dedication (optional)

Summary

Write your summary here...

Preface

Write your preface here...

Table of Contents

List of Tables

List of Figures

Abbreviations

AMS	=	Agent Management System
DF	=	Directory Facilitator
IMDB	=	Internet Movie Database
JADE	=	Java Agent framework for Distance learning Environments
PII	=	Personally Identifiable Information
PINQ	=	Privacy Integrated Queries
PPDM	=	Privacy-Preserving Data Mining
PPDP	=	Privacy-Preserving Data Publishing
QID	=	Quasi Identifier

Symbols

ϵ	=	Privacy parameter that represents the privacy budget
ϵ_A	=	Parameter which represents the privacy level of our aggregation mechanism
λ	=	Regularization parameter
α	=	Learning rate parameter
A	=	Represents our aggregation mechanism
D	=	Represents a dataset
M	=	Represents a general privacy mechanism, like the Laplacian or the Exponential.

Chapter 1

Introduction

All over the world people are interacting with technology more than ever; when using their cell phone, shopping online, visiting a doctor who uses electronic records, and in countless other acts. This usage generates a massive amount of information, leading to data being more deeply integrated into our daily lives than ever before. Sintef published a report in 2013 which stated that: ”A full 90% of all the data in the world has been generated over the last two years [13].” With this massive influx of information, new fields of both academic study and commercial interest have appeared to find out how to best analyze this data.

The terms ”big data” and ”analytics” have been widely used as common designations for this emerging field of technology. The communal definition for describing big data stems from a 2001 research report, in which analyst Doug Laney defined the problem of being a three-dimensional challenge: ”Big data is high-volume, -velocity and -variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.” The first part of his challenge, commonly known as the 3 Vs of big data, deals with the necessary qualifications for data to be called ”big data”, while part two and three is the how and why.

The wide variety of the potential applications of big data analytics have also raised essential questions about whether our social and ethical norms are sufficient to protect privacy in a world which has entered ”the era of big data”. Both in the European Union and in the United States there have been efforts made to create new laws for handling data privacy. The Council to the President, an advisory group to the US President, concluded in their 2014 report [21] that preserving privacy values would be their number one recommendation when designing a new policy framework for big data. Furthermore, they advised that more than 70 million USD should be made available to federal research in privacy-enhancing technologies.

1.1 Objective and Scope / Problem Statement

The objective of this study is to contribute to the aforementioned field of study, more specifically in the area of Privacy-Preserving Data Mining (PPDM). In our work we ex-

Right word?

plore the usefulness of employing a privacy-preserving technique called differential privacy. Relying on previous work in homomorphic encryption and peer to peer communication, we would like to create an architecture that allows for distributed, scalable machine learning while preserving the privacy of the participants.

R1: How big is the loss of accuracy in a distributed, differentially private system, compared to a centrally trained model?
While there have been research on both distributed and differentially private machine learning system, there have been very little research done on a combination of both. Results from research on differential privacy indicate that there often is a trade-off between privacy and a loss of accuracy. We want to study this trade-off in our distributed approach and analyze which factors comes into play and how they can be handled in a way that leads to an optimal result.

R2: How can the variance in accuracy between participants be minimized?
Our system architecture is based on a notion of independent peers which collaborate to create aggregated logistic regression models which is used for classification. Due to there not being one single centralized classifier, there will most likely be a variance in the accuracy of the classifiers each peer hold. We want to explore options on how to reduce this variance, so that we can reduce the likelihood of one peer having a well-performing classifier while another produces poor classification results.

1.2 Research Question/Goal

Can we do distributed machine learning while still provide a differential privacy guarantee?

1.3 Thesis Structure

Here there will a short summary of each chapter

Chapter 2

Background and Motivation

In this section we will first explain some basic concepts and expressions that are used in the privacy context such anonymization operations and Personally Identifiable Information(PII). Then we will have a look at some classic examples of failure to preserve privacy when data publishing and how these attacks motivated us to choose our topic for this thesis.

2.1 Concepts and Expressions

In the most basic form of privacy-preserving data publishing (PPDP), the data holder has a table of attributes from the following four categories: Explicit Identifier, Quasi Identifier, Sensitive Attributes, and Non-Sensitive Attributes [15]. A summary of each category can be found in Table 2.1.

Attribute name	**Definition**	**Example**
Explicit Identifier	Explicitly identifies record owners	Government identity number (e.g SSN)
Quasi Identifier(QID)	Potentially identifies record owners	Birth date and gender
Sensitive Attributes	Sensitive information about a person	Income, disability status
Non-Sensitive Attributes	All other attributes	Favorite band

Table 2.1: Table of basic categories of database attributes

From these categories, it would be easy to think that Personally Identifiable Information (PII) would only be found in the first attribute. As we will see in the next section, this is not the case. Recent privacy laws have defined PII in a much broader way. They account for the possibility of deductive disclosure and do not lay down a list of attributes that constitutes as PII. For example, the European Parliament made a set of directives known as the Data Protection Directive, in which personal data is defined as: ”any information relating to an [] natural person [] who can be identified, directly or indirectly, in particular by reference [] to one or more factors specific to his physical, physiological, mental, economic, cultural, or social identity”‘[33].

In order to remove any PII from a dataset, it needs to go through a process called anonymization. This constitutes a series of modifications/manipulations of with the ultimate end goal of protecting the privacy of the dataset’s participants. Fung et al.[15] operates with a number of five basic operations which might be applied for this purpose. These operations are shortly described in Table 2.2.

Anonymization Operation	**Definition**
Generalization	Replaces the value with more general value, such as a mean value
Suppression	Replaces the value with a special value, indicating that the replaced values are not disclosed
Anatomization	De-associates the relationship between the quasi-identifier and sensitive information
Permutation	Partitions a set of data records into groups and shuffles their sensitive values
Perturbation	Replace the original value with a synthetic value that keep the statistical characteristics

Table 2.2: Table of anonymization operations (adapted from [15])

2.2 Privacy Breaches

In the recent years there have been many failures in privacy preserving data publishing. Many companies have been faced with a PR disaster after releasing data about their customers thinking them being anonymized, only to have people de-anonymize their data and breaching the privacy of the datasets’ participants. In this section we will have a look at some of these privacy failures.

2.2.1 Netflix Prize Competition

Netflix, the world’s largest online movie streaming website, decided in 2006 to crowdsource a new movie suggestion algorithm and offered a cash prize of 1 million dollar for the most efficient algorithm. To help the research, they released 100 million supposedly anonymized movie ratings from their own database. In order to protect the privacy of their users, Netflix removed all user level information; such as name, username, age, geographic location, browser used, etc. They also deliberately perturbed ”some of the rating data for some customers[...] in one or more of the following ways: deleting ratings; inserting alternative ratings and dates, and modifying random dates”[5]. The released data records included an anonymized user ID, movie name, date of rating, and the user’s rate on a scale from 1 to 5.

Two researchers from the University of Texas,Narayanan and Shmatikov[29], demonstrated that an adversary who knows only a little bit about an individual subscriber can easily identify this subscriber's record in the dataset. Using the publicly available dataset from the Internet Movie Database (IMDB) as the source of background knowledge, they matched certain subscribers with their Netflix records, and uncovered their apparent political preferences and other potentially sensitive information. The paper also offered a formal mathematical treatment of how a small amount of auxiliary knowledge about an individual can be used to do a fairly reliable re-identification. In the case of the Netflix dataset, the authors [29] found that with only 8 movie ratings, 99% of the records could be uniquely identified. Furthermore, they proved that the de-anonymization algorithm they employed is robust to discrepancies in the rating and dates.

2.2.2 Group Insurance Commission

In 1997, Latanya Sweeney wrote a paper on how she had identified the medical records of Massachussets governor William Weld based on publicly available information from the database of Group Insurance Commission. She achieved this analyzing data from a public voter list, and linked it with patient-specific medical data through a combination of birth date, zip code, and gender[41]. As these columns were similar in both databases, their combination could be used to identify medical records that belong to either one person, or a small group of people. Sweeney hypothesized that 87% of the US population could be identified by having the combination of the three aforementioned records. It's worth noting here that this theory is not conclusive. A paper by Daniel Barth-Jones suggests that the re-identification of Weld may have been a fluke due to his public figure, and that ordinary people risk of identification is much lower[4].

