Anonify: Decentralized Dual-level Anonymity for Medical Data Donation

Sarah Abdelwahab Gaballah
Ruhr University Bochum
sarah.gaballah@rub.de

Lamya Abdullah
Technical University of Darmstadt
abdullah@tk.tu-darmstadt.de

Mina Alishahi
Open Universiteit
mina.sheikhalishahi@ou.nl

Thanh Hoang Long Nguyen
Technical University of Darmstadt
long.nguyen@stud.tu-darmstadt.de

Ephraim Zimmer
Technical University of Darmstadt
zimmer@privacy-trust.tu-darmstadt.de

Max Mühlhäuser
Technical University of Darmstadt
max@tk.tu-darmstadt.de

Karola Marky
Ruhr University Bochum
karola.marky@rub.de

ABSTRACT

Medical data donation involves voluntarily sharing medical data with research institutions, which is crucial for advancing healthcare research. However, the sensitive nature of medical data poses privacy and security challenges. The primary concern is the risk of de-anonymization, where users can be linked to their donated data through background knowledge or communication metadata. In this paper, we introduce *Anonify*, a decentralized anonymity protocol offering strong user protection during data donation without reliance on a single entity. It achieves dual-level anonymity protection, covering both communication and data aspects by leveraging Distributed Point Functions, and incorporating k-anonymity and stratified sampling within a secret-sharing-based setting. *Anonify* ensures that the donated data is in a form that affords flexibility for researchers in their analyses. Our evaluation demonstrates the efficiency of *Anonify* in preserving privacy and optimizing data utility. Furthermore, the performance of machine learning algorithms on the anonymized datasets generated by the protocol shows high accuracy and precision.

KEYWORDS

Medical Data Donation, Data Anonymity, Anonymous Communication, Distributed Point Functions, k-anonymity, Stratified Sampling

Proceedings on Privacy Enhancing Technologies 2024(3), 94–108

https://doi.org/10.56553/popets-2024-0069

1 INTRODUCTION

Medical data donation[1] is a voluntary act where individuals share their health-related information with researchers to support scientific research, medical advancements, and public health initiatives [4]. However, medical data donation faces significant challenges primarily due to privacy and security concerns based on the sensitivity of medical data. The most critical concern revolves around the potential for the risk of individuals being identifiable through their data [35, 40]. Therefore, it is crucial to provide strong protection guarantees for users when they donate their medical data.

Medical data is typically provided to researchers in the form of a relational (tabular) structure. A table is composed of columns (attributes) and rows (records), with attributes categorized as *direct identifiers*, *quasi-identifiers* (QIDs), *sensitive attributes* (SAs), or *non-sensitive attributes* [26]. Direct identifiers, such as names or social security numbers, explicitly identify record owners. QIDs, like age, job, sex, or zip code, may not identify individuals on their own but could if combined. SAs encompass sensitive person-specific information, such as diseases. Non-sensitive attributes include those that do not fit into the above categories.

Simply removing direct identifiers from data is not sufficient to prevent re-identification [38]. It has been shown that if an individual's record is unique based on QIDs, an attacker with this information can directly link the record to its owner, leading to *identity disclosure*.

Even in cases where a group of individuals in a dataset shares identical QID values, the absence of diversity in the SA values within the records of these individuals can make them vulnerable to *attribute disclosure attacks* [26]. In Table 1, we present an example of such attacks, illustrating two groups where individuals in each exhibit similarity in QID values. Specifically, the first three records belong to one group, while the remaining records belong to another group. An adversary can deduce that any individual with a record in the first group has hepatitis, as all records in the group share

[1] In medical data donation scenarios, data is gathered by research institutes from individuals through apps, such as the Corona-Datenspende app [30]. The set of data donors may include both those currently experiencing health issues and those in good health. This differs from traditional medical data collection scenarios, where data is usually collected by hospitals or medical institutions as part of the treatment process.

Table 1: Example for the attribute disclosure attacks.

No.	Age	Sex	Zip Code	Disease
1	43	Male	56126	Hepatitis
2	43	Male	56126	Hepatitis
3	43	Male	56126	Hepatitis
4	35	Female	56121	Coronary Heart Disease
5	35	Female	56121	Arrhythmia
6	35	Female	56121	Valve Disease

identical SA values. Similarly, each individual in the second group can be inferred to have a heart-related disease due to SA values in the records that imply a shared trait.

To mitigate de-anonymization, various techniques have been introduced, with the most popular ones being k-anonymity for addressing identity disclosure attacks, and ℓ-diversity and t-closeness for mitigating attribute disclosure attacks. These techniques are designed for a centralized setting where a single entity is responsible for aggregating all individuals' medical data [6]. This implies that the entity has knowledge about each individual and their corresponding medical information, requiring users to place trust in this entity. Such a requirement may potentially reduce users' willingness to donate their medical data. Furthermore, the centralized nature of this entity introduces a single point of failure. In the event of a data breach, the privacy of all users could be compromised.

This paper addresses de-anonymization attacks, specifically identity and attribute disclosure, and issues arising from centralized medical data sharing. It proposes *Anonify*, a decentralized anonymity protocol designed for medical data donation. It offers dual-level anonymity protection: (1) at the communication and (2) at the data level. *Anonify* guarantees anonymous communication to prevent any linkability between users and their communicated records, enabling users to donate data without the need to place their trust in a single entity. This protection is achieved through a secret-sharing-based method for anonymous writing called *Distributed Point Functions* (DPF)[7], coupled with a broadcasting-based approach for anonymous data retrieval. Additionally, to defend against de-anonymization risks associated with donated medical data, it employs k-anonymity[38] and *stratified sampling* [28], all within decentralized settings.

Anonify consists of two phases: the registration phase and the publishing phase. In the registration phase, users employ DPF to anonymously submit records containing their QID values to an aggregator operated by multiple collaborating servers. To guard against identity disclosure attacks, *Anonify* applies k-anonymization to these records and organizes them into groups based on QID similarity. Using a broadcast-based approach, users can then anonymously learn their corresponding group. In the second phase, using DPF, users anonymously transmit their medical data (SA values) associated with their group identifier to the aggregator. To protect against attribute disclosure attacks, *Anonify* conducts stratified sampling on encrypted data, revealing only a portion of records in each group. Finally, the protocol disseminates the anonymized sampled donated data to researchers.

Contributions: In this paper, we make the following contributions:

- We introduce *Anonify*, a decentralized protocol for anonymous medical data donation, ensuring protection at communication and data levels. By leveraging DPF, *Anonify* aggregates data from users and applies k-anonymity and stratified sampling without relying on a single entity. This prevents the aggregator from de-anonymizing users, even with communication metadata or identity and attribute disclosure attacks. Additionally, our novel application of DPF facilitates the employment of stratified sampling without requiring trust in the aggregator with the entire set of records.
- We conduct a security analysis to demonstrate that *Anonify* achieves our security goals in terms of anonymous communication and data anonymity.
- We assess the efficiency of *Anonify*, utilizing a realistic medical dataset to simulate user-submitted records. Our evaluation incorporates multiple utility and privacy metrics, along with an examination of the data distribution characteristics post-application of *Anonify*. To ensure the anonymized data allows accurate analysis, we tested eight well-known machine learning classifiers on the anonymized dataset, revealing results closely resembling those of the original non-anonymized dataset.

The remainder of this paper is organized as follows: Section 2 provides the necessary background knowledge about anonymous communication and data anonymity. In Section 3, we explain our system and threat model, along with the security properties provided by our protocol. Section 4 introduces our protocol for decentralized anonymous medical data donation. In Section 6, we describe the evaluation of our approach. Section 7 presents related work, and in Section 8, we conclude our paper.

2 BACKGROUND

In this section, we explain the methods we use in our protocol, namely DPF, k-anonymity, generalization, and sampling.

2.1 Distributed Point Functions

Distributed Point Functions (DPF) [7, 13] are cryptographic constructs designed to facilitate secure and privacy-preserving computations in distributed or decentralized environments. DPF enables users to write in a database D distributed across a set of servers S without any of the servers being able to link any user to the specific message they wrote. This guarantee is achieved if at least one of the servers is honest.

To describe how anonymous writing can be done using DPF, we first introduce a basic secret-sharing-based approach, then explain how DPF improves this approach to enable efficient anonymous writing.

Suppose there are n servers, with each server storing a full copy of a database D. All servers collectively maintain the contents of this database. A user aims to submit a message m_i to D without the n servers storing D being able to link the message to the user's identity. The naive approach to achieve this is as follows:

Initially, the user computes a vector v with the same length as the database D. This vector contains the message m at a randomly selected index t (chosen locally by the user) and 0 at all other

indices. Subsequently, the user generates n secret shares $v_1, \ldots, v_n$ that satisfy the following properties:

(1) $\sum_{i=1}^{n} v_i = v$
(2) Any combination of $n-1$ secret shares does not reveal information about m or the index where m is located.

The user then distributes these shares to the servers, with the i-the server, $s_i \in S$, receiving v_i. Each s_i adds v_i to its database instance D^i using the operation $D^i \leftarrow D^i + v_i$.

After processing requests from multiple users, the servers collaborate to compute a combined database $D = \sum_{i=1}^{n} D^i$. Assuming each user selected a unique index for their messages, D contains all original messages.

For this method to work, transmitting a vector with the same size as the database for each write request is needed, making this method inefficient. To address this inefficiency, Corrigan-Gibbs et al. [7] proposed DPF to *compress* the shares transmitted to the servers.

