption.

Similarly, secure multiparty computation permits several parties to work together to jointly compute a function over their private inputs. Moreover, Blockchain technology is used in privacy preservation of data, employing 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 of 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.

2.6. Multidimensional Sensitivity-Based Anonymization

Multidimensional sensitivity-based anonymization is an improved kind of anonymization that can be used to outperform more conventional anonymization methods [11]. It identifies which attributes are sensitive in the datasets. It includes both quasi-identifiers (which can identify individuals when combined) and direct identifier attributes. It evaluates the sensitivity of each attribute. Some attributes may be more sensitive than others, and this sensitivity can be quantified. Anonymization strategies such as generalization, suppression, or noise addition are implemented to make sure the data cannot be traced back to individuals. The level of anonymization that is used can change depending on how sensitive each attribute is. Interactions between multiple attributes are considered. Even if individual attributes are anonymized, 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 enhanced privacy by considering the sensitivity of multiple attributes and their interactions. It allows for different levels of anonymization based on the sensitivity of each attribute and minimizes the risk of re-identification through the use of combinations of attributes. This technique is better suited for large scales an static data. Moreover, it is not applicable to streaming data.

2.7. Data Distribution Technique

This technique involves the splitting of data over multiple sites. There are two main methods for distributing data. Both strategies—horizontal distribution and vertical distribution [12]—decentralize data processing and storage in an effort to reduce the possibility of privacy breaches. In horizontal distribution, a subset of a dataset's records or rows is stored 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 across different sites, a site can implement its own privacy policies and controls according to their specific requirements, with less chance of single-point failure. Only a single subset of data is compromised, regardless of whether a site is compromised. Queries of 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. Multiple websites may need to maintain various kinds of data on the same people, for instance, financial data may be stored on one website and personally identifiable information on another. In the case of a breach, this method reduces the risk of complete data exposure by separately storing sensitive attributes.

3. Related Works

This section reviews relevant research concerning privacy preservation in healthcare data. Kumar et al. [13] focused on the necessity of large datasets for the 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 highlighted the potential of federated learning to overcome these barriers by allowing data to remain with the local party (such as a hospital), ensuring confidentiality and compliance with data protection regulations. The authors specifically focused on two algorithms: Federated Averaging (FedAvg) and FedProx. Using federated learning in healthcare highlights several limitations; for example, privacy risks still exist, as model updates sent to a central server could be intercepted. Communication costs are also notable due to frequency of data exchanges between clients and the server.

A hybrid strategy was presented by Joshi et al. [14] and 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 of frequent patterns and anonymization processes to conceal sensitive information.

In order to solve privacy problems in healthcare big data, Suneetha et al. [15] offered 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. [16] 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. [17] thoroughly implemented an attribute-based access control model to protect electronic healthcare records (EHRs) in an XML-based system. In their model, sensitive data are partially encrypted by the system using XML encryption after access control. Next, the data are secured against unauthorized changes and access by utilizing XML digital signatures.

Research by Abdullah et al. [18] 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 are 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. [19] implemented a machine learning approach using Extremely Randomized Trees (ERTs) that is specifically designed for health data with distributed structures. This distributed ERT technique modifies the traditional approach to adapt to a distributed setting, ensuring that data privacy is upheld by avoiding a direct data environment. Instead, data insights are derived through secure multi-party computation methods that allow entities to collaborate without exposing their private data.

Charles, V., et al. [20] used the improved ElGamal and ResNet classifiers to maintaining privacy in a heart disease database. In their model, patients use wearable devices, and sensors connected to these devices gather data and transfer them to a microprocessor; the collected data are then sent to the cloud. The upgraded ElGamal encryption technique is used by the trusted cloud to safely protect patient data from outside threats. To accurately predict whether a patient is suffering with heart disease or not, a CNN Classifier with ResNet-50 is employed for data categorization and refinement. However, key generation and encryption add to the computational cost, and implementation depends on a Trusted Authority (TA).

Research by Wang, K., et al. [21] outlined 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 them. Forward privacy ensures that updates to the dataset do not reveal any information about the contents of past search queries, thereby 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. [22] described a methodology that combines Federated Learning (FL) with Physical Layer Security (PLS) to enhance the privacy and efficiency in medical records. 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. [23] described uses cloud computing to facilitate the distribution of a Hierarchical Long Short-Term Memory (HLSTM) architecture among distributed dew servers. Before the data are utilized to train the model, they are pre-processed to assure the quality of 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 phrase vectors 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.

Shabbir, M., et al. [24] implemented a Modular Encryption Standard (MES) to secure health data in Mobile Cloud Computing (MCC) environments. Health data are categorized and recognized according to their 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 are treated in accordance with their security classification at every stage, starting with the user's mobile device and continuing to the cloud.

Krall et al. [25] explored an innovative way to maintain privacy in predictive healthcare analytics by utilizing 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 for the gradient parts of the objective function linked to sensitive characteristics than for 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. [26]. 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 ranges and multi-keyword searches using dynamic searchable encryption techniques. By using this technique, patient privacy is protected because, guaranteeing that the data are encrypted during all operations. Numerical analysis queries of 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 erroneous data and confirm the accuracy of PHI files.

A technique of attribute-focused anonymization for publishing healthcare data was proposed by Onesimu, J. A., et al. [27]. 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. [28] focused 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. For access control, it allows patients to assign access rights to their EHRs for doctors and relatives. For data hiding, it employs anonymization to protect sensitive EHR data from unauthorized access. Table 2 presents a comparison of existing privacy preservation mechanisms in healthcare.

With smart phones becoming more common nowadays, authentication methods are critical processes that require strong and secure authentication to maintain user privacy. Existing behavioural biometrics for authentication in smart phones can be compromised. To improve this, Cong Wu et al. has presented a new technique called the "BIOHOLD" method for user authentication [29]. BIOHOLD uses natural gestures for authentication, effectively mitigating behavioural variability. This proposed method records hand shape and finger movements during regular phone use to authenticate users. Evaluation of this approach on a dataset collected from 20 participants showed that BIOHOLD has a very low error rate and is also protected against common security threats.

Table 2. Study of existing privacy preservation mechanisms in healthcare.

Ref.	System Model	Goals	Limitations	Privacy-Preserving Technique(s)	Trust Model
Joshi et al., 2020 [2]	Hybrid method using the FP-Growth algorithm and anonymization	Hide sensitive patient data in healthcare datasets using hybrid approaches	Increased time and memory requirements for large datasets	FP-Growth algorithm	Anonymization and association rule-hiding techniques
Suneetha et al., 2020 [3]	Uses Apache Spark for privacy preservation in healthcare big data	Use of K-anonymity and L-diversity for the protection of patient data in healthcare	Potential data segregation issues for transfer to HDFS	K-anonymity and L-diversity	Handling of healthcare big data with Apache Spark for faster processing
Zhang et al., 2022 [6]	Federated learning in combination with homomorphic encryption	Ensure privacy preservation of patient data in IoT-enabled healthcare systems	Increased computation and communication overhead and dropout clients not handled	Homomorphic encryption, Shamir secret sharing, and Diffie–Hellman key agreement	Honest but curious; semi-honest participant
Seol et al., 2018 [7]	Attribute-Based Access Control (ABAC) using XACML	Provide restricted access and protect patient privacy in EHR systems	Increased complexity and computational overhead due to encryption and access control mechanisms	XML encryption and digital signatures	Assumes a semi-trusted cloud environment and authorized users to access EHR data
Abdullah et al., 2017 [8]	Uses the MediBchain framework based on Blockchain	Ensure privacy, security, and integrity of healthcare data using blockchain	Increased complexity and computational overhead and requires secure key management	Blockchain and public key encryption (ECC)	Decentralized patient-centric model

Table 2. *Cont.*

Ref.	System Model	Goals	Limitations	Privacy-Preserving Technique(s)	Trust Model
Aminifar, A., et al., 2022 [9]	Uses distributed extremely randomized trees for privacy preservation	Ensure privacy-preserving machine learning for distributed health data	Increased complexity and computational overhead and handling of missing values	Secure Multi-Party Computation (SMC) and encryption	Semi-honest model that assumes no collusion among k parties
Wang, K., et al., 2021 [13]	Uses forward-privacy searchable encryption in electronic healthcare data	Ensures the privacy and security of healthcare data while enabling efficient search and data sharing	Potential exposure of search patterns and requires efficient key management	Searchable Encryption (SE), Pseudo-Random Function (PRF), and trapdoor permutation	Semi-honest adversaries; trust in cloud service provider to follow protocol without collusion
Ahmed, J., et al., 2021 [14]	Federated learning (FL) combined with physical layer security (PLS) in IoMT networks	Enhance privacy and security in IoMT networks by using FL and PLS	Increased complexity and computational overhead and potential for localized eavesdroppers	Homomorphic encryption, PLS, and blockchain	Assumes a semi-trusted central server and devices in a hierarchical network
Singh, P., et al., 2022 [15]	Dew–cloud-based Hierarchical Federated Learning (HFL) using hierarchical LSTM (HLSTM) for IoMT networks	Enhance data privacy, availability, and intrusion detection accuracy in IoMT networks using HFL and HLSTM	Complexity in managing hierarchical models and potential latency in federated learning updates	Homomorphic encryption and federated learning	Trust in decentralized dew and cloud servers; assumes secure communication channels
Shabbir, M., et al., 2021 [14]	Modular encryption standard (MES) in mobile cloud computing (MCC)	Secure health information in mobile cloud computing environments	Increased complexity and computational cost and layered modelling performance issues	Modular encryption standard (MES)	Assumes trust in cloud service providers and mobile devices
Krall et al., 2020 [16]	Mosaic gradient perturbation (MGP) in IoT-enabled healthcare systems using predictive modelling	Preserve privacy and reduce the possibility of model inversion attacks with model accuracy	Increased complexity in fine-tuning trade-offs and potential computational overhead in large-scale implementations	Differential privacy and gradient perturbation	Semi-trusted entities within a decentralized framework; assumes honest but curious adversaries
Xu et al., 2019 [18]	E-healthcare system with cloud assistance that includes wearables, cloud servers, IoT gateways, and health service providers (HSPs)	Enable secure and efficient sharing of patient health information (PHI) using searchable encryption	The 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	Searchable encryption, privacy-preserving equality test (PET) protocol, Variant Bloom Filter (VBF), and Message authentication codes (MACs)	The trusted authority (TA) is fully trusted, cloud servers are honest but curious, and IoT gateways and health service providers (HSPs) are trusted

Table 2. *Cont.*

Ref.	System Model	Goals	Limitations	Privacy-Preserving Technique(s)	Trust Model
Onesimu, JA., et al., 2020 [22]	Publishing of healthcare data using l-diverse slicing and a fixed-interval technique for attribute-focused anonymization	Privacy preservation when releasing EHR data, providing maximum data utility while ensuring privacy	Increased computational complexity with large datasets and vulnerability to certain privacy attacks with fixed methods	Enhanced l-diverse slicing for the grouping of attributes and fixed-interval anonymization for numerical attributes	Internal data controllers are trusted, and data analysts are considered potential adversaries

4. Differential Privacy

Differential privacy is a mathematical mechanism that offers a robust privacy guarantee throughout data analysis, enabling the public exchange of data. This idea was first presented by Cynthia Dwork and associates in the early 2000s [30]. It protects an individual's privacy by making sure the the presence or absence of an individual in the dataset cannot impact the results of any search. It helps to make guarantee that private information about an individual is kept hidden upon aggregated data analysis. The fundamental concept of differential privacy is the introduction of controlled randomness into the data analysis process. Differential privacy guarantees that no single data point's privacy is compromised in the output by carefully adding the noise to query results [31].