2.2.3 New York Taxi dataset

The New York City Taxi and Limousine Commission released a dataset in 2013 containing details about every taxi ride that year, including pickup and drop-off times, location, fare, as well as anonymized (hashed) versions of the taxi's license and medallion numbers. Vijay Pandurangan, a researcher for Google, wrote a blog-post where he showed how he exploited a vulnerability in the hashing-function to re-identify the drivers. He then showed how this could be potentially used to calculate any driver's personal income[32].

Another researcher, called Anthony Tockar, wrote an article during his internship at Neustar Research where he proved that the dataset also contained an inherent privacy risk to the passengers which had been riding New York Taxis. Even though there was no information in the dataset on who had been riding the taxis, Tockar showed that by using auxiliary information such as timestamped pictures, he could stalk celebrities and figure out to where they were driving, and how much they tipped the driver. He also used map data from Google Maps to create a map of drop-off locations for people that had exited a late night visit from gentleman's club and taken a cab home. He then used websites like Spokeo and Facebook to find the cab customer's ethnicity, relationship status, court records, and even a profile picture[42].

2.3 Attack Vectors

2.3.1 Linkage Attacks [Rename]

In each of the examples in the previous section, the privacy breach was achieved through an attack model called linkage attacks. These types of attacks are characterized that they create a decision rule which link at least one data entry in the anonymized dataset with public information which contain individual identifiers, given that the probability of these two matching exceeds a selected confidence threshold.

In the literature[6, 15], they broadly classify the attack models into two categories: Record linkage and attribute linkage. In both these types of attack, we need to assume that the attacker knows the QID of the victim.

Record Linkage[rename]

In the case of attribute linkage, some quasi-identifier value QID identifies a small number of records in the original dataset, which is called a group. If the victim's QID is the same, he or she is then vulnerable to being linked to this much smaller number of records in the group. With the help of some additional information, there is then a chance that the attacker could uniquely identify the victim's records in the group. This is what happened to governor William Weld as mentioned in section2.2.2. Sweeney linked medical data with a voter list, which both included the QID= <Zip,Birth date,Sex >. She then employed the background knowledge that governor Weld was admitted to the hospital at the certain date, which allowed her to uniquely identify him from the small group of people that shared the same QID as him.

k-anonymity Sweeney[41] proposed a notion called k-anonymity in order to try and prevent record linkage through QID. She defined that a table T with a quasi-identifier QI_T would satisfy k-anonymity if and only if each sequence of values $T[QI_T]$ appears with at least k occurrences in $T[QI_T]$. From that definition it appears that k-anonymity is designed to prevent record linkage through hiding the record of the victim in a big group of records with the same QID. This method has a weakness however, as an attacker can still infer a victim's sensitive attribute, such as having the attribute hasDisease=true, if most records in a group have similar values on those sensitive values.

Attribute Linkage[rename]

The aforementioned weakness is an example of an attribute linkage attack. An attacker might not be able to precisely identify the victim through a record, but can still infer his or her sensitive values from the published data. The attacker does this based on the set of sensitive values associated to the group the victim belongs to.

To prevent this type of attack, Machanavajjhala et al[25] proposed an idea based on diminishing the correlation between the QID attributes and the sensitive values, which they called l-diversity. The method requires each group with similar QID to have l distinct values for the sensitive attributes.

2.3.2 Background Information

2.4 Challenges in Data Privacy

Several studies have been performed to assess which privacy risks exists in fields such as mobile applications citations, health care data, and in social networks, and all of them found deficiencies in either the collection or handling of individuals' data. A study run by the European Data Protection Authorities (DPA) found that out of 1211 mobile applications surveyed, 59% caused concern with respect to pre-installation privacy communications, and that 31% requested permissions exceeding what the surveyors would expect based on their understanding of the applications functionality[3].

The law might not necessarily be enough to sufficiently prevent the misuse of personally sensitive information, such as patient's health care data. A study performed by Yale's center for bioethics concluded that: "Law likely cannot catch up with burgeoning data collection, data aggregation, and data mining activities, nor with technological advance, let alone adequately anticipate it." Yet the author also argued that technological progress would lead to "Better alternatives to identification and de-identification; means of tracking data; [...] improved data security; and returning benefit to data originators"[23].

2.5 Motivation

We hope to show that a competitive solution can be created in a distributed learning setting, which also can provide a privacy guarantee for the people who supply the data required for learning. If we are successful, our research can open an avenue of practical solutions where the paradigm in data mining shifts from collecting data in massive centralized databases, to a distributed approach where the data producers also become data owners.

2.5.1 Data security

This project is motivated by the aforementioned challenges and breaches of data privacy, and wish to contribute to the development of privacy-preserving technology. In a world where massive amounts of sensitive personal data are being collected, attacks on the individual's privacy are becoming more and more of a threat.

2.5.2 Data ownership

Addtionally, we are strongly motivated by the idea that there should be a reversal in data ownership. Currently, companies offering services to users collect the data stream generated by a user and store it centrally in a data center owned by the company. The user has to trust that these data centers will not be breached or leaked. Furthermore, the user has to trust that the company policies or ethical standards will not change in the future and that the company or their data will not be bought by a independent third party. If data streams were instead collected in some user-controlled repository, risk of breaches would be reduced and the user would maintain full access control and monitoring. Tim Berners-Lee voiced his support for this idea at the IP EXPO in 2014[11]: "I would like us to build a world in which I have control of my data. I can sell it to you and we can negotiate a price, but more importantly I will have legal ownership of all the data about me,”. He also brought up another compelling reason to ensure that users retain the data they produce[19]: ”In general if you put together all that data, from my wearable, my house, from other companies like the credit card company and the banks, from all the social networks, I can give my computer a good view of my life, and I can use that. That information is more valuable to me than it is to the cloud.” A user will have multiple applications that gather information from their daily life, such as exercise, social and office applications. While each of these data streams on their own can be useful for the companies that collect them, they can have even more powerful uses when put together to give a more complete context. Instead of each company pulling user data to their data centers, users could push data stream to their personal storage, and offer. The user then has control of who accesses the data and how, while also allowing for data analysis across completely separate applications.

2.5.3 Future legal requirements

The European Parliament is working towards new legislation that will create a set of common data protection rules for all EU member states[30]. This legislation offers right to erasure and right to portability. The matter of portability is a step in the direction of the ideas of data ownership discussed in Section 2.5.2. Perhaps most significantly, the regulation requires that all companies operating from the EU or having customers in the EU will be required to comply with. Companies that do not comply can risk being imposed periodic data protection audits or fines up to 100 million or 5% of annual worldwide turnover.

The European Parliament is not alone in looking new laws to regulate data privacys. The Council to the President, an advisory group to the US President, concluded in their 2014 report [21] that preserving privacy values would be their number one recommendation when designing a new policy framework for big data. Furthermore, the so called ”Privacy Bill of Rights” outlined by the Obama administration in 2012 is moving forward, and a new discussion draft was published in 2015[1]. Among the requirements put forward in this bill is transparency about how data is used, the degree of control a person has over how their data is used.

2.6 Related Work

Boutet et al. worked on a privacy-preserving distributed collaborative filtering which relied on user profile obfuscation and randomized response.[7].

Boutsis et al developed a participatory sensing system for smartphones which assumed that the data was distributed locally. Their system was called LOCATE, which handled ensured the privacy of the users, but they did not provide a differential privacy guarantee. [8]

Chaudhuri and Monteloni designed a logistic regression algorithm which guaranteed differential privacy in 2009, but their algorithm was designed to run on a single centralized database[10].

