MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

# COMPARISON OF THE EFFECTS OF DATA PRIVACY PRESERVING METHODS ON MACHINE LEARNING ALGORITHMS IN IOT

TAJ ELDEEN SALEH

**MASTER THESIS**

Department of Electronics and Computer Engineering

**ADVISOR**

Assoc. Prof. Ömer Korçak

ISTANBUL, 2022

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

# COMPARISON OF THE EFFECTS OF DATA PRIVACY PRESERVING METHODS ON MACHINE LEARNING ALGORITHMS IN IOT

TAJ ELDEEN SALEH

**MASTER THESIS**

Department of Electronics and Computer Engineering

**ADVISOR**

Assoc. Prof. Ömer Korçak

ISTANBUL, 2022

# DECLARATION OF AUTHORSHIP

I, Taj Eldeen Saleh, hereby declare that this thesis titled “Comparison of the Effects of Data Privacy Preserving Methods on Machine Learning Algorithms in IoT” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

**I’d like to dedicate my work to the memory of my grandfather, Shukri Saleh, a father, a grandfather and a teacher who always been my inspiration for seeking knowledge.**

# ACKNOWLEDGMENT

First and foremost, I would like to express my deepest appreciation to my parents for their endless and continuous support and encouragement in my education journey, I'm also grateful for all the support of my two sisters.

I'm very thankful to my thesis advisor Prof. Ömer Korçak who offered his continuous support and assistant, who enabled and encouraged me to explore and research my work and provided me with much appreciated insights.

I would like to mention Prof. Ali Çakmak who I worked with him the better part of my masters in bioinformatic lab, where I had the chance to research and gain experience that truly helped me during the journey.

At last, I would like to take this chance to especially thank my friends for being there for me throughout the process, who always believed in me, and cheered me up, I couldn't have done it without you guys.

# TABLE OF CONTENTS

# ÖZET

## NESNELERİN İNTERNETİNDE VERİ GİZLİLİĞİNİ KORUMA YÖNTEMLERİNİN MAKİNE ÖĞRENME ALGORİTMALARINA ETKİLERİNİN KARŞILAŞTIRILMASI

Veri gizliliğini korumak, birçok kuruluş ve birey için çok önemli ve artan bir endişe kaynağıdır. Gizlilik konusunu ele almak için, veriye dayalı hizmetler araştırma ve geliştirme üzerinde doğrudan etkileri olan birçok düzenleme uygulanmaktadır. Verilerin anonimleştirilmesi, belirli gizlilik düzenlemelerine uymak için kişisel olarak tanımlanabilir bilgileri kaldırarak bu sorunla başa çıkmanın bir yoludur. Ancak, anonimleştirme süreci tek başına verilere bir miktar gürültü getirir.

Bu çalışmada, anonimleştirme algoritmalarının uygulanmasının makine öğrenmesi modellerinin performansı üzerindeki etkilerini anlamayı amaçlıyoruz. *K*-anonimliği ve l-diversity ve t-closeness gibi farklı varyasyonlarını sağlamanın etkilerini bir dizi sınıflandırıcı ve gerçek hayat veri kümesi üzerinde karşılaştırıyoruz. Karşılaştırmalarımızda, makine öğrenimi için özelleşmiş bir bilgi kaybı metriği kullanıyoruz. Ayrıca, bilgi kaybını en aza indiren ve *k*-anonimlik özelliğini uygulayan optimal genelleme hiyerarşi ağaçlarını oluşturabilen ve seçebilen otomatik bir genelleme ve bastırma çerçevesi sunuyoruz. Sonuçlarımız, her *k*-anonimlik varyasyonunun farklı bir gizlilik düzeyi sunduğunu ve anonimleştirme sürecinde farklı kısıtlamalar getirdiğini göstermektedir. Genel olarak, anonimleştirme sürecinde ne kadar fazla kısıtlamamız olursa, verilerde o kadar fazla gürültü alırız.

Ayrıca, kullanıcıların ham verilerini toplamadan veya paylaşmadan ML modellerinin merkezi olmayan bir şekilde eğitilmesine izin veren federe öğrenme isimli yeni bir başka yaklaşımı da araştırdık. *K*-anonimleştirilmiş verileri kullanmaya adapte olabilen, silolar arası federe bir öğrenme çerçevesi tasarladık. Veri anonimleştirme entegrasyonunun daha iyi gizlilik sağlarken minimum bilgi kaybı sağlayabileceğini ve her iki yaklaşımı tek bir çerçevede kullanmanın her iki yaklaşımın avantajlarından yararlanmamızı sağladığını gösteriyoruz.

# ABSTRACT

## COMPARISON OF THE EFFECTS OF DATA PRIVACY PRESERVING METHODS ON MACHINE LEARNING ALGORITHMS IN IOT

Maintaining data privacy is a crucial and rising concern for many organizations and individuals. To address the issue of privacy, many regulations are enforced, which have direct impacts on data-driven services research and development. Data anonymization is one way to deal with this issue, by removing personal identifiable information to abide by certain privacy regulations. However, the anonymization process by itself introduces a level of noise to the data.

In this study we aim to understand the effects of applying anonymization algorithms on the performance of the machine learning models. We compare the effects of enforcing *k*-anonymity and its different variations (known as l-diversity and t-closeness) on a number of classifiers and real-life datasets. In our comparisons, we utilize an information loss metric specialized for machine learning. Furthermore, we introduce an automatic generalization and suppression framework that can build and choose the optimal generalization hierarchy trees that minimize information loss and enforce the *k*-anonymity property. Our results show that each *k*-anonymity variation offers a different level of privacy and introduces different constraints on the anonymization process. In general, the more constraints we have on the anonymization process the more noise we get in the data.

We also investigated another recent approach, that is federated learning where it allows for training of ML models in a decentralized manner without collecting or sharing users' raw data. We designed a cross-silo federated learning framework that can adapt to use *k*-anonymized data. We show that integration of data anonymization can give minimal information loss while providing better privacy, and that utilizing both approaches in one framework does enable us to benefit from both approaches' advantages.

# SYMBOLS

| | |
|---|---|
| $\boldsymbol{H}$ | : Conditional Entropy |
| $\boldsymbol{I}$ | : Entropy coefficient |
| $\boldsymbol{S}$ | : Set of columns for a given dataset |
| $\boldsymbol{C_j}$ | : Set of unique values in a given column |
| $\boldsymbol{T}$ | : Set of unique values in target column |
| $\boldsymbol{\rho}$ | : Percentile Range Value |
| $\boldsymbol{f}$ | : Frequency of coupled values |
| $\boldsymbol{R}$ | : Set of Mondrian approximated records |
| $\boldsymbol{L_i}$ | : Set of rows in a record |
| $\boldsymbol{E_i}$ | : Subset of a record that satisfies *k* criterion |
| $\boldsymbol{A_i}$ | : Subset of records that don't satisfies *k* criterion |

# ABBREVIATIONS

| | |
|---|---|
| **ML** | : Machine Learning |
| **FL** | : Federated Learning |
| **DP** | : Data Privacy |
| **TFDS** | : TensorFlow data builder |
| **TP** | : True Positive |
| **TN** | : True Negative |
| **FP** | : False Positive |
| **FN** | : False Negative |
| **QI** | : Quasi Identifiers |
| **GDPR** | : General Data Protection Regulation |
| **KVKK** | : Personal Data Protection Authority |
| **VGH** | : Value Generalization Hierarchy |
| **DGH** | : Domain Generalization Hierarchy |

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

In this digital age, there is an unprecedented growth in the amount of data that is being collected, stored, and shared, with the utilization of new technologies integrated into our daily lives such as smartphones and IoT devices. Naturally a large portion of this data contains considerable amounts of personal information, e.g., identity, salaries, addresses, and health conditions. This data is continuously being collected and processed by the companies for a multitude of purposes, e.g., research-related and services such as personalized ads and recommendation systems. With this growth, we face tremendous challenges related to securing and keeping the data private. Here, the largest challenge is trying to preserve the usefulness of the data while only allowing access to it by authorized parties.

Given those challenges, many countries are concerned and started addressing the issue of individual privacy. Modern regulations and standards did appear, e.g., General Data Protection Regulation (GDPR) [1] in the EU and Personal Data Protection Authority (KVKK) [2] in Turkey. Such regulations are carefully revised and enforced to prevent the unauthorized collecting, processing, and sharing of data. Such regulations and laws are put to maintain personal information privacy, but they also limit the useability and accessibility of essential data for many organizations that depend on it while creating and providing their services. Therefore, data anonymization [3] is being heavily utilized to abide by those regulations while maintaining a degree of data usefulness.

Data anonymization is a process that ensures that individuals collected data can no longer be traced back to its original owners by removing or encrypting its identifiers. While there exist a multitude of different anonymization models, by far the most common one is *k*-anonymity [4], which ensures that data records with similar quasi-identifiers are grouped together, where each group contains at least *k*-1 samples that contain the exact same sequence of Quasi-identifiers values. To ensure this property, data manipulation techniques are used, such as generalization, suppression, and aggregation. However, such manipulations can introduce a significant information loss [5] that affect the performance of the machine learning (ML) models.

Although many studies related to anonymization exist, most of them investigate the anonymization process optimization, and very few investigate the effects of such methods on classifiers performance. We take a step further in this study and we investigate and compare the effects of *k*-anonymity with its different variations, namely l-diversity [6], and t-closeness [7] on a number of classifiers and four real-life datasets. We also provide and evaluate a fully automated and optimized method for generalization and suppression [8] of data. In addition, we utilize an entropy-based information loss metric to understand the effects of anonymization on data predictive performance. This is a novel and useful approach to predict the ML performance before training, while previous methods primarily used metrics that count the number of generalization levels done through anonymization.

Another approach was recently introduced by Google research group is federated learning (FL) for ML applications [35] where all the data processing happens at the user level and no raw data is collected or shared but instead the resulted models' weights will be shared with a central server. Both data anonymization and FL face a number of challenges and limitations that could potentially compromise privacy. In this study we also investigated the integration of *k*-anonymization in federated learning frameworks, more specifically in cross-silo federated learning framework [36]

. We provide a detailed explanations for the proposed framework where we believe it provides a step further towards protecting privacy. We test our framework on *k*-anonymized and non-anonymized data to properly compare between them. Our investigation shows that with proper usage of generalization in the anonymization process, usefulness of data can be preserved. Our framework results shows that the integration of *k*-anonymity protect better against privacy attacks with limited effects on models' performance due to anonymization.

## 1.1 Previous Studies

There is a significant amount of work that handles the issue of how to anonymize data [6-10]. There exist some studies on the integration of machine learning techniques into anonymization process [11-13]. Some efforts have been made on anonymization for specific datasets and IoT domains [14-16], also some papers worked on adapting general purpose and specialized information metrics to the anonymization process [6, 18]. We noticed a lack of work on the effects of anonymization on ML models performance and we came across few studies that handled this issue [17, 20].

Last et al. [18] introduced a new anonymization algorithm named NSVDist (Non-homogeneous generalization with Sensitive Value Distributions) with an information loss metric. In their study they worked with 8 datasets and 4 ML models, they also compare their algorithm to three other algorithms, Mondrian [21], PAIS [22] and SeqA [23]. In comparison to our work, we introduce an automatic generalization framework with specialized information loss metric for ML, we compare three *k*-anonymity extensions, and we use more classifiers to compare. This study shows the usage of Naive Bayes and SVM models, which are reported to perform badly in general for anonymized datasets by [19] and our study.

Slijepcevic et al. [19] compared 4 different algorithms for *k*-anonymity, Optimal Lattice Anonymization (OLA) [24], Mondrian [21], Top-Down Greedy Anonymization (TDG) [25], and k-nearest neighbor clustering-based (CB) algorithm [26]. They used 4 ML models and 4 datasets, and their work is extensive on studying the effects of *k*-anonymization on classification models. This study is close to our research topic, but to compare, in our work we explored different extensions of *k*-anonymity (namely *l*-diversity and *t*-closeness) and not just one. We also propose and use a novel automatic generalization framework that utilizes information metric specialized for ML.

Fung et al. [20] introduced a new method for anonymization that satisfies *k*-anonymity by Micro-aggregation based Classification Tree (MiCT). They tested their methods on two datasets, the standard Adult dataset, and German credit dataset. They trained and tested using two classifiers. Although their work is utilizing different technique (micro aggregation) to achieve anonymity, it is close to our study in the sense that both studies aim to investigate the effect of anonymization on the classifier performance. However, we tested over more *k*-anonymity extensions and more classifiers, and we believe this study approach should also be investigated with more classifiers and datasets with different distributions to understand the behavior of the proposed method.

Wimmer et al. [17] aimed to investigate the effects of *k*-anonymity on ML models performance. They used 6 different ML models and 3 different datasets, and they extensively investigated the ML models parameters and reported accuracy, F1 score, and confusion matrix results for each model. However, they did not mention the *k*-anonymity algorithm that is used nor did they talk about how the generalization and suppression are utilized. In comparison to our work, we extensively work and report on both *k*-anonymity algorithm design and its utilization of generalization and suppression over a number of *k*-anonymity extensions. Also, in addition to reporting performance of the models in terms of the same metrics they used, we used an information loss metric in the generalization framework specialized for ML classifiers.

## 1.2 Data Privacy

Data privacy refers to the notion that each individual should be able to determine how and when their personal information or any data related to them is shared or processed to other parties. The increase in internet usage led to the rise of data collection across all platforms for various purposes. This increase naturally led to an increased attention towards data privacy. Unauthorized usage of personal information would lead to many undesired results such as:

- Entities that gather personal information may sell them to third parties without user approval.
- Private data can be used to defraud or harass users.
- Personal information can be used to know sensitive information about people and to monitor their activity.

Such exploitation of personal information can affect individuals and compromise entire organizations. Due to this regulation and laws appeared, many individuals became more conscious about this. Many studies handled this topic, but more work is still needed since many aspects of those concerns are still not addressed.

## 1.3 Anonymization

### 1.3.1 *k*-anonymity

*k*-anonymity is a data anonymization technique that ensures that data samples cannot be re-identified while maintaining data usefulness. Assume we have a dataset with *n* number of records, each record contains *d* number of attributes. These attributes contain two types of values, quasi-identifiers and a sensitive value. A group of quasi-identifiers combine together into super-identifiers, and these super-identifiers can sometimes directly trace back the data record to its identifier. For example, a combination of the attributes {Age, Gender, ZIP, Race} may directly be linked to a person. For the dataset to have the *k*-anonymity property, each quasi-identifier tuple should occur at least k times, hence each record should be indistinguishable from *k*-1 other records in the dataset. To ensure the property of *k*-anonymity, generalization and suppression methods are often used. More details will be given in the later sections.

### 1.3.2 *k*-anonymity Variations

*k*-anonymity is achieved by guaranteeing that in each data group, each record should be indistinguishable from *k*-1 other records, this protect the subject identity in the sense that even if an adversary were to know all the quasi values of a subject, the adversary will only be able to find out which data group the subject belongs to but nothing more.

In the next subsections we will address a few issues behind this logic and describe the proposed solutions.

***l*-diversity:** A major problem to the standard *k*-anonymity is that in a data group, all of the records may contain the same sensitive attribute, an adversary who found out that a subject is in a specific group will be able to know for certain the value of the sensitive attribute [9]. To deal with this a variation of *k*-anonymity called *l*-diversity [6] was proposed, it adds an extra criteria to *k*-anonymity which dictate that in each data group should at least include I number of sensitive attribute values, with this in mind even if an adversary managed to find out the data group of the subject, he won't be able to find out the sensitive attribute for certain for a given subject.

**t-closeness:** An issue with *l*-diversity variation that we discussed earlier is that an adversary could use probabilistic reasoning to find out the sensitive attribute for a subject, if in a data group 4 out of 5 subjects have the same sensitive attribute an adversary could have a degree of certainty that the intended subject belongs to most of the subjects with the same sensitive attribute [9]. t-closeness variation [7] takes a step further and adds another criteria for the standard *k*-anonymity, which dictates that the statistical distribution of the sensitive attributes in the entire dataset should be reasonably close to its distribution in each data record, which is measured with Kullbac*k*-Leibler divergence.