Definition 1. *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 [32].*

$$\Pr[R(O_1) \in Q] \leq e^{\epsilon} \Pr[R(O_2) \in Q] + \delta \tag{1}$$

Neighbouring datasets O_1 and O_2 are adjacent datasets that differ by no more than one element. Here, a positive privacy parameter called epsilon (ϵ) is used to evaluate the loss of privacy. A smaller value of epsilon indicates stronger privacy. While δ is a positive parameter, which is usually close to zero, that permits a minor relaxation of the strict privacy guarantee. It is pure differential privacy if the value of δ = 0; then, we obtain a stricter definition of differential privacy.

$$\frac{\Pr[R(O_1) \in Q]}{\Pr[R(O_2) \in Q]} \leq e^{\epsilon} \tag{2}$$

4.1. Mechanisms of Differential Privacy

4.1.1. Laplace Mechanism

This mechanism used in differential privacy to add a controlled amount of noise to the output of computations [33]. The Laplace mechanism can be applied to achieve differential privacy to make sure that the presence or absence of an individual will not significantly alter the result of a calculation or analysis. The amount of noise added in computation's output is evaluated according to the Laplace distribution using the Laplace mechanism [34].

Definition 2. *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 follows:*

$$P(O) = f(O) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right) \tag{3}$$

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

$$f(u) = \frac{\epsilon}{2b} \exp\left(-\frac{\epsilon|u|}{b}\right) \tag{4}$$

The privacy budget ϵ is a privacy parameter that is responsible obtaining differential privacy [36]. A lesser value of ϵ provides higher privacy protection. In Figure 1, b is a scale parameter used to determines the spread of the distribution ($b > 0$). The larger the value of b, the greater the increase in the amount of added noise, leading to more fluctuations in the final results. $|u|$ defines the absolute value of u to ensure that the Laplace distribution is symmetric around its mean. The value of $|u|$ is often calculated as

$$|u| = \frac{b\Delta f}{\epsilon} \tag{5}$$

Here, privacy and utility are both trade-offs with values larger than $|u|$, providing stronger privacy but reducing the utility of the output and vice versa. The change in the output of a function applied to two adjacent datasets, i.e., neighbouring datasets that vary in terms of the presence or absence of a single individual's record, is known as the sensitivity of the Δf function.

$$\Delta f = \max_{d_1, d_2 : |d_1 \Delta d_2| = 1} ||f(d_1) - f(d_2)||_1 \tag{6}$$

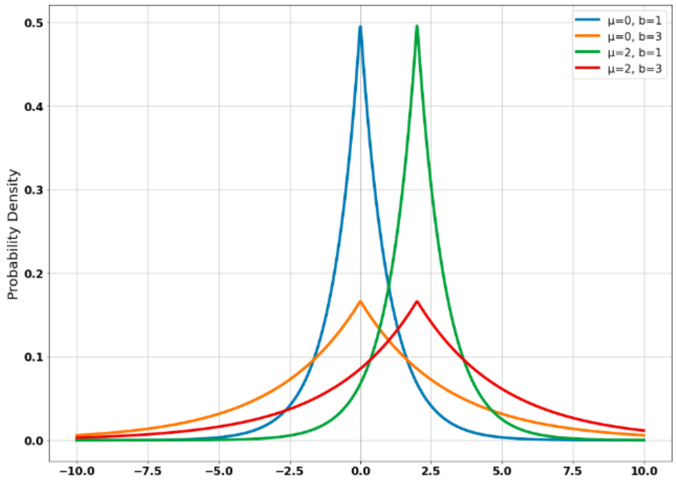


Figure 1. Laplace distribution.

4.1.2. Gaussian Mechanism

The Gaussian mechanism is an alternative to the Laplace mechanism used to inject noise into the results of a function to ensuring privacy while preserving data utility in differential privacy. Because of its bell-shaped distribution, the Gaussian mechanism [37] 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 3. *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 follows:*

$$P(O) = f(O) + \mathcal{N}(0, \sigma^2) \tag{7}$$

where $f(O)$ is the exact result of the f function on dataset O. $\mathcal{N}(\sigma^2)$ represents the noise that is evaluated through a Gaussian distribution. σ refers to a parameter that is evaluated according to the sensitivity of the f function. It measures how much the output of $f(O)$ can change when one element of O is altered. Here, s is the sensitivity of the function, and log represents the natural logarithm [38].

$$\sigma^2 = \frac{2s^2 \log\left(\frac{1.25}{\delta}\right)}{\epsilon^2} \tag{8}$$

The privacy parameter that regulates the quantity of generated noise is ϵ. This mechanism balances privacy and utility by controlling the ϵ and σ parameters. Larger ϵ and smaller σ values provide weaker privacy guarantees, as they add less noise while providing higher utility. In the same way, smaller ϵ and larger σ values add more noise, strengthening privacy but potentially reducing utility.

4.1.3. 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 proposed by McSherry and Talwar [39]. This mechanism deals with non-quantitative queries.

Definition 4. *Given a set of N of acceptable outputs and a utility function ($u : N \times O \rightarrow \mathbb{R}$) that quantifies the desirability of every outcome ($n \in N$), given a dataset (O), this mechanism [40] probabilistically selects an output (y) to ensure ϵ-differential privacy:*

$$P(O) = \Pr[n \mid O] \propto \exp\left(\frac{\epsilon u(n, O)}{2\Delta u}\right) \tag{9}$$

where $u(n, O)$ is the utility of output n given dataset O [35]. 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, the amount of noise added depends on the privacy parameter (ϵ).

4.2. 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 Figure 2 [32]. The choice between local and global differential privacy depends on the specific application, the level of trust in the central server, and the desired privacy guarantees.

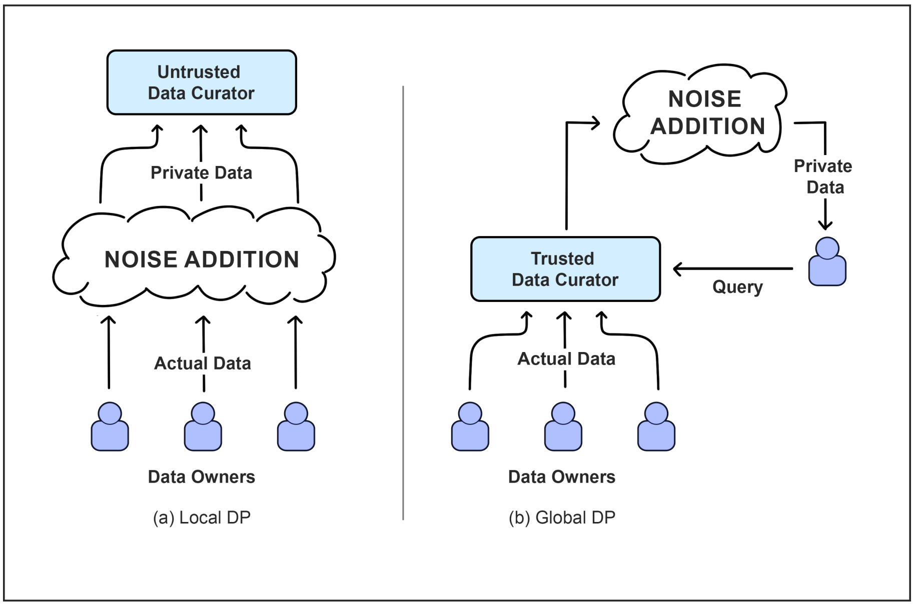


Figure 2. Local vs. global differential privacy.

4.2.1. Local Differential Privacy

In Local Differential Privacy (LDP), the data contributor is responsible for adding noise to their data before sharing them with a central aggregator and data collector, so it does not require any trusted party [41]. In LDP, noise is introduced to 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 a probability of $\frac{1+\epsilon}{2}$ to report b_i or with a probability of $\frac{1-\epsilon}{2}$ to report $1 - b_i$.

This ensures 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 trust in the data aggregator, as it is unaware of the real values. But problem is that every user will have to introduce noise in personal information, which will increase the total amount of added noise. But this problem can be mitigated by using high values of epsilon ϵ.

4.2.2. Global Differential Privacy

GDP generates noise to be added to the final results of a query by the central aggregator before sharing them with any third party [42]. In this model, each user shares their actual data with a central aggregator without adding noise. To add noise to the entire dataset, the central aggregator uses a differential privacy method. Global differential privacy makes sure that the presence or absence of an individual in the dataset does not alter the probability distribution in the final output. Consider a function (f) that calculates a sum over a dataset (O) defined as $f(O) = \sum_{i=1}^{n} x_i$. The Laplace mechanism adds noise taken from the Laplace distribution to achieve global differential privacy [43].

$$P(O) = f(O) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right) \tag{10}$$

As the central aggregator has access to the real dataset, requires trust in the 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, the trust of users in the data collector is required. If the data aggregator is compromised, the data can be leaked, increasing the risk of privacy failure.

4.3. Selection of Privacy Parameter ϵ

Setting the value for epsilon is a challenging task in effectively implementing differential privacy in any application [44]. The desirable balance between privacy and utility can be controlled by adjusting the value of epsilon (ε). Typical values ranging from 0.01 to 1 are used for strong privacy, but higher values might be used depending on the application or context.

Loss Function (L):

Consider a 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:

$$L(\phi, O) = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2 \tag{11}$$

where z_i represents the actual values and $\hat{z}_i$ represents the predicted values.

Privacy Loss (PL):

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

$$\text{PL}(\varepsilon) = \frac{K}{\varepsilon} \tag{12}$$

where K is a constant representing the baseline privacy risk when ϵ = 1.