Han et al. investigated the problem of preserving differential privacy in distributed constrained optimization. By creating an algorithm based on stochastic gradient descent, which preserved privacy by adding noise to the public coordination signals (i.e gradients). [18]

Ji et al. [22] recently proposed a distributed solution using logistic regression, which learned from both private and publicly available medical datasets. Their solution differ from our own as they employ a globally synchronized structure, whereas our own solution works asynchronously. Their approach also requires a global aggregation of gradients, compared to our own which employ distributed ensemble learning,

Pathak et al [34] proposed a privacy-preserving protocol for composing a differentially private aggregate classifier. Their protocol trained classifiers locally in different parties, and the parties would then interact with an curator through a homomorphic encryption scheme to create a perturbed aggregate classifier. We took inspiration from their protocol when we created our own ensemble classifier.

Look at the last line and see if it's correct.

Chapter 3

Basic Theory

Common to all the attack vectors described in the previous chapter is that the attacker rely on background knowledge, often also called auxiliary information, to perform their linkage attacks. Protecting a database against this threat has long been a major challenge in database design. Already back in 1977 Tore Dalenius [12] defined a desideratum for data privacy which says that:

> Access to the published data should not enable the adversary to learn anything extra about target victim compared to no access to the database, even with the presence of any adversarys background knowledge obtained from other sources.

This privacy goal was rejected by Cynthia Dwork, who showed the general impossibility of Dalenius' goal due to the existence of auxiliary information. Instead she chose to formulate a probabilistic privacy goal, which places an upper bound on how much the risk of privacy breach can increase by participating in a database.

3.1 Differential Privacy

The term ”differential privacy” was defined by Dwork as a description of a promise, made by a data holder to a data subject: ”You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.” [14] In an ideal situation, databases which implement differential privacy mechanisms can make confidential data widely available for accurate data analysis, without resorting to data usage agreements, data protection plans, or restricted views. Nevertheless, the Fundamental Law of Information Recovery states that overly accurate answers to too many questions will destroy privacy in a spectacular way [14], meaning that data utility will eventually be consumed.

3.1.1 Definition of Differential Privacy

The classic example for explaining a security breach is the case of Mr White: Suppose you have access to a database that allows you to compute the income of all residents in a specified area. If you knew that Mr White was going to move, simply querying the database before and after his relocation would allow you to deduce his income.

Definition 1. The distance of two datasets, $d(D_1, D_2)$, denotes the minimum number of sample changes that are required to change D_1 into D_2.

Formally, differential privacy is defined as follows: A randomized function M gives ϵ-differential privacy if for all data sets D_1 and D_2 where $d(D_1, D_2) = 1$, and all $S \subseteq Range(M)$,

$$Pr[M(D_1) \in S] \leq \mathrm{e}^{(\epsilon)} \times Pr[M(D_2) \in S] \tag{3.1}$$

That is, the presence or absence of a particular record should not affect the probability of any given output of $M(D)$ by more than some multiplicative factor.

Informally, the presence or absence of a single record in a database should not have a noticeable impact on the output of any queries sent to it. Though the existence of the database itself might allow attackers to learn information about a person, opting out of the database will not significantly help reduce the risk of information disclosure. Conversely, participating in the database does not significantly increase the risk of disclosure either, thus fulfilling Dworks promise quoted in the beginning of Section 3.1.

Privacy preserving data analysis platforms such as PINQ[26], Airavat[37] and Fuzz[17] have all implemented features such as privacy budgeting and noise mechanisms to compute useful queries while fulfilling Equation 3.1.

3.1.2 Privacy Budget

The quotient $\frac{Pr[M(D_1) \in S]}{Pr[M(D_2) \in S]}$ measures the extent to which an attacker can ascertain the difference between the two datasets[2]. Sarathy and Muralidhar[38] calls this ratio the ”knowledge gain ratio”. Differential privacy requires that this ratio is limited to e^{ϵ}. This is because as the ratio grows larger, an attacker can determine with greater probability that the query result was obtained from one dataset over the other.

Privacy budgeting was introduced to limit the amount of information a data analyst can obtain about any individual with data records in the dataset. The data analysis platform will track every query to ensure that both individual queries and aggregation queries do not exceed the given budget. This privacy standard forbids further queries to the database once the budget has been consumed.

Defining and depleting a privacy budget is possible due to the sequential composition property of ϵ-differentially private mechanisms, as shown by McSherry[26]. Given N mechanisms M_i that offer $\frac{\epsilon}{N}$-differential privacy, applying each mechanism M_i in sequence offers ϵ-differentially privacy.

3.1.3 Noise Mechanisms

Given a target function f to compute on a database D, it is necessary to design a randomized function M which fulfills Equation 3.1 while yielding a useful approximation to the true f. This randomized function M can be created by adding noise to the computation of f. There are many different mechanisms for applying this noise, but the two most common are the Laplace mechanism and the Exponential mechanism.

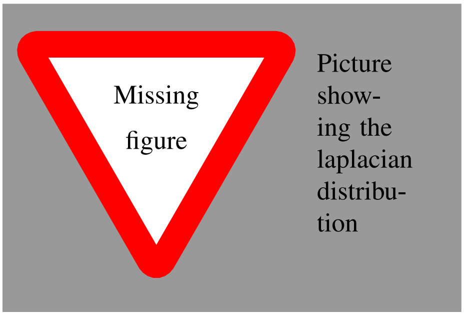
Laplace Mechanism

The Laplace mechanism involves adding random noise which follows the Laplace statistical distribution. The Laplace distribution centered around zero has only one parameter, its scale b, and this is proportional to its standard deviation.

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x|}{b}) \tag{3.2}$$

When using the Laplace mechanism it is necessary to choose a suitable value for the parameter b. Increasing values of b results in increased noise variance. The scale of b is naturally dependent on the privacy parameter ϵ, and also on the effect the presence or absence of a single record can have on the output of function f. This risk is called the sensitivity of the function, and is defined mathematically as:

$$\Delta f = \underset{D_1,D_2}{max} ||f(D_1) - f(D_2)||_1 \tag{3.3}$$



Not sure if we need the pic. Alex: I think we do.

This equation states that the sensitivity Δf is the maximum difference in the values that the function f may take on any pair of databases that differ on only one row. Dwork proved that adding a noise drawn from $Lap(\Delta f/\epsilon)$ to a query, ϵ-differential privacy[14] is guaranteed.

add a section explaining that while it would seem that sensitivity is would often be high, some functions like count are naturally low sensitivity, and that steps such as clamping can be taken to reduce sensitivity

Exponential Mechanism

The exponential mechanism proposed by McSherry and Talwar [27] is a method for selecting one element from a set, and is commonly used if a non-numeric value query is used. An example would be: ”What is the most common eye color in this room?”. Here it would not make sense to perturb the answer by adding noise drawn from the Laplace distribution. The idea of the exponential mechanism is to select the output from all the possible answers at random, with the probability of selecting a particular output being higher for those outputs that are ”closer” to the true output.

More formally, let A be the range of of possible outputs for the query function f. Also, let $u_f(D, a)$ be a utility function that measures how good an output $a \in A$ is as an answer to the query function f given that the input dataset is D (Note that higher values of u_f represents better outputs). The sensitivity function will then be defined as the maximum possible change in the utility function's value u_f due to the addition or removal of one person's data from the input, i.e:

Definition 4: the sensitivity of score function u_f is defined as

$$S(u_f) = \max_{d(D_1,D_2)=1,a\in A} ||u_f(D_1, a) - u_f(D_2, a)|| \tag{3.4}$$

‘

3.2 Logistic Regression

Logistic regression is a popular probability model developed by D.R Cox in 1958. The most basic form is used to predict a binary response based on a set of features.