### 1.3.3 Attacks on anonymization

**Homogeneity [34]-[46]:** An attacker might have previous knowledge about a subject such as Gender, Address, Age, wor*k*-field from a multitude of methods such as comparing different published data records from different organizations or even living next to the subject and gaining some information through social interactions. A problem with standard *k*-anonymity is that in some data groups that have been grouped together in the anonymization process may contain the same sensitive values. Assuming the attack had access to the anonymized and published data records for an organization the subject belongs to (for example hospitals or schools), the attacker can use a combination of the information he knows about the subject such as Age, Address, Ethnicity, and Gender, and filter the records based on those quasi-identifiers. By doing this the attacker get a set of records with identical quasi-identifiers and very possibly all share the same sensitive value, then the attacker will know with absolute certainty the sensitive value of the subject.

**Background Knowledge Attack [34]-[46]:** Background attacks and Homogeneity attacks are similar in assuming the attacker have access to some quasi-identifiers about the subject. However, they differ in that Background attacks don't assume the grouped data records share the same sensitive value, instead the subject may know that the subject is statically more likely to have one of the sensitive values in the data records. An example of this is that the subject medical data is published and the attacker already knows his gender and the sensitive values contains a value (for example a disease) that occur more frequently in women than men. Such knowledge can easily enable the attacker to filter the data records and gain exact knowledge of the sensitive records.

## 1.4 Federated Learning

Federated learning [35] in simple terms is a decentralized approach to train machine learning models, where all the training happens at the user level in edge devices. The data itself is not transferred out of the device and thus reducing data privacy risks that may occur during the transfer or in any stage at the data centers processing or training. In the base version of federated learning a central server exists with a global model. At each edge device that will be used for training, there exist a local model. The local models train the data from the edge devices and transfer the trained local model weights to the central server and continues to update the global model with all the devices received weights. In the case that the edge devices need to use the fully trained global model it can be shared with the devices, no raw data will be shared with devices or servers. This approach was introduced in 2016 by google, although since that time many enhancements have been studied and applied to address multiple challenges related to data privacy and making Federated learning more efficient. The core structural theory is still the same, and we will describe it more in detail below:

- A set of Edge devices (clients) are chosen to carry out the local models training. The selection criteria vary, there are multiple algorithms and techniques that are being used to select the best available clients.
- Global model weights exchange, in the case that clients need to start training their local models starting from a previous session that was already aggregated to the current Global model. We can think of this as a starting point.
- Local models training on Clients' devices. Training the local models with the data each device has locally or collected locally and related to the task the global model is aimed for.
- Local Models exchange their newly generated weights after training with the global model. Next the global model will be updated from those weights.
- The process can be repeated until reaching the desired goal has been achieved.

## 1.4.1 Types of Federated Learning

**Types related to Data distribution:**

**Horizontal FL** [37],[38]: This type is used when each client's device has the same feature set in the data that will be used for training, but with different records. An example of Horizontal FL is such that multiple hospitals collect the same exact type of medical data, but each data record represents a different subject.

**Vertical FL** [37],[38]: This type is used when each client device has a different feature set of data that will be used in training. An example of Vertical FL is such that hospitals have a feature set of data for a number of subjects related to their medical history and an outside laboratory has another set of features of data related to blood tests results for the same subjects.

### Types related to Clients Selection:

**Cross-silo FL** [36], [38]-[40]: This type of federated learning is used when there is a limited number of clients but are available for all training rounds, the data distribution can be horizontal or vertical and this type of FL is usually used by Hospitals and organizations.

**Cross-Device FL** [36], [38]: This type of federated learning is used when there is a large number of clients but not all are reliable, selection algorithms and techniques are heavily utilized in this type, other issues also may raise such as client's interruption of training, thus this type is usually under careful monitoring during the training process.

### Fully Decentralized Federated Learning

In normal federated learning, a central server is still needed to handle all the weights and contributions exchange from a wide set of clients and edge devices, so naturally, such a central server is the single most important component in the FL. Potentially the weak point of this approach that if it crashes or fails at any point will have serious consequences.

A solution to this issue is the peer-to-peer distributed learning [36], [41], where there is no central server and the state of global model no longer exists. The way it works is that each client has a small set of neighboring clients that it will be communicating with, each training round is considered a successful when each client provides a local update in the training process and exchange them with their neighboring clients. Through this process the clients end up with roughly the same local model weights through the whole network.

## 1.4.2 Attacks on Federated Learning

**Privacy and security Related Attacks**

**Membership Inference Attacks**: ML and FL applications are vulnerable to Membership Inference Attacks [42] which make it possible to detect the data records used in the training process where the attacker doesn't have access to the training data and in many cases, not even the models' parameters or weights. Those type of attacks exploits the fact that ML and FL Models behave differently on data records that it has seen before (in the training process) and other records that it has never seen before. When the attacker gains those observations, he can determine which data records were a part of the training process and which were not. A possible scenario is that if an app is using FL in a service publicly provided an attacker could use the service and provide a data record of his own and then observe the service results (FL model output) and gain insightful info about the model and data records. Many solutions were proposed to deal with those types of attacks, such as data encryption and training on encrypted data, data anonymization, and differential privacy methods. While the first one adds a computational overhead and is very time consuming especially when we need to train a large amount of data, the latter two introduce noise in the data that could affect the models' performance and considered a compromise between usability and privacy.

**Data and Models Poisoning Attacks:** Data poisoning attacks [38] involve polluting and adding data records to the training data in one or more client devices with the aim to corrupt or influence the model behavior. Then those trained local models proceed to the central server to update the global mode. In the case that the attacker aimed to influence the training process and model outputs this attack belongs to the targeted poisoning attacks group, if the attacker only means to introduce false data to deem the model prediction results false, it belongs to the non-targeted attacks group. Those attacks are dependent on whether the models rely on clients' data for training. Models poisoning attacks [38] are very similar to data poisoning attacks but the difference is that instead of corrupting the training data, it aims to corrupt the local models' parameters to introduce errors that will be updated to the global model. There is no easy way to deal with such attacks, especially if it was detected after the global model has been updated, in this case, a comprehensive analysis of local model updates would take place and would take much time and resources to identify corrupted updates especially when an attacker poison data or models within different cycles of training. To prevent those attacks, detection of corrupted clients' devices should take place before using their updates. This could be achieved through clients' selection strategies and algorithms, clients' validation, and anomaly detection.

**Training Related drawbacks**

**Connections and Communication:** FL by design requires a carefully optimized communication strategy since all the clients will be sharing their local model's weights with the central server for it to update the global model. Client devices' connection usually varies in-network transfer rates and are almost always slower than that of central servers' connection rates. This can add huge bottlenecks and a serious and expensive challenge for FL. The solution to this could be addressed by minimizing the communication bandwidth for local models' updates, due to its importance of this specific issue it naturally gained a lot of attention and has been a topic of interests for many studies.

## 1.5 Motivation Behind This Work

Data privacy is a rising concern for many individuals and organizations alike. Naturally regulations to protect data privacy appeared and are imposed in many countries. Such regulation provides a challenge for organizations ability to collect and use data. To face those challenges, data anonymization appeared, and many data driven organization adapted this approach. Another recent approach appeared is FL where it allows for training of ML models in a decentralized manner without collecting or sharing users' raw data. In this study we designed a cross-silo federated learning framework that can adapt to use *k*-anonymized data. We show that integration of data anonymization can give minimum information loss while providing better privacy, and that utilizing both approaches in one framework does enable us to benefit from both approaches' advantages. This work topic is done based on the fact that data privacy despite the new advances, can still be compromised and we need to continue exploring options to prevent that. Our approach of data anonymization will guarantee the privacy of data, and the optimal use of resources while processing. This study is designed to assess the hypothesis that data privacy methods' drawbacks on ML algorithms' performance in terms of computation or quality of output can be minimized by using the proper data privacy techniques and choice of machine learning algorithms in IoT.

## 1.6 Thesis Structure

The thesis is structured as follows:

- The First part of this Thesis handles the logic behind anonymization, and our anonymization framework setup. We address several key points and important components of *k*-anonymization and other variations that we will use through this work.
- In Chapter 2 we describe an information loss metric that we use in this work, an entropy coefficient based, used to estimate the information loss after introducing anonymization to the data.
- In Chapter 3 we discuss the ML models and classifiers we worked with, we describe the theoretical logic behind each model, we also talk about the evaluation metrics we used in this study in detail.
- In Chapter 4 we talked about our anonymization framework final design where we combine our *k*-anonymization, l-diversity and t-closeness variations and describe the way we introduce them to the data, and how we use them to train the models and evaluate them,
- In Chapter 5 we report our Anonymization results, our information loss results for each dataset and anonymization variation. We also show our classifiers results with the metrics described earlier, we end this chapter with few remarks on both the models and anonymization performance.

- In Chapter 6, we talk about our FL Framework, where we talk about each aspect in detail, we discuss data distribution among clients, models weights aggregation and the integration of anonymization into federated learning to achieve maximum privacy.
- In Chapter 7 we report our FL framework results in terms of information loss after the anonymization process and the classification results on the final version of the global model that used a number of clients in the training process.
- In Chapter 8, we conclude this work by describing our work and final findings and we offer future directions for this field and study.

# 2. *K*-ANONYMITY SETUP

## 2.1 Mondrian *k*-anonymity

Mondrian [21] is a top-down greedy approximation algorithm for strict multidimensional partitioning that is used to achieve \$k\$-anonymity. The Mondrian algorithm works as follows. Let us assume that we have final partitions set *F* and a working set *S* that initially contains the entire dataset as a partition. Next, we calculate and sort the relative spans of the dimensions (columns) in the partition. For each dimension we split the partition along the columns space based on a split point, which is the median in this case. After that, we check if the new partition satisfies the *k*-anonymity criteria. If it is so, we add the resulted partitions to the set *S* and iterate through the same previous process on the new partitions. If no valid split was achieved and no regions left in the columns space, we add the partition to set *F*. When all of this is finished, we end up with an approximate *k*-anonymized dataset.

In this work we explore the Mondrian algorithm capability in achieving *k*-anonymity and its other extensions, *l*-diversity and *t*-closeness. In the implementation of the Mondrian algorithm, we also allowed the choosing of dimensions that the partitioning should be based on. This is important when we have non quasi-identifiers and we do not wish to include them in the anonymization process. We also integrated an optimization step to the Generalization and Suppression part that allows for a better reduction in information loss.

## 2.2. Loss metric

In order to assess the quality of an anonymization approach (including the proposed generalization and suppression methods) and measure how information loss effects ML models performance, we ought to measure the data that is lost due to the anonymization process. We decided to use the entropy coefficient [5], [10], which is a measure of association that tells us how much of X can we predict using Y, where X and Y are two random variables.

To assess the quality of our $k$-anonymity approach (including the proposed generalization and suppression methods) and measure how information loss effects ML models performance, we ought to measure the data that is lost due to the anonymization process.

We start by calculating the conditional entropy, where P(x,y):

$$H(X \mid Y) = -\sum_{x,y} P_{x,y}^{(x|y)}(x \mid y) \log P_{x|y}^{(x|y)}$$

Then we proceed to calculate the Entropy coefficient:

$$U(X \mid Y) = \frac{H(X) - H(X \mid Y)}{H(X)} = \frac{\mathrm{I(X;Y)}}{\mathrm{H(X)}}$$

We will end up with an association value ranging between 0 and 1, where 1 means total association (no data loss) and 0 means no association (complete data loss).

Since we aim to use our datasets in ML models, we focused on minimizing the data loss while maintaining privacy as much as possible. We will mainly use this metric to measure the association between Domain Variables and Target Variable.

## 2.3 Optimization Step

We introduce an optimization step in the anonymization process that tries to better approximate the Mondrian algorithm outputs within each group of split records. Let us assume we have a Records set *R* that contains all Mondrian split records. Each record $i \in R$ has a set of rows $L_i$, and each row has a sequence of attribute values (quasi-identifiers).
For each set $L_i$ we do the following:

1. Check whether there are at least *k* rows that have the exact same sequence of values, hence it satisfies *k* criterion. We add these rows into subset $E_i \in L_i$.

2. Check for the rows that have an approximate match for the sequence of values that satisfies $k$ criterion. We add these rows into subset $A_i \in L_i$.

We end up with subset $E_i$ which requires no further processing, since they satisfy the $k$ criterion and are an exact match, and subset $A_i$ which will be forwarded to the Generalization and Suppression step.

## 2.4 Generalization and Suppression

Since Mondrian is a greedy approximation algorithm that aims to partition datasets into $k$-anonymized records, we must use Generalization and Suppression methods to enforce $k$-anonymity criterion on records which the algorithm could not fully enforce. Generalization is the process of replacing the individual values of each domain with a less specific broader value, e.g. if we had a dataset with column Age that has the values 18, 19, 21, 25, 31, and 38 we can generalize them to <20, 20-30, and 30-40. While suppression is the process of partially or fully replacing certain domain values with an asterisk '*' to indicate the complete loss of that value in a given row,e.g. if we had a dataset with column ZIP that has the values 114554 , 114553, and 114543 we can suppress them partially to 1145**, 1145**, and 1145**. Table 2.1 shows generalization in Age Column, and Suppression in Nationality column.

**Table 2.1:** Data records that has been anonymized with generalization and suppression

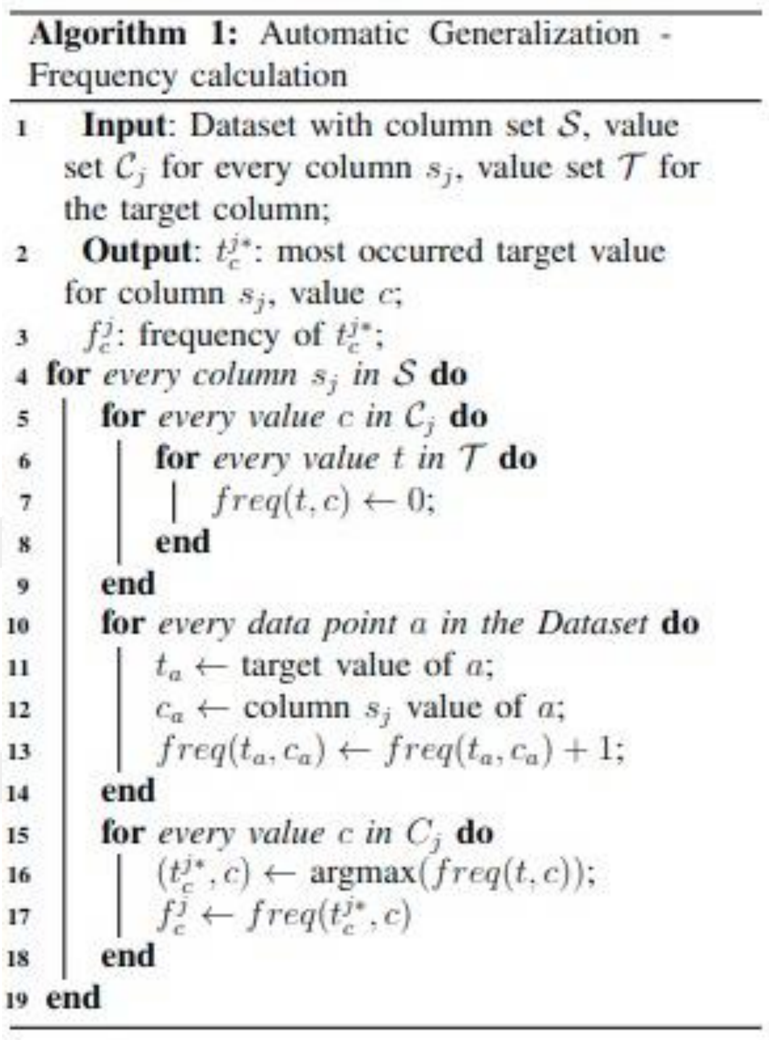
| # | ZIP | Age | Nationality | Disease |
|---|---|---|---|---|
| 1 | 130** | < 25 | * | Flu |
| 2 | 130** | < 25 | * | Heart Disease |
| 3 | 130** | < 25 | * | Cancer |

In this study, we use generalization including suppression strategy. Both generalization and suppression methods are used to enforce *k*-anonymity criterion. While enforcing anonymity we aim to reduce data distortion. To achieve this we adopt both minimum suppression policy and minimum relative distance policy defined in [4]. In other words, our generalization approach prefers the generalization that suppresses less and  minimizes the total number of relative steps with respect to the height of the hierarchy.