Utility Measure (U):

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

$$U(\varphi, O) = \frac{1}{L(\varphi, O)} \tag{13}$$

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).

$$F(\varepsilon) = \alpha \cdot \mathrm{PL}(\varepsilon) + \beta \cdot L(\varphi, O) \tag{14}$$

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

Combined Cost Function:

Substituting $\mathrm{PL}(\varepsilon)$ and $L(\varphi_\varepsilon, O)$ into the cost function yields

$$F(\varepsilon) = \alpha \cdot \frac{K}{\varepsilon} + \beta \cdot L(\varphi_\varepsilon, O) \tag{15}$$

Selecting Optimal ϵ:

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

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

Solving for ϵ yields:

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

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

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

$$\varepsilon = \sqrt{\frac{\alpha \cdot K}{\beta \cdot \frac{\partial L(\varphi_\varepsilon, O)}{\partial \varepsilon}}} \tag{16}$$

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

By using this formula, one can select ε in such a way that balances both privacy (represented by α and K) and accuracy (represented by β and the sensitivity of the loss function). A lower value of ε provides stronger privacy guarantees. Selecting a lower value of ε requires increasing the value of α, which increases the emphasis on minimizing privacy loss, and decreasing the value of β, which reduces the emphasis on preserving 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.

5. Proposed Method for Privacy Preservation in Healthcare Data

This section describes the proposed method for practical implementation of DP in electronic healthcare data as demonstrated in Figure 3. In this model, global differential privacy is implemented on sensitive data to achieve privacy [45].

In this part of the framework, users or data analysts connect with the database through a user interface. The user requests the desired data in the form of queries and obtains differentially private results. The protected system receives queries made by data analysts or the involved user; then, it pulls 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 was selected as the programming language on the basis of the need to process large datasets within the minimum time period and its ability to deal with computational tasks. Moreover, to handle large datasets, the PyDP [46], Pandas, Numpy, and matplotlib libraries are used. PyDP is a differential privacy project from Google in which all computation methods use Laplace noise only.

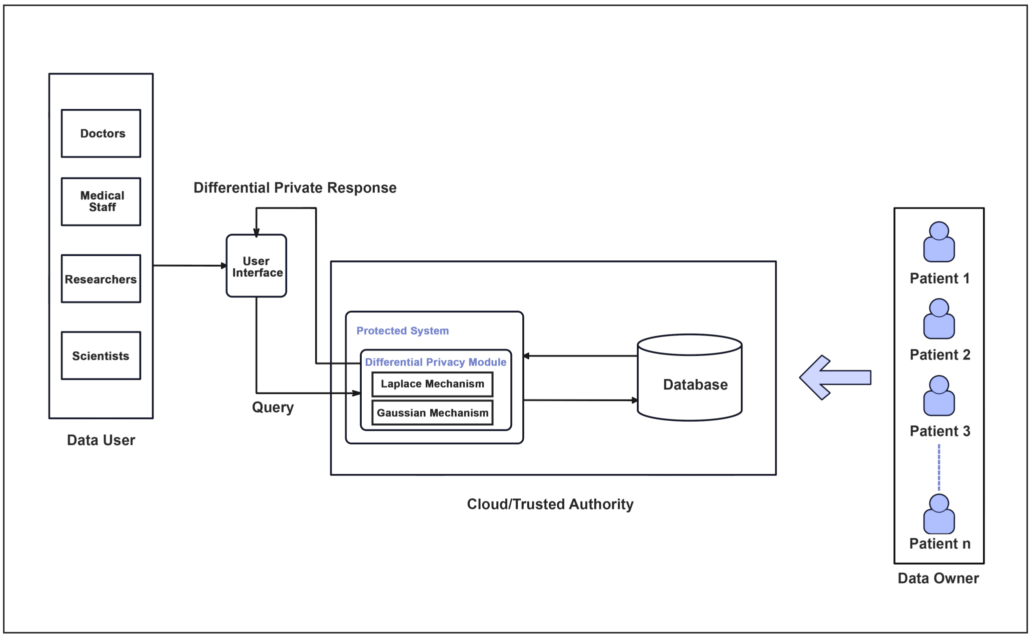


Figure 3. Architecture of the system.

5.1. System and Software Requirements

5.1.1. Programming Environment and Libraries

- Programming language: Python;
- Libraries used: PyDP (Version 1.1.1), Pandas (Version 1.4.2), NumPy (Version 1.22.4), and Matplotlib (Version 3.5.1).
- IDE: Jupyter Notebook

5.1.2. Device Specifications

- Processor: Core i5 8th Generation;
- RAM: 16 GB;
- System type: x64-based processor

5.2. Algorithm Details

Two different primary methods, namely Laplace and Gaussian mechanisms, are used for the purpose of introducing noise to implement differential privacy. As described below, both Algorithms 1 and 2 [32] were applied in this research fro the implementation of DP in healthcare data.

Algorithm 1: Laplace Mechanism Algorithm

```
Input: $O$, $Q$, $\epsilon$
Output: Noised output
$\Delta Q \leftarrow \mathrm{GS}(Q)$;
noise $\leftarrow [0] \times k$;
for $a$ in $range(k)$ do
    noise[a] $\leftarrow \mathrm{Lap}(\Delta Q/\epsilon)$;
return $Q(O) + noise[a]$;
```

Algorithm 2: Gaussian Mechanism Algorithm

```
Input: $D$, $Q$, $\epsilon$, $\delta$, lower_limit, upper_limit
Output: Noisy_count
Function GAUSSIAN_MECHANISM($D$, $Q$, $\epsilon$, $\delta$, lower_limit, upper_limit):
    filtered_data ← filter($D$, lower_limit, upper_limit)
    actual_count ← count(filtered_data)
    $\sigma \leftarrow \sqrt{2 \cdot \ln\left(\frac{1.25}{\delta}\right)}/\epsilon$
    noise ← sample_normal(0, $\sigma$)
    noisy_count ← actual_count + noise
return Noisy_count
```

5.3. Dataset Description

In order to implement differential privacy in healthcare data, we performed experiments on two different healthcare datasets. Both datasets are publicly available and further described below.

5.3.1. Breast Cancer Prediction Dataset

The Breast Cancer Prediction Dataset used in this research is publicly available on Kaggle [47]. The dataset contains information from 20,000 digital and 20,000 film-screen mammograms collected from women in the age range of 60–89 years for breast cancer prediction. It has almost 30,000 instances (patient records) with 13 attributes.

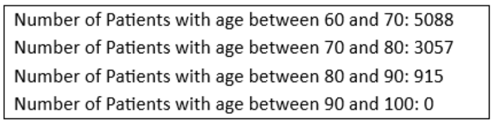
5.3.2. COVID-19 Home Nursing Dataset

Another dataset, the "COVID-19 Home Nursing Data", was used to perform the experiment by applying differential privacy in electronic healthcare data. This dataset is also publicly available on the data.cms.gov and Kaggle [48] websites. It consists of around 510,000 records with 39 attributes.

5.4. Experimental Results on the Breast Cancer Prediction Dataset

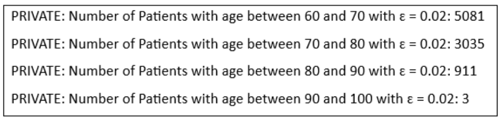
In this implementation, we compared the difference between the actual count and differential private outcome. First, we imported a CSV file of the Breast Cancer Prediction Dataset into IPython Jupyter Notebook Version 6.4.8 (It is developed by the Project Jupyter team, the project originated at Stanford University in the United States). After this, we performed multiple queries on the data to extract the counts of patients in different age groups during mammography with true values for history of breast biopsy. Figure 4 shows the actual counts of patients in different age groups without applying differential privacy.

Figure 5 shows the counts of patients in different age groups during mammography with true values of history of breast biopsy after applying differential privacy. Differential privacy was implemented through PyDP using the Laplace mechanism. The experiment was performed by selecting different values for epsilon; here, the selected value for epsilon is $\varepsilon = 0.2$.



```
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 4. Actual counts without DP.