$$h_\theta(x) = \theta^T x = \sum_{i=0}^{n} \theta_i x_i,$$

The logistic regression model is

$$p(y = 1|\mathbf{x}, \theta) = \frac{1}{1 + exp(-\theta^T \mathbf{x})} \tag{3.5}$$

where θ is the parameters we wish to learn. Following this we want to maximize the following:

$$argmax_\theta \sum_{i=1}^{m} logPr(y_i|x_i, \theta) - \tag{3.6}$$

(1) It can be used for binary classification or for predicting the certainty of a binary outcome. See Cox and Snell (1970) for the use of this model in statistics. This note focuses only on computational issues related to maximum-likelihood or more generally maximum a-posteriori (MAP) estimation. A common prior to use with MAP is: p(w) N (0, 1 I) (2) Using ¿ 0 gives a regularized estimate of w which often has superior generalization performance, especially when the dimensionality is high (Nigam et al., 1999). Given a data set (X, y) = [(x1, y1), ...,(xN , yN)], we want to find the parameter vector w which maximizes:

$$log(1 + exp(y_i \mathbf{w}^T \mathbf{x}_i)) - \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \tag{3.7}$$

The gradient of this objective is

$$g = \Delta_w l(\mathbf{w}) = \sum_i (1 - \sigma(y_i \mathbf{w}^T \mathbf{x}_i)) y_i \mathbf{x}_i - \lambda \mathbf{w} \tag{3.8}$$

Include learning rate in update rule

When m, the number of observations or training examples, is not large enough compared to n, the number of feature variables, simple logistic regression leads to over-fit. A standard technique to prevent over-fitting is regularization, in which an extra term that penalizes large weights is added to the average logistic loss function. The l_2-regularized logistic regression problem is

$$minimize : l_{avg}(v, w) + \lambda ||w||_2^2 = (1/m) \sum_{i=1}^{m} f(w^T a_i + v b_i) + \lambda \sum_{i=1}^{n} w_i^2 \tag{3.9}$$

Here $\lambda > 0$ is the regularization parameter, used to control the trade-off between the average logistic loss and the size of the weight vector, as measured by the l_2-norm. The objective function in this logistic regression problem is smooth and convex, and can be minimized by standard gradient descent.

3.2.1 Sensitivity of Logistic Regression Aggregation Mechanism

In order to built logistic regression models in a way that preserves privacy, it is necessary to determine the sensitivity of the output model. Chaudhuri and Monteleoni[10] showed that the sensitivity of logistic regression is at most

$$\frac{2}{n\lambda} \tag{3.10}$$

where n is the size of the training set and λ is the regularization parameter used in model training.

This solves only the case of training a privacy-preserving logistic regression model on local data. Our research goal involves any number of peers cooperating to build useful models without compromising the privacy of their local data. Pathak, Rane and Raj[34] proposed an approach where locally trained logistic regression classifiers are aggregated by averaging. Secrecy is achieved by using an encryption method to compute the aggregate classifier, ensuring that local data is not shared while allowing a differentially private model to be published. This encrypted computation method is presented in more detail in Section 3.6. It is important to note that the approach of Pathak et al. assumes that the participants are honest-but-curious. This assumption means that participants will follow the established protocol, but will read any information that is somehow available to it. Their method is not robust against malicious sabotage.

When aggregating K locally trained models, their approach computes the final model

$$\boldsymbol{w} = 1/K \sum_{j=1}^{K} \boldsymbol{w_j} + \boldsymbol{\eta} \tag{3.11}$$

where $\boldsymbol{\eta}$ is a noise vector that guarantees ϵ-differential privacy. This noise vector is drawn from the Laplace distribution with parameter $\frac{2}{n_j \epsilon \lambda}$, where n_j is the size of the smallest dataset used in training of the K models. This means that they use a bound on output sensitivity of

$$\frac{2}{n_j \epsilon \lambda} \tag{3.12}$$

which is the same as in Equation 3.10, except that the lowest n_j is used. Since the lowest n_j corresponds to the highest noise variance, this gives protection to all the participants regardless of the size of their dataset.

It is important to note that this does not offer full protection to participants. It only offers differential privacy guarantee for individual records in their data set. This means that aggregate information about a participants data set will go, and in principle sufficient knowledge about the other $d_{i \neq k}$ datasets would allow a third party to learn information. This is a very critical issue, and must be rectified before the system can be applied to data sets were even . For example, an individual might not want insurance companies to know about the averages of features in their biometric records. The current system would not help with that concern - it only protects against specific knowledge about individual biometric records.

3.3 Ensemble learning

In ensemble learning, the predictions of individually trained models are combined to form a final prediction[**?**]. In this project we will be using a variant of ensemble learning called bootstrap aggregating or bagging, as presented by Breiman[9]. Breiman showed that when changes in the training set has a significant effect on the trained model, bagging can give better performance than training a single model on the learning set. Bootstrap aggregating involves creating new learning sets by sampling from the original set with replacement, and training a model on each new set produced. These models are all added to the ensemble, which then makes predictions by taking a majority vote.

We did not strictly use bagging according to its formal definition, as the bootstrap step was not used. In our approach models are instead trained on disjoint subsets of the training set, which are then published after being aggregated according to Equation 3.11. There can be many such models published, so they are added to ensembles and prediction is done in the same fashion as in bagging.

3.4 Cross validation

We initially divided the data sets into a training set and a testing set, the latter being intended to evaluate the performance and properties of our approach. We needed to explore many different combinations and variations of our experiment during, but the test set should only be used as a final step. If the test set is used for repeated validation of different parameters, we would risk overfitting it and getting unrealistic test results.

One way to do reliable accuracy estimation is with cross validation, which makes more efficient use of training data than creating a separate holdout set and has less bias than as shown by Kohavi [**?**]. Cross validation involves partitioning the training set into K disjoint sets. Then, for each $t \in [1, K]$ partition t is used as the test set, and the remaining partitions are combined to form the training set. Accuracy is reported as number of correctly classified instances divided by the total number of instances over all K partitions.

Kohavi recommends 10-fold stratified cross validation. Stratified cross validation involves ensuring that each fold has the same class distribution as the original data set. . Since the data sets we tested with have thousands of records and close to uniform class distribution, we concluded that stratified folds was not necessary. Our experiments were evaluated with 10-fold cross validation with each fold being a random, disjoint subset of the training set.

3.5 Programming Frameworks Used

3.5.1 JADE

To minimize the risk of errors we wanted to implement the experiment in such a way that it was easy to reason about the behavior of the components and identify mistakes. Since the core of our experiment involves peers communicating and cooperating to create predictive models, we decided an agent-based model was suitable.

The Java Agent framework for Distance learning Environments(JADE) is a middleware which facilitates the development of multi-agent systems. An application based on JADE is made of a set of components called agents, where each one has an unique name. Agents execute tasks and interact by exchanging messages between each other. Agents execute on top of a platform that provides them with basic services such as message delivery. A platform is composed of one or more containers, where the containers can be executed on different hosts thus achieving a distributed platform. The Main container is a special container which exists in the platform, as it has two special properties. 1: It must be the first container to start in the platform, and all other containers must register to it. 2: Two special agents are included; the Agent Management System (AMS) which represents the single authority in the platform, and is an agent tasked with platform management actions such as starting and killing other agents. The other special agent is the Directory Facilitator (DF), which provides a directory which announces which agents are available on the platform. This acts like a yellow pages service where agents can publish the services they provide and find other agents providing services they need.

Note that while all containers in a single platform must register with the Main container in that platform, multiple Jade platforms can be instantiated separately and communicate with each other, allowing for scalability of Jade deployments.