In the next subsection, we propose a new Generalization method that can automatically generate rules for both categorical and numerical data and an additional optimization step will be integrated as described in the previous subsection to guarantee the strategy goals.

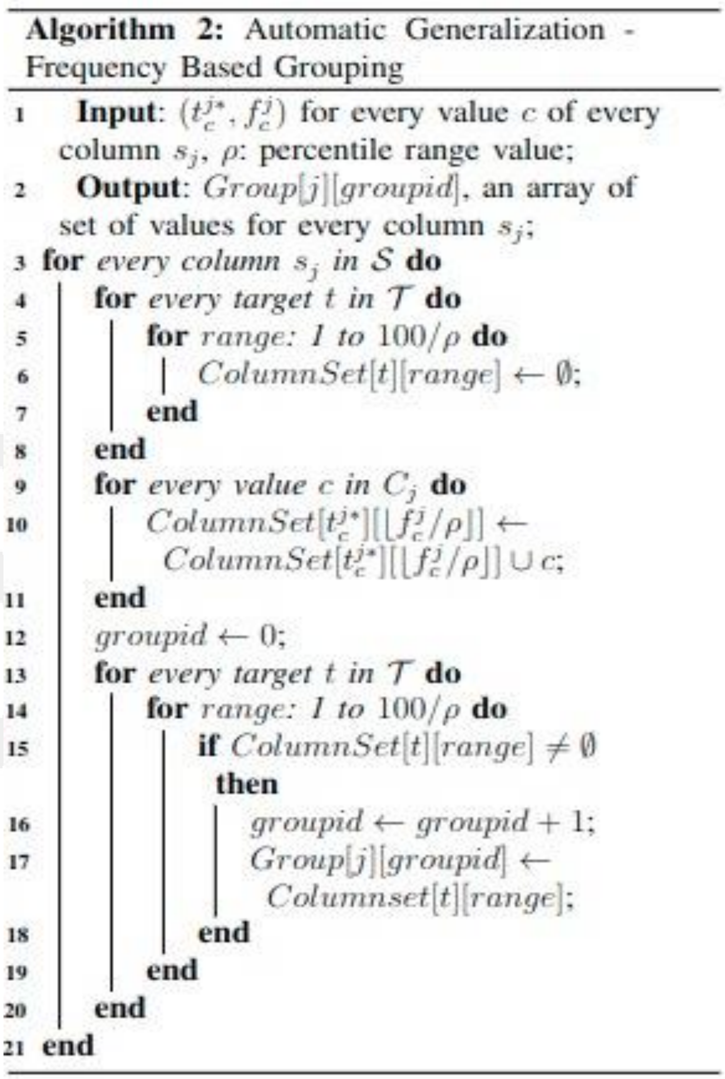
## 2.5 Automatic Generalization

Generalization is a very powerful method to enforce *k* criterion, and in most cases declarative generalization is utilized, which involves manually deciding the records generalization hierarchies. Manually built hierarchies are more associated with outer-bound logic, it can sometimes distort or bias the data due to outliers often being excluded entirely, also it completely ignores the correlation of the records values to the target attributes. We propose an automatic aggregation-based generalization hierarchy, which is based on the records value's frequencies in correlation to the target attributes. The proposed automatic generalization algorithm consists of two main steps. The first step is frequency calculation per column value-target value pairs, as illustrated in Figure 2.2. The second step is frequency-based grouping which is illustrated in Figure 2.3.



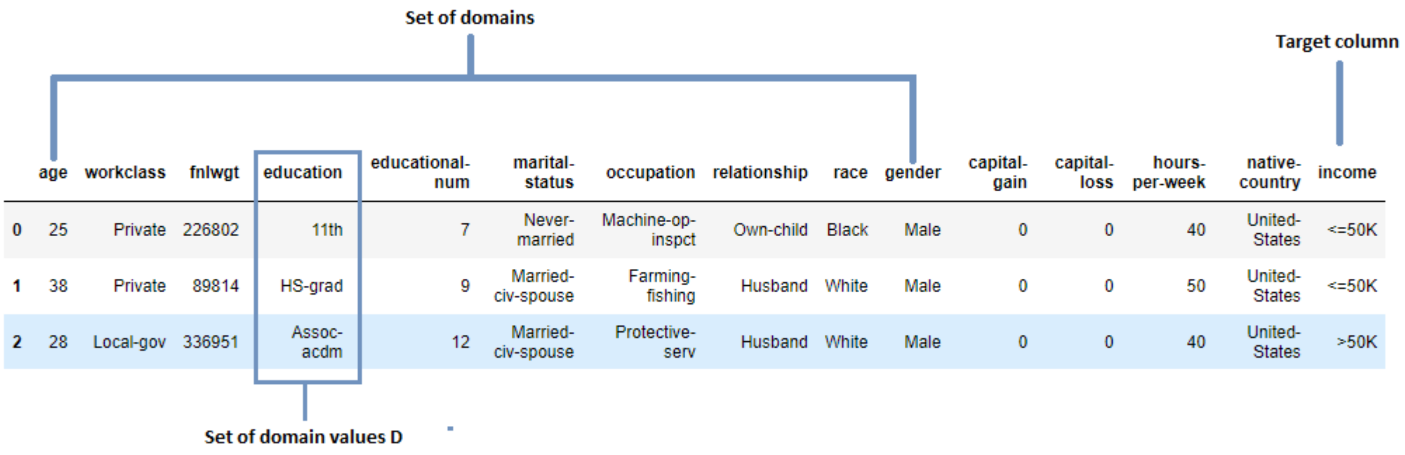
```
Algorithm 1: Automatic Generalization - Frequency calculation

Input: Dataset with column set $\mathcal{S}$, value set $\mathcal{C}_j$ for every column $s_j$, value set $\mathcal{T}$ for the target column;
Output: $t_c^{j*}$: most occurred target value for column $s_j$, value $c$;
$f_c^j$: frequency of $t_c^{j*}$;
for every column $s_j$ in $\mathcal{S}$ do
    for every value $c$ in $\mathcal{C}_j$ do
        for every value $t$ in $\mathcal{T}$ do
            $freq(t, c) \leftarrow 0$;
        end
    end
    for every data point $a$ in the Dataset do
        $t_a \leftarrow$ target value of $a$;
        $c_a \leftarrow$ column $s_j$ value of $a$;
        $freq(t_a, c_a) \leftarrow freq(t_a, c_a) + 1$;
    end
    for every value $c$ in $C_j$ do
        $(t_c^{j*}, c) \leftarrow$ argmax$(freq(t, c))$;
        $f_c^j \leftarrow freq(t_c^{j*}, c)$
    end
end
```

**Figure 2.1:** Automatic Generalization Algorithm 1 – Frequency Calculations



```
Algorithm 2: Automatic Generalization - Frequency Based Grouping
Input: $(t_c^{j*}, f_c^j)$ for every value $c$ of every column $s_j$, $\rho$: percentile range value;
Output: $Group[j][groupid]$, an array of set of values for every column $s_j$;
for every column $s_j$ in $\mathcal{S}$ do
    for every target $t$ in $\mathcal{T}$ do
        for range: 1 to $100/\rho$ do
            $ColumnSet[t][range] \leftarrow \emptyset$;
        end
    end
    for every value $c$ in $C_j$ do
        $ColumnSet[t_c^{j*}][\lfloor f_c^j/\rho \rfloor] \leftarrow ColumnSet[t_c^{j*}][\lfloor f_c^j/\rho \rfloor] \cup c$;
    end
    $groupid \leftarrow 0$;
    for every target $t$ in $\mathcal{T}$ do
        for range: 1 to $100/\rho$ do
            if $ColumnSet[t][range] \neq \emptyset$ then
                $groupid \leftarrow groupid + 1$;
                $Group[j][groupid] \leftarrow Columnset[t][range]$;
            end
        end
    end
end
```

**Figure 2.2:** Automatic Generalization Algorithm 2 – Frequency Based Grouping



|  | age | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender | capital-gain | capital-loss | hours-per-week | native-country | income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 25 | Private | 226802 | 11th | 7 | Never-married | Machine-op-inspct | Own-child | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 1 | 38 | Private | 89814 | HS-grad | 9 | Married-civ-spouse | Farming-fishing | Husband | White | Male | 0 | 0 | 50 | United-States | <=50K |
| 2 | 28 | Local-gov | 336951 | Assoc-acdm | 12 | Married-civ-spouse | Protective-serv | Husband | White | Male | 0 | 0 | 40 | United-States | >50K |

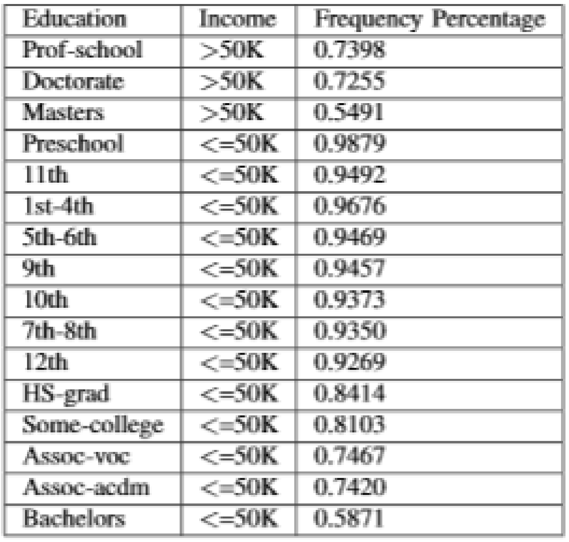
**Figure 2.3:** Sample data records that are used in automatic generalization.

Assume we have a data-set with a set of $\boldsymbol{C}$ columns $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_c}\} \in \boldsymbol{S}$, each column $\boldsymbol{s_j}$ has a set of $n_j$ unique values $\left\{c_1^j, c_2^j, \dots, c_{n_j}^j\right\} \in C_j$. Also we have a set of $\boldsymbol{m}$ unique values $\{t_1, t_2, \dots, t_m\} \in T$ in the Target column.

For every column $s_j \in S$, we start by coupling each value in $C_j$ with each value in $\boldsymbol{T}$, i.e., we couple all the pairs $\left(c_i^j, t_k\right)$, $\forall i \in \{1,2, \dots, n_j\}$, $\forall k \in \{1,2, \dots, m\}$. Next, we calculate the frequency of the coupled pairs over the entire dataset (i.e. how many times the two values appeared in the same row). For each column $s_j$ and value $c_i^j$ we pick the couple $\left(c_i^j, t_k\right)$ with the highest frequency. This highest frequency for value $c_i^j$ is denoted by $f_i^j$ and the corresponding target value is denoted by $t_i^{j*}$.

Let us consider the "education" column in the adult data set. There are two values in the target column (income). For every value pair in the Education column and the target column, we calculate the frequencies and the couples with highest frequencies are illustrated in Table 2.2.

**Table 2.2:** Most frequent education column unique values coupled with target columns in the dataset.



| Education | Income | Frequency Percentage |
|---|---|---|
| Prof-school | >50K | 0.7398 |
| Doctorate | >50K | 0.7255 |
| Masters | >50K | 0.5491 |
| Preschool | <=50K | 0.9879 |
| 11th | <=50K | 0.9492 |
| 1st-4th | <=50K | 0.9676 |
| 5th-6th | <=50K | 0.9469 |
| 9th | <=50K | 0.9457 |
| 10th | <=50K | 0.9373 |
| 7th-8th | <=50K | 0.9350 |
| 12th | <=50K | 0.9269 |
| HS-grad | <=50K | 0.8414 |
| Some-college | <=50K | 0.8103 |
| Assoc-voc | <=50K | 0.7467 |
| Assoc-acdm | <=50K | 0.7420 |
| Bachelors | <=50K | 0.5871 |

Next, we continue with the second step, frequency-based grouping. Initially, we set a percentile range value $\rho$, such that 100/ $\rho$ is an integer. Then we divide the whole range between 0 and 100 to 100/$\rho$ percentile ranges with equivalent sizes, i.e. (0 – $\rho$,  $\rho$ – 2$\rho$, ... , 100 - $\rho$  – 100). As an example, if $\rho = 10$, then there would be 10 percentile ranges, and if $\rho = 20$, there would be 5 ranges. Each percentile range corresponds to a different group. Then we group every column value $c_i^j$ according to its mostly encountered target value and the corresponding percentile range of the frequency $f_i^j$. Then we generalize the column values into these representative groups.

Now, let us revisit the "education" column in the adult dataset shown in Figure 2.3, and set the $\rho$  value to 10. The frequency values illustrated in Table 2.2 suggest that the education column values would be included in the following six groups with the given percentile ranges.

- Group 1 = (HS-grad, some-college)

- Group 2 = (Assoc-acdm, Assoc-voc)
- Group 3 = (Prof-school, Doctorate)
- Group 4 = (Bachelor)
- Group 5 = (Master)
- Group 6 = (11th, 10th, 7th-8th, 5th-6th, 9th, 12th, 1th-4th, preschool)

We can notice that this method was able to automatically generalize 16 values into 6 representative groups, which is pretty impressive. It is evident that the smaller the $\rho$ value is, the more similar the values in one group will be, and with very low $\rho$ value (such as 2), it would mostly result in no generalization at all. On the other hand, the larger the condition value, the less groups we will have and more generalization will happen. A natural question at this point is how to decide on the best $\rho$ value. The answer here is to try different $\rho$ values and create a hierarchy for each value. Then, during anonymization process we apply the generalization trees to the data records and measure the information loss by our proposed metric for each tree.

This approach with the given percentile range condition will output the following groups: We can notice that this method was able to automatically generalize 16 values into 6 representative groups, which is impressive. It is evident that the smaller the $\rho$ value is, the more similar the values in one group will be, and with very low $\rho$ value (such as 2), it would mostly result in no generalization at all. On the other hand, the larger the condition value, the less groups we will have, and more generalization will happen.

A natural question at this point is how to decide on the best $\rho$ value. The answer here is to try different $\rho$ values and create a hierarchy for each value. Then, during anonymization process we apply the generalization trees to the data records and measure the information loss by our proposed metric for each tree. We are trying to minimize information loss by choosing the best generalization tree for each column, i.e. choosing the $\rho$ value that offers the least loss and satisfies k criterion. Note that we may come up with a different $\rho$ value for different columns.

Although in this work our focus was on Value Generalization Hierarchy (VGH), we also provide the choice for Domain Generalization Hierarchy (DGH), which can be used for domains like ZIP numbers.

Time complexity of the Algorithm in Figure 2.2 is O(*cn*) where n is the number of data points in the dataset. Since the number of columns (*c*) is typically a small number, this complexity can be considered as linear. Time complexity of Algorithm in Figure 2.3 is even lower, because it is a function of number of columns, number of values per column, number of target values and number of range values, all are independent from the size of the dataset.

# 3. MACHINE LEARNING

## 3.1 Machine Learning Models

In this study, we worked with a number of ML models in order to understand the effects of the anonymization process on Machine Learning.

**Random Forest:** It is an ensemble technique that utilizes decision trees and bagging techniques [27], it starts by creating a set of decision trees from a randomly selected subset of the data (training set) and collects its results to take part in determining the final result.

**Extremely Randomized Trees (Extra Trees) Classifier:** It belongs to the ensemble learning techniques that aggregate various decision trees results [28]. It is very similar to random forest and only differs in the method it constructs its decision trees.