```
PRIVATE: Number of Patients with age between 60 and 70 with ε = 0.02: 5081
PRIVATE: Number of Patients with age between 70 and 80 with ε = 0.02: 3035
PRIVATE: Number of Patients with age between 80 and 90 with ε = 0.02: 911
PRIVATE: Number of Patients with age between 90 and 100 with ε = 0.02: 3
```

Figure 5. Counts with DP.

Table 3 shows a comparison between actual results and differentially private results. It can be seen that noise was introduced in the actual count to make data private while maintaining the data's utility and accuracy. Therefore, these data can be used by data analysts for research purposes.

Table 3. Comparison of actual results and DP results.

No. of Patients	Actual Results	DP Results	Bias
Patients between age 60 and 70	5088	5081	−7
Patients between age 70 and 80	3057	3035	−22
Patients between age 80 and 90	915	911	−4
Patients between age 90 and 100	0	3	3

Figure 6 displays a comparison between true values and differentially private values by setting ϵ = 0.2. The Y axis shows the counts of patients in different age groups during mammography with true value of history of breast biopsy, and the X axis shows patient age groups. To provide a more intuitive idea of the precision of our estimations, we computed the ratios between the actual and estimated values. For instance, for the 60–70 patient age group, for which the actual value is 5088 and the estimated value is 5081, the ratio can be calculated as follows:

$$\text{Ratio} = \frac{\text{Actual Value}}{\text{Estimated Value}} = \frac{5088}{5081} \approx 0.999$$

This shows that our estimate has a high degree of accuracy and is relatively close to the actual values. The ratio of the 60–70 age group of approximately 0.999 shows that the estimated value (5081) is nearly identical to the actual value (5088), suggesting that our estimation method performs exceptionally well for this group. With a ratio of the 70–80 age group of about 0.992, the estimate (3035) is slightly lower than the actual value (3057), indicating a minor discrepancy. This still reflects a strong performance, and the ratio of the 80–90 group of approximately 0.996, again, indicates high accuracy, with the estimated value (911) being very close to the actual value (915). Overall, the high ratios across the majority of age groups show that our estimation method is robust and strong, effectively capturing actual values with a small margin of error. The bias is calculated as the difference between the actual value and the estimated value for each age group. A positive bias indicates overestimation, while a negative bias indicates underestimation.

The mean squared error can be calculated using the following formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Sum of Squared Errors:

$$\text{Sum of Squared Errors} = 49 + 484 + 16 + 9 = 558$$

MSE Calculation**:**

$$\text{MSE} = \frac{558}{4} = 139.5$$

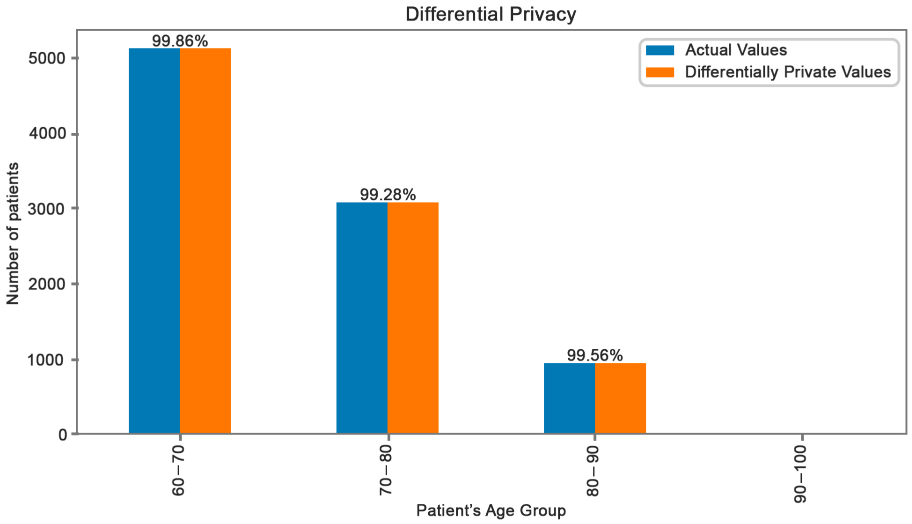
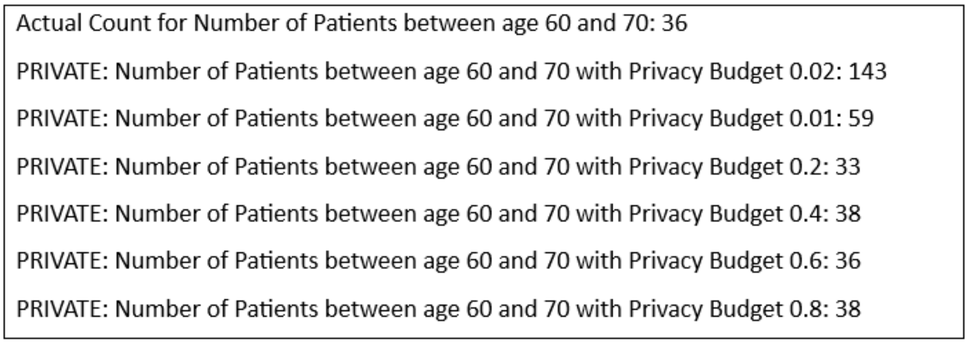


Figure 6. Comparison of results using the Breast Cancer Prediction Dataset.

5.4.1. Varying Privacy Budget Using Breast Cancer Prediction Dataset

Experiments were performed with different values of epsilon to examine the protection level provided by the DP mechanism with identical attributes but setting different values for privacy parameter. Results were evaluated by selecting different values for epsilon (0.02, 0.01, 0.2, 0.4, 0.6, and 0.8). It can be seen in Figure 7 that decreasing ε value added more noise and vice versa.



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

Figure 7. Varying epsilon values on the Breast Cancer Prediction Dataset.

For further demonstration, a graph was plotted between the privacy parameter (epsilon) and the results of queries to compare the exact results and data with introduced noise. In the context of differential privacy, epsilon ϵ is a privacy parameter that balances the trade-off between accuracy and privacy. The amount of noise introduced to the query results grows significantly as the ϵ value decreases toward zero. This improves privacy but may result in less accurate results (higher count variance). In Figure 8, it can be seen that the actual count for the number of patients between ages 60 and 70 at the time of mammography with true values of history of breast biopsy is 36. Data points such as (ϵ = 0.02, count = 143) demonstrate that a smaller epsilon yields a higher count with increased noise. After decreasing the value of the privacy parameter (epsilon) by applying DP, more noise is added to the actual data. In contrast, the quantity of noise introduced into the results decreases as ϵ increases, enabling more precise query outputs (e.g., the points (ϵ = 0.40, count = 38) and (ϵ = 0.60, count = 36) demonstrate that the count values get less noisy as ϵ increases). This line chart illustrates the relationship whereby higher ϵ values are associated with less distortion in the query results but at cost of less privacy.

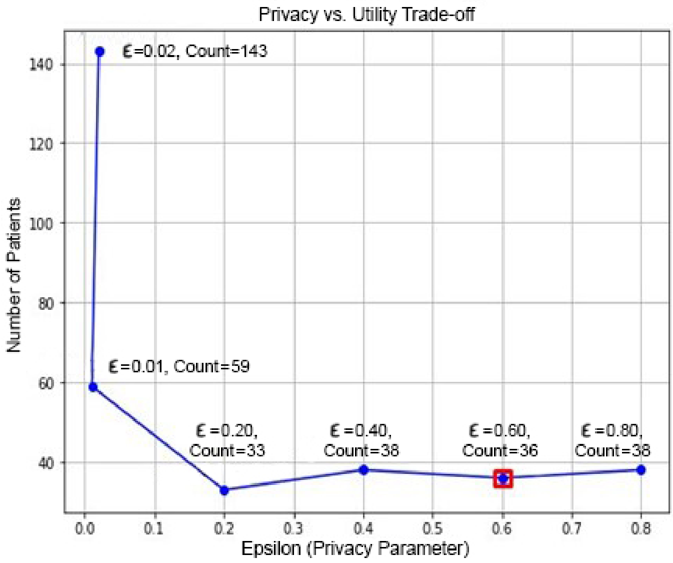
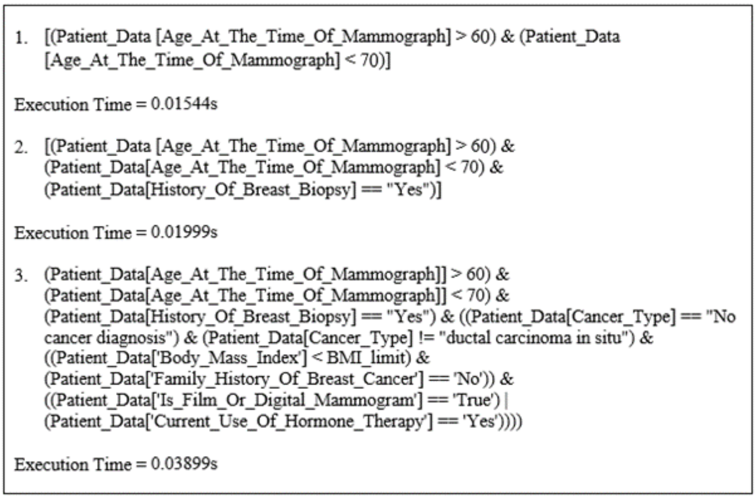


Figure 8. Analysis of privacy parameter using the Breast Cancer Prediction Dataset.

5.4.2. Time Complexity Analysis with the Breast Cancer Prediction Dataset

The execution time of queries somewhat increases with an increase in the conditions in the query, as seen in Figure 9. The first query filters the data to include only patients whose age at the time of the mammogramwas between 60 and 70 years. It involves a simple range filter on one attribute. The 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. The next query is the most complex, combining multiple conditions across several attributes, including logical operations and comparisons. The time increases from 0.01544 s to 0.01999 s, then to 0.03899 s with increasing conditions. A slight rise in execution time observed with each additional attribute, but it is not drastic, suggesting that the filtering operations scale reasonably well with the application of differential privacy and an increase in the number of conditions. The time complexity in practice suggests that the operations are manageable within the given execution times.

Large-scale healthcare systems, which frequently handle massive volumes of data like patient records and clinical data, can benefit from the use of hierarchical models, which arrange data in layers. Hospitals, for instance, might gather data locally, then aggregate them throughout the region and apply DP to maintain patient privacy. DP algorithms are designed to scale with the dataset. For example, methods such as the Gaussian or Laplace mechanisms introduce controlled noise proportionate to the sensitivity of queries and the quantity of the dataset. This guarantees that personal privacy remains protected, even as data volumes increase. While DP is effective in maintaining privacy, it can introduce latency due to the additional noise that needs to be calculated and added during data processing. Systems can reduce the time taken to respond to user requests by pre-computing commonly requested queries. For instance, if certain aggregate statistics that are frequently needed, they can be computed and stored with the necessary DP adjustments made in advance. Similarly, response times can be greatly decreased by putting in place a caching mechanism for computed results. The system can quickly respond to repeated queries without having to perform all calculations by storing previously computed results with their associated noise.

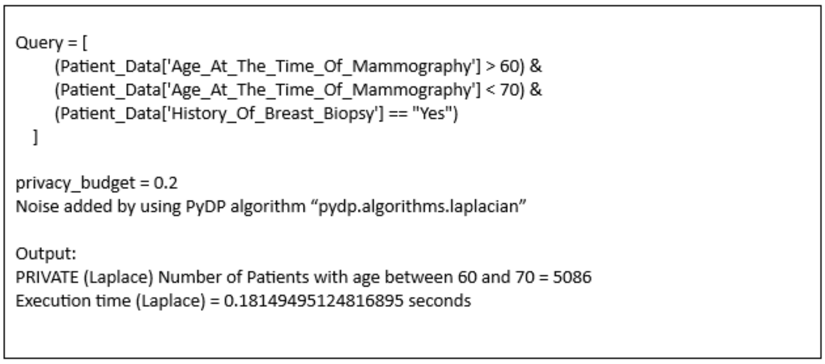


```
1.  [(Patient_Data [Age_At_The_Time_Of_Mammograph] > 60) & (Patient_Data
    [Age_At_The_Time_Of_Mammograph] < 70)]

Execution Time = 0.01544s

2.  [(Patient_Data [Age_At_The_Time_Of_Mammograph] > 60) &
    (Patient_Data[Age_At_The_Time_Of_Mammograph] < 70) &
    (Patient_Data[History_Of_Breast_Biopsy] == "Yes")]

Execution Time = 0.01999s

3.  (Patient_Data[Age_At_The_Time_Of_Mammograph]] > 60) &
    (Patient_Data[Age_At_The_Time_Of_Mammograph]] < 70) &
    (Patient_Data[History_Of_Breast_Biopsy] == "Yes") & ((Patient_Data[Cancer_Type] == "No
    cancer diagnosis") & (Patient_Data[Cancer_Type] != "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'))))

Execution Time = 0.03899s
```

Figure 9. Queries with time comparison using the Breast Cancer Prediction Dataset.

5.4.3. Comparison Analysis of Laplace vs. Gaussian Mechanism

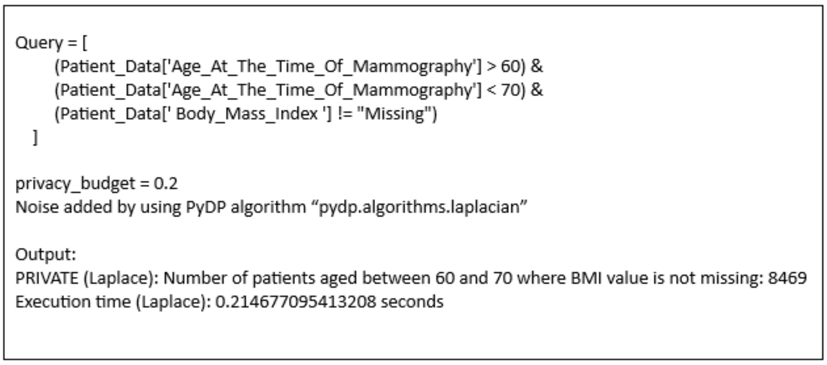
The implementation of the Laplace mechanism using PYDP adds noise sampled from the Laplace distribution based on the privacy budget to provide results with provable privacy guarantees under differential privacy as shown in Figure 10. To generate noise, the Gaussian mechanism uses a Gaussian distribution based on the privacy budget, which also provides differentially private results but typically is used for scenarios where smoothness and sensitivity are key considerations. The Laplace mechanism [49] is generally efficient due to the simplicity of sampling from a Laplace distribution. The 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, ease of implementation, and computational complexity. The choice of between them depends on the particular needs and conditions of the differential privacy application and the nature of queries being performed on the datasets. A comparison between the two mechanisms is presented the figures below.