Write how we adapted the use of JADE here

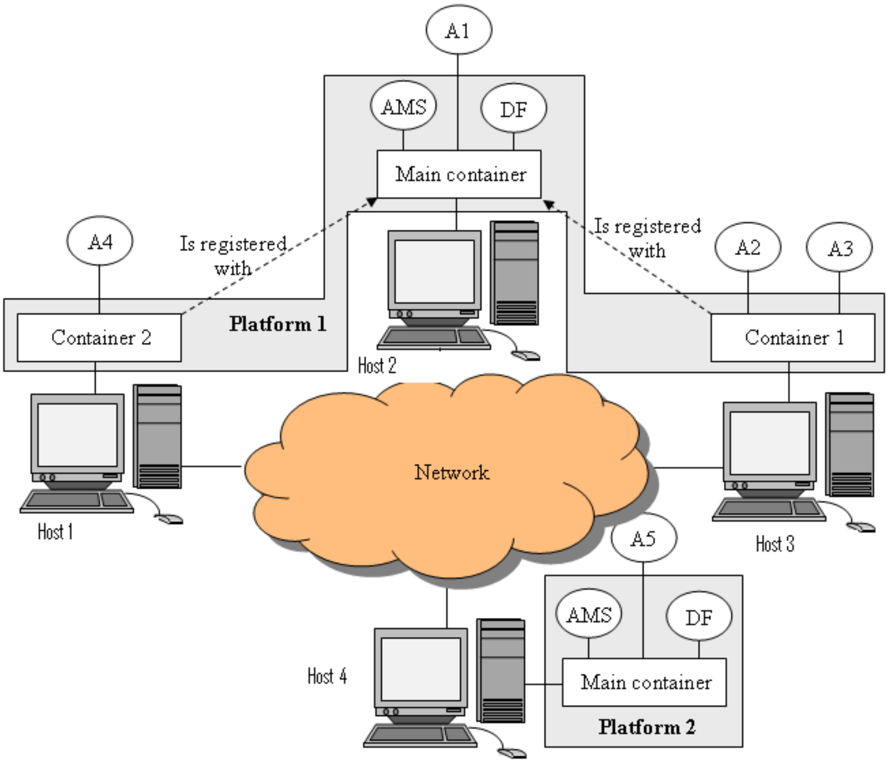


Figure 3.1: JADE Architechture

Consider remaking this figure so we don't have to cite it

3.6 Homomorphic Encryption

Homomorphic encryption is an encryption scheme which allows computations to be carried out on ciphertext, meaning plaintext that has been encrypted using an algorithm and a public key. The result of the computations is also encrypted, and can be deciphered back to plaintext using a private key. This has long been considered as crypthography's holy grail [28], as this would allowing operating on encrypted text without knowing the decryption key. For example, given ciphertexts $C_1 = Enc(Data1)$ and $C_2 = Enc(Data2)$, an additively homomorphic encryption scheme would allow to combine C_1 and C_2 to obtain $Enc_K(Data1 + Data2)$. More concretely this means that if you encrypt your data using such an encryption scheme, you can transfer your data to an untrusted server which can perform some arbitrary computations on that data without being able to decrypt the data itself.

Up until recently, all published homomorphic encryption schemes only supported one basic operation, most commonly addition. These schemes could only be called partially homomorphic, as they did not provide any extensive functionality. The notion of a fully homomorphic encryption schemes was first proposed by Rivest, Adleman, and Dertouzos in 1978 [36], but it wasn’t until 2009 that Craig Gentry published a doctoral thesis where he proved that he had constructed a fully homomorphic scheme[16]. Gentry’s solution was based on ”ideal lattices” as well as a method to double-encrypt the data in such a way that the errors could be handled ”behind the scenes”. By periodically unlocking the inner layer of encryption underneath an outer layer of scrambling, the computer could hide and recover from errors without ever analyzing the secret data.

Do we need to talk about a method of HE that doesn’t really work for our requirements?

The downside of Gentry’s two-layered approach is that it requires a massive computational effort. Bruce Schneier, a leading American cryptographer, pointed out ”Gentry estimates that performing a Google search with encrypted keywords – a perfectly reasonable simple application of this algorithm – would increase the amount of computing time by about a trillion. Moore’s law calculates that it would be 40 years before that homomorphic search would be as efficient as a search today, and I think he’s being optimistic with even this most simple of examples[39].”

3.7 Resource consumption

Give analysis of time and space complexities and how much this taxes a device.

Chapter 4

Experiment

4.1 Overview

As presented in Section 1.1, we wanted to do test an architecture that allows fully decentralized machine learning that maintains the privacy of the participants.

We consider a setting with N peers that each have a local data set. These data sets are assumed to be independently sampled for each peer, but may be sampled from the same distribution. When the system initializes, each peer trains a logistic regression model on its local dataset. The data set and the trained model is private and should only be known by its owner.

After the initialization phase, the aggregation phase begins. In this phase, the aggregation mechanism described in Section 3.2.1 is applied one or more times, using the private model held by each peer. The mechanism is not applied to all peers at the same time. Instead, subsets of peers are selected randomly to form aggregation groups, each group producing a single aggregate model. Many such groups can be formed, and the group size can vary from including all the available peers to including just a single peer. In our experiments, we specify a constant group size which is used until the end of the experiment.

While the output of each mechanism application is an average of the input models, produced in way that guarantees differential privacy, the computation itself must be done in a central manner. This is possible to do securely using the protocol detailed by Pathak et al., which uses homomorphic encryption to compute the model aggregate without allowing the any of the participants to know the original, private model of another participant[34]. Since this protocol requires some central computation, one of the peers is chosen at random to be the curator, responsible for acting as the central party described in the solution by Pathak et al. The other peers in the group will submit the necessary information to this curator, including their private model, in an encrypted fashion. Once the peer acting as curator has received a model from all participants, it computes the average model, adds noise sufficient to guarantee ϵ-differential privacy and publishes the final result. The target of this publish step can vary. In our experiments we have tested one version that publishes a model to all available peers and one that only publishes the model to the peers in the group that helped create it.

Each peer holds a privacy budget, as discussed in Section 3.1.2, that limits how many times it can be involved in a mechanism application. If the budget of a peer is depleted, it is no longer a candidate for the randomly formed aggregation groups. When there no longer is enough peers to form a group with the size specified for a particular experiment, the experiment terminates.

4.1.1 Limitations of current implementation

Certain parts of this system is not implemented in this project, and are replaced by black-box substitutes that simulate the required behavior. We have only done this for components that are already described and tested in other work.

The protocol created by Pathak et al. for computing aggregates securely was not implemented. In our implementation models are sent unencrypted to the peer acting as curator. While this part would need to be replaced with a full implementation of the approach by Pathak et al. the output returned by the curator is exactly the same, using the computation in Equation 3.11.

Finally, the selection of random groups is done in a non-scalable manner. A centralized actor that has full knowledge of all participating peers randomly selects groups of these peers and sends a message to each peer with the list of participants. This should be replaced by a decentralized method for the system to be scalable. How we intend to do this is discussed further in Section 6.2 on Future Work.

4.2 Dataset

This section will introduce the dataset(s) used. What features it contains, what we try to learn/classify, and why we chose to use it.

4.2.1 Spambase

The Spambase dataset [20] was used as a baseline training set. This dataset is publicly available from the UCI machine learning directory, and contains 57 input attributes of continuous format which serves as input features for spam detection and 1 target attribute in discrete format which represents the class.

We chose this dataset as it is a popular dataset to analyze the performance of binary classifiers, so that we could compare the results of other logistic regression classifiers against our own. While this dataset might not seem like the ideal choice for testing a differentially private classifier due to its lack of personal information, we argue that it still fits well for the purpose of demonstration. In a spam-classifying system based on our distributed model, a logistic regression model can be built by training it locally in each user's personal mail folder and then aggregated into an ensemble. That way you can build a diverse spam-classifier without the users having to give up their personal email to a centralized database.

Before we could use the dataset, we needed to use normalization to scale the data to 0-1 range. This is due to the proof in Chaudhuri paper which states the assumption $|X_i| < 1$.

The scaling was based on the formula

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

We appended a feature with constant value 1.0 to all data records, to act as the intercept or bias term.

4.2.2 Australian Credit Approval

Another dataset we used, was the Australian Credit Approval (Statlog), which concerns the approval or disapproval of credit card applications. It is publicly available from the UCI machine learning directory. This is a much smaller set of data, with only 690 samples spread over 14 attributes. The data is useful for binary classification as it contains a good mix of attributes: continuous, nominal with small numbers of values, and nominal with larger numbers of values.

We mostly used this dataset to confirm and double-check the conclusions we drew from the Spambase dataset, as it contains too few data records for it to be 100% ideal for our use. Due to the sparsity of data we had to scale down the amount of peers in each experiment, so that each peer could get enough data to create a decent local classifier. We chose to keep on using this data as one of the motivating examples in Dwork's book[14] of differential privacy, is to protect data holders from insurance and credit card companies.