**Support Vector Machines (SVM):** It is a well-known supervised learning algorithm that is mainly used for classification. It works by finding the optimal decision boundaries (hyper-plane) that best separate the data to help in decision making [29].

**Gradient Boosting:** It is an ensemble technique that is known for utilizing boosting in the training process. It works by building a number of models where the first model is built with the training data, then the proceeding models are trained with its predecessor's residual error [30].

**Artificial Neural Network (ANN):** In this work we used a multi-layer perception classification algorithm that utilized feed forward and back propagation in the training process [31]. We also used the Grid Search algorithm for parameters optimization.

In addition to these algorithms, we also study other ML models such as **K-Nearest Neigbour (KNN)**, **Logistic Regression, Gaussian Naive Bayes (GaussianNB)** and **Stochastic Gradient Descent (SGD) classifier.**

## 3.2 Machine Learning Evaluation Metrics

In the final stages, to evaluate our results we compare different performance metrics. We obtain Classification Accuracy, Precision Score, Recall and F1 scores.

**Classification Accuracy:** Accuracy is a metric used to evaluate classification models performance, where it refers to the number of predictions the model got right. Simply it can be expressed as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Number\ of\ Total\ Predictions}$$

Also, can simply expressed as: $\frac{(TP+TN)}{(TP+TN+FP+FN)}$

**Precision Score:** it tries to answer the following question, what proportion of positive identifications was actually correct? While considering TP, TN, FP, FN logic, we can express it as:

$$Precision = \frac{TP}{TP + FP}$$

**Recall:** it tries to answer this question, what proportion of actual positives was identified correctly? And can be expressed as:

$$Recall = \frac{TP}{TP + FN}$$

**F1 Score:** is a metric that used to find balance between Precision and Recall values, and can be expressed as:

$$F1 = 2\ X\ \frac{precision\ *\ Recall}{precision\ +\ Recall}$$

# 4. ANONYMIZATION FRAMEWORK DESIGN

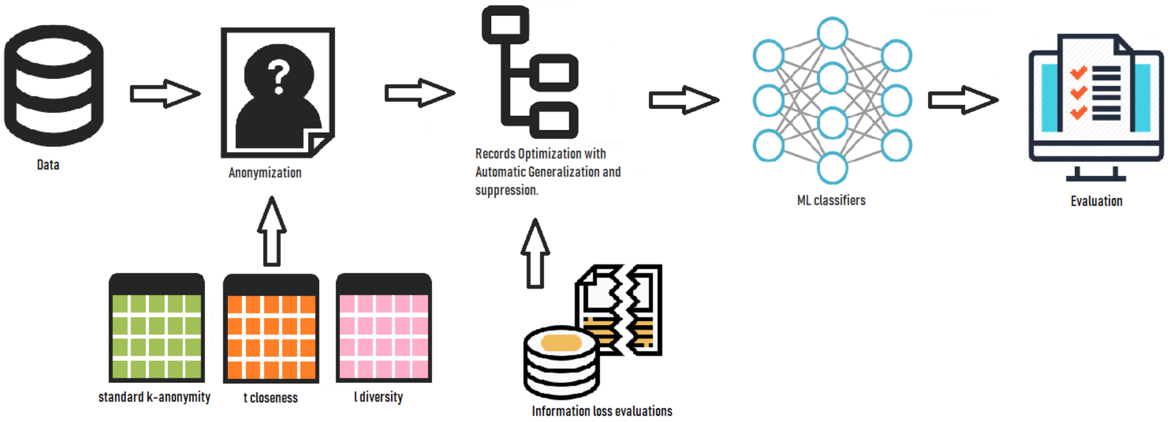

Data
Anonymization
Records Optimization with Automatic Generalization and suppression.
ML classifiers
Evaluation
standard k-anonymity
t closeness
l diversity
Information loss evaluations

**Figure 4.1:** Anonymization framework

## 4.1 Overview

Figure 4.1 illustrates an overview of the anonymization framework and its evaluation, which can be described in detail as follows.

- After accessing the data, quasi-identifiers and sensitive attributes are identified. These are stored in the main file along with the selected *k* value and the *k*-anonymity extension to be used.
- The anonymized records are generated using the Mondrian algorithm and all generalization trees are generated for each column in a fully automated way.
- The optimization step starts and finds the best generalization hierarchy for each column, and the best one is picked according to the presented information loss metric. Suppression is implemented if needed in this step. Automation ends after this step.
- The resulting records are pre-processed and prepared for Models Training.
- Training starts and the evaluation metrics are received.

We implemented all the steps in Python 3.7. We have a core services directory and automated generalization and suppression services directory, where the latter directory is flexible and enables us to easily integrate any new suppression or generalization techniques by only providing similar inputs and outputs structure. We also provide the option for multiprocessing during the optimization process, all of this is controlled by a main script in the root directory which can be easily executed from the command line.

## 4.2 Datasets

**Adult dataset:** This dataset contains 48843 data points, with 15 columns. It is derived from the 1994 USA census database. We used age, workclass, education, education-num, material status, occupation, relationship, race, gender, and native country columns as the quasi-identifiers and income column as the target column.

**California Housing Dataset:** This dataset contains 20640 data point. It is derived from the USA 1990 census data. We used latitude, longitude, housing median age, median house value, and median income columns as quasi-identifiers and ocean proximity as the target column.

**The Mammographic Mass Dataset**: This dataset contains 830 data points (after cleaning) with 6 columns. It is reported from breast cancer screening systems and contains BI-RADS assessments and used to predict the severity of a mammographic mass lesion. We used all of the columns as quasi-identifiers (age, shape, bi-rads assessment, density and margin) and severity as the target column.

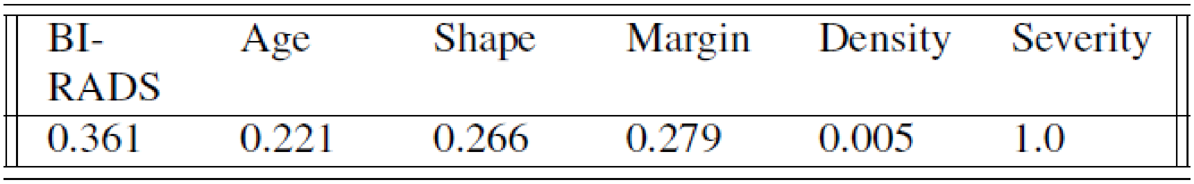
# 5. MACHINE LEARNING AND ANONYMIZATION RESULTS

We applied the anonymization techniques discussed in the previous sections and summarized in Figure 4.1. In this section we discuss the results obtained for each of the datasets separately. We start by describing the data loss obtained when we apply *k*-anonymity and its extensions, compared to the original datasets. Then we interpret the ML results when the anonymized datasets are used.

## 5.1 Anonymization and information loss Results

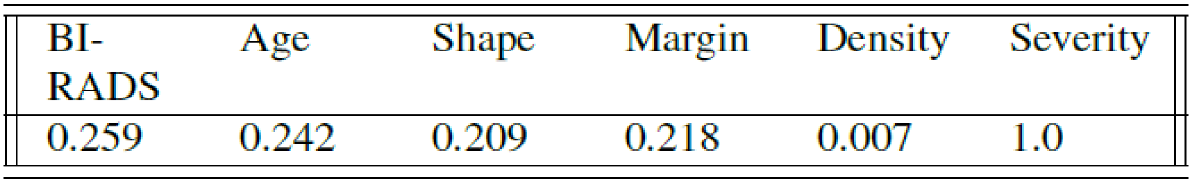
We calculate total information loss by calculating the difference between two datasets, the anonymized one and the original one. We calculate the total columns conditional entropy to the target column in both datasets and measure how similar the anonymized dataset results are to the original dataset.

**Table 5.1:** MGM dataset: Entropy results with target column 1Severity” (original dataset)



| BI-RADS | Age | Shape | Margin | Density | Severity |
|---|---|---|---|---|---|
| 0.361 | 0.221 | 0.266 | 0.279 | 0.005 | 1.0 |

**Table 5.2:** MGM dataset: Entropy results with target column “Severity” (standard *k*-anonymity)



| BI-RADS | Age | Shape | Margin | Density | Severity |
|---|---|---|---|---|---|
| 0.259 | 0.242 | 0.209 | 0.218 | 0.007 | 1.0 |

For example, Table 5.1 shows the association values (entropy coefficient) of each column of original MGM dataset to the target column, and Table 5.2 shows the association values for the MGM dataset after application of standard *k*-anonymity. When we compare the average association results, we observe that the anonymization process managed to preserve 82.6% of the data and the information loss is 17.4%.

In general, the expected theoretical performance should show that the standard version would perform the best, and the *l*-diversity should be the second best while *t*-closeness would cause highest information loss. The reason is that each extension adds more restrictions to the anonymization groups splitting process which results for each groups QI values to be more diverse than it would be in the standard *k*-anonymity. Note that each extension have different level of privacy and *t*-closeness is the most private one. Hence, the choice of extension is a compromise between privacy and data usefulness.

To better understand the behavior of *k*-anonymity and its extensions, we applied anonymization on all the datasets with different conditions (for eg, k=10, l=3, t=0.4). Then we proceeded to calculate information loss for each anonymized dataset and reported the results in Tables 5.3, 5.4, 5.5 and Figures 5.1, 5.6, 5.11.

In Figures 5.1, 5.6, 5.11, we applied standard *k*-anonymity and our optimized version on all the datasets over different k conditions that ranged from 3 to 15 and reported the loss results as well as the number of partitions.

Tables 5.3, 5.4, 5.5 show the outcomes for standard *k*-anonymity, optimized version, *l*-diversity, and *t*-closeness. We apply the previously mentioned extensions to each dataset and report the information loss results and number of partitions. Below we describe the information loss results for the individual datasets.

1) **MGM Dataset:**

The MGM dataset has 830 entry and the resulted partition groups and loss results are shown in Table 5.3 and Figures 5.1 – 5.5.

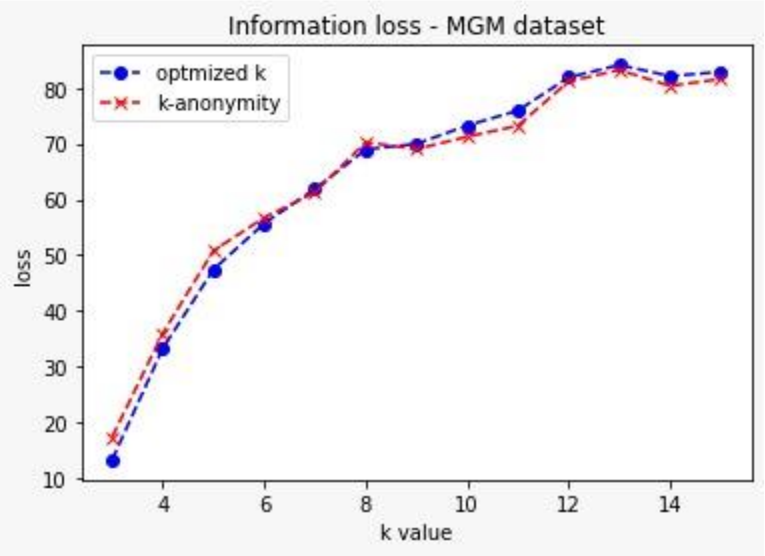
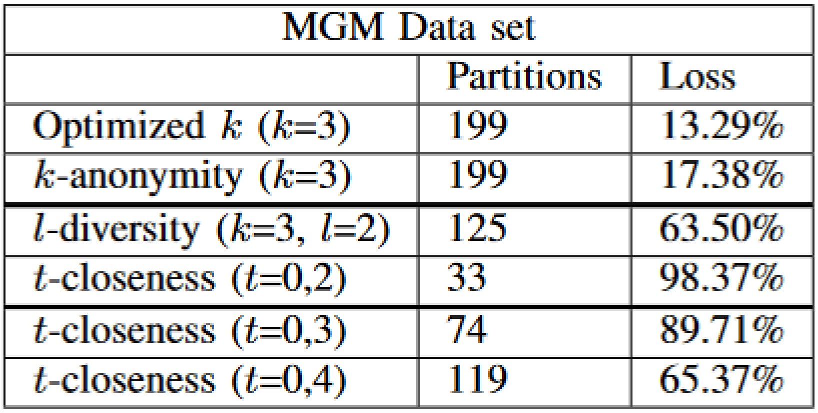


**Figure 5.1:** Information loss results for MGM dataset over different *k* values

**Table 5.3:** MGM dataset partition and information loss results



| MGM Data set | | |
|---|---|---|
| | Partitions | Loss |
| Optimized $k$ ($k$=3) | 199 | 13.29% |
| $k$-anonymity ($k$=3) | 199 | 17.38% |
| $l$-diversity ($k$=3, $l$=2) | 125 | 63.50% |
| $t$-closeness ($t$=0,2) | 33 | 98.37% |
| $t$-closeness ($t$=0,3) | 74 | 89.71% |
| $t$-closeness ($t$=0,4) | 119 | 65.37% |

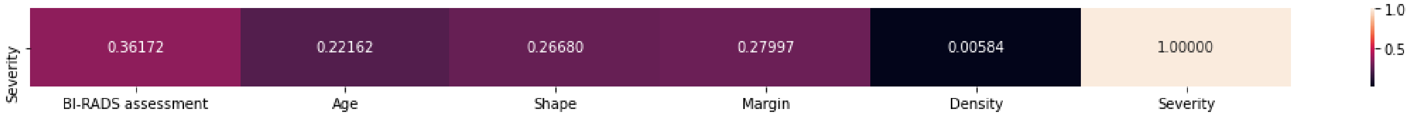


**Figure 5.2:** Entropy result for MGM dataset (no anonymization)

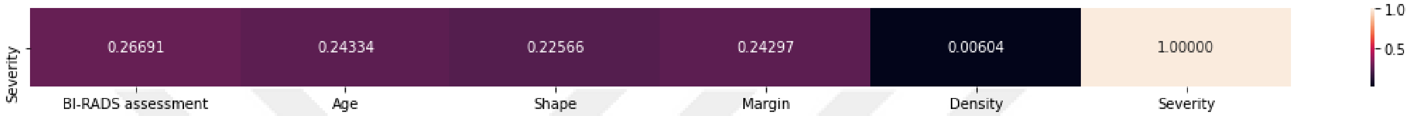


**Figure 5.3:** Entropy result for MGM dataset (standard *k*-anonymity)

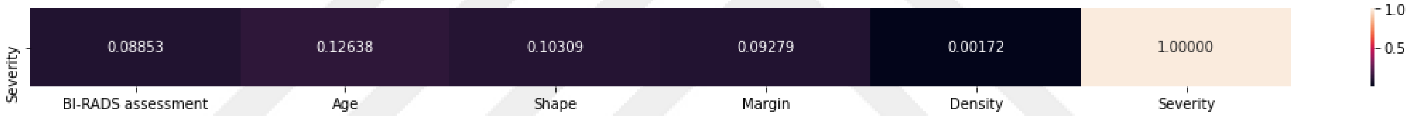


**Figure 5.4:** Entropy result for MGM dataset (*l*-diversity)

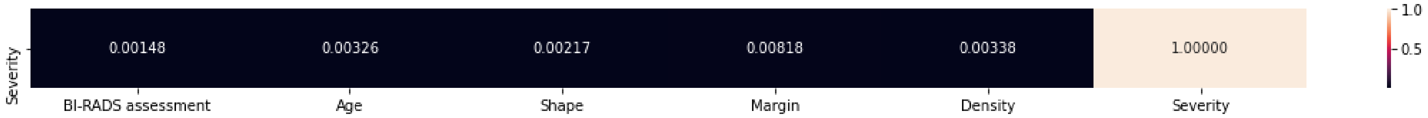


**Figure 5.5:** Entropy result for MGM dataset (*t*-closeness)

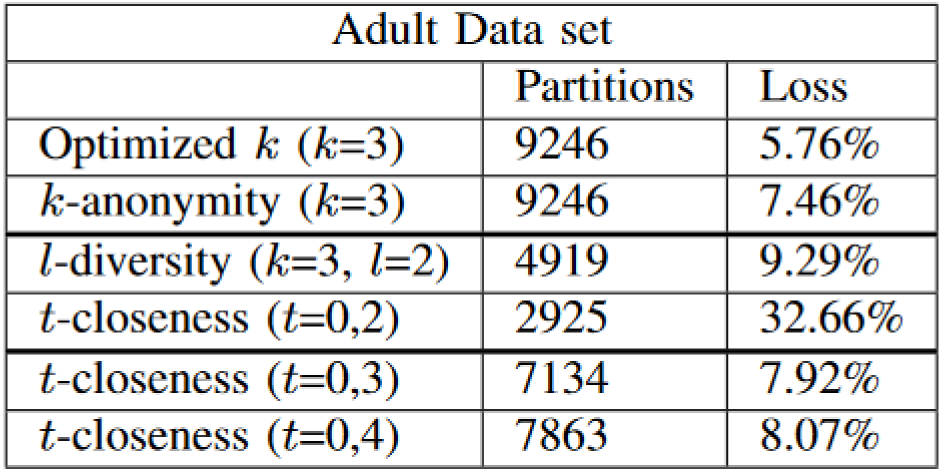
The results are consistent with our expected theoretical results. We notice that *k*-anonymity performed the best, *l*-diversity came as second best while t-closeness performed badly. We notice that the increase of *k* value reduces the number of partitions and increase information loss. *l*-diversity also shows huge information loss due to its more restrictive nature. *t*-closeness results varies when we change the *t* value.