Definition 1 (DPF). *Let $f_{t,m} : \{0, \ldots, \ell\} \mapsto \mathbb{F}$ be a point function with*

$$f_{t,m}(j) = \begin{cases} m & \text{for } j = t \\ 0 & \text{for } j \in \{0, \ldots, \ell\} \setminus t \end{cases}$$

$f_A, f_B : \{0, \ldots, \ell\} \mapsto \mathbb{F}$ are distributed point functions of $f_{t,m}$ if:

(1) neither f_A nor f_B individually reveal any information about m or t, and
(2) $\forall\, j \in \{0, \ldots, \ell\} : f_A(j) + f_B(j) = f_{t,m}(j)$

To generate n DPF-shares $f_1, \ldots, f_n$ containing m at a random index t, the user employs $\mathrm{GenDPF}(m, t)$. These shares are distributed among the servers, with each server s_i receiving f_i. The server s_i can derive v_i by evaluating $f_i(j)$ at every point $j \in \{0, \ldots, \ell\}$. Current research indicates that sending a DPF share instead of v_i reduces the communication cost to $O(\lambda \cdot \log \ell + |m|)$ bits, where λ is a security parameter [5].

2.2 k-anonymity

To mitigate identity disclosure attacks, it is crucial to anonymize the QID values in datasets. A widely used concept for this is *k-anonymity* [38], which modifies the dataset to ensures that at least k records in the dataset share the same QID values. k-anonymity guarantees that even if an attacker knows the QID values of a record owner, that record owner remains anonymous within a group, known as an *equivalence class*. The parameter k acts as a control for the level of anonymity offered, with larger values enhancing anonymity but potentially reducing data utility. Table 1 provides an example of k-anonymity with $k = 3$.

The main limitation of k-anonymity arises from the potential of similar or identical SA values in the equivalence classes, which can constrain the anonymity protection it offers. Therefore, k-anonymity cannot protect against attribute disclosure attacks.

2.3 Generalization

Generalization stands out as the main non-perturbative technique employed with k-anonymity. Non-perturbative techniques, in general, reduce data information by minimizing or suppressing detail without altering the content, preserving data truthfulness. Generalization enhances data anonymity by replacing QID values with more generalized yet semantically consistent values [37]. For categorical attributes, such as gender or job, specific values can be replaced with more general values using a value generalization hierarchy (VGH) [32]. For each attribute, a VGH is described as a tree structure whose leaves contain the values of the attribute and non-leaf nodes define generalized values. Figure 1 shows a VGH for the attribute *Job*, as an example. In this figure, jobs like surgeon and nurse are generalized to the broader category of medical professional. For numerical attributes, such as age, exact values can be replaced by intervals containing the exact values. For example, the age 45 could be generalized to the interval "[41-60]".

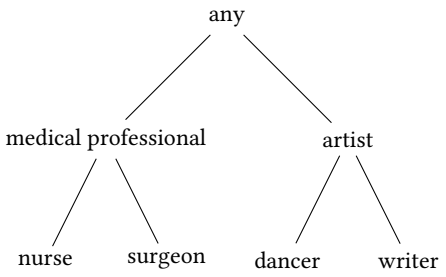

Figure 1: Example VGH for the categorical attribute *job*

While there are various approaches to achieving k-anonymity, our focus in this paper is on k-anonymization based on generalization. This choice is driven by the non-perturbative nature of generalization, which aids in preserving data truthfulness compared to perturbative techniques that often introduce new information. Although suppression (deleting specific QID values and replacing them with a special symbol, e.g., *) is non-perturbative, we exclude it due to its tendency to lower data utility [20]. Therefore, to maintain high data utility, generalization appears to be a more favorable option than suppression. It is important to note that over-generalization can negatively impact data analysis results [26].

2.4 Sampling

Sampling involves retaining only a portion of records from an original dataset and can occur either before (pre-sampling) or after (post-sampling) k-anonymizing the dataset [39]. Pre-sampling reduces the input dataset size for k-anonymization, thereby lowering computing power requirements. Post-sampling allows for advanced techniques that leverage the k-anonymous dataset generated after generalization. Combining both pre-sampling and post-sampling is feasible for very large datasets.

Sampling can be employed using different methods. The most common methods are simple random sampling and stratified sampling [28]. Simple random sampling involves randomly removing the desired number of records, ensuring no bias by definition. However, it does not guarantee equal removal from each equivalence class, leading to an unfair distribution in protection against de-anonymization attacks [39]. Stratified sampling addresses this issue by proportionally removing records from each equivalence class based on its size. This ensures a certain level of uncertainty for each record. As this method relies on equivalence classes, it is only applicable as a post-sampling method.

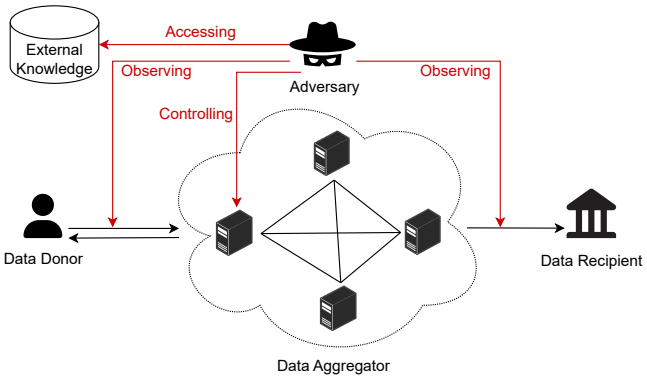

Figure 2: An overview of the system and threat model.

In our protocol, we employ post-stratified sampling, a choice that aligns well with the assumptions we outline for our system model and threat model, as detailed in the next section.

3 MODELS & PROPERTIES

This section describes the system model and the design assumptions of our proposed protocol, *Anonify*. It also discusses the adversary's goal and capabilities and presents the security properties of our protocol.

3.1 System Model

We consider a model that consists of three main components: the data donors (users), the data aggregator, and the data recipients (researchers), see Figure 2.

Data Donors. We assume a set of users U that represent the data donors who participate by sending their data to the data aggregator. Each user $u_i \in U$ has a record r_i that consists of a set of QIDs (personal information) and a set of SAs (medical data). Each QID is denoted as qid_j, and each SA is denoted as $sens_j$. That means r_i can be represented as $r_i = \{qid_1, qid_2, \ldots, qid_x, sens_1, sens_2, \ldots, sens_y\}$.

To guarantee truthfulness at the record level, we assume that every user u_i is truthful and does not falsify their data. Each record r_i collected by the system matches with an existing individual in real life [15].[2]

Data Aggregator. The data aggregator $\mathcal{G}$ is responsible for collecting users' data and making it accessible to the data recipient(s). The responsibilities of the aggregator $\mathcal{G}$ are distributed across n servers, implying that $\mathcal{G}$ is managed by multiple servers to avoid dependence on a single entity. Each of these servers maintains a copy of a database D, and collectively, the servers manage the contents of this database. The servers are assumed to employ methods that enable anonymous communication. Furthermore, the aggregator $\mathcal{G}$ is expected to ensure data anonymity for users through the implementation of k-anonymity and stratified sampling in decentralized settings.

[2]Data truthfulness is crucial in the context of medical data donation because the donated data can be used for treatments or the analysis of the effects of medicines [8]. Falsified data can lead to incorrect treatments and analyses.

Data Recipient. The entity that needs the donated data is referred to as the data recipient $\mathcal{T}$. In some scenarios, there might be several data recipients, such as multiple research institutes. We assume that the recipient gets an anonymized dataset of the users' records (R'). Each record in R' corresponds to a unique user and contains an anonymized information about the user's personal information and medical data. It is important to note that we only consider the case where the exchanged data between users and researchers is unidirectional. In other words, we focus on the situation where researchers collect users' data but do not provide anything in return, such as feedback or analysis results, to the users.

3.2 Threat Model

We consider the presence of an adversary, denoted as $\mathcal{A}$, whose goal is to de-anonymize users by identifying their SA values to gain insights into their health details. $\mathcal{A}$ can have background knowledge about some of the users, specifically information about their QID values. The set U' represents users about whom $\mathcal{A}$ has no knowledge of their QID values. Additionally, since each user is assumed to generate their SA values locally and these values are only known to the user, $\mathcal{A}$ is assumed to have no information about the SA values of any users in the set U.

$\mathcal{A}$ is a global passive adversary who is capable of observing all incoming and outgoing network traffic in the system. However, $\mathcal{A}$ is not able to manipulate the traffic, such as dropping, altering transmitted messages, or injecting new messages. Neither can it gain any information about the actual content of messages while transmission due to message end-to-end encryption. Further, it lacks the capability to associate users based on message size, given our assumption of fixed message size.

We assume that the adversary collaborates with the data recipients; therefore, we consider the data recipient(s) as untrusted. Besides, $\mathcal{A}$ can have control over a subset of $\mathcal{G}$'s servers, meaning at least one of the servers must be honest. Even under the assumption that a subset of $\mathcal{G}$'s servers may be malicious, $\mathcal{G}$ is expected to maintain honesty in executing the protocol. Without this assumption, anonymity is unaffected (see Section 5), but availability could be compromised as malicious servers may deny the service by manipulating their database instances. All users are also assumed to be honest, as this is important for ensuring data truthfulness and maintaining system availability. It is worth noting that a malicious user cannot manipulate others' data but can submit false data or compromise availability by submitting corrupted DPF shares.

3.3 Security Properties

Anonify aims to provide the following properties:

3.3.1 Anonymous Communication. Our protocol is designed to protect communication between users (data donors) and $\mathcal{G}$ by providing the following two anonymity properties: *sender anonymity* and *receiver anonymity*. These properties prevent $\mathcal{A}$ from de-anonymizing users based on communication metadata such as IP addresses, or the time of sending or receiving.