```
Query = [
      (Patient_Data['Age_At_The_Time_Of_Mammography'] > 60) &
      (Patient_Data['Age_At_The_Time_Of_Mammography'] < 70) &
      (Patient_Data['History_Of_Breast_Biopsy'] == "Yes")
  ]

privacy_budget = 0.2
Noise added by using PyDP algorithm "pydp.algorithms.laplacian"

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

Figure 10. Laplace mechanism.

The Laplace mechanism is ideal for situations with bounded sensitivity. The Laplace mechanism provides noise proportionate to the function's sensitivity; hence, it might be useful when the query has a known, finite sensitivity (such as counting queries). In a healthcare dataset, using the Laplace mechanism can effectively mask the exact count while ensuring privacy; the query returns the number of patients within a specific condition as shown in Figure 11. For queries with low sensitivity, the Laplace method can offer more precise privacy guarantees, as the noise can be adjusted according to the sensitivity, resulting in less distortion in the output.

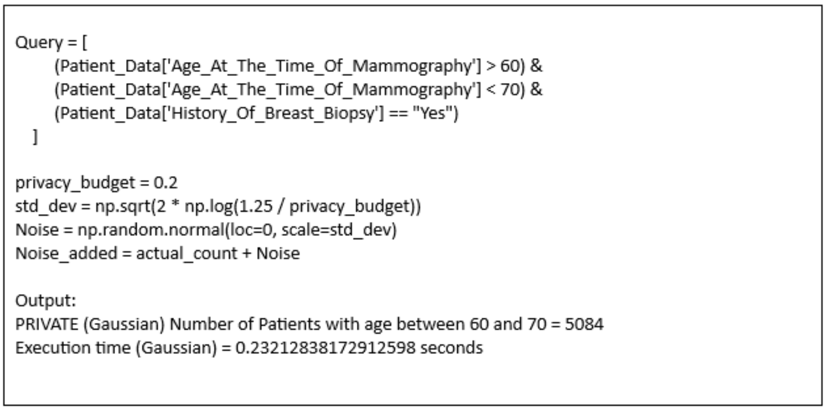


```
Query = [
      (Patient_Data['Age_At_The_Time_Of_Mammography'] > 60) &
      (Patient_Data['Age_At_The_Time_Of_Mammography'] < 70) &
      (Patient_Data[' Body_Mass_Index '] != "Missing")
  ]

privacy_budget = 0.2
Noise added by using PyDP algorithm "pydp.algorithms.laplacian"

Output:
PRIVATE (Laplace): Number of patients aged between 60 and 70 where BMI value is not missing: 8469
Execution time (Laplace): 0.214677095413208 seconds
```

Figure 11. Laplace mechanism.

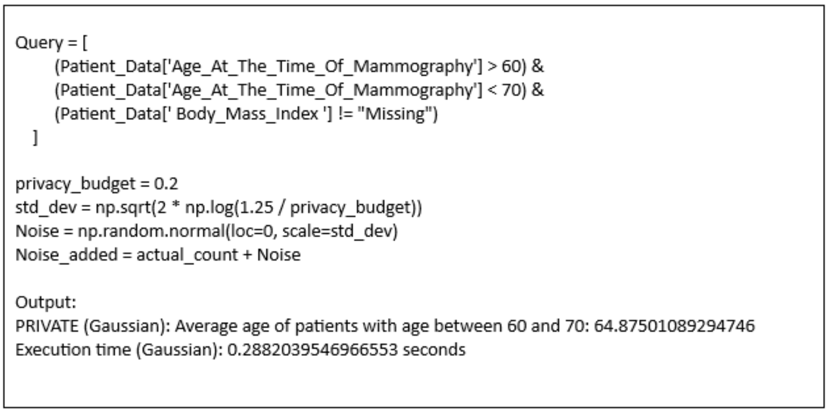
The Gaussian mechanism well-suited for scenarios with unbounded sensitivity or when dealing with aggregate statistics over large datasets. For queries that involve averages or continuous outcomes, the Gaussian mechanism might be preferable, as it can handle the potential for large deviations more gracefully. The Gaussian mechanism can manage higher sensitivity with less effort and does not require significant noise levels like the Laplace mechanism; therefore, it is advantageous for guaranteeing privacy with higher-dimensional queries.

It can be noticed that the Gaussian mechanism [50] is slightly more computationally intensive due to the nature of sampling from a Gaussian distribution, which involves more complex calculations. By performing time complexity analysis, it can be seen in Figure 12 and Figure 13 that the Gaussian mechanism takes slightly more time in processing queries with differential privacy.



```
Query = [
      (Patient_Data['Age_At_The_Time_Of_Mammography'] > 60) &
      (Patient_Data['Age_At_The_Time_Of_Mammography'] < 70) &
      (Patient_Data['History_Of_Breast_Biopsy'] == "Yes")
  ]

privacy_budget = 0.2
std_dev = np.sqrt(2 * np.log(1.25 / privacy_budget))
Noise = np.random.normal(loc=0, scale=std_dev)
Noise_added = actual_count + Noise

Output:
PRIVATE (Gaussian) Number of Patients with age between 60 and 70 = 5084
Execution time (Gaussian) = 0.23212838172912598 seconds
```

Figure 12. Gaussian mechanism.



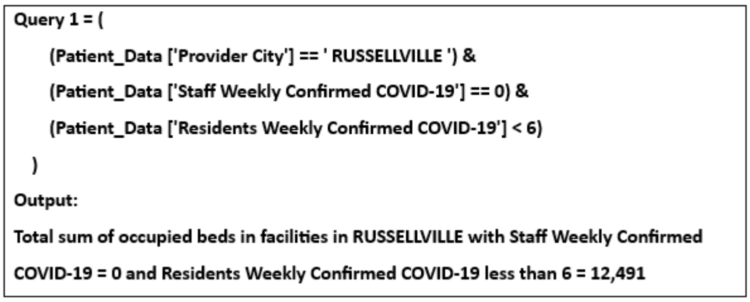
```
Query = [
      (Patient_Data['Age_At_The_Time_Of_Mammography'] > 60) &
      (Patient_Data['Age_At_The_Time_Of_Mammography'] < 70) &
      (Patient_Data[' Body_Mass_Index '] != "Missing")
  ]

privacy_budget = 0.2
std_dev = np.sqrt(2 * np.log(1.25 / privacy_budget))
Noise = np.random.normal(loc=0, scale=std_dev)
Noise_added = actual_count + Noise

Output:
PRIVATE (Gaussian): Average age of patients with age between 60 and 70: 64.87501089294746
Execution time (Gaussian): 0.2882039546966553 seconds
```

Figure 13. Gaussian mechanism.

5.5. Experimental Results on the COVID-19 Home Nursing Dataset

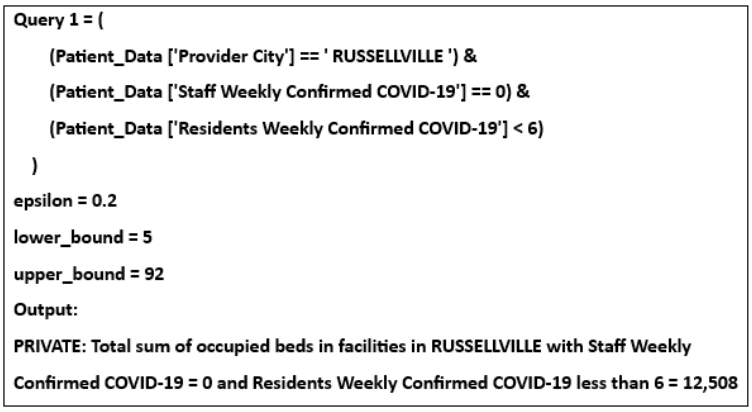
To implement differential privacy, we performed different queries on another dataset to compare real outcomes and differential private outcomes of queries. The first query shows the overall count of beds in use in facilities of the city of "RUSSELLVILLE", with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents. Figure 14 shows the actual count for this query without implementing differential privacy.