Our default distance value for *t*-closeness is 0.2, which is very strict to guarantee the extension criteria. We know that *t*-closeness use a distance measurement to try to have the same data distribution in each grouped data records as the original dataset's distribution. We also used higher values of *t*, and the more we increase it the less constraints it will have leading to less privacy and less information loss. Results obtained from our optimized *k*-anonymity seems to have slightly reduced the information loss when we compare it to the standard *k*-anonymity.

**2) Adult Dataset**

The Adult dataset has 48843 entry and the resulted partition groups and loss results shown in Table 5.4  and Figures 5.6 – 5.10.

**Table 5.4:** Adult dataset partition and information loss results



| Adult Data set | | |
|---|---|---|
| | Partitions | Loss |
| Optimized *k* (*k*=3) | 9246 | 5.76% |
| *k*-anonymity (*k*=3) | 9246 | 7.46% |
| *l*-diversity (*k*=3, *l*=2) | 4919 | 9.29% |
| *t*-closeness (*t*=0,2) | 2925 | 32.66% |
| *t*-closeness (*t*=0,3) | 7134 | 7.92% |
| *t*-closeness (*t*=0,4) | 7863 | 8.07% |

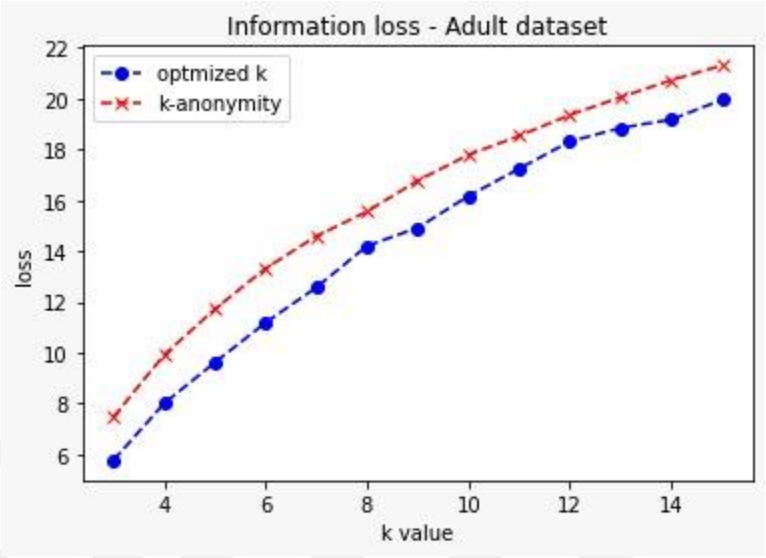


**Figure 5.6:** Information loss results for Adult dataset over different *k* values

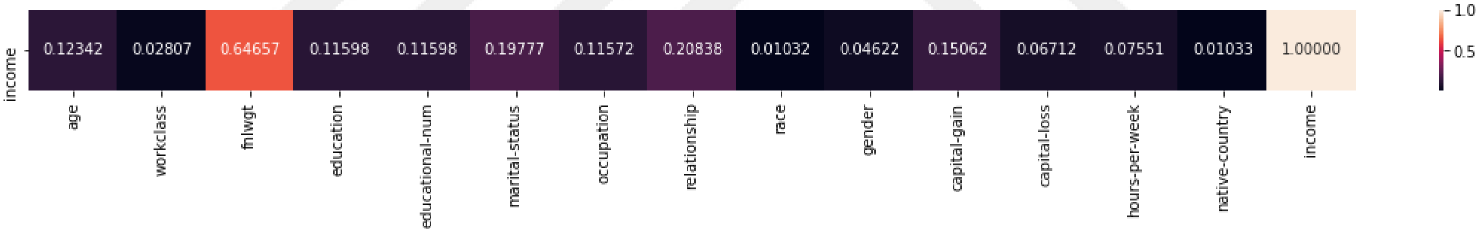


**Figure 5.7:** Entropy result for Adult dataset (no anonymization)

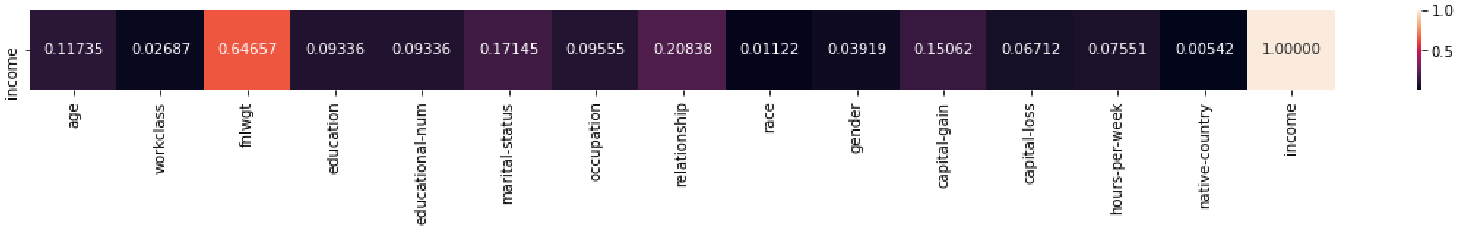


**Figure 5.8:** Entropy result for Adult dataset (standard *k*-anonymity)

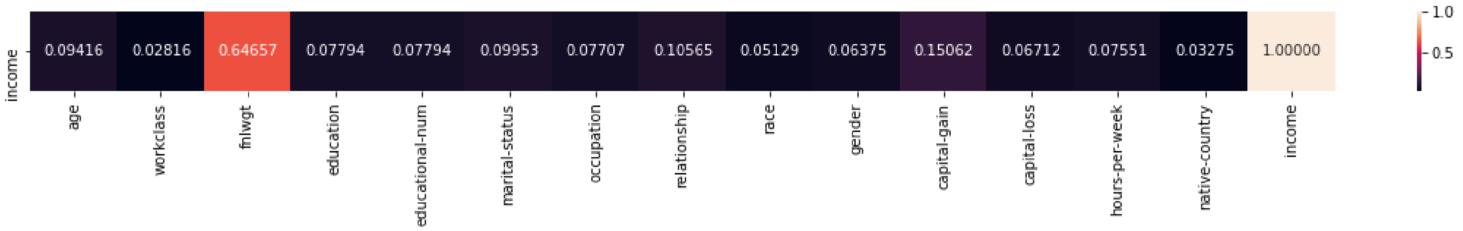


**Figure 5.9:** Entropy result for Adult dataset (*l*-diversity)

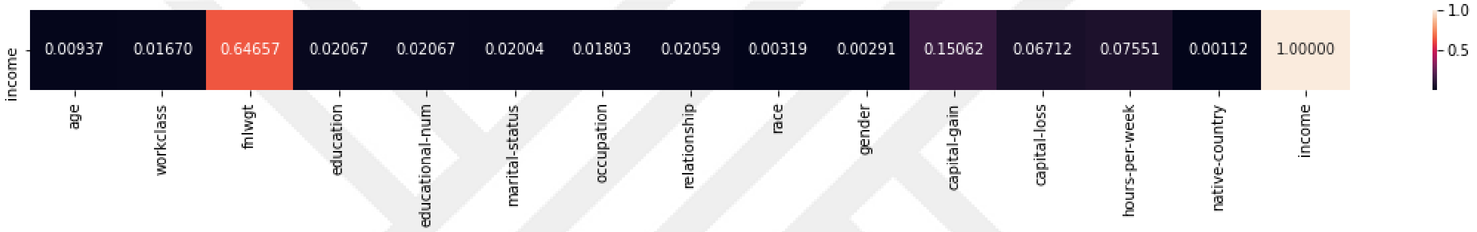


**Figure 5.10:** Entropy result for Adult dataset (*t*-closeness)

Adult dataset is one of the most commonly used dataset in *k*-anonymity studies. From the results illustrated in Table 5.4 we can notice that our theoretical expectations are met across all extensions. We notice that the higher the *k* value the more information loss it has. Similarly, information loss increases when we adopt *l*-diversity extension, and highest loss is observed when *t*-closeness is applied. However, information loss is minimal compared to other datasets, and it appears that this dataset is very suitable for anonymization. It requires very little generalization and suppression during anonymization process. These results also explain probably why most of the work we came across use this dataset for *k*-anonymity analysis. Results obtained from our optimized *k*-anonymity shows a noticeable reduction in information loss when we compare it to the standard *k*-anonymity.

### 3) Housing Dataset

The Housing dataset has 20640 entry and the resulting partition groups and loss values for each variation are shown in Table 5.5 and Figures 5.11 – 5.15.

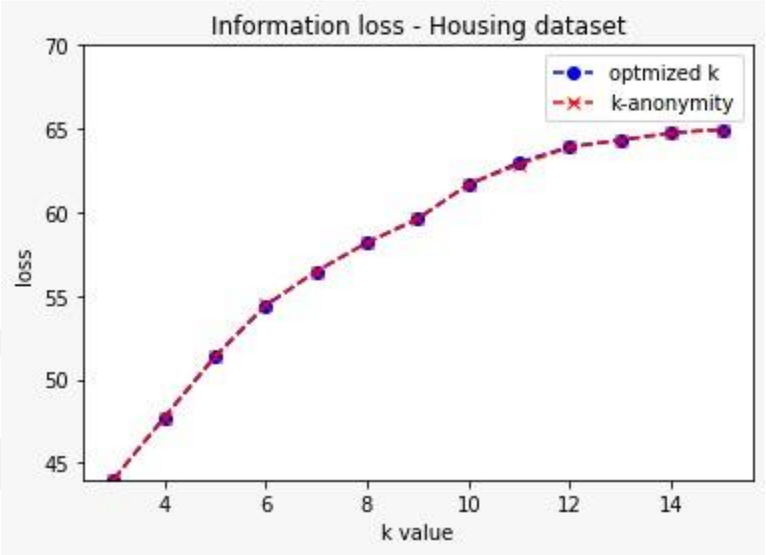
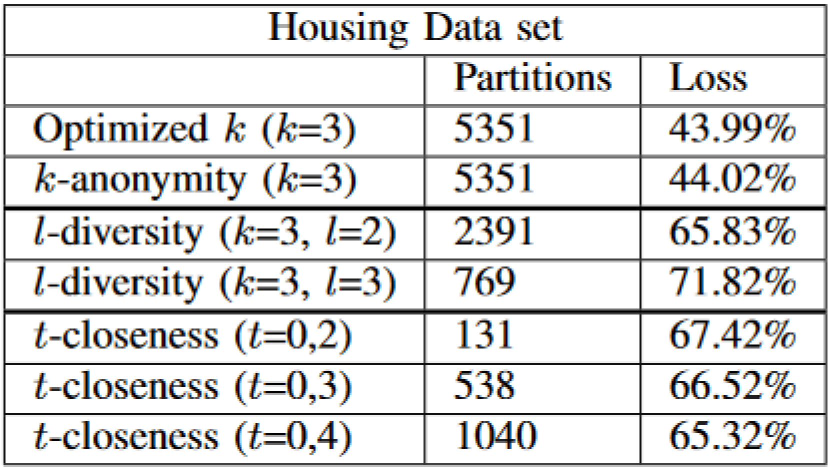


**Figure 5.11:** Information loss results for Housing dataset over different k values

**Table 5.5:** Housing dataset partition and information loss results



| Housing Data set | | |
|---|---|---|
| | Partitions | Loss |
| Optimized $k$ ($k$=3) | 5351 | 43.99% |
| $k$-anonymity ($k$=3) | 5351 | 44.02% |
| $l$-diversity ($k$=3, $l$=2) | 2391 | 65.83% |
| $l$-diversity ($k$=3, $l$=3) | 769 | 71.82% |
| $t$-closeness ($t$=0,2) | 131 | 67.42% |
| $t$-closeness ($t$=0,3) | 538 | 66.52% |
| $t$-closeness ($t$=0,4) | 1040 | 65.32% |