Informally, sender anonymity is the property where, $\mathcal{A}$ cannot determine which user in U' wrote specific messages in the database D any better than making random guesses.

Definition 2 (sender anonymity). *When $\mathcal{A}$ lacks prior knowledge about the message's content, the protocol guarantees sender anonymity. For each message within D, sender anonymity is achieved if the protocol ensures that $\mathcal{A}$ cannot significantly reduce the probability of accurately identifying the user $u_i \in U'$ responsible for writing a specific message in D to less than $1/|U'|$.*

Essentially, this implies that the adversary cannot de-anonymize u_i (i.e., associate a message with a specific user) more effectively than random guessing from the set of users U'. The strength of the sender anonymity property depends on the size of U', where a larger $|U'|$ implies a larger anonymity set size, thereby ensuring stronger anonymity. To ensure a minimum level of anonymity, the size of U' should be larger than or equal to a certain threshold.

During the protocol execution, particularly in the registration phase, each user will need to retrieve specific information from $\mathcal{G}$ to proceed to the publishing phase. This information should be obtained anonymously. As a result, the protocol provides the receiver anonymity property for users, ensuring that no adversary can learn which piece of information a user is interested in retrieving from $\mathcal{G}$.

Definition 3 (receiver anonymity). *When $\mathcal{A}$ lacks prior knowledge regarding the content that the user u_i wants to retrieve from $\mathcal{G}$, the protocol provides receiver anonymity. This is achieved when the protocol ensures that $\mathcal{A}$ has only a negligible probability of successfully deducing the specific data $u_i \in U'$ intends to retrieve, with this probability being close to $1/|U'|$.*

Our protocol guarantees that all users can access the data of interest from the aggregator while minimizing the likelihood of $\mathcal{A}$ successfully discovering which data u_i is retrieving down to $1/|U'|$.

3.3.2 Data Anonymity. Our protocol protects users at the data level by providing the following two anonymity properties: *k-anonymity* and *sensitive attribute (SA) uncertainty*. These properties protect users from data de-anonymization attempts, particularly by defending against identity and attribute disclosure attacks that may target the donated data.

k-anonymity necessitates that a minimum of k individuals have identical QID values in the anonymized dataset R'. In this manner, even if $\mathcal{A}$ acquires knowledge of the QID values associated with a particular record owner, that individual still retains anonymity concerning QID values within their equivalence class.

Definition 4 (k-anonymity). *The protocol provides k-anonymity if for each unique combination of QID values that is present in R', there exist at least k-1 other records in R' with the identical combination of these QID values.*

The k-anonymity property serves as a safeguard against identity disclosure attacks. The k parameter represents the minimum size of an equivalence class within the anonymized dataset R'. The value of k determines the level of anonymity offered, where higher values of k lead to stronger anonymity protection.

Relying solely on the k-anonymity property may not provide sufficient data protection for users. While k-anonymity can guarantee protection against identity disclosure attacks, it cannot ensure protection against attribute disclosure attacks. These can compromise the data anonymity of users when there is a lack of diversity in SA values within equivalence classes in R'. In a scenario where $\mathcal{A}$ knows the QIDs for a user u_i, it can identify the equivalence class to which u_i belongs. If all records within this class share identical SA values, $\mathcal{A}$ can then learn the SA values of u_i. To address this, our protocol introduces an additional property known as SA uncertainty. This property ensures that $\mathcal{A}$ cannot determine whether u_i's record is among the records in R'. Consequently, even if all records in R' share a specified SA value, $\mathcal{A}$ cannot ascertain if u_i has this value. This property is achieved in our protocol by applying stratified sampling.

Definition 5 (sa uncertainty). *The protocol provides SA uncertainty if the probability that $\mathcal{A}$ can guess that u_i's record is one of the records in R' is ρ. The value of ρ is calculated as the ratio of the number of records in u_i's equivalence class before sampling to the number after sampling.*

The parameter ρ defines the protection level granted by the SA uncertainty property. Smaller ρ values indicate greater uncertainty for $\mathcal{A}$ regarding the SA values of users, resulting in higher protection.

SA uncertainty, like t-closeness and ℓ-diversity, protects against attribute disclosure attacks. However, both t-closeness and ℓ-diversity require centralization, whereas SA uncertainty can be achieved by a decentralized system. Additionally, SA uncertainty ensures that adversaries cannot determine the presence of a user's SA values in R', whereas t-closeness and ℓ-diversity do not conceal the presence of a user record in R'.

4 PROTOCOL ARCHITECTURE

In this section, we describe *Anonify*, our decentralized protocol designed to enable medical data donation. *Anonify* operates in two phases: the registration phase and the publishing phase. The main steps of the protocol are illustrated in Figure 3.

4.1 Registration Phase

The protocol begins with a registration phase, during which users are required to submit their personal information (QID values) to $\mathcal{G}$. To prevent $\mathcal{G}$ from linking users to their submitted data through the exploitation of communication metadata, our protocol employs the DPF method. Other anonymous communication systems, such as onion routing or mix networks, can also be used to maintain unlinkability between users and the data they send to the aggregator. However, these alternatives may have weaker security guarantees compared to the DPF method we use, as they are known to be vulnerable to traffic analysis attacks by an adversary controlling a substantial portion of the network or monitoring multiple links within it [9].

Algorithm 1 depicts the details of the registration phase. The users first write their data anonymously in D using DPF. After a specific number of users contribute their records into D, the $\mathcal{G}$'s servers collaboratively share and combine their database instances to unveil the content in D. Subsequently, they apply k-anonymity, categorizing records into equivalence classes based on the similarity of QID values.[3] This categorization guarantees that each class $EC_j \in$

[3] All servers are assumed to execute the protocol honestly, allowing any individual server to perform k-anonymization. However, advocating for uniform execution by all servers ensures process integrity and consistent results.

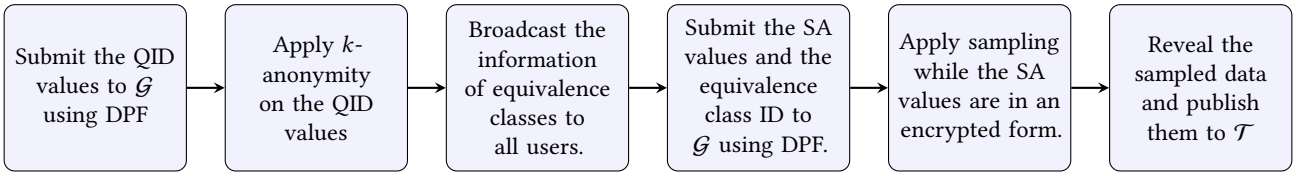

Figure 3: The main steps of the *Anonify* protocol, with the initial three steps corresponding to the registration phase and the subsequent three steps associating with the publishing phase.

EC consists of a minimum of k records. The servers allocate an identifier e_j to each class EC_j, followed by broadcasting a message to all users. [4] This message contains a list of class identifiers and the personal identifiers of users assigned to each class. Each user independently determines their class identifier by checking which equivalence class their personal identifier p_i is associated with. It is important to note that each user generates their personal identifiers locally.

Algorithm 1 Registration Phase

1. **Prepare Registration Message.** For each user u_i, let p_i be a randomly generated identifier, and $qr_i = \{qid_1, qid_2, \ldots, qid_x\}$ represents a record of QID values. The user u_i creates a message $m_i = (p_i, qr_i)$.
2. **Submit DPF Shares.** Each user u_i:
   - generates DPF shares: $\{f_1, \ldots, f_n\} = \mathsf{GenDPF}(m_i, t)$, where t is a random index in D.
   - submits one share to each of the n servers of $\mathcal{G}$. The share is encrypted using the receiving server's public key.
3. **Reveal the Database Content.** Each server of the n servers of $\mathcal{G}$:
   - uncompresses the u_i's DPF share and adds the result to its instance of D.
   - If the number of new users registering on the servers exceeds a predefined threshold, U is defined as this group of new users.
   - exchanges its database instances with the other servers and combines all instances to get the messages written to D by users in U.
4. **Apply k-anonymity.** The servers:
   - group the users' records (i.e., qr_i) into equivalence classes based on the similarity of QID values.
   - apply generalization to the records in each equivalence class $EC_j \in EC$.
   - allocate a distinct identifier e_j to each equivalence class EC_j.
5. **Broadcast Classes IDs.** The servers send, to all users, a message that contains each class's identifier e_j along with the set of users' identifiers attached to the records in EC_j.
6. **Get Class ID.** Each user u_i looks for her identifier p_i in the broadcasted message and finds the corresponding equivalence class identifier (e_j).

[4] Any of the servers can perform this step. Another approach involves dividing the set of users among the servers, with each server broadcasting the message to only a specific subset of users.

In Figure 4, Table (a) presents an example of the data that the servers have after revealing the messages users wrote in D, while Table (b) illustrates the equivalence classes generated through the k-anonymization step. Each equivalence class is associated with a class identifier e_j, QID values representing users within the class, and the identifiers of users belonging to that specific class. As shown in the table, users belonging to each class share the same generalized personal information (QID values) with all other users in the same class, making them indistinguishable within this specific equivalence class in terms of QID values.

Collisions. It is crucial to consider the problem of collision when users generate their random personal identifiers. Since each user u_i generates p_i locally, this can potentially create a situation where two users end up with the same identifiers. To significantly reduce the likelihood of this happening, we recommend that each user generates a random 128-bit number as their identifier.