Figure 15 shows the overall actual count of beds in use in facilities of the city of "RUSSELLVILLE", with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents after the implementation of differential privacy. Here, differential privacy was implemented through PyDP using the Laplace mechanism with a selected epsilon value of $\epsilon = 0.2$.



```
Query 1 = (
     (Patient_Data ['Provider City'] == ' RUSSELLVILLE ') &
     (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
     (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
Output:
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 = 12,491
```

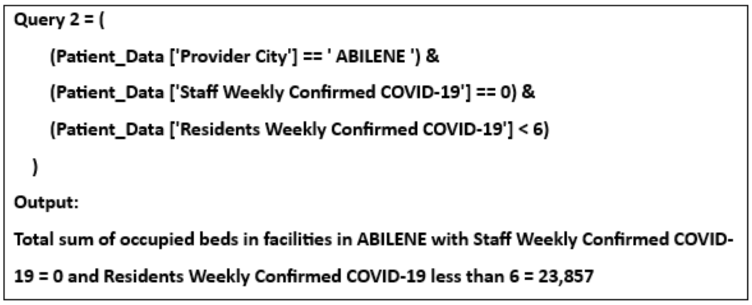
Figure 14. Query 1 result without DP.



```
Query 1 = (
     (Patient_Data ['Provider City'] == ' RUSSELLVILLE ') &
     (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
     (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
epsilon = 0.2
lower_bound = 5
upper_bound = 92
Output:
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 = 12,508
```

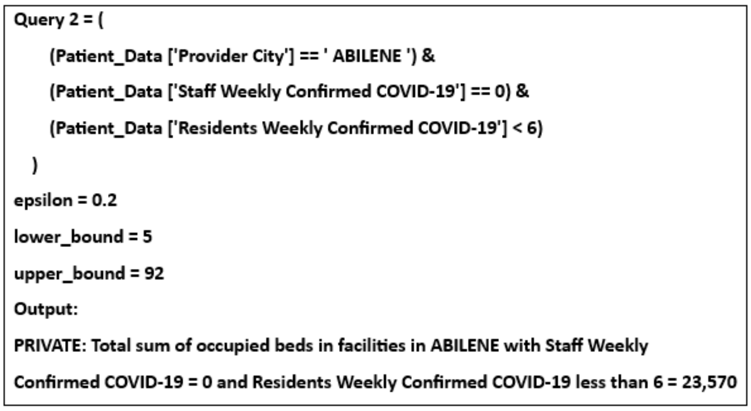
Figure 15. Query 1 result with DP.

The second query shown in Figures 16 and 17 shows the overall count of beds that in use in facilities of the city of "ABILENE", with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents both without and with implementing differential privacy.



```
Query 2 = (
     (Patient_Data ['Provider City'] == ' ABILENE ') &
     (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
     (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
Output:
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 = 23,857
```

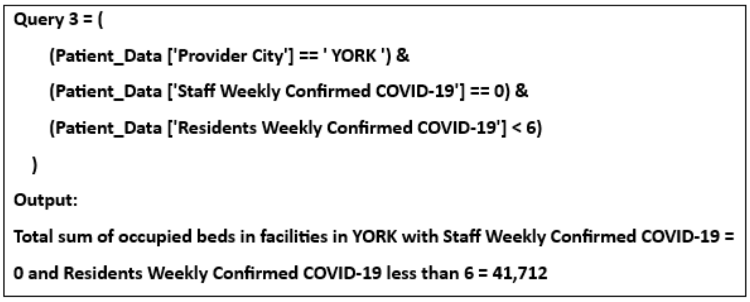
Figure 16. Query 2 rResult without DP.



```
Query 2 = (
      (Patient_Data ['Provider City'] == ' ABILENE ') &
      (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
      (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
epsilon = 0.2
lower_bound = 5
upper_bound = 92
Output:
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 = 23,570
```

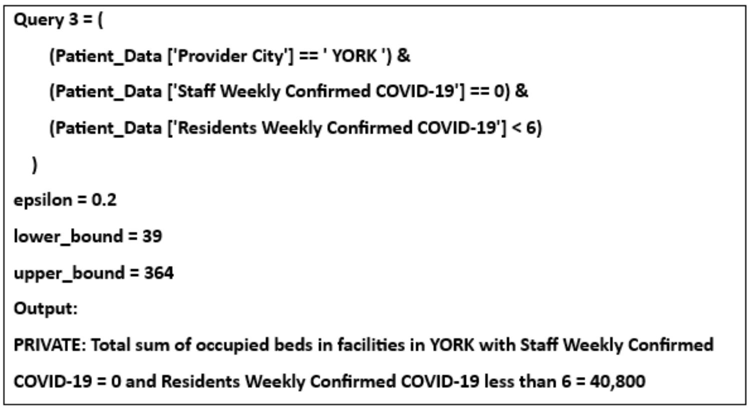
Figure 17. Query 2 result with DP.

The third query shown in Figures 18 and 19 represents the overall count of beds in use in facilities of the city of "YORK", with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents both with and without implementing differential privacy.



```
Query 3 = (
      (Patient_Data ['Provider City'] == ' YORK ') &
      (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
      (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
Output:
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 = 41,712
```

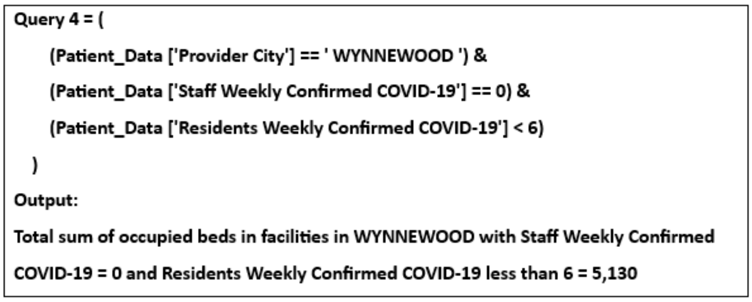
Figure 18. Query 3 result without DP.



```
Query 3 = (
      (Patient_Data ['Provider City'] == ' YORK ') &
      (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
      (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
epsilon = 0.2
lower_bound = 39
upper_bound = 364
Output:
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 = 40,800
```

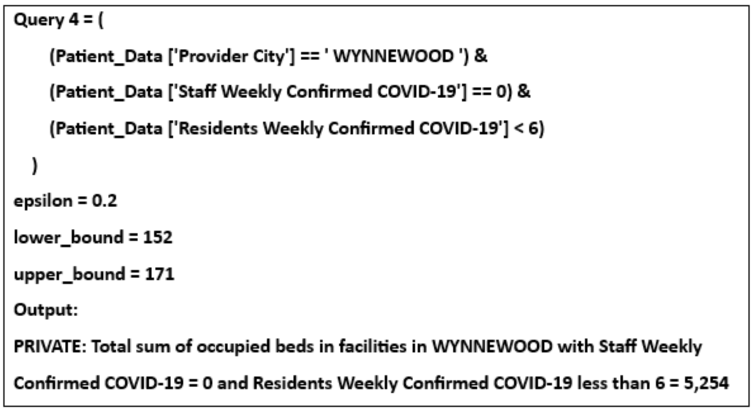
Figure 19. Query 3 result with DP.

The fourth query shown in Figures 20 and 21 shows the overall count of beds in use in facilities of the city of "WYNNEWOOD", with with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents both with and without differential privacy.



```
Query 4 = (
    (Patient_Data ['Provider City'] == ' WYNNEWOOD ') &
    (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
    (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
Output:
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 = 5,130
```

Figure 20. Query 4 result without DP.



```
Query 4 = (
    (Patient_Data ['Provider City'] == ' WYNNEWOOD ') &
    (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
    (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6)
  )
epsilon = 0.2
lower_bound = 152
upper_bound = 171
Output:
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 = 5,254
```

Figure 21. Query 4 result with DP.

Table 4 shows a comparison between actual results and differentially private results for multiple queries. It can be noticed that noise was added to the actual outcomes of queries while maintaining data utility and data accuracy.

Table 4. Comparison of results using the COVID-19 Home Nursing Dataset.

City	Overall Occupied Beds	Overall Occupied Beds with DP	Bias
RUSSELLVILLE	12,491	12,508	17
ABILENE	23,857	23,570	−287
YORK	41,712	40,800	−912
WYNNEWOOD	5130	5254	124

For comparison, we again plotted a graph comparing true values and differentially private values by setting $\epsilon = 0.2$, as shown in Figure 22. The Y axis shows the count for the number of occupied beds with zero weekly confirmed COVID-19 cases among staff and fewer than six weekly confirmed COVID-19 cases among residents in different cities, while the X axis represents the statistics for different cities. We calculated the ratios between the actual and estimated values to provide a more intuitive explanation for the accuracy of our estimations. A ratio greater than 100% (e.g., RUSSELLVILLE (100.14%) and WYNNEWOOD (102.42%)) means that the estimated value is marginally higher than the actual value. This implies a slight overestimation of the given data. A ratio less than 100% (e.g., ABILENE (98.80%) and YORK (97.84%)) indicates that the estimated value is lower than the actual value, reflecting an underestimation. Ratios near 100% (such as RUSSELLVILLE and ABILENE) indicate that the estimated values are reasonably accurate in comparison to the real values, demonstrating that the estimation process is effective in maintaining utility while adhering to privacy requirements.

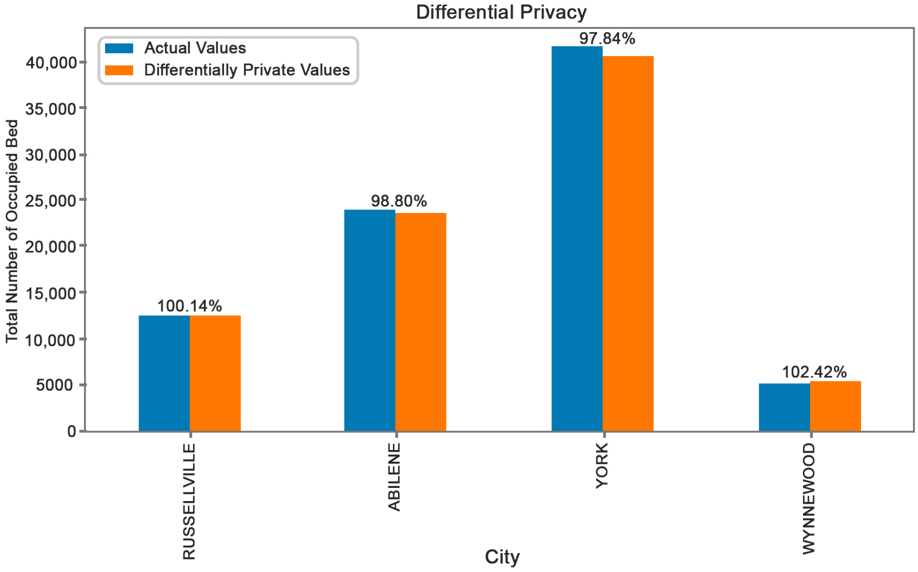
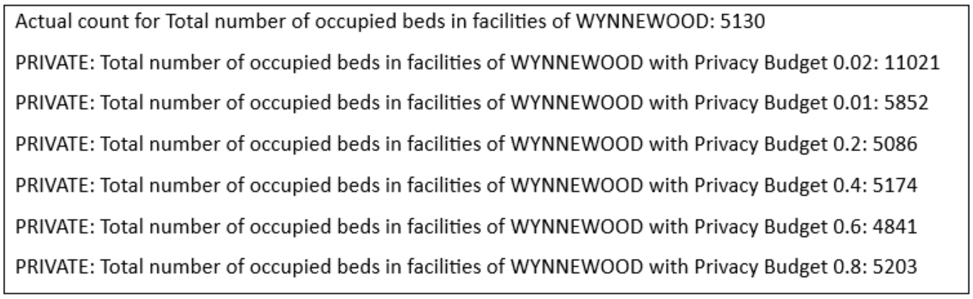


Figure 22. Comparison of results using the COVID-19 Home Nursing Dataset.