This dataset was preprocessed in the same manner as was described in Section 4.2.1.

4.3 Parameter tuning

Number of peers P specifies how many different peers participate in the experiment, and necessarily the number of partitions of the training data sets. The training is divided into P parts of equal size.

Rationalize why we have included 1 in the 10-inteval experiments - it is because 1 is a very interesting edge case. Should also talk about the significance of group size 1 in analysis.

Aggregate models are created from local models at each peer through an aggregation process that is performed one or more times with subsets of peers. The parameter g specifies how many peers will participate in a single model aggregation. Since each peer has a unique subset of data, this parameter determines how many partitions of the training set contribute to the published aggregate models. These data partitions do not contribute directly, but indirectly through the aggregation of models trained locally on each partition.

Each peer trains a local logistic regression classifier on its data partition. This requires selection of a learning rate α, a regularization constant λ and a maximum number of iterations of gradient descent I. The learning rate is sensitive to the size of the local training set[43], and should be tuned individually by each peer. We did this by running 3-fold cross validation when each peer fits its local model to identify the best α in the range $[-7, 0]$. 3-fold cross validation was chosen because of both project computer time constraints and experimental data constraints. Each experiment in its entirety is tested with 10-fold cross validation, so it was necessary to reduce local model training time in order to run in a reasonably short time on a single computer. The data constraints is a part of the domain we want to explore. When the amount of data is very small, 3-fold cross validation offers a balance between parameter search reliability and validation set sizes.

In usual data mining applications the regularization λ would be tuned in this manner as well, but the sensitivity of the aggregation mechanism depends on λ, as seen in Equation 3.12. This means that the peers will have to communicate to either agree on a regularization level or to determine the smallest regularization constant to identify the worst case noise level. In our experiments we chose a global regularization level, which was used by all peers. We identified the best λ by testing a coarse grid of powers of 2 whenever we changed the per-peer number of training samples.

Talk about the problem of tuning regularization. We are essentially doing a global selection of regularization. This could be difficult. Better to communicate the minimum, perhaps? But then, which regularization should peers pick locally? The highest one that stil has good performance?

The privacy parameter ϵ determines the level of privacy for each data partition. Note that this parameter does not apply to the original training set as a whole - each peer has its own private database, which is protected by ϵ-differential privacy.

Finally, the parameter ϵ can be divided across several applications of the aggregation mechanism, as described in Section 3.1.2. This was achieved with a per-aggregation parameter ϵ_i. Each data partition can participate the aggregation mechanism n times, where $n\epsilon_A \leq \epsilon$.

4.4 Validation

The test sets set aside could not be used when tuning and evaluating system hyperparameters. In order to explore the effects of the various hyperparameters we used cross validation with number of folds $n = 10$. For a given combination of hyperparameters, performance metrics were measured as their average across ten repetitions. In repetition i, data fold i was used as validation set and the remaining $n - 1$ data folds were combined to form the test set.

4.5 Algorithm

This section explain the logistic regression algorithm, how it is commonly used, and what modifications are needed when used in a distributed setting. Explanation on how it is used in a differentially private manner is explained in the architecture section.

4.5.1 Application of Aggregation Mechanism

The central element in our experiments is the aggregation mechanism A, which takes a set of models. This mechanism is given in Algorithm 1. As presented in Section 3.2.1, the sensitivity of logistic regression depends on the sizes of the data sets used to train the models. Specifically, the mechanism needs to know the size of the smallest training set in order to guarantee differential privacy. It is important to note that the method we are testing assumes honest-but-curious participants, as assumed by Pathak et al[34].

```
Input: $\epsilon$ - privacy parameter $M$ - set of models trained by participating peers;
$N$ - set of peer training set sizes;
$\lambda$ - regularization level used when training each model in $M$;
Output: Perturbed aggregate of the models in $M$
$n_{min} \leftarrow min(N)$;
$\eta \leftarrow Laplace(0; \frac{2}{n_{min}\epsilon\lambda})$;
$model_{agg} \leftarrow 1/K \sum_{j=1}^{|N|} w_j + \eta$;
return $model_{agg}$
```

Algorithm 1: ϵ-differentially private aggregation mechanism

```
Input: $P$ - the set of peers;
$\epsilon$ - privacy parameter;
$\epsilon_A$ - privacy level of a mechanism application;
$A$ - the $\epsilon_A$-differentially private aggregation mechanism;
$group\_size$ - number of peers in a single mechanism application
for $peer \in P$ do
    $budget_{peer} \leftarrow \epsilon$;
end
while $|P| \geq group\_size$ do
    $group \leftarrow randomSample(P, group\_size)$;
    $model_{agg} \leftarrow A(group)$;
    for $peer \in group$ do
        $budget_{peer} \leftarrow budget_{peer} - \epsilon_A$;
        if $budget_{peer} < \epsilon_A$ then
            $P \leftarrow P \setminus peer$;
        end
    end
    $publish(P, model_{agg})$
end
```

Algorithm 2: Distributed training process

4.5.2 Propagation Of Published Models

Originally in our system, aggregated models were only propagated to the peers that had participated in creating that model, as can be seen in Figure **??**. What resulted from this, especially when epsilon was set to a low amount such as 0.1 or lower, was that the high amount of noise made the classifiers have a big standard deviation on their mean classification rate. What this meant was that while the classifiers could be very accurate in some peers, classifying up towards 90% accuracy, it could also be significantly worse in other peers.

We theorized that we could improve the ensemble classifier in each peer if we could propagate the aggregated models to all the peers in the network, instead of just those who had participated in making them. Our hypotheses was that this would lead to more stable classifiers with lower standard deviation, due to a smoothing effect in having more models in the ensemble classifier in each peer. This is basically the same idea as bootstrap aggregating, or bagging, which has been proven to lead to improvements in unstable procedures[9].

Definitely talk more about the bagging effect, either here or in the analysis section

For this reason, we decided to run experiments to compare the different possible model. In all cases, the published models will have been perturbed with Laplacian noise to give ϵ-differential privacy. In the group publication setting, only the peers that join together to produce a perturbed model will receive the final result. In the full publication setting, all peers active in the network will receive all perturbed models.

Note that there is no selection or pruning of the ensemble classifier owned by each peer. If a peer receives a model, it will blindly add it to the ensemble. This means each peers ensemble model will grow much faster in the full publishing setting, and they will all contain essentially the same models, the only exception being the unperturbed model produced by the peer locally. We anticipated that this would lead to a reduction in ensemble model accuracy variance.

4.6 Architecture

We designed a distributed system using the JADE framework. The core component in this system is a PeerAgent, which represents a participant in the distributed learning setting. This agent contains what would be the local data of a person using some application. In the remaining sections, whenever we say ”peer” we are refering to the PeerAgent described here, holding a local data set and with means of communcating with other PeerAgent instances.

To form aggregate models it is necessary to select groups of peers to create each model. In our experiment, this is implemented with a singleton agent we named the GroupAgent. This agent draws random subset. The size and number of groups formed is given by the parameters selected at the beginning of the experiment, as specified in Section 4.3. It is this agent that is responsible for keeping track of

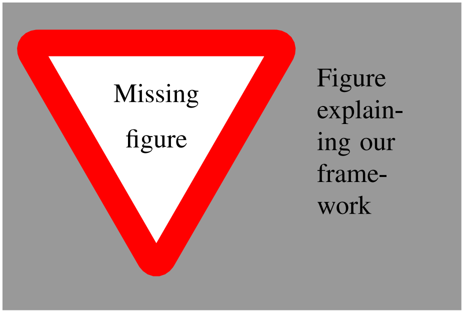


Figure explaining our framework

Chapter 5

Analysis

Make a table with baseline classification rate for logistic regression, put in our values. 1 Peer with no noise, 1 peer with noise, best case from other experiments that we want to highlight. This can lead to a discussion of why our results are not as good, where they show promise etc

All prediction results given in this section are presented with mean and standard deviation values. These values are computed by evaluating each combination of parameters with 10-fold cross validation and taking the mean and standard deviation of accuracy across the 10 data folds.