Another collision issue can arise when users write their messages anonymously using DPF. As each user independently and randomly chooses the index to write their message in the database D, there is a risk of concurrent selections leading to two users attempting to write messages to the same index. In such cases, the content of these messages becomes irretrievable as one user's message overwrites another's. To address this issue, the size of D should be configured in a manner that significantly minimizes the probability of collisions. In other words, the size of D should be sufficient to accommodate the anticipated number of messages while maintaining a high probability of writing success without encountering collisions. In the work presented in [7], a formula is provided to compute the expected writing success rate for a given database size ℓ. This formula can assist in choosing a size that minimizes the likelihood of collisions:

$$SuccessRate \approx 1 - \frac{ReqCount}{DBSize} + \frac{1}{2}\left(\frac{ReqCount}{DBSize}\right)^2$$

For example, to handle writing requests from 100,000 users ($ReqCount = 100,000$ with each user having one writing request) and achieve an expected success rate of 90%, the database size $DBSize$ should be set to 1,000,000.

4.2 Publishing Phase

After users complete the registration phase, they proceed to the publishing phase, where they send their SA values (i.e., medical data) to $\mathcal{G}$. In this phase, communication between users and $\mathcal{G}$ is also established through the DPF method. Further, *Anonify* safeguards against the inference of users' SA values during this phase by employing protection against attribute disclosure attacks. This

PID	Sex	Age
3	F	32
5	F	26
8	M	45
9	M	37
12	M	42
16	F	28

(a)

CID	Sex	Age	PIDs
1	F	25-35	3, 5, 16
2	M	35-45	8, 9, 12

(b)

Figure 4: The information that $\mathcal{G}$'s servers have during the registration phase. PID stands for the user's random identifier (p_i), while CID stands for the class identifier (e_j).

CID	Test result
1	Positive
1	Negative
1	Positive
2	Negative
2	Negative
2	Negative

(a)

CID	Test result
1	Negative
1	Positive
2	Negative
2	Negative

(b)

Figure 5: The information that $\mathcal{G}$'s servers have during the publishing phase includes Table (a) in encrypted form and Table (b) in plain text.

protection is achieved through the use of stratified sampling, where servers only reveal a portion of records within each equivalence class. Leveraging DPF, *Anonify* applies stratified sampling in a decentralized manner and on encrypted records, eliminating the necessity to trust the aggregator with entire records before sampling.

Algorithm 2 describes the steps for the publishing phase. First, each user selects a random index t' in the database D and anonymously writes their class identifier at this index. Next, servers randomly select a fixed number of indices in D associated with each class. They designate these indices as locations where records will not be revealed (note that this occurs before users submit their actual records containing their SA values). Subsequently, each user anonymously writes their records to D at the same index t'. The servers then delete the records at indices in D marked not to be revealed; this process occurs while the records are still in an encrypted form. Therefore, servers remain unaware of the content of unreleased (deleted) records. After the sampling process, servers create a dataset from the remaining records in D and transmit this dataset to $\mathcal{T}$.

An example of the sampling is illustrated in Figure 5, where Table (b) represents a sample of the data in Table (a). In this example, only two records are released for each class, indicating that one record is deleted in each class.

Multiple Iterations of Medical Data Donation. In certain scenarios, data recipients (researchers) may require users to periodically submit their medical data (SA values). For instance, consider

Algorithm 2 Publishing Phase

1. **Reserve Index.** Each user $u_i \in U$:
   - generates DPF shares: $\{f_1, \ldots, f_n\} = \mathsf{GenDPF}(e_j, t')$, where t' is a random index in D.
   - submits one share to each of the n servers of $\mathcal{G}$. The share is encrypted using the receiving server's public key.
2. **Identify Indices Related to Each Class.** The servers:
   - add the received shares to their database instances and collaboratively reveal the content of D after all users in U submit their shares.
   - identify indices that contain a similar class identifier e_j.
3. **Choose the Unconsidered Indices.** The servers randomly select μ indices associated with each e_j. The set of all selected indices is defined as Z.
4. **Send SA Values.** Each user $u_i \in U$:
   - generates DPF shares: $\{f_1, \ldots, f_n\} = \mathsf{GenDPF}(sr_i, t')$, where $sr_i = \{sens_1, sens_2, \ldots, sens_y\}$ represents the u_i's record of SA values.
   - submits one share to each of the n servers of $\mathcal{G}$. The share is encrypted using the receiving server's public key.
5. **Apply Stratified Sampling.** Each server:
   - waits until adding to its database instance the shares received from every $u_i \in U$.
   - deletes the content of the indices in its database instance that are part of Z (refer to step 3).
   - exchanges its updated instance with the other servers and combines all instances to unveil the remaining records in D.
6. **Create the Anonymized Dataset.** The servers create the dataset R' by grouping together the records in D written in indices associated with the same class identifier e_j.
7. **Forward to the Recipient.** The servers transmit R' to the data recipient $\mathcal{T}$ along with the corresponding QID values that represent each class.

the "safevac" app introduced by the Paul-Ehrlich Institute (PEI) during the COVID-19 pandemic for a study on the vaccine's effects [25]. This study involves users downloading the app and responding to surveys at specific intervals following vaccination. In such cases, the typical single execution of the publishing phase is replaced by multiple iterations. This necessitates certain adjustments to the publishing phase. One key requirement is enabling data recipients to link data points from the same source, as this is pivotal for effective data analysis. To achieve this while safeguarding user anonymity, the following updates should be made to the publishing phase:

In the initial iteration where users submit their SA values, the steps of the publishing phase as outlined in Algorithm 2 must be followed. However, a crucial modification is required in step 4. Specifically, the message passed to the GenDPF function should be $m_i = (p'_i, sr_i)$ instead of passing sr_i only, where p'_i denotes a locally generated random identifier by a user u_i. Importantly, p'_i should be distinct from the identifier p_i created by u_i during the registration phase. This differentiation is important, as any similarity between these identifiers would enable servers to link the QID values submitted by the user in the registration phase with

the SA values provided during the publishing phase. Such a linkage would compromise user anonymity, undermining the protocol's fundamental objective.

In the following iterations where users need to provide new SA values to the data recipients, each user must consistently submit their message to the same index denoted as t', which they initially selected during the first iteration of the publishing phase. Additionally, in these subsequent iterations, only steps 4 through 7 of Algorithm 2 should be carried out. This implies that, across all iterations, the protocol always removes the records submitted to the indices that are part of the set Z, and Z is defined only once in the first iteration of the publishing phase. As a result, the protocol releases records from the same set of users in every iteration. This approach effectively prevents any information leakage to $\mathcal{A}$ across iterations regarding which users have records that have been published and which users have had their records removed by the protocol, thus protecting against intersection attacks [16, 17].

5 SECURITY ANALYSIS

In this section, we show that *Anonify* reaches the security properties that are defined in Section 3.3. *Anonify* achieves sender anonymity and receiver anonymity only for the set of users $U' \in U$ (see Section 3.2). It achieves k-anonymity and SA uncertainty for the set of all users U.

Sender Anonymity. *Anonify* ensures sender anonymity for users in U' during both the registration and publishing phases. To achieve sender anonymity, the users should be unlinked to the messages they write in D. In our protocol, this is achieved through secret sharing based on DPF. The anonymity guarantees of DPF were proven in [7].

We prove the sender anonymity property in our protocol by showing that $\mathcal{A}$ cannot use any of its abilities to compromise this property.

- *Passive Observation.* In each protocol phase, all users adhere to the protocol, ensuring that they each send the same number of shares to the servers. All messages exchanged between users and servers have the same size and are encrypted using the public key of the receiving server. The servers add all incoming shares to the database D and disclose all messages written by all users in D at once. Therefore, $\mathcal{A}$ is unable to link messages to senders through passive observation of requests between users and servers.
- *Server Corruption.* As per our assumptions, $\mathcal{A}$ has the capability to corrupt all but one of the $\mathcal{G}$'s servers. $\mathcal{A}$ can link a user with the share they send. However, $\mathcal{A}$ can learn at most $n-1$ out of the n DPF-shares from users. Existing literature provides formal proof that combining $n-1$ shares does not disclose any information about the content of the enclosed message [19].

Receiver Anonymity. *Anonify* guarantees receiver anonymity for users in U'. This property is only required during the registration phase, as users need to retrieve their corresponding equivalence class identifier. Receiver anonymity is achieved through a broadcast mechanism that ensures $\mathcal{A}$ cannot link users to their class identifiers.

We prove the receiver anonymity property in our protocol by demonstrating that $\mathcal{A}$ is incapable of utilizing any of its abilities to break this property.

- *Passive Observation.* Messages are only delivered to users when the servers broadcast a message containing the equivalence classes identifiers and the list of the personal identifiers of users assigned to each equivalence class. Since all users receive the same message from the servers, $\mathcal{A}$ cannot determine the class identifier of a user $u_i \in U'$ by simply observing communication between users and servers.
- *Server Corruption.* Since users generate their personal identifiers locally, $\mathcal{A}$ is unable to associate $u_i \in U'$ with their class identifier in the message by leveraging u_i's personal identifier. Furthermore, since $\mathcal{A}$ lacks prior knowledge about the QID values of u_i, it cannot establish a link between $u_i \in U'$ and their equivalence class identifier in the message by exploiting u_i's QID values.

k-anonymity. *Anonify* ensures that the dataset R' achieves k-anonymity. With this property, $\mathcal{A}$ cannot link a user $u_i \in U$ to their SA values in the dataset R' by exploiting the u_i's QID values, subject to the restriction that the size of the equivalence class of u_i is at least k. [5]

We prove the k-anonymity property in our protocol by showing that $\mathcal{A}$ cannot use any of its abilities to break this property.