5.5.1. Varying Privacy Budget Using the COVID-19 Home Nursing Dataset

In order to examine how noise affects the same query, we ran an experiment where we varied the value of epsilon (0.8, 0.6, 0.4, 0.2, 0.01, and 0.02; Figure 23). In the given results, we can notice that with a decrease in the epsilon value, the amount of added noise also increases. Therefore, the smaller the value of epsilon, the greater the privacy required and the more noise is added. A compromise between privacy and utility exists. Adding more noise increases privacy but also reduces data utility. In differential privacy, the epsilon parameter (ϵ) is used to control this trade-off between privacy and accuracy.



```
Actual count for Total number of occupied beds in facilities of WYNNEWOOD: 5130
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.02: 11021
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.01: 5852
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.2: 5086
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.4: 5174
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.6: 4841
PRIVATE: Total number of occupied beds in facilities of WYNNEWOOD with Privacy Budget 0.8: 5203
```

Figure 23. Varying epsilon values in the Nursing Home COVID-19 Dataset.

For further demonstration, a graph was plotted for different values of epsilon (ϵ), as shown in Figure 24. The amount of noise introduced to the query results grows significantly as the ε value decreases toward zero. This enhances privacy but may result in less accurate results. Increasing the epsilon value results in less noise added to the output of query results, which typically provides higher accuracy and utility, as data remain closer to their true values. This helps data analysts to make informed decision to maintain the privacy level while also considering the usability and reliability of the data.

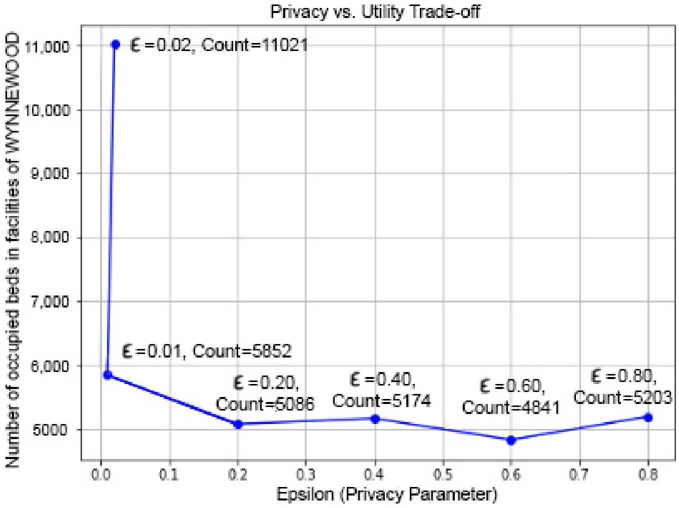
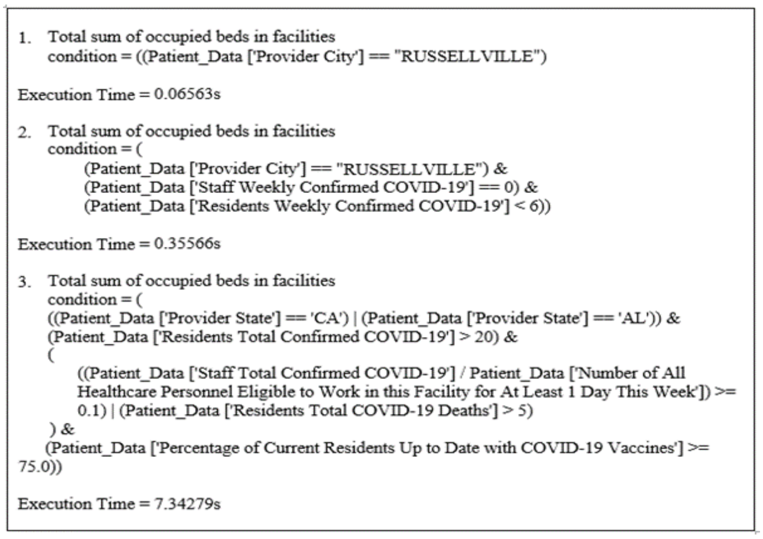


Figure 24. Analysis of the privacy parameter using the Nursing Home COVID-19 Dataset.

5.5.2. Time Complexity Analysis with the 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. The first query involves filtering the dataset based on a single condition, while the second and third queries involve complex filtering conditions including logical AND and OR operations across multiple columns. It can be noticed that increasing the number of conditions in queries also increases the execution time for queries, as shown in Figure 25. The time increases from 0.06563 s to 0.35566 s, then to 7.34279 s due to the increase in conditions. A slight 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 the previous "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, the "Nursing Home COVID-19 Dataset", with around 510,000 records and 39 attributes, execution times are slightly higher due to the sheer volume of data being processed.



```
1.  Total sum of occupied beds in facilities
    condition = ((Patient_Data ['Provider City'] == "RUSSELLVILLE")

Execution Time = 0.06563s

2.  Total sum of occupied beds in facilities
    condition = (
          (Patient_Data ['Provider City'] == "RUSSELLVILLE") &
          (Patient_Data ['Staff Weekly Confirmed COVID-19'] == 0) &
          (Patient_Data ['Residents Weekly Confirmed COVID-19'] < 6))

Execution Time = 0.35566s

3.  Total sum of occupied beds in facilities
    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 in this Facility for At Least 1 Day This Week']) >=
         0.1) | (Patient_Data ['Residents Total COVID-19 Deaths'] > 5)
     ) &
    (Patient_Data ['Percentage of Current Residents Up to Date with COVID-19 Vaccines'] >=
75.0))

Execution Time = 7.34279s
```

Figure 25. Queries with time comparison using the Nursing Home COVID-19 Dataset.

5.6. Limitations of Implementing Differential Privacy

- By implementing DP in real-time healthcare systems, computational delays can arises due to the need to add noise and adjust parameters dynamically. This can be crucial in scenarios where immediate data analysis is required, such as in any emergency situation.
- Continuous streams of data are produced by dynamic systems (e.g., from monitoring sensors). In this situation, maintaining differential privacy necessitates frequent modifications to privacy guarantees, which can be difficult and resource-intensive.
- The parameters for differential privacy must be continuously adjusted due to the continuous changes in patient data. This continual adjustment can make the system architecture more difficult and raise the possibility of inconsistent privacy protections.
- The requirement for real-time privacy guarantees in EHRs may encourage the addition of excessive noise, which could reduce the usefulness or accuracy of the data. This distortion in data can negatively impact clinical insights and outcomes.
- Ensuring that DP approaches scale well without appreciable performance deterioration can be challenging in dynamic systems with growing data volumes. When real-time analysis is required, this becomes especially difficult.
- Systems for providing real-time healthcare frequently face severe resource limitations. Maintaining DP comes with a computational cost that can strain system resources, potentially impacting other critical functions.

6. Conclusions and Future Work

This study proposed a differential privacy-based method for protecting healthcare data on the Internet of Medical Things. Initially, this study examined conventional approaches that were employed in the electronic healthcare data privacy process prior to the application of differential privacy. Then, we performed an 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 the privacy parameter, i.e. the privacy budget, were analysed to illustrate their impact on the preservation of privacy while maintaining data utility. Comparative analyses involving Laplace and Gaussian mechanisms were conducted by analysing both schemes in terms of their ability to meet privacy and security requirements with minimal computational overhead. Furthermore, we carried out a thorough examination of time complexity through the application of differential privacy to complex queries on datasets of various sizes.

Even the DP mechanism is sufficiently effective in providing the necessary data privacy protection, but there exist still some limitations that need to be addressed to maintain privacy. In adversarial scenarios, attackers can attempt to infer sensitive information by analysing the outputs of multiple queries. Even with differential privacy, if they have prior knowledge or can infer relationships within the data, they may extract sensitive insights. To prevent such query inference attacks, robust logging and auditing mechanisms can be implemented for queries. This can help identify suspicious patterns and flag potentially adversarial queries. Restrictions can be put on how many queries a user can send in a certain amount of time to reduce the possibility of inference attacks caused by excessive querying. When complex queries are executed under differential privacy, SQL injection attacks can compromise the privacy mechanism. Attackers can modify the behaviour of queries by injecting SQL code to access raw data rather than the differentially private output, effectively bypassing privacy protections. Strict input validation and sanitization should be used to mitigate this by ensuring that only valid data types and formats are accepted. However, differential privacy is not always adaptable enough to use in every real-world situation, which 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 in the future. Applying data-dependent differential privacy to real-world datasets, where databases comprising tuple correlations that signify relations between different tables in the database, may expose limitations in the underlying assumptions of this privacy model. In such cases, inference attacks may exist under the differential privacy mechanism. Thus, future research should take this into account to create a better mechanism that enhances the current approach.

Author Contributions: conceptualization, B.M., F.K. and S.T.; methodology, B.M., F.K., S.I.S. and Q.H.A.; investigation, B.M., F.K., S.I.S. and Q.H.A.; supervision, F.K., S.T. and S.A.S.; funding acquisition, S.A.S.; writing—original draft, preparation, B.M.; writing—review and editing, B.M., F.K., S.T., S.A.S., S.I.S. and Q.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported, in part, by the Engineering and Physical Sciences Research Council (EPSRC) (grant # EP/W037076/1).