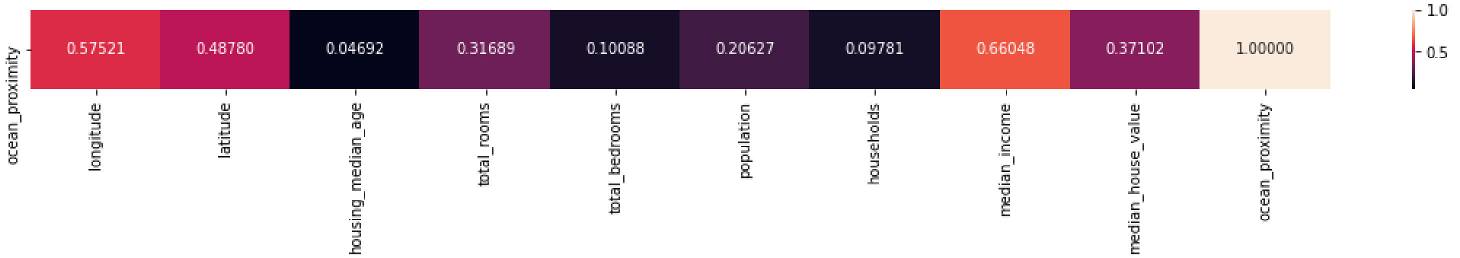


**Figure 5.12:** Entropy result for Housing dataset (no anonymization)

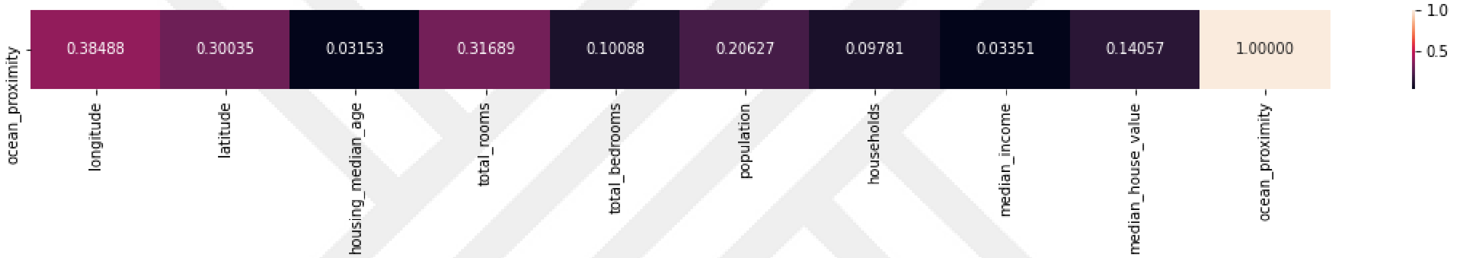


**Figure 5.13:** Entropy result for Housing dataset (standard *k*-anonymity)

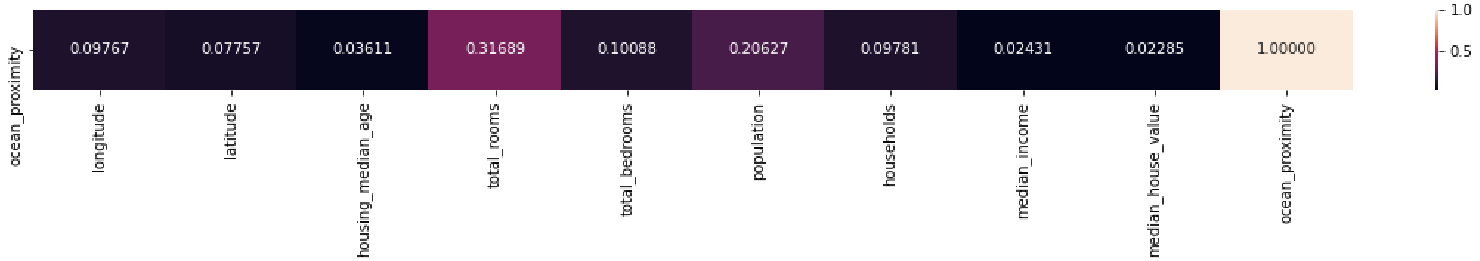


**Figure 5.14:** Entropy result for Housing dataset (l-diversity)

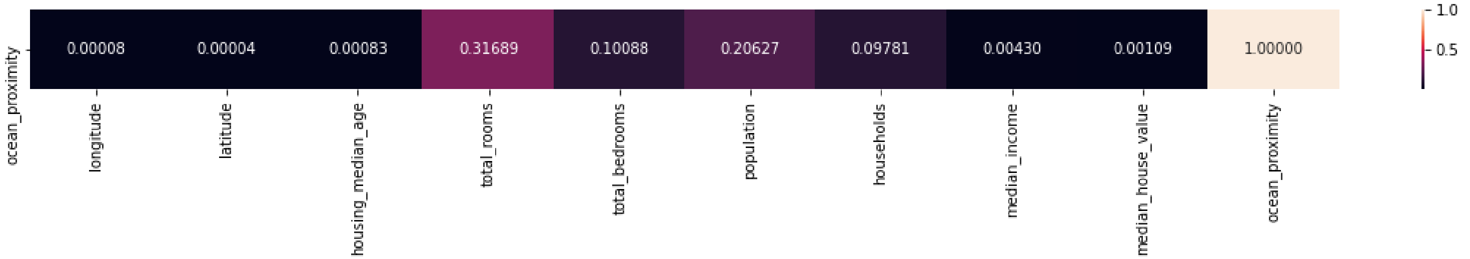


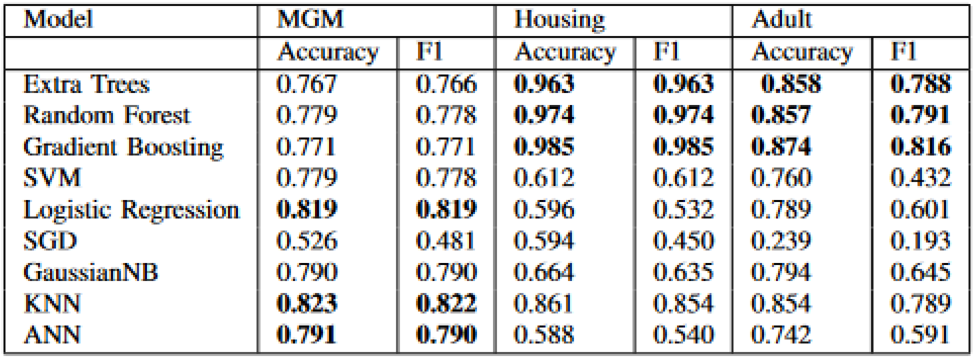
**Figure 5.15:** Entropy result for Housing dataset (t-closeness)

The Housing dataset has 3 unique target column values, with frequencies of 0.44 for 1H-OCEAN, 0.317 for INLAND and 0.23 for NEAR-OCEAN. We notice that there were a significant data loss in all extensions. This is mainly due to the generalization process of the first two QIs, longitude and latitude. This behavior was also reported in other papers [19]. In some datasets, anonymization may cause higher information loss. To be exact, some data column values within a dataset are significantly dependent than others and have high correlation to target columns values, which makes it very hard to generalize and preserve data privacy at the same time. While we noticed a similar behavior when we experimented with different *k*, *l* and *t* values, we concluded that this dataset is exposed to highest information loss after the anonymization process.

## 5.2 Machine Learning Classifiers Results

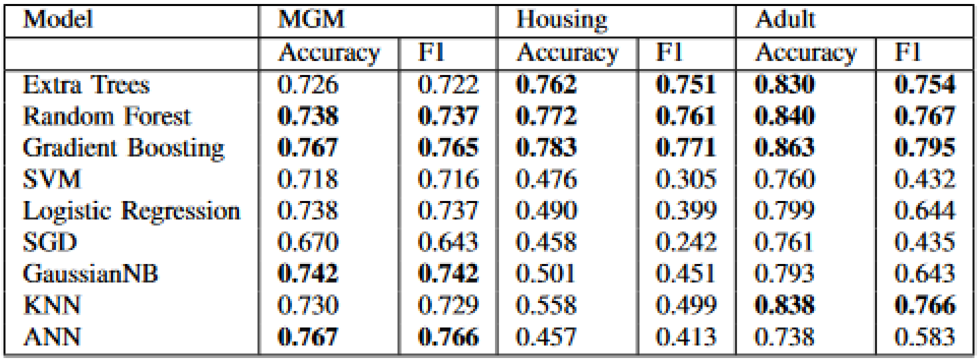
To be able to understand the effect of anonymization we decided first to train our models on the original datasets with no anonymization, report their performance and to use them to compare with models that were trained with anonymized data for each dataset.

**Table 5.6:** Original datasets machine learning results



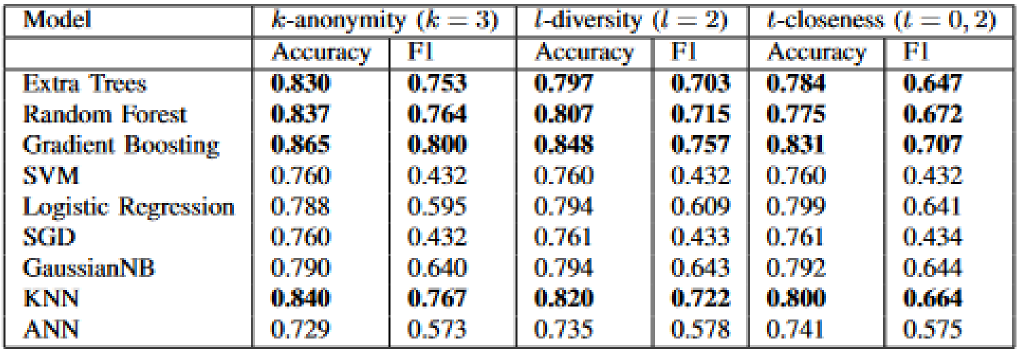
| Model | MGM | | Housing | | Adult | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Extra Trees | 0.767 | 0.766 | **0.963** | **0.963** | **0.858** | **0.788** |
| Random Forest | 0.779 | 0.778 | **0.974** | **0.974** | **0.857** | **0.791** |
| Gradient Boosting | 0.771 | 0.771 | **0.985** | **0.985** | **0.874** | **0.816** |
| SVM | 0.779 | 0.778 | 0.612 | 0.612 | 0.760 | 0.432 |
| Logistic Regression | **0.819** | **0.819** | 0.596 | 0.532 | 0.789 | 0.601 |
| SGD | 0.526 | 0.481 | 0.594 | 0.450 | 0.239 | 0.193 |
| GaussianNB | 0.790 | 0.790 | 0.664 | 0.635 | 0.794 | 0.645 |
| KNN | **0.823** | **0.822** | 0.861 | 0.854 | 0.854 | 0.789 |
| ANN | **0.791** | **0.790** | 0.588 | 0.540 | 0.742 | 0.591 |

**Table 5.7:** Datasets machine learning results, *k*-anonymity with optimization



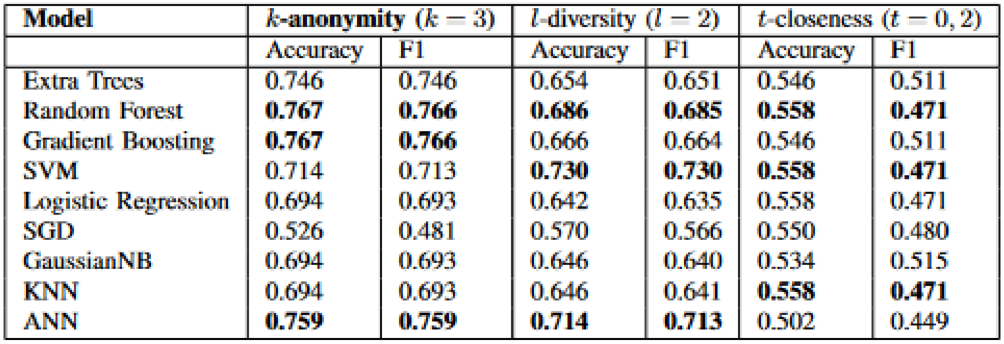
| Model | MGM | | Housing | | Adult | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Extra Trees | 0.726 | 0.722 | **0.762** | **0.751** | **0.830** | **0.754** |
| Random Forest | **0.738** | **0.737** | **0.772** | **0.761** | **0.840** | **0.767** |
| Gradient Boosting | **0.767** | **0.765** | **0.783** | **0.771** | **0.863** | **0.795** |
| SVM | 0.718 | 0.716 | 0.476 | 0.305 | 0.760 | 0.432 |
| Logistic Regression | 0.738 | 0.737 | 0.490 | 0.399 | 0.799 | 0.644 |
| SGD | 0.670 | 0.643 | 0.458 | 0.242 | 0.761 | 0.435 |
| GaussianNB | **0.742** | **0.742** | 0.501 | 0.451 | 0.793 | 0.643 |
| KNN | 0.730 | 0.729 | 0.558 | 0.499 | **0.838** | **0.766** |
| ANN | **0.767** | **0.766** | 0.457 | 0.413 | 0.738 | 0.583 |

**Table 5.8:** Adult datasets machine learning results, without optimization



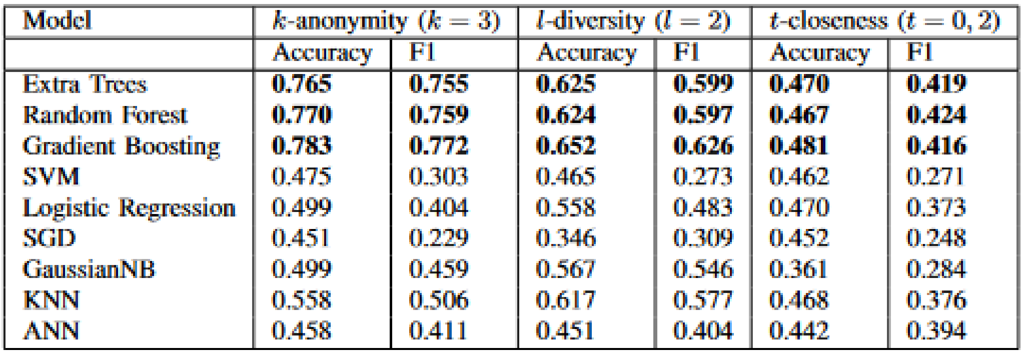
| Model | *k*-anonymity ($k = 3$) | | *l*-diversity ($l = 2$) | | *t*-closeness ($t = 0,2$) | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Extra Trees | **0.830** | **0.753** | **0.797** | **0.703** | **0.784** | **0.647** |
| Random Forest | **0.837** | **0.764** | **0.807** | **0.715** | **0.775** | **0.672** |
| Gradient Boosting | **0.865** | **0.800** | **0.848** | **0.757** | **0.831** | **0.707** |
| SVM | 0.760 | 0.432 | 0.760 | 0.432 | 0.760 | 0.432 |
| Logistic Regression | 0.788 | 0.595 | 0.794 | 0.609 | 0.799 | 0.641 |
| SGD | 0.760 | 0.432 | 0.761 | 0.433 | 0.761 | 0.434 |
| GaussianNB | 0.790 | 0.640 | 0.794 | 0.643 | 0.792 | 0.644 |
| KNN | **0.840** | **0.767** | **0.820** | **0.722** | **0.800** | **0.664** |
| ANN | 0.729 | 0.573 | 0.735 | 0.578 | 0.741 | 0.575 |

**Table 5.9:** Mammographic mass datasets machine learning results, without optimization



| **Model** | ***k*-anonymity** ($k = 3$) | | *l*-diversity ($l = 2$) | | *t*-closeness ($t = 0,2$) | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Extra Trees | 0.746 | 0.746 | 0.654 | 0.651 | 0.546 | 0.511 |
| Random Forest | **0.767** | **0.766** | **0.686** | **0.685** | **0.558** | **0.471** |
| Gradient Boosting | **0.767** | **0.766** | 0.666 | 0.664 | 0.546 | 0.511 |
| SVM | 0.714 | 0.713 | **0.730** | **0.730** | **0.558** | **0.471** |
| Logistic Regression | 0.694 | 0.693 | 0.642 | 0.635 | 0.558 | 0.471 |
| SGD | 0.526 | 0.481 | 0.570 | 0.566 | 0.550 | 0.480 |
| GaussianNB | 0.694 | 0.693 | 0.646 | 0.640 | 0.534 | 0.515 |
| KNN | 0.694 | 0.693 | 0.646 | 0.641 | **0.558** | **0.471** |
| ANN | **0.759** | **0.759** | **0.714** | **0.713** | 0.502 | 0.449 |

**Table 5.10:** California Housing datasets machine learning results, without optimization



| Model | $k$-anonymity ($k = 3$) | | $l$-diversity ($l = 2$) | | $t$-closeness ($t = 0, 2$) | |
|---|---|---|---|---|---|---|
| | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Extra Trees | **0.765** | **0.755** | **0.625** | **0.599** | **0.470** | **0.419** |
| Random Forest | **0.770** | **0.759** | **0.624** | **0.597** | **0.467** | **0.424** |
| Gradient Boosting | **0.783** | **0.772** | **0.652** | **0.626** | **0.481** | **0.416** |
| SVM | 0.475 | 0.303 | 0.465 | 0.273 | 0.462 | 0.271 |
| Logistic Regression | 0.499 | 0.404 | 0.558 | 0.483 | 0.470 | 0.373 |
| SGD | 0.451 | 0.229 | 0.346 | 0.309 | 0.452 | 0.248 |
| GaussianNB | 0.499 | 0.459 | 0.567 | 0.546 | 0.361 | 0.284 |
| KNN | 0.558 | 0.506 | 0.617 | 0.577 | 0.468 | 0.376 |
| ANN | 0.458 | 0.411 | 0.451 | 0.404 | 0.442 | 0.394 |

## 5.3 General Remarks on the Models

The Machine Learning models results are shown in Tables 5.3 – 5.7, which are consistent with the data loss results. In general, trees and ensemble-based models tended to perform better on the anonymized datasets and the best performed models are mainly Gradient Boosting, Random Forests and Extra Trees. We notice that the behavior of these three models remained consistent and they performed the best across all the datasets. The reason of this is due to the nature of the tree-based models since they tend to work well with missing or non-normalized data and they are good at detecting connections between different variables. Hence, they performed good in generalized and suppressed data. We also observe that ANN performs good in MGM dataset in all extensions, and it performs poorly in other datasets. We believe that this behavior is related to parameter tuning. Both SVM and KNN models performed badly in Adult and Housing datasets and this is mainly due to the way they deal with generalized data, however in the MGM dataset they performed consistently better across all extensions.