- *Background Knowledge about QID values.* The dataset R' consists of equivalence classes, each containing a minimum of k records, with each record within an equivalence class containing SA values. Due to the shared QID values among all records within the same class, $\mathcal{A}$ is unable to identify a specific record for a user u_i, even if $\mathcal{A}$ has knowledge of the QID values of u_i and can specify the class to which u_i belongs.
- *Server Corruption.* Although $\mathcal{A}$ may have control over specific servers, it is unable to manipulate or interfere with the protocol execution, as even malicious servers are obligated to honestly execute the protocol. As a result, the dataset R' is constructed in an honest manner, with each record sharing identical QID values with at least k other records.

SA Uncertainty. *Anonify* ensures SA uncertainty, leaving $\mathcal{A}$ uncertain about the inclusion of SA values for user $u_i \in U$ in R'. The capability of $\mathcal{A}$ is limited to probabilistically guessing whether the SA values of user u_i are present in R', with a probability denoted by ρ. We prove the SA uncertainty property in our protocol by demonstrating that $\mathcal{A}$ is unable to leverage any of its capabilities to break this property.

- *Background Knowledge about QID values.* $\mathcal{A}$, with background knowledge of u_i's QID values, can identify the equivalence class in R' to which u_i belongs. However, even with knowledge of all classes and their associated users, $\mathcal{A}$ cannot be certain about the inclusion of u_i's specific record within the records released in an equivalence class in R'. This uncertainty arises due to the protocol's employment of stratified sampling. The protocol selects random records from each

[5] Due to sampling, the sizes of equivalence classes in R' are smaller than their sizes in the registration phase. Therefore, this should be considered when setting the value of k during registration to ensure that the final k value in R' is not very low.

equivalence class to be released in R'. Thus, the presence of the u_i's record among the released records becomes probabilistic, with the likelihood determined by the ratio ρ (see Section 3.3).

- *Server Corruption.* All servers, including potentially malicious ones, adhere to honest execution of the protocol. The use of DPF and the local selection of the message index in D by each user u_i prevents $\mathcal{A}$ from identifying the specific index to which u_i wrote their SA values in D. The stratified sampling step is done while the records are in an encrypted form in D which means the $\mathcal{G}$'s servers cannot determine which records that are deleted. Single servers cannot decrypt records before the stratified sampling step because the DPF method ensures that only collaborative decryption is possible. The stratified sampling protects against the $\mathcal{A}$'s ability to determine whether or not u_i's record containing their SA values is among the records in R'. Therefore, even if all the records in R' share a specified value for SA, $\mathcal{A}$ cannot be sure if u_i has this value because u_i's records might be one of the deleted records.

Impact of Weakening Assumptions on Anonymity. We assume that all servers, including potentially malicious ones, faithfully adhere to the protocol during execution (see Section 3.2). This assumption is made to ensure system availability, not anonymity, as the anonymity protection remains uncompromised without this assumption. The reason behind this is that the protocol relies on the following:

- DPF: This method guarantees that the content of submitted messages can only be revealed when all servers collaborate, with the condition that at least one server is honest. In the scenario where n-1 servers cooperate, the disclosure of message content becomes impossible. Therefore, sender anonymity and SA uncertainty cannot be broken, even in the presence of malicious servers deviating from the protocol.
- Equivalence classes agreement: The protocol requires that all servers reach the same set of equivalence classes. This ensures that malicious servers cannot manipulate the k-anonymization step or the list containing each class's identifier and the associated personal identifiers without detection by honest servers. Thus, malicious servers are unable to compromise k-anonymity and receiver anonymity.

To demonstrate the deterministic guarantees of *Anonify*, we assume user honesty and $\mathcal{A}$'s lack of knowledge regarding SA values. A malicious user cannot compromise the anonymity of other users unless they collude with $\mathcal{A}$ and disclose their SA values to it. However, if the malicious users refrain from collusion with $\mathcal{A}$, their influence is limited to affecting system availability and data truthfulness, not anonymity.

When $\mathcal{A}$ lacks knowledge of SA values, the certainty of $\mathcal{A}$ regarding u_i's record being in R' is calculated deterministically. It is expressed as the ratio of the number of records in u_i's equivalence class before sampling to the number after sampling. When $\mathcal{A}$ is aware of the SA values of some users in the equivalence class to which u_i belongs, the certainty about a u_i's record being in R' becomes entirely probabilistic and complex. It depends on factors such as the number of users in the class whose SA values are unknown to $\mathcal{A}$, and whether the records of these users are part of the sampled dataset R'. As sampling is done randomly within each class, the records in R' may be exclusively drawn from users with unknown SA values, from users with known SA values, or from a combination of both sets of users. Adding to the complexity, the certainty of $\mathcal{A}$ is also affected by the potential scenario where users with unknown SA values may share similar values with users whose values are known. This can significantly complicate the differentiation between records in R' belonging to users with known SA values and those belonging to users with unknown SA values.

6 EVALUATION

Anonymization approaches that alter data to meet anonymity needs often do so at the expense of data utility. Therefore, to evaluate such approaches, it's vital to assess both their anonymity impact and the utility of the anonymized data. In this section, we provide the performance results of *Anonify* in terms of anonymity and data utility.

As mentioned earlier, our protocol aims to ensure both anonymous communication and data anonymity. In our evaluation, we focus on assessing the data anonymity aspect. On the one hand, the protocol's guarantees regarding anonymous communication are deterministic, and their verifiability is demonstrated in Section 5. On the other hand, the bandwidth overhead in our protocol generally remains low, given that the number of messages users need to submit or receive is limited (see Section 4). Moreover, the use of DPF does not introduce high bandwidth overhead, as each share a user needs to send should have a bitlength of $O(\lambda \cdot \log \ell + |m|)$[13]. For example, the share size is expected to be 10.096 KB when the security parameter λ is set to 128 (as in[13]), the message size $|m|$ is 10 KB, and the database size ℓ is 1,000,000. Additionally, concerning latency overhead, it is crucial to note that the medical data donation scenario typically tolerates higher latency. Furthermore, employing DPF for anonymous writing has been proven to introduce low latency, even with a large user base, as demonstrated in [13, 18].

Nevertheless, it is important to emphasize that multiple parameters influence the latency and bandwidth overhead resulting from employing *Anonify*. A key parameter is n, representing the number of servers running $\mathcal{G}$. With an increase in the number of servers, more shares must be sent, and more database instances need to be combined, resulting in higher bandwidth overhead and latency. Another significant factor is the size of U, as it directly impacts the size of the database D. A larger U leads to a substantially larger database, causing increased latency when adding shares to a database instance and necessitating more computation time to combine all instances.

6.1 Dataset

In our evaluation, we simulate the records that the users send to $\mathcal{G}$ using the diabetes prediction dataset [6]. This dataset contains 100,000 records, each representing a distinct individual. It includes medical and demographic data, along with the diabetes status (i.e., whether individuals have diabetes or not). The dataset consists of features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose

[6] https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset

level. It can be used to train machine learning models to predict diabetes in patients based on their medical history and demographic information. Further, researchers can use this dataset to investigate the links between various medical and demographic factors and the chance of developing diabetes.

In our evaluation, we use the entire dataset, i.e., the size of U is 100,000 individuals. We designate the attributes—age, gender, and BMI—as QIDs because we assume that they can potentially be acquired by an adversary, for example, through information available on social networks. Conversely, we classify the remaining attributes as SAs since they pertain to vital health-related information that we presume the adversary does not have access to.

6.2 Data Utility Assessment

We assess the effectiveness of our protocol by measuring data utility using the Normalized Certainty Penalty (NCP) metric [43]. Additionally, we evaluate data utility by examining the change in data distribution after anonymization. Furthermore, we assess the performance of machine learning classification algorithms on the anonymized datasets produced by our protocol.

6.2.1 Information Loss. The NCP metric can quantify the information loss resulting from anonymization. This metric captures the uncertainty created by generalization [43]. It is very suitable to be used when the exact usage of the data is not well defined yet by the recipients [22], i.e., general-purpose data.

Let a dataset R with quasi-identifiers (num_1, num_2, ..., num_x, cat_1, cat_2, ..., cat_y), where (num_1, num_2, ..., num_x) are numerical attributes and (cat_1, cat_2, ..., cat_y) are categorical attributes. NCP defines the uncertainty for both of these attribute types. For a record $r_i \in R$, the NCP is computed as follows:

$$NCP(r_i) = \frac{\sum_{j=1}^{x} NCP_{num_j(r_i)} + \sum_{j=1}^{y} NCP_{cat_j(r_i)}}{x + y}$$

Where $NCP_{num_j(r_i)}$ represents the NCP of r_i with respect to num_j, and $NCP_{cat_j(r_i)}$ denotes the NCP of r_i with respect to cat_j.

The NCP for the entire dataset R is the sum of NCP values for all records:

$$NCP(R) = \frac{\sum_{i=1}^{|R|} NCP(r_i)}{|R|}$$

The data utility of the k-anonymized dataset is influenced by both the value of k and the specific k-anonymization method applied. Therefore, in our evaluation using NCP, we conducted tests with various k values. Further, we considered three different k-anonymity methods: Mondrian [23], one-pass k-means (OKA) [24], and ARX [29].