Data Availability Statement: Data are available upon request to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chenthara, S.; Ahmed, K.; Wang, H.; Whittaker, F. Security and Privacy-Preserving Challenges of E-Health Solutions in Cloud Computing. *IEEE Access* **2019**, *7*, 74361–74382. [CrossRef]
2. Nelson, G.S. Practical implications of sharing data: A primer on data privacy, anonymization, and de-identification. In *Proceedings of the SAS Global Forum*; SAS Institute Inc.: Cary, NC, USA, 2015; pp. 1–23.
3. Dwork, C.; Kohli, N.; Mulligan, D. Differential Privacy in Practice: Expose Your Epsilons! *J. Priv. Confidentiality* **2019**, *9*, 5–8. [CrossRef]
4. Inan, A.; Gursoy, M.E.; Saygin, Y. Sensitivity Analysis for Non-Interactive Differential Privacy: Bounds and Efficient Algorithms. *IEEE Trans. Dependable Secur. Comput.* **2017**, *17*, 194–207. [CrossRef]
5. Zhang, M.; Chen, Y.; Susilo, W. PPO-CPQ: A Privacy-Preserving Optimization of Clinical Pathway Query for E-Healthcare Systems. *IEEE Internet Things J.* **2020**, *7*, 10660–10672. [CrossRef]
6. Majeed, A.; Khan, S.; Hwang, S.O. Toward Privacy Preservation Using Clustering Based Anonymization: Recent Advances and Future Research Outlook. *IEEE Access* **2022**, *10*, 53066–53097. [CrossRef]
7. Zhu, T.; Li, G.; Zhou, W.; Yu, P. *Differential Privacy and Applications*; Springer: Berlin/Heidelberg, Germany, 2017. [CrossRef]
8. Kar, T.S. A Study on Privacy Preserving Data Publishing with Differential Privacy. Ph.D. Thesis, University of Saskatchewan, Saskatoon, SK, Canada, 2017.
9. Mir, D.J. Differential Privacy: An Exploration of the Privacy-Utility Landscape. Available online: https://rucore.libraries.rutgers.edu/rutgers-lib/41872/ (accessed on 21 July 2024).
10. Kaaniche, N.; Laurent, M. Data Security and Privacy Preservation in Cloud Storage Environments Based on Cryptographic Mechanisms. *Comput. Commun.* **2017**, *111*, 120–141. [CrossRef]
11. Al-Zobbi, M.; Shahrestani, S.; Ruan, C. A Multidimensional Sensitivity-Based Anonymization Method of Big Data. In *Networks of the Future*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017; pp. 415–430.
12. Ram Mohan Rao, P.; Murali Krishna, S.; Siva Kumar, A.P. Privacy Preservation Techniques in Big Data Analytics: A Survey. *J. Big Data* **2018**, *5*, 33. [CrossRef]
13. Kumar, B.; Shukla, P.; Mohan, K.; Bharadwaj, A.; Shivam, Y.; Kumar, C. Medical Dataset Preparation and Privacy Preservation for Improving the Healthcare Facilities Using Federated Learning Approach. In Proceedings of the 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM), Roorkee, India, 26–27 August 2023; IEEE: Piscataway, NJ, USA, 2023.
14. Joshi, A.; Gautam, P. An Implementation of Hybrid Method Towards the Privacy of HealthCare Record. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; IEEE: Piscataway, NJ, USA, 2020.
15. Suneetha, V.; Suresh, S.; Jhananie, V. A Novel Framework Using Apache Spark for Privacy Preservation of Healthcare Big Data. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; IEEE: Piscataway, NJ, USA, 2020.
16. Zhang, L.; Xu, J.; Vijayakumar, P.; Sharma, P.K.; Ghosh, U. Homomorphic Encryption-Based Privacy-Preserving Federated Learning in IoT-Enabled Healthcare System. *IEEE Trans. Netw. Sci. Eng.* **2022**, *10*, 2864–2880. [CrossRef]
17. Seol, K.; Kim, Y.G.; Lee, E.; Seo, Y.D.; Baik, D.K. Privacy-Preserving Attribute-Based Access Control Model for XML-Based Electronic Health Record System. *IEEE Access* **2018**, *6*, 9114–9128. [CrossRef]
18. Al Omar, A.; Rahman, M.S.; Basu, A.; Kiyomoto, S. Medibchain: A Blockchain Based Privacy Preserving Platform for Healthcare Data. In Proceedings of the Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2017 International Workshops, Guangzhou, China, 12–15 December 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017.

19. Aminifar, A.; Shokri, M.; Rabbi, F.; Pun, V.K.I.; Lamo, Y. Extremely Randomized Trees with Privacy Preservation for Distributed Structured Health Data. *IEEE Access* **2022**, *10*, 6010–6027. [CrossRef]
20. Charles, V.B.; Surendran, D.; SureshKumar, A. Heart Disease Data Based Privacy Preservation Using Enhanced ElGamal and ResNet Classifier. *Biomed. Signal Process. Control.* **2022**, *71*, 103185. [CrossRef]
21. Wang, K.; Chen, C.-M.; Tie, Z.; Shojafar, M.; Kumar, S.; Kumari, S. Forward Privacy Preservation in IoT-Enabled Healthcare Systems. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1991–1999. [CrossRef]
22. Ahmed, J.; Nguyen, T.N.; Ali, B.; Javed, M.A.; Mirza, J. On the Physical Layer Security of Federated Learning Based IoMT Networks. *IEEE J. Biomed. Health Inform.* **2022**, *27*, 691–697. [CrossRef] [PubMed]
23. Singh, P.; Gaba, G.S.; Kaur, A.; Hedabou, M.; Gurtov, A. Dew-Cloud-Based Hierarchical Federated Learning for Intrusion Detection in IoMT. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 722–731. [CrossRef] [PubMed]
24. Shabbir, M.; Shabbir, A.; Iwendi, C.; Javed, A.R.; Rizwan, M.; Herencsar, N.; Lin, J.C.W. Enhancing Security of Health Information Using Modular Encryption Standard in Mobile Cloud Computing. *IEEE Access* **2021**, *9*, 8820–8834. [CrossRef]
25. Krall, A.; Finke, D.; Yang, H. Mosaic Privacy-Preserving Mechanisms for Healthcare Analytics. *IEEE J. Biomed. Health Inform.* **2020** *25*, 2184–2192. [CrossRef]
26. Xu, C.; Wang, N.; Zhu, L.; Sharif, K.; Zhang, C. Achieving Searchable and Privacy-Preserving Data Sharing for Cloud-Assisted E-Healthcare System. *IEEE Internet Things J.* **2019**, *6*, 8345–8356. [CrossRef]
27. Onesimu, J.A.; Karthikeyan, J.; Eunice, J.; Pomplun, M.; Dang, H. Privacy Preserving Attribute-Focused Anonymization Scheme for Healthcare Data Publishing. *IEEE Access* **2022**, *10*, 86979–86997. [CrossRef]
28. Zala, K.; Thakkar, H.K.; Jadeja, R.; Singh, P.; Kotecha, K.; Shukla, M. PRMS: Design and Development of Patients' E-Healthcare Records Management System for Privacy Preservation in Third Party Cloud Platforms. *IEEE Access* **2022**, *10*, 85777–85791. [CrossRef]
29. Wu, C.; Cao, H.; Xu, G.; Zhou, C.; Sun, J.; Yan, R.; Liu, Y.; Jiang, H. It's All in the Touch: Authenticating Users with HOST Gestures on Multi-Touch Screen Devices. *IEEE Trans. Mob. Comput.* **2024**, *23*, 10016–10030. [CrossRef]
30. Dwork, C. Differential Privacy: A Survey of Results. In *International Conference on Theory and Applications of Models of Computation*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–19.
31. Sun, Z.; Wang, Y.; Shu, M.; Liu, R.; Zhao, H. Differential Privacy for Data and Model Publishing of Medical Data. *IEEE Access* **2019**, *7*, 152103–152114. [CrossRef]
32. Asseffa, S.; Seleshi, B. A Case Study on Differential Privacy. Master's Thesis, Department of Computer Science, Umeå University, Umeå, Sweden, 2017.
33. Phan, N.H.; Wu, X.; Hu, H.; Dou, D. Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; IEEE: Piscataway, NJ, USA, 2017.
34. Garfinkel, S.L.; Abowd, J.M.; Powazek, S. Issues Encountered Deploying Differential Privacy. In Proceedings of the 2018 Workshop on Privacy in the Electronic Society, Toronto, ON, Canada, 15 October 2018; ACM: New York, NY, USA, 2018.
35. Zhu, T. Differential Privacy and Its Application. Ph.D. Thesis, Deakin University, Victoria, Australia, 2014.
36. Dandekar, A.; Basu, D.; Bressan, S. Differential Privacy at Risk: Bridging Randomness and Privacy Budget. *arXiv* **2020**, arXiv:2003.00973. [CrossRef]
37. Nguyen, T.T. Differential Privacy for Survival Analysis and User Data Collection. Ph.D. Thesis, Nanyang Technological University, Singapore, 2019.
38. Thissen, K.K.K.; Schoenmakers, I.L.; Koster, I.R.; van Liesdonk, I.P. Achieving Differential Privacy in Secure Multiparty Computation. Master's Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2019.
39. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07), Providence, RI, USA, 21–23 October 2007; IEEE Computer Society: Washington, DC, USA, 2007; pp. 94–103. ISBN 0-7695-3010-9.
40. Dong, J.; Durfee, D.; Rogers, R. Optimal Differential Privacy Composition for Exponential Mechanisms. In Proceedings of the International Conference on Machine Learning (ICML), PMLR, Virtual, 13–18 July 2020.
41. Yang, M.; Guo, T.; Zhu, T.; Tjuawinata, I.; Zhao, J.; Lam, K.Y. Local Differential Privacy and Its Applications: A Comprehensive Survey. *Comput. Stand. Interfaces* **2023**, *89*, 103827. [CrossRef]
42. Wang, H.; Zhao, Q.; Wu, Q.; Chopra, S.; Khaitan, A.; Wang, H. Global and Local Differential Privacy for Collaborative Bandits. In Proceedings of the 14th ACM Conference on Recommender Systems, Virtual, 22–26 September 2020; ACM: New York, NY, USA, 2020.
43. Holohan, N.; Antonatos, S.; Braghin, S.; Mac Aonghusa, P. The Bounded Laplace Mechanism in Differential Privacy. *arXiv* **2018**, arXiv:1808.10410. [CrossRef]
44. Hsu, J.; Gaboardi, M.; Haeberlen, A.; Khanna, S.; Narayan, A.; Pierce, B.C.; Roth, A. Differential Privacy: An Economic Method for Choosing Epsilon. In Proceedings of the 2014 IEEE 27th Computer Security Foundations Symposium, Vienna, Austria, 19–22 July 2014; IEEE: Piscataway, NJ, USA, 2014.
45. Mohammed, N.; Chen, R.; Fung, B.C.; Yu, P.S. Differentially Private Data Release for Data Mining. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; ACM: New York, NY, USA, 2011.

46. OpenMined. PyDP. Available online: https://github.com/OpenMined/PyDP (accessed on 20 June 2024).
47. 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).
48. Kennedy, C. Nursing Home COVID-19 Data. Available online: https://www.kaggle.com/datasets/corykennedy/nursing-home-covid19-data (accessed on 20 June 2024).
49. Huang, W.; Zhou, S.; Zhu, T.; Liao, Y.; Wu, C.; Qiu, S. Improving Laplace Mechanism of Differential Privacy by Personalized Sampling. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December–1 January 2020; IEEE: Piscataway, NJ, USA, 2020.
50. Balle, B.; Wang, Y.-X. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In Proceedings of the International Conference on Machine Learning (ICML), PMLR, Stockholm, Sweden, 10–15 July 2018.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.