Stochastic gradient descent (SGD) and logistic regression classifiers belong to the Linear models family and they performed badly in Adult and Housing datasets. We notice that Logistic Regression classifier performed somewhat close to the other models in MGM dataset in standard *k*-anonymity. The behavior of SGD was expected since it also performed badly on the original datasets. We believe that those models failed due to their linear nature and their inability to deal with generalized data. Gaussian Naive Bayes also performed badly in Housing and Adult datasets and inconsistent in MGM dataset. We notice it is not the best when we are dealing with anonymized and generalized datasets.

It is important to note that, while the information loss metric is very helpful in providing insights into anonymized data performance in ML, it should not be confused with the actual ML results. We notice that the results of the classifiers are explainable and strongly linked with the loss results. As an example, when we look at the loss results and classifier results for Adults dataset, we observe that they are strongly linked. As another example, in MGM dataset, when using *t*-closeness extension with $t = 0.2$, loss reaches 98% which means almost all the association is lost. However, the classifiers accuracy scores are around 50% percent. This is not an unexpected result, because MGM dataset has two unique values in target column, each with approximately 52% percent and 48% percent value frequency. The classifiers may continue to predict one class over the other, resulting in the reported results. In general, the obtained test results suggest that the utilized loss metric is a correct indicator for the ML performance of a dataset, and it can be used as an indicator for the suitability of the anonymized data for ML applications in advance.

In some cases, we notice similar accuracy results for optimized and normal *k*-anonymity due to the fact that the reduction in information loss was not significant. However, we believe that with better models tuning *k*-anonymity with optimization could show promising results that reflects the difference in information loss.

In this work, we highly emphasize the usability of the anonymized data, with data generalization and suppression being the most critical parts. To better generalize the data with minimal loss, an automated generalization approach was proposed, which is integrated with an information loss metric.

When we compare our results to those of other studies, for instance in [19] we see that in the Adult dataset, we outperform them noticeably, in the MGM dataset, we achieve slightly higher results, and in Housing dataset, we achieve similar results overall. When we compare to the Mondrian results presented in [18], again we observe that our approach performs better in Adult dataset and perform very similarly in MGM dataset. On the other hand, compared to their proposed NSVDist approach, our approach slightly outperforms in Adult dataset and again performs quite similarly in MGM dataset. Our results appear promising, demonstrating that the proposed automated anonymization approach can outperform previous standard approaches or, in some cases, be on par with them.

## 5.4 General Remarks on *k*-anonymity variations

Previously we mentioned that we have a theoretical expected results for each of the extension, being that the more restriction the anonymization process is subjected to the more data loss is frequent, *l*-diversity restrictions comes from the choice of *l* value where it determines the number of target column values in each grouped data records, the value effects performance by putting extra restrictions on the anonymization process. *t*-closeness restrictions come from the chosen accepted distance criterion between original dataset target value distribution and that of each record distribution. We experimented with various conditions values for each extension and showed its loss results and concluded that our theoretical expected results are met.

In this work we used default values for the conditions on each extension when we needed to use their resulted data for training the ML models. Our default condition values being *k* = 3, *l* = 2 for *l*-diversity and distance criterion of 0.2 for *t*-closeness extension. The *l* value we choose tends to be the least restrictive value that ensures *l*-diversity property and its important to note that it also should not be higher than the number of unique target class values. The choice of 0.2 for distance measurement in *t*-closeness extension limits the grouped data records and add an extra layer of restriction to achieve maximum privacy and ensures the distribution of each record is very similar to the original dataset, the larger the value the greater the distance between distributions. The choice of the distance value does affects the splitting process, anonymization process, information loss and privacy level and its value should be chosen carefully.

# 6. CROSS-SILO FEDERATED LEARNING WITH ANONYMIZATION FRAMEWORK

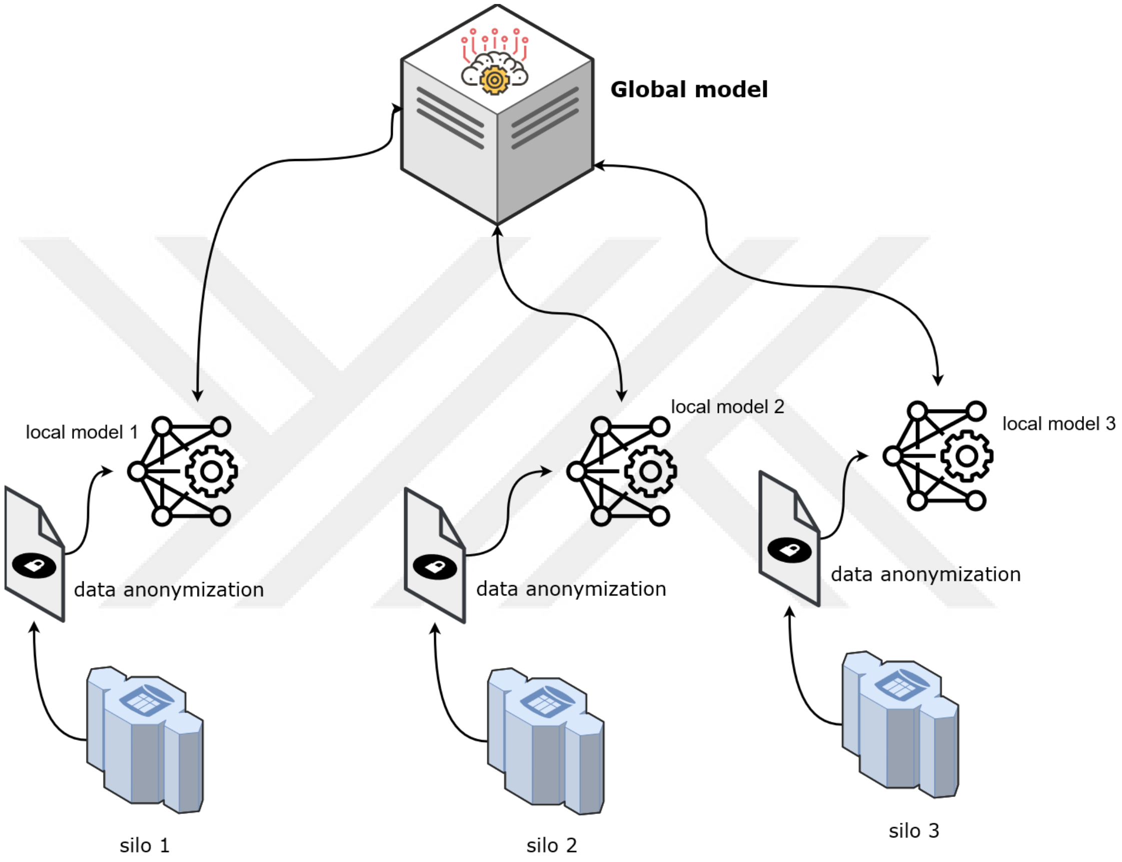


**Figure 6.1:** Cross-Silo federated learning with anonymization framework workflow.

In this study we investigate and apply k-anonymity on data that is meant to be used for federated learning to improve privacy and help protect against potential attacks, as shown in Figure 6.1. Cross-silo federated learning relies on a small number of trusted clients to carry on the training process. It is mostly used in large organizations and hospitals with secure and controlled environment for the training purposes. Now we will discuss the framework components.

## 6.1 Data Distribution

To simulate a federated learning workflow, we need to distribute data among different clients. In this study we used Census Income dataset from UC Irvine Machine Learning Repository. The dataset contains 48843 record and 15 columns.

Our data distribution strategy is as follows:

- We start by splitting the entire dataset into two sets, testing and training sets where testing set contains 30% of the data and the rest is kept for training.

- We get the training set and split it into a previously decided number of partitions, noting that the number of partitions is same number of clients we aim to use.

- The training partitions are forwarded to the anonymization process.

## 6.2 *k*-anonymity

We utilize a Mondrian based *k*-anonymity approximation algorithm [46]. To guarantee that the *k* criterion is satisfied, we use both generalization and suppression. Mondrian is reported to be the best performing algorithm for *k*-anonymization [47]. We developed an automatic generalization approach that generates generalization trees for each column and use an information loss metric to find the best generalization tree for each partition that minimize data loss and achieve *k*-anonymity. An important step is quasi-identifiers selection for anonymization process. We select a subset of features from the data sets we believe if grouped together could lead to a direct identification of subjects. We apply the previous approach for each training partition we have and move to the data distribution among clients.

## 6.3 Federated Learning Clients

After finishing the anonymization process for each training partition, we create a client for each partition. Each client partition will be processed and prepared for training. Each partition will be transformed into a tfds object (TensorFlow data builder) with a batch size of 32.

## 6.4 Federated Learning Global and Local Models

Both global and local models share the same model architecture. We use a Multi-layer Perception (MLP) algorithm. We utilize a 5 layers architecture where 3 of them are hidden layers, the other two are input and output layers respectively. We used a learning rate of 0.01 and Stochastic gradient descent (SGD) [48] as our optimization algorithm.

## 6.5 Models Aggregation

We achieve local models' weights updates to global models by utilizing an iterative model averaging algorithm derived from [49]. The mentioned approach is achieved by this equation:

$$f(w) = \sum_{k=1}^{K} \frac{\mathrm{n_k}}{\mathrm{n}} F_k(w).$$

where:

$$F_k(w) = \frac{1}{\mathrm{n_k}} \sum_{i \in p_k} f_i(w).$$

after we prepared and distributed our partitioned sets among clients. We start the training process as follows:

- We start by creating a global model, then we start creating local models and assign them to each client.

- We start the training process, in each epoch (training round) the local models send their weights to the global model for it to be updated.

- Each epoch (round) before updating the global model we apply federated learning averaging for local models' weights.

- After the global model has been updated in each epoch (round), global model shares its weights with the local models.

- With each global model we calculate global accuracy by testing on the non-anonymized testing partition.

# 7. FEDERATED LEARNING FRAMEWORK RESULTS

To be able to compare the results, we need to apply the previous mentioned approach on non-anonymized data and anonymized data under the same training process and client's data distribution conditions. We will also calculate global accuracy, precision, recall and F1 score at the end of the training. Global accuracy refers to the accuracy resulted from testing on the final version of the global model after it finishes all training epochs (rounds) on all clients and finish updating. We split the data among 3 silos, the dataset is split into 3 parts, then they are moved to the anonymization stage. We use a non-anonymized partition as our testing set. We will refer to each data partition as group1, group2 and group3.

## 7.1 Information Loss Results

Information loss results for each column for group1, group2 and group3 are illustrated in Figures  7.1 – 7.6.

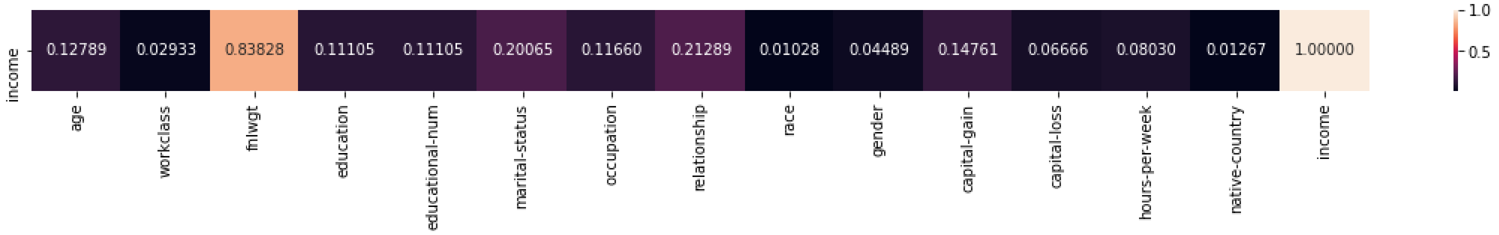


**Figure 7.1:** Information loss results for Group 1 before anonymization

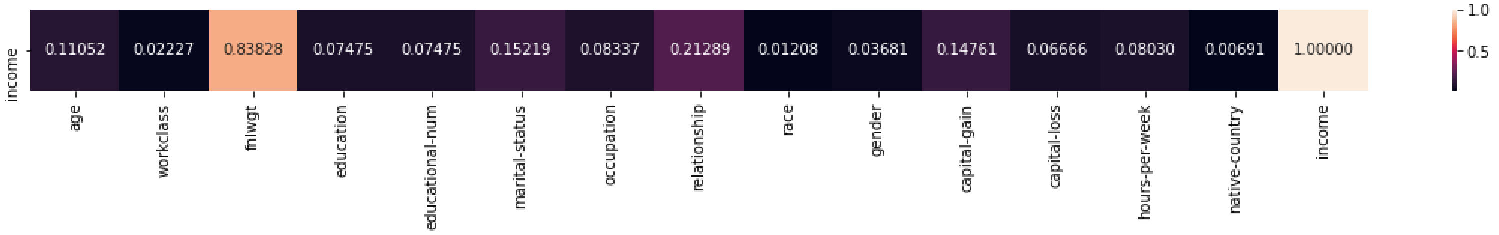


**Figure 7.2:** Information loss results for Group 1 after anonymization

If we compare the information loss result for each column (comparing each one with its original value) we notice an information loss of 6.13%, hence it was able to preserve 93.87% of the information.

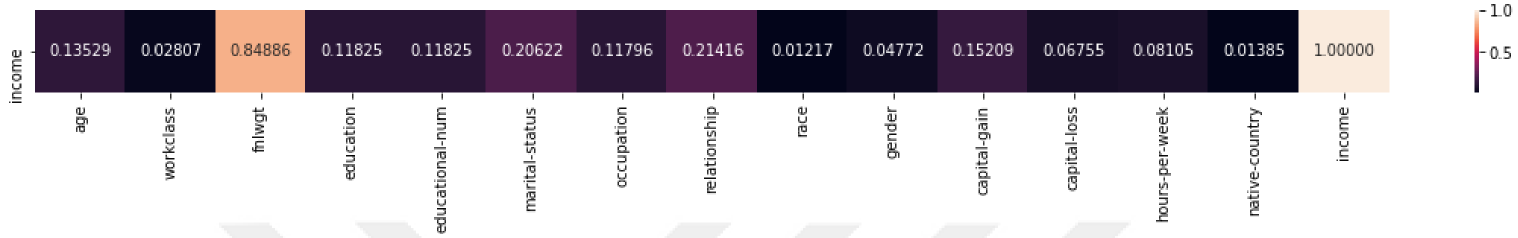


**Figure 7.3:** Information loss results for Group 2 before anonymization

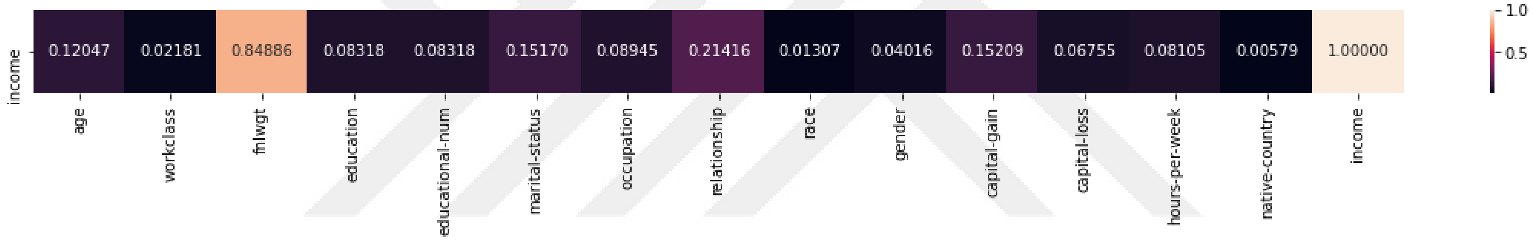


**Figure 7.4:** Information loss results for Group 2 after anonymization

If we compare information loss result for each column (comparing each one with its original value) we notice an information loss of 5.97%, hence it was able to preserve 94.03 of the information.