Figure 6 shows the results of our data utility experiments using the NCP metric. As depicted in the figure, we systematically varied the value of k, starting with an initial value of 50 and incrementing it in steps of 50. These experiments encompassed the entire dataset. The results obtained with the ARX method significantly outperformed those of the Mondrian and the OKA, even with higher k values. Moreover, our observations revealed a consistent increase in the NCP values across all methods as the k value increased. This phenomenon can be attributed to the fact that larger k values lead to larger minimum size of equivalence classes. Thus, it becomes increasingly challenging for records within the same equivalence class to reach an agreement on attribute values. Consequently, more generalization is required to satisfy the k-anonymity criterion, resulting in a reduction in data utility. These findings underscore the inherent trade-off between data utility and anonymity.

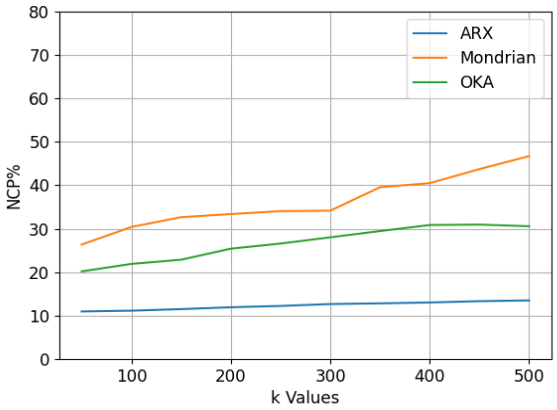

Figure 6: The impact of the k value and the k-anonymization method on data utility.

Since ARX produces the best results, we base our analysis on its results in all the experiments discussed in the following subsections. Furthermore, in all the upcoming experiments, we set the value of k to 250, as this choice strikes a favorable balance between anonymity and data utility.

6.2.2 Data Distribution. We compare the data distribution among three versions of the dataset: the original dataset, a k-anonymized dataset (referred to as "Anonymized" in the figures, representing the original dataset after k-anonymization), and a sampled anonymized dataset (representing R', i.e., displaying only a percentage of records from every equivalence class in the k-anonymized dataset). This data distribution comparison provides an indication of how the data has changed and illustrates the impact of anonymization.

In the experiments, when the sampling percentage is set at 70%, it indicates that μ (see Section 4.2) equals $100 - 70 = 30$, meaning 30% of the records in each equivalence class in the k-anonymized dataset were deleted. Lower sampling percentages imply more protection but typically at the expense of data utility. It's worth highlighting that in all our sampling experiments, we ran each experiment 10 times and computed the average results. This is necessary because the selection of records to be released within each equivalence class is performed randomly.

Figure 7 illustrates the average blood glucose levels for different age ranges across the original dataset, the k-anonymized dataset, and the sampled anonymized dataset (with a sampling percentage of 50%). Notably, the results show that both the k-anonymized dataset and the sampled anonymized dataset closely resemble the original dataset. This suggests a minimal impact of anonymization on the data distribution, as even when only 50% of records in the k-anonymized dataset are sampled, the distribution is still not noticeably affected. A similar finding is demonstrated in Figure 8 which depicts the number of individuals with hypertension based

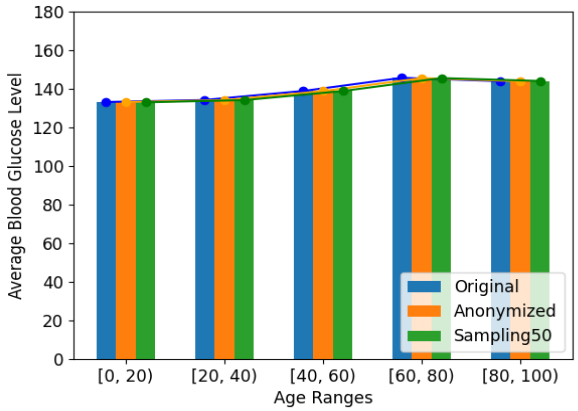

Figure 7: The average blood glucose level within different age ranges (k=250, Sampling percentage=50%).

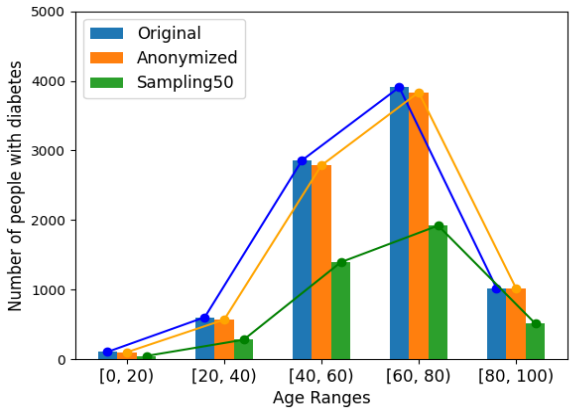

Figure 9: The average number of people with diabetes within different age ranges (k=250, Sampling percentage=50%).

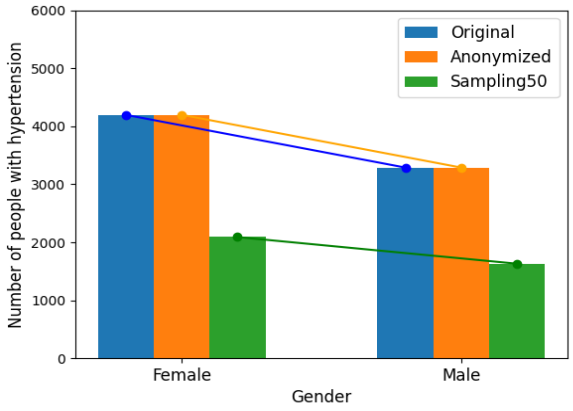

Figure 8: The average number of people with hypertension within different gender groups (k=250, Sampling percentage=50%).

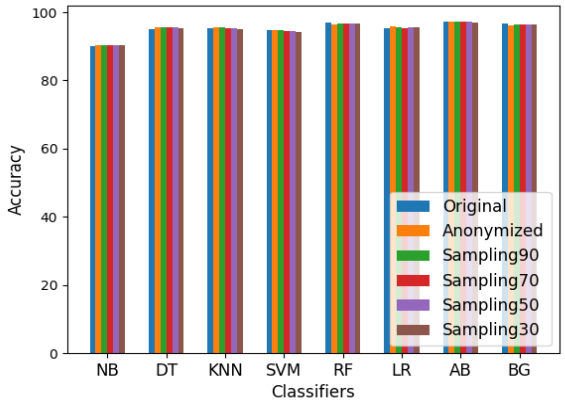

Figure 10: Accuracy (k=250, Sampling percentage=90%, 70%, 50%, and 30%).

on the gender attribute. The sampled anonymized dataset exhibits slight variations but remains well-aligned with the other datasets.

However, the sampled anonymized dataset appears to induce more noticeable alterations in the distribution of individuals with diabetes compared to the original dataset for older age ranges, particularly for age 60 years and above, as shown in Figure 9. This difference in data distribution is influenced by the chosen sampling percentage. Lower percentages, such as 50%, result in more deviations, while higher (e.g., 80%) percentages may yield only slight differences. This highlights the importance of carefully selecting the sampling percentage to achieve anonymity preservation without undermining data quality. In summary, the results demonstrate the effectiveness of our protocol, which integrates k-anonymity and stratified sampling, in maintaining anonymity without jeopardizing data distribution, especially when the parameters are carefully chosen.

6.2.3 Machine Learning Classifiers' Performance. Data recipients may require training machine learning models on the dataset generated by *Anonify* for diabetes prediction. Thus, we assess the performance of machine learning classification algorithms on the anonymized datasets produced by *Anonify*. More specifically, we compare the performance of eight well-known machine learning classification algorithms on the original dataset, the k-anonymized dataset, and different sampled anonymized datasets (with sampling percentages: 90%, 70%, 50%, and 30%). The considered algorithms include Decision Tree (DT), Naïve Bayes (NB), k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), AdaBoost (AB), and Bagging (BG). These algorithms offer various trade-offs in terms of simplicity, accuracy, robustness, and sensitivity to noise. For a detailed understanding of each, refer to [1, 3].

We assess the classifiers' performance in terms of Accuracy, Precision, Recall, and F1-score, which are defined based on True

Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) values. These four metrics are computed as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad , \quad \text{Precision} = \frac{TP}{TP+FP} \quad ,$$
$$\text{Recall} = \frac{TP}{TP+FN} \quad , \quad \text{F1-Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

In our experiment, we designated the "diabetes" attribute in the dataset as the class label, with values "1" indicating diabetes and "0" representing the absence of diabetes. We partitioned each dataset version into a 75% training set and a 25% test set. Subsequently, we trained the classifiers using the training data and evaluated their performance on the test dataset.

Figure 10 shows the accuracy scores, serving as indicators of the classifiers' proficiency in data classification, with higher values denoting superior performance. The results for the original dataset, k-anonymized dataset, and different sampled datasets show no noticeable differences, as they all exhibit close accuracy scores. Even when sampling only 30% of records from each equivalence in the k-anonymized dataset, the accuracy score remains high and is comparable to or slightly better than the result of the original data. This demonstrates the protocol's capability to be employed without negatively impacting classification accuracy.

Furthermore, this trend extends to recall, precision, and F1 score (as shown in Figure 13, 14 and 15 in Appendix A), reinforcing the fact that applying k-anonymity and sampling maintains the good performance of these metrics, as the results for sampled anonymized datasets remaining closely aligned with or better than those of the original dataset.

6.3 Data Anonymity Assessment

Data anonymization techniques (e.g., k-anonymity and differential privacy) involve altering or removing information in datasets to prevent the identification of individuals [27]. While effective in reducing direct identification risks, they cannot guarantee complete immunity from re-identification risks [14]. To assess re-identification risks in datasets anonymized by Anonify, we employed the Journalist Risk and Certainty metrics, which are commonly considered for this type of assessment [29, 39].