As explained in section 3.1, differential privacy works by disguising an individual's data in a dataset by adding noise to their records. The amount of noise added is determined by the privacy parameter ϵ, and grows exponentially the closer the parameter gets to zero.

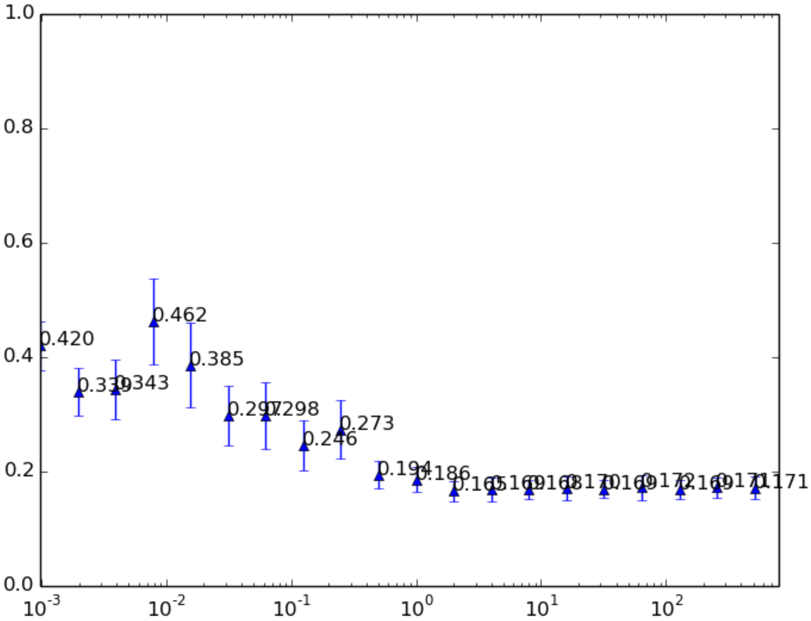


Figure 5.1: $\epsilon = [10^{-3}, 10^{3}]$, $\lambda = 2^{-4}$, 50 peers, 1 aggregation

Figure 5.1 shows the effect of the privacy parameter ϵ in our experiment. We wanted to test the effect of varying the ϵ-value in the range from 2^{-10} to 2^{9}, especially to find out how the classifier would perform when faced with data with high amount of noise added to it. The positive class rate in the UCI Spambase dataset is 0.4, so any error rate at this level is no better than a random classifier. The plot shows how sensitive output is to the values of ϵ.

Figure 5.3

5.1 The importance of data

One of the more important findings for Data is important, the more data a peer have, the greater the chance is that it will make a decent local classifier. J

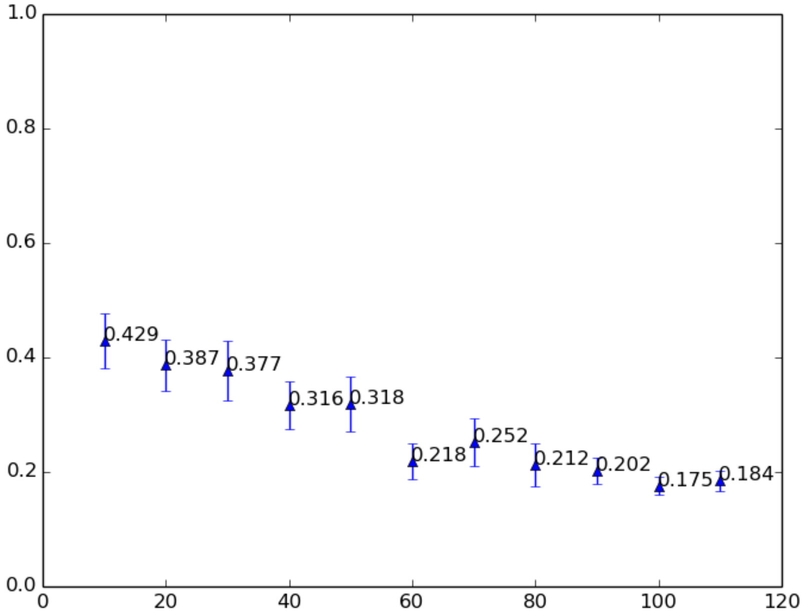


Figure 5.2: $\epsilon = [0.3], \lambda = 2^{-2}$, 30 peers, 1 aggregation

Figure 5.2 indicate how the classification performance improves as the amount of data records available to each peer grows. When each peer only have a small amount of data to create their local classifier, the classifier tends to have display terrible performance. As peers gain more data, both the performance and the variance of the classifier improves.

The reason for this improvement is two-fold. 1: A bigger sample size for the logistic regression model generally leads to better performance [35]. 2: The sensitivity of logistic regression (see equation 3.10) is bounded by the size of the training set. What this means is that the more data a peer have available, the less amount of noise is needed to obfuscate their logistic model.

The observation we've made is therefore thoroughly grounded in theory, and is important to highlight when discussing our main research question. Until a certain amount of data has been gathered, a system based on our distributed architecture will display very poor performance. What is interesting however, is that the amount of data needed seems to be a relatively low number. In the paper written by Pathtak et al[34], they report that each party was given at least 3256 data records. In our experiments we found that a much smaller amount of data could still be used and create decent classifiers. We believe this is due to our choice of propagating every model out, so that each peer can create an ensemble of classifiers.

Fix this last sentence. It is not that good

5.2 Importance of regularization

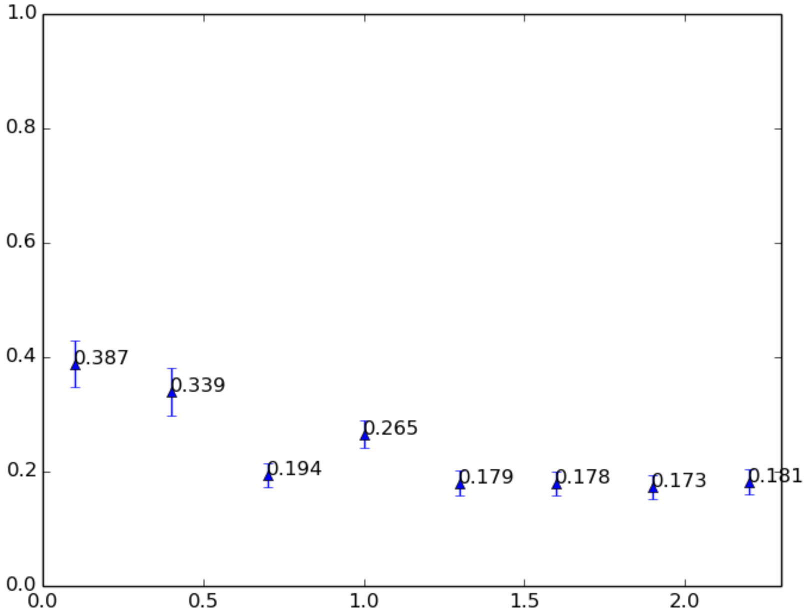


Figure 5.3: $\epsilon = [0.1, 2.2], \lambda = 2^{-4}$, 50 peers, 1 aggregation

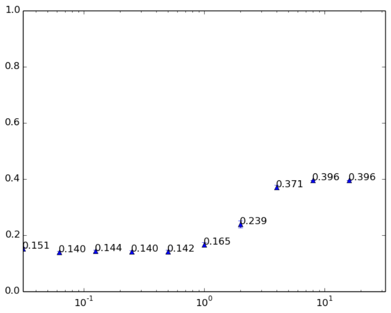


Figure 5.4: $\epsilon = 2^{10}, \lambda = [2^{-5}, 2^{4}]$, 50 peers, 25 aggregations, publish to participants

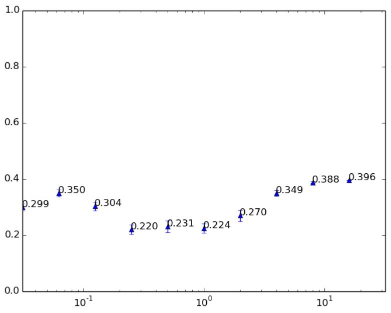


Figure 5.5: $\epsilon = 0.1, \lambda = [2^{-5}, 2^{4}]$ 50 peers, 25 aggregations, publish to all

Figure 5.5 shows the normal effect regularization has on accuracy for the spambase data set, by setting ϵ so high that noise is essentially nonexistent. As the regularization parameter λ grows large, the model becomes less able to fit the training data, eventually resulting in models predicting only the negative class, which constitutes 60% of the data set. This happen because the high regularization forces the parameter vector to the zero vector, resulting in uniform class probability for all samples.