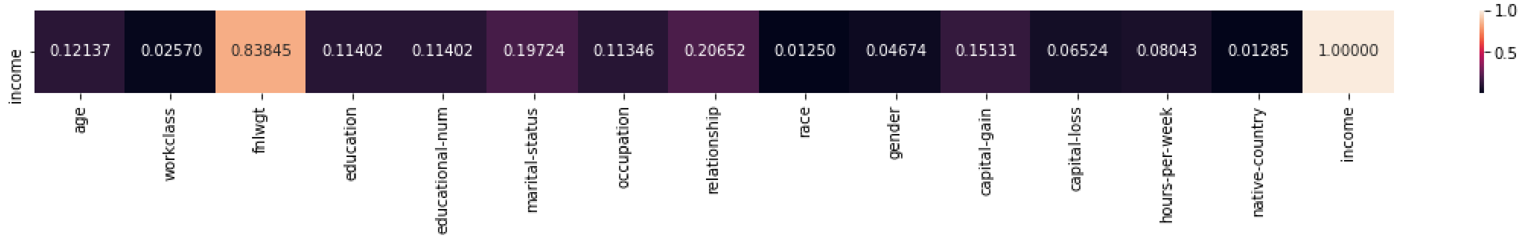


**Figure 7.5:** Information loss results for Group 3 before anonymization

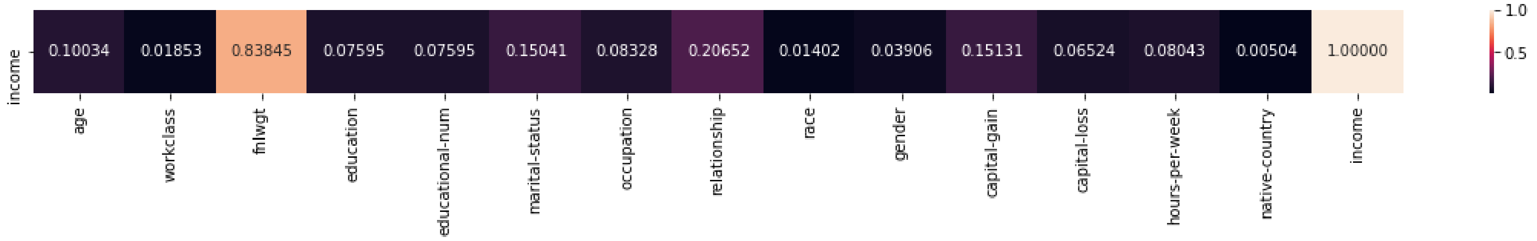


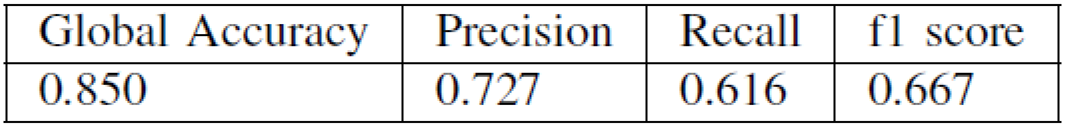
**Figure 7.6:** Information loss results for Group 3 after anonymization

If we compare information loss result for each column (comparing each one with its original value) we notice an information loss of 6.30%, while it was able to preserve 93.70% of the information.

## 7.2 Training Results

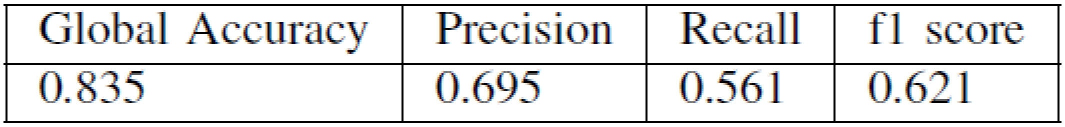
Table 7.1 illustrates FL results for non-anonymized data, while Table 7.2 illustrates FL results for anonymized data.

**Table 7.1:** Non-Anonymized data federated learning results



| Global Accuracy | Precision | Recall | f1 score |
|---|---|---|---|
| 0.850 | 0.727 | 0.616 | 0.667 |

**Table 7.2:** Anonymized data federated learning results



| Global Accuracy | Precision | Recall | f1 score |
|---|---|---|---|
| 0.835 | 0.695 | 0.561 | 0.621 |

## 7.3 Federated learning framework remarks

We can notice that the anonymized data FL model have a slight decrease in global accuracy. This is also reflected in the F1 score and other metrics we calculated. This decrease is a result of the anonymization process. This behavior was expected since applying anonymization introduces some degree of information loss. We can infer that anonymization with proper generalization and suppression techniques can minimize data loss. Introducing anonymization to FL provides more privacy than using any of them alone.

Let us address the attacks of *k*-anonymity. the attacks we discussed in this paper, homogeneity attacks and background knowledge attacks work under the assumption that an attacker can have access to anonymized data and exploit some limitation on *k*-anonymity to gain insights on certain subjects, compromising the data privacy. Such attacks are no longer applicable with federated learning since no raw or anonymized data is being shared, instead it shares the local models' weights.

Addressing attacks on FL, while anonymizing data is a step further to protect from membership inference, data and models poisoning attacks, focus is also being done on identifying malicious clients and local models' updates rejections based on many error rates they introduce. Both attacks assume an attacker have a possibility to add or temper with local models training data. In Cross-Silo Federated Learning framework we studied those attacks are hard to implement since this type of FL is usually used by large organization and hospitals where the training happens in secured and controlled environments.

# 8. Conclusion and Future Work

## 8.1 Thesis Conclusion

In this thesis, we investigated the anonymization effects on machine learning model performance, as well as focused on developing an automated generalization framework that can work on any dataset without requiring manually built generalization hierarchy trees. We used an information loss metric, which is a novel approach that provides useful insights on ML model performance before training. We chose the Mondrian algorithm to implement our *k*-anonymity framework because it has been reported to be the best performing overall [15]. We also worked with two different extensions of *k*-anonymity to understand the behavior of anonymization in each one, considering that each extension offers a different level of privacy. We tested on three different datasets, experimented with different condition values for each extension, and discussed our results.

Our results show that standard *k*-anonymity performs the best when it comes to ML model performance, followed by *l*-diversity and *t*-closeness. We also notice that the results can vary with different condition values. An interesting case we tested is adding an optimization step to the anonymization process, which showed that it can improve standard *k*-anonymity performance without compromising its privacy. When the anonymization process is subjected to strict constraints, the performance of ML models degrades in general; however, the degree of degradation is dependent on the *k*-anonymity extension, condition value, generalization, and suppression choices. Our findings from the Housing dataset show that some dataset columns cannot be anonymized without significantly degrading ML performance, particularly when those columns have a high correlation with the target values.

We also investigated data anonymization, federated learning, and their role in data privacy in detail. We discussed threats on both approaches. We defined a cross-silo federated learning framework with *k*-anonymity adaptation that improves data privacy and introduce minimal information loss for classification tasks. Our proposed framework benefits from both approaches advantages and better protects against data privacy threatening attacks that each approach alone is vulnerable to. We showed that federated learning performance can be improved in terms of preserving privacy while getting minimal information loss with proper integration of anonymization techniques. We noticed that little focus is put on investigating anonymization effects in federated learning frameworks, so we discussed our implementation of Federated learning in detail. We hope that this work will be a starting point for more future studies.

## 8.2 Future Work Directions

Future work for anonymization framework can include an extensive testing on new real-life datasets. While we tested on 3 different datasets with different distributions and a variety of data types, there is a general lack of publicly available real-life datasets that are suitable for this type of study. Also, an extension of this work can be the comparison between *k*-anonymized datasets trained ML models and FL models in terms of processing time, privacy level and performance.

For our FL framework, we noticed lack of studies and implementation for categorical datasets in FL, while most work we found is using FL for images-based classification tasks. So, testing on more datasets is vital to investigate the role of FL further. For some types of FL, investigating more strategies for clients' selections to better protect against malicious clients would be the next step.

# REFERENCES

**[1]** European Union, “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),” p. 1–88, 2016. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

**[2]** Turkey, “Regulation (tr) 2016/6698,” 2016.[Online]. Available: https://www.kvkk.gov.tr/Icerik/6649/Personal-Data-Protection-Law

**[3]** A. Majeed and S. Lee, “Anonymization techniques for privacy preserving data publishing: A comprehensive survey,” IEEE Access, vol. 9, pp. 8512–8545, 2021.

**[4]** P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression,” Tech. Rep., 1998.

**[5]** A. Gionis and T. Tassa, “*k*-anonymization with minimal loss of information,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 2, pp. 206–219, 2009.

**[6]** A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “L-diversity: privacy beyond *k*-anonymity,” in 22nd International Conference on Data Engineering (ICDE’06), 2006, pp. 24–24.

**[7]** N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond *k*-anonymity and l-diversity,” in 2007 IEEE 23rd Inter-national Conference on Data Engineering, 2007, pp. 106-115.

**[8]** L. Sweeney, “Achieving *k*-anonymity privacy protection using generalization and suppression,” Int. J. Uncertain. Fuzziness Knowl. Based Syst., vol. 10, pp. 571–588, 2002.

**[9]** C. C. Aggarwal and P. S. Yu, A General Survey of Privacy- Preserving Data Mining Models and Algorithms. Boston, MA: Springer US, 2008, pp. 11–52.

**[10]** T. De Waal and L. Willenborg, “Information loss through global recoding and local suppression,” 01 1999.

**[11]** J.-W. Byun, A. Kamra, E. Bertino, and N. Li, “Efficient *k*- anonymization using clustering techniques,” in Advances in Databases: Concepts, Systems and Applications, R. Kotagiri, P. R. Krishna, M. Mohania, and E. Nantajeewarawat, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 188– 200.

**[12]** B. Malle, P. Kieseberg, and A. Holzinger, “Interactive anonymization for privacy aware machine learning,” in IAL@PKDD/ECML, 2017.

**[13]** S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, “Privacy preserving location data publishing: A machine learn- ing approach,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, pp. 3270–3283, 2021.

**[14]** F. Liu and T. Li, “A clustering *k*-anonymity privacy-preserving method for wearable iot devices,” Secur. Commun. Networks, vol. 2018, pp. 4 945 152:1–4 945 152:8, 2018.

**[15]** W. Mahanan, W. A. Chaovalitwongse, and J. Natwichai, “Data anonymization: a novel optimal *k*-anonymity algorithm for identical generalization hierarchy data in iot,” Serv. Oriented Comput. Appl., vol. 14, no. 2, pp. 89–100, 2020.

**[16]** R. Khan, X. Tao, A. Anjum, T. Kanwal, S. u. R. Malik, A. Khan, W. u. Rehman, and C. Maple, “-sensitive *k*- anonymity: An anonymization model for iot based electronic health records,” Electronics, vol. 9, no. 5, 2020.

**[17]** H. Wimmer and L. M. Powell, “A comparison of the effects of *k*-anonymity on machine learning algorithms,” International Journal of Advanced Computer Science and Applications, vol. 5, pp. 155–160, 2014.

**[18]** M. Last, T. Tassa, A. Zhmudyak, and E. Shmueli, “Improving accuracy of classification models induced from anonymized datasets,” Information Sciences, vol. 256, pp. 138–161, 2014.

**[19]** D. Slijepcevic, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, “*k*-anonymity in practice: How generalisa- tion and suppression affect machine learning classifiers,” ArXiv, vol. abs/2102.04763, 2021.

**[20]** B. C. Fung, K. Wang, and P. S. Yu, “Anonymizing classification data for privacy preservation,” IEEE Transactions on Knowl- edge and Data Engineering, vol. 19, no. 5, pp. 711–725, 2007.

**[21]** K. LeFevre, D. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional *k*-anonymity,” in 22nd International Conference on Data Engineering (ICDE’06), 2006, pp. 25–25.

**[22]** N. Mohammed, B. C. Fung, P. C. Hung, and C.-k. Lee, “Anonymizing healthcare data: A case study on the blood transfusion service,” in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1285–1294.

**[23]** J. Goldberger and T. Tassa, “Efficient anonymizations with enhanced utility,” in 2009 IEEE International Conference on Data Mining Workshops, 2009, pp. 106–113.

**[24]** K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley, “A Globally Optimal *k*-Anonymity Method for the De-Identification of Health Data,” Journal of the American Medical Informatics Association, vol. 16, no. 5, pp. 670–682, 09 2009.

**[25]** J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.- C. Fu, “Utility-based anonymization using local recoding,” in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’06. New York, NY, USA: Association for Computing Machinery, 2006, p. 785–790.

**[26]** J.-L. Lin and M.-C. Wei, “An efficient clustering method for *k*-anonymization,” ser. PAIS ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 46–50.

**[27]** L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

**[28]** P. Geurts, “Extremely randomized trees,” in MACHINE LEARN- ING, 2003, p. 2006.

**[29]** N. Cristianini and J. Shawe-Taylor, “An introduction to support vector machines and other kernel-based learning methods,” 2000.

**[30]** T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system.” Tech. Rep., 2016.

**[31]** E. Wilson and D. Tufts, “Multilayer perceptron design algorithm,” in Proceedings of IEEE Workshop on Neural Networks for Signal Processing, 1994, pp. 61–68.

**[32]** European Union, “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),” p. 1–88, 2016.

**[33]** P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression,” Tech. Rep., 1998.

**[34]** I. Buratovi´c, M. Miliˇcevi´c, and K. ˇZubrini´c, “Effects of data anonymization on the data mining results,” in 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1619– 1623.

**[35]** J. Koneˇcn ´y, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in NIPS Workshop on Private Multi-Party Machine Learning, 2016.

**[36]** P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D’Oliveira, S. E. Rouayheb, D. S. Zhao, “Advances and open problems in federated learning,” CoRR, vol. abs/1912.04977, 2019.

**[37]** Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” arXiv: Artificial Intelligence, 2019.

**[38]** P. M. Mammen, “Federated learning: Opportunities and challenges,” ArXiv, vol. abs/2101.05428, 2021.

**[39]** K. Nandury, A. Mohan, and F. Weber, “Cross-silo federated training in the cloud with diversity scaling and semi-supervised learning,” in ICASSP 2021 - 2021 IEEE International Confer- ence on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3085–3089.

**[40]** A. Durrant, M. Markovic, D. Matthews, D. May, J. A. Enright, and G. Leontidis, “The role of cross-silo federated learning in facilitating data sharing in the agri-food sector,” CoRR, vol. abs/2104.07468, 2021. [Online]. Available: https: //arxiv.org/abs/2104.07468

**[41]** C. Che, X. Li, C. Chen, X. He, and Z. Zheng, “A decentralized federated learning framework via committee mechanism with convergence guarantee,” CoRR, vol. abs/2108.00365, 2021.

**[42]** R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Member- ship inference attacks against machine learning models,” 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18, 2017.

**[43]** A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “L-diversity: privacy beyond *k*-anonymity,” in 22nd International Conference on Data Engineering (ICDE’06), 2006.

**[44]** N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond *k*-anonymity and l-diversity,” in 2007 IEEE 23rd Inter- national Conference on Data Engineering, 2007, pp. 106–115.

**[45]** S. Yoo, M. Shin, and D. Lee, “An approach to reducing infor- mation loss and achieving diversity of sensitive attributes in *k*- anonymity methods,” Interactive Journal of Medical Research, vol. 1, 2012.

**[46]** K. LeFevre, D. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional *k*-anonymity,” in 22nd International Conference on Data Engineering (ICDE’06), 2006.

**[47]** D. Slijepcevic, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, “*k*-anonymity in practice: How generalisation and suppression affect machine learning classifiers,” CoRR, vol. abs/2102.04763, 2021.

**[48]** S. Ruder, “An overview of gradient descent optimization algorithms,” ArXiv, vol. abs/1609.04747, 2016. [18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in AISTATS, 2017.

**[49]** H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in AISTATS, 2017.