6.3.1 Journalist Risk (JR). It measures the probability of linking a specific record r_i to a targeted user [12]. Let the equivalence class $EC_j \in EC$ is denoted as $EC_j^{\mu} \in EC^{\mu}$ after the sampling step, where $EC^{\mu} \subset EC$. JR is calculated by multiplying the probability that r_i remains in the equivalence class EC_j after sampling (i.e., $\frac{|EC_j^{\mu}|}{|EC_j|}$) with the probability that the attacker selects the correct record from that sampled equivalence class (i.e., $\frac{1}{|EC_j^{\mu}|}$) :

$$JR(r_i) = \frac{|EC_j^{\mu}|}{|EC_j|} \cdot \frac{1}{|EC_j^{\mu}|} = \frac{1}{|EC_j|}, \text{ where } r_i \in EC_j^{\mu}$$

The journalist risk serves as an indicator of the level of risk associated with the anonymization process. A lower index implies more protection, and vice versa. Figure 11 provides an evaluation of the journalist risk index on the k-anonymized dataset and different sampled anonymized datasets. The journalist index results for all

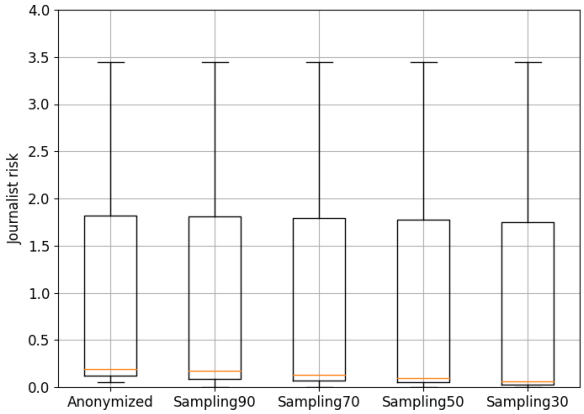

Figure 11: Journalist risk (k=250, Sampling percentage=90%, 70%, 50%, and 30%).

versions of the datasets exhibit very low values, indicating a higher level of anonymity. Additionally, as expected, reducing the number of records released, as seen in lower sampling percentages, leads to lower journalist index values. For example, at a 30% sampling percentage, the average journalist index reaches a significantly low value of 0.057075, indicating a substantially reduced risk.

6.3.2 Certainty Metric. This metric represents the likelihood that a record r_i is contained within the anonymized dataset [39]. It is computed as follows:

$$C(r_i) = \frac{|EC_j^{\mu}|}{|EC_j|}, \text{ where } r_i \in EC_j^{\mu}$$

For each record r_i in the original (non-anonymized) dataset, certainty is 100%, given that all records naturally exist in this dataset. The same principle applies to the k-anonymized dataset, where we solely rely on generalization without employing suppression (i.e., removing outlier records in terms of QIDs). This ensures the inclusion of records from all users in the k-anonymized dataset.

Figure 12 displays the percentage values of certainty. As expected, in the k-anonymized dataset (referred to as "Anonymized" in the figure), the highest level of certainty is maintained at 100%. This value indicates that all records are retained in the released dataset, ensuring the utmost level of confidence in data presence and predictability. In contrast, the sampled anonymized datasets, particularly as the sampling percentages increase, introduce uncertainty and reduce the confidence in the presence of records within the dataset sent to data recipients. Striking a balance between anonymity concerns and the necessity for certainty in the outcomes suggests the importance of adjusting the sampling percentage to a value that isn't excessively low. This ensures a better trade-off between data anonymity and the reliability of the results.

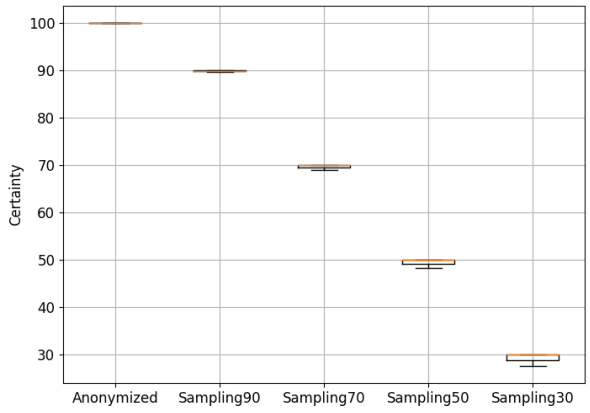

Figure 12: Certainty (k=250, Sampling percentage=90%, 70%, 50%, and 30%).

7 RELATED WORK

In this section, we discuss techniques commonly applied in practice [14] to safeguard privacy and anonymity within the context of medical data sharing.

Differential privacy (DP) is a technique that protects privacy by introducing noise to data, minimizing the impact of individual records on analysis results. Unlike other methods such as k-anonymity, ℓ-diversity, and t-closeness, DP doesn't rely on assumptions about attackers' knowledge [10]. Initially designed for interactive queries [10]—where data recipients can query the system interactively without access to the complete dataset—DP was later extended to non-interactive scenarios [11]. DP can employ noise addition in a local or global manner [26]. In local DP, noise is applied to individual data points independently, potentially causing more distortion than necessary due to a lack of consideration for the overall dataset. In contrast, global DP adds noise to the overall output, allowing characteristics of the dataset to be considered and generally leading to more accurate results. However, global DP, like k-anonymity, requires trust in the data aggregator from record owners to ensure privacy preservation. While effective against certain privacy threats, DP has drawbacks [21], including its inability to prevent all linkage attacks, introduced communication overhead, and potential hindrance of pattern detection in small populations or rare events due to added noise. Additionally, repeated queries on differentially private datasets can lead to privacy risks over time. Similarly to the challenges in k-anonymity, where selecting appropriate QIDs and finding a balanced k value is crucial, determining a suitable epsilon (ϵ) in DP presents a challenge due to the inevitable trade-off between privacy and utility.

Many data sharing protocols have been proposed, leveraging secure multi-party computation (SMPC) [34, 36, 41] or homomorphic encryption (HE) [31, 33, 42]. These protocols aim to ensure strong protection guarantees with minimal trust requirements. SMPC allows computations on distributed datasets without exposing raw data, ensuring each party retains control while defending against malicious attacks. Drawbacks of SMPC include computational overhead, increased communication complexity, and limited scalability [44]. On the other hand, HE enables computations on encrypted data, ensuring end-to-end confidentiality and eliminating the need for a central trusted authority. However, it faces challenges such as computational intensity, and slowing down processing, which limits its practicality [2]. Another common issue in data sharing protocols relying on SMPC or HE is their tendency to support specific or limited types of statistical analyses, hence limiting flexibility for researchers in conducting their studies.

Given the limitations of DP, SMPC, and HE, we assert that our protocol design offers superior flexibility to researchers compared to solutions based on any of these methods. We provide anonymized data to researchers without assuming the exact usage or nature of the studies, enhancing adaptability to diverse research needs. Furthermore, our protocol ensures better anonymity and data utility when compared to DP. The dataset anonymized by *Anonify* mitigates privacy risks associated with repeated queries, a concern inherent in differentially private datasets. *Anonify* depends on DPF, whose capability to support scalability without imposing high bandwidth and computational burdens has been demonstrated in the literature [13, 18]. That can position *Anonify* as a potentially superior option over solutions based on SMPC or HE in terms of scalability, bandwidth and computational efficiency.

8 CONCLUSION

Anonify is a decentralized anonymity protocol specifically designed for medical data donation. It offers users anonymous communication and data anonymity when donating their data without the need to trust a single entity. This is achieved through the utilization of a secret-sharing-based method called DPF, facilitating the anonymous sending of records, complemented by a broadcasting-based approach for anonymous data retrieval. Moreover, to mitigate data de-anonymization risks, we have employed k-anonymity and stratified sampling within a decentralized setting. Our comprehensive evaluation, encompassing various privacy and data utility metrics, demonstrates the effectiveness of *Anonify* in ensuring strong protection without compromising data utility.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

REFERENCES

[1] CC Aggarwal. 2014. Data classification: algorithms and applications.(1stedn).
[2] Bechir Alaya, Lamri Laouamer, and Nihel Msilini. 2020. Homomorphic encryption systems statement: Trends and challenges. *Computer Science Review* 36 (2020), 100235.
[3] Mina Alishahi and Nicola Zannone. 2021. Not a Free Lunch, But a Cheap One: On Classifiers Performance on Anonymized Datasets. In *Data and Applications Security and Privacy (DBSec)*. Springer, 237–258.
[4] Matthew Bietz, Kevin Patrick, and Cinnamon Bloss. 2019. Data donation as a model for citizen science health research. *Citizen Science: Theory and Practice* 4, 1 (2019).
[5] Elette Boyle, Niv Gilboa, and Yuval Ishai. 2016. Function Secret Sharing: Improvements and Extensions. (2016), 1292–1303. https://doi.org/10.1145/2976749.2978429
[6] Tânia Carvalho, Nuno Moniz, Pedro Faria, and Luís Antunes. 2022. Survey on privacy-preserving techniques for data publishing. *arXiv preprint*

arXiv:2201.08120 (2022).
[7] Henry Corrigan-Gibbs, Dan Boneh, and David Mazières. 2015. Riposte: An Anonymous Messaging System Handling Millions of Users. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*. IEEE Computer Society, 321–338. https://doi.org/10.1109/SP.2015.27
[8] D. Mentzer D. Oberle and G. Weber. 2020. Befragung zur Verträglichkeit der Impfstoffe gegen das neue Coronavirus (SARS-CoV-2) mittels Smartphone-App SafeVac 2.0. https://www.pei.de/SharedDocs/Downloads/EN/newsroom-en/pharmacovigilance-bulletin/single-articles/2020-safevac-app-en.pdf?__blob=publicationFile&v=3.
[9] George Danezis and Andrei Serjantov. 2004. Statistical disclosure or intersection attacks on anonymity systems. In *International Workshop on Information Hiding*. Springer, 293–308.
[10] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.
[11] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.
[12] Khaled El Emam. 2013. *Guide to the de-identification of personal health information*. CRC Press.
[13] Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, and Dan Boneh. 2021. Express: Lowering the cost of metadata-hiding communication with cryptographic privacy. In *30th USENIX Security Symposium (USENIX Security 21)*. 1775–1792.
[14] European Medicines Agency. 2017. Report on Data Anonymisation as a Key Enabler for Clinical Data Sharing. https://www.ema.europa.eu/en/documents/report/report-data-anonymisation-key-enabler-clinical-data-sharing_en.pdf. Accessed: June 25, 2024.
[15] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)* 42, 4 (2010), 1–53.
[16] Sarah Gaballah, Thanh Hoang Long Nguyen, Lamya Abdullah, Ephraim Zimmer, and Max Mühlhäuser. 2023. Mitigating Intersection Attacks in Anonymous Microblogging. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*. 1–11.
[17] Sarah Abdelwahab Gaballah, Lamya Abdullah, Minh Tung Tran, Ephraim Zimmer, and Max Mühlhäuser. 2022. On the Effectiveness of Intersection Attacks in Anonymous Microblogging. In *Secure IT Systems - 27th Nordic Conference, NordSec 2022, Reykjavic, Iceland, November 30-December 2, 2022, Proceedings (Lecture Notes in Computer Science, Vol. 13700)*. Springer, 3–19. https://doi.org/10.1007/978-3-031-22295-5_1
[18] Sarah Abdelwahab Gaballah, Christoph Coijanovic, Thorsten Strufe, and Max Mühlhäuser. 2021. 2PPS - Publish/Subscribe with Provable Privacy. In *40th International Symposium on Reliable Distributed Systems, SRDS 2021, Chicago, IL, USA, September 20-23, 2021*. IEEE, 198–209. https://doi.org/10.1109/SRDS53918.2021.00028
[19] Nethanel Gelernter and Amir Herzberg. 2013. On the limits of provable anonymity. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. 225–236.
[20] Aris Gkoulalas-Divanis, Grigorios Loukides, and Jimeng Sun. 2014. Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of biomedical informatics* 50 (2014), 4–19.
[21] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2019. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 746–789.
[22] Hyukki Lee, Soohyung Kim, Jong Wook Kim, and Yon Dohn Chung. 2017. Utility-preserving anonymization for health data publishing. *BMC medical informatics and decision making* 17, 1 (2017), 1–12.
[23] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. 2006. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*. IEEE, 25–25.
[24] Jun-Lin Lin and Meng-Cheng Wei. 2008. An efficient clustering method for k-anonymization. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*. 46–50.
[25] D. Oberle, D. Mentzer, and G. Weber. 2020. https://www.pei.de/SharedDocs/Downloads/EN/newsroom-en/pharmacovigilance-bulletin/single-articles/2020-safevac-app-en.pdf?__blob=publicationFile&v=3. Accessed: 2022-10-03.
[26] Iyiola E Olatunji, Jens Rauch, Matthias Katzensteiner, and Megha Khosla. 2022. A review of anonymization for healthcare data. *Big data* (2022).
[27] European Parliament. 2016. Regulation (EU) 2016/679 of the European Parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32016R0679 (Accessed 02-February-2021).
[28] Van L Parsons. 2014. Stratified sampling. *Wiley StatsRef: Statistics Reference Online* (2014), 1–11.
[29] Fabian Prasser and Florian Kohlmayer. 2015. Putting statistical disclosure control into practice: The ARX data anonymization tool. *Medical data privacy handbook* (2015), 111–148.
[30] RKI. 2019. Corona Data Donation Project. https://corona-datenspende.de/science/en/.
[31] Bharath K Samanthula, Gerry Howser, Yousef Elmehdwi, and Sanjay Madria. 2012. An efficient and secure data sharing framework using homomorphic encryption in the cloud. In *Proceedings of the 1st International Workshop on Cloud Intelligence*. 1–8.
[32] Pierangela Samarati. 2001. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering* 13, 6 (2001), 1010–1027.
[33] Hossein Shafagh, Anwar Hithnawi, Lukas Burkhalter, Pascal Fischli, and Simon Duquennoy. 2017. Secure sharing of partially homomorphic encrypted iot data. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–14.
[34] Haoyi Shi, Chao Jiang, Wenrui Dai, Xiaoqian Jiang, Yuzhe Tang, Lucila Ohno-Machado, and Shuang Wang. 2016. Secure multi-pArty computation grid LOgistic REgression (SMAC-GLORE). *BMC medical informatics and decision making* 16 (2016), 175–187.
[35] Joanna Sleigh. 2018. Experiences of donating personal data to mental health research: an explorative anthropological study. *Biomedical Informatics Insights* 10 (2018), 1178222618785131.
[36] Haris Smajlović, Ariya Shajii, Bonnie Berger, Hyunghoon Cho, and Ibrahim Numanagić. 2023. Sequre: a high-performance framework for secure multiparty computation enables biomedical data sharing. *Genome Biology* 24, 1 (2023), 5.
[37] Latanya Sweeney. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 571–588.
[38] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.
[39] Jenno Verdonck, Kevin De Boeck, Michiel Willocx, Jorn Lapon, and Vincent Naessens. 2023. A hybrid anonymization pipeline to improve the privacy-utility balance in sensitive datasets for ML purposes. In *the 18th International Conference on Availability, Reliability and Security*. 1–11.
[40] Torsten H Voigt, Verena Holtz, Emilia Niemiec, Heidi C Howard, Anna Middleton, and Barbara Prainsack. 2020. Willingness to donate genomic and other medical data: results from Germany. *European Journal of Human Genetics* 28, 8 (2020), 1000–1009.
[41] Felix Nikolaus Wirth, Tobias Kussel, Armin Müller, Kay Hamacher, and Fabian Prasser. 2022. EasySMPC: a simple but powerful no-code tool for practical secure multiparty computation. *BMC bioinformatics* 23, 1 (2022), 531.
[42] Alexander Wood, Kayvan Najarian, and Delaram Kahrobaei. 2020. Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Computing Surveys (CSUR)* 53, 4 (2020), 1–35.
[43] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. 2006. Utility-based anonymization using local recoding. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data*. ACM, 785–790. https://doi.org/10.1145/1150402.1150504
[44] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. 2019. Secure multi-party computation: theory, practice and applications. *Information Sciences* 476 (2019), 357–372.

A APPENDIX

The following figures show the eight machine learning classifiers' performance in terms of Recall, Precision, and F1-score, respectively.

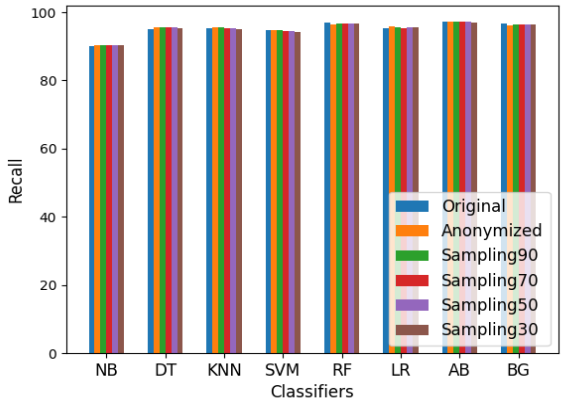

Figure 13: Recall (k=250, Sampling percentage=90%, 70%, 50%, and 30%).

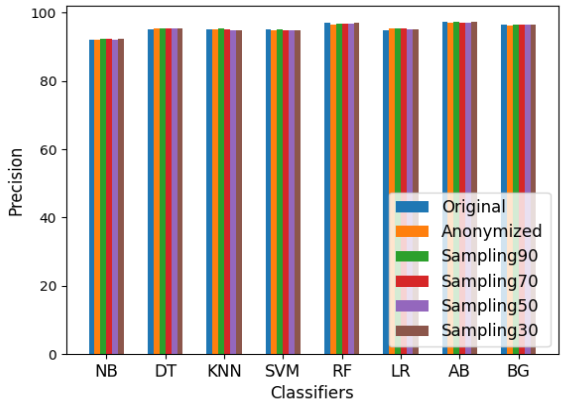

Figure 14: Precision (k=250, Sampling percentage=90%, 70%, 50%, and 30%).

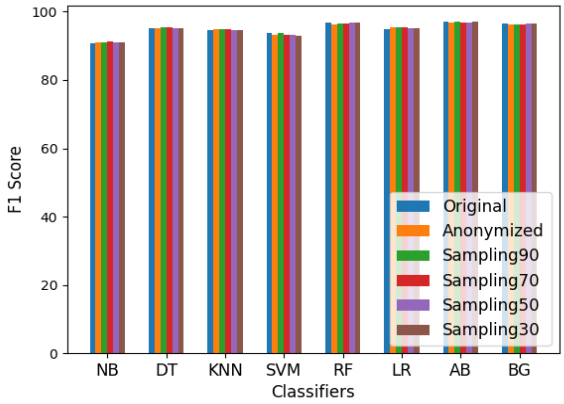

Figure 15: F1 Score (k=250, Sampling percentage=90%, 70%, 50%, and 30%).