On this particular dataset it appears that a logistic regression model is not at risk of overfitting, since the cross validated error does not increase when the level of regularization is very low. Ignoring the effects of privacy mechanisms, this would mean that selecting some regularization parameter in the range $[10^{-5}, 10^{-2}]$ could be acceptable. Choosing a level at the high end of this range could be a good idea, to reduce the risk of overfitting

Figure 5.4 shows mean accuracy over a range of λ similar to Figure 5.5, where ϵ is set to a level where the noise variance still has an effect on prediction accuracy, as seen in figure 5.1.

When noise with significant variance is added to the model creation process, tuning λ will adjust noise variance as well. Equation 3.12 states that the noise variance is inversely proportional to λ. The choice of regularization then must balance the model flexibility at lower levels of λ with the decreased noise at higher levels of lambda.

Talk about how it is interesting that regularization now must respect something other than just model performance, and offer some guidelines on how regularization should be considered, especially in a setting like ours, with many independent parties

5.2.1 Analysis of Propagation and group size

As mentioned in Section 4.5.2, we hypothesized that we could improve the overall classification accuracy of our system by publishing the aggregated models generated in phase 2.In this section we present the effects of publishing method. Figure 5.6 shows the case where each group of peers only share the perturbed, aggregate model among themselves, while Figure 5.7 shows the results the model is sent to all existing peers.

Rewrite this if we don't use phases

The most obvious effect of globally publishing models is that the standard deviation is much lower than the group publishing case. This is not surprising.

The truth is probably somewhere in the middle - we can't expect to easily publish models globally in all settings. Additionally, there might be situations were global publishing could be detrimental to real world performance - for instance, if there are strong geographical, temporal or demographic trends, it might be better to limit the amount of model sharing to suitable subsets.

As the previous experiment indicates that publishing newly made models to as many peers as possible is better , we performed an additional experiment to determine whether it in such a scenario would be better to perform many aggregations with fewer models included in each aggregate or performing few aggregations including many models. This was achieved by testing performance with a range of values. In the experiment, each peer can only participate in a single aggregation before reaching the limit set by the privacy guarantee. For example, given a set of 50 peers, a aggregation size of 25 can only publish two aggregated models.

Add a reflection talking about why this makes sense.

The results of this experiment is seen in Figure 5.8. It is clear that a smaller group size and consequently more aggregated models published resulted in a strong reduction in accuracy variance. While the variance is too high at larger group sizes to know much about the actual mean value after only 10 repetitions of the experiment, it is unlikely that it is smaller than the mean accuracy observed when the group size is one.

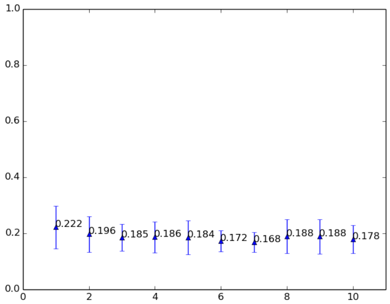


Figure 5.6: $\epsilon = 1.0, \lambda = [2^{-5}, 2^{4}]$ 50 peers, 25 aggregations, publish to participants

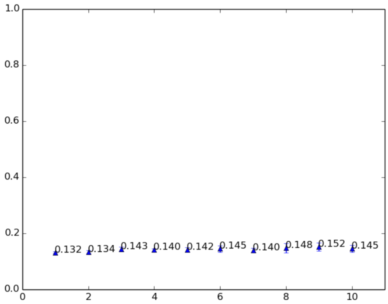


Figure 5.7: $\epsilon = 1.0, \lambda = [2^{-5}, 2^{4}]$ 50 peers, 25 aggregations, publish to all

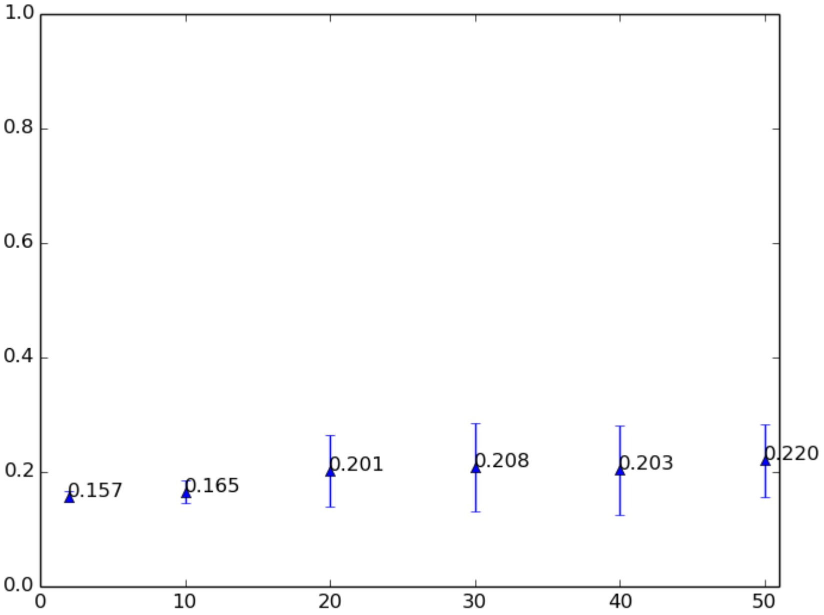


Figure 5.8: Spambase. $\epsilon = 1.0, \lambda = [1.0]$, 50 peers, aggregation sizes in range [2, 50], 66 samples per peer

A possible explanation of this result is that the effects of boosting counters the loss in accuracy that results from the addition of noise. Since models will have noise added to them before being published, expending all data to produce a single model might yield a worse classifier than partitioning data, adding noise to each separate, weaker model and combining them in an ensemble. On the other hand, this effect could be caused by some leakage of privacy which becomes visible over repeated applications of the aggregation mechanism.

Figure out if there is some way we could test or prove that this is not the case.

The interesting observation that can be made from this is that the best situation is when there is no aggregation. A group size of one results in only a single model being contributed, and is equivalent to each peer publishing its local model with noise. One possible reason for this could be that there simply is no value in averaging models in the way done in our experiments and by Pathak et al.[34]. Neither our experiment or the experiment by Pathak et al. help distinguish between these two possibilities. The experiment by Pathak et al. only demonstrate that their method for creating aggregated models has comparable performance to adding noise to a centrally computed model. Additionally, this is only demonstrated with large data sets. In their experiment, the minimum data set size for any participant is 3256. With data sets this large, it is possible they would have gotten similar results by testing a model produced by a single participant, without performing additional aggregation. However, no experiment evaluating this possibility remains. Their theoretical conclusions stand, but experimentally validating the value of aggregation is necessary.

Thus the key question is whether or not aggregation is worth the complexity of a homomorphic encryption protocol. If similar performance can be achieved solely by ensemble classifiers of differentially private models published by each peer, one could skip the complexity and risk of relying on a cryptographic protocol to maintain privacy.

Another possible explanation for this observation is that aggregation might be useful, but not when the peers all have samples of data from the exact same distribution. This is the case with the Spambase data set used in the experiment in Figure 5.8.

One way to answer question would be by finding a data set which has subsets that are produced by distinct distributions, and partitioning data by source distribution. Due to time constraints we did not have time to locate and prepare data sets that fit this requirement.

An alternative avenue for validating the value of doing aggregation could be to reduce the amount of data given to each peer. If just one peer has data sufficient to train a good classifier, it is sufficient. For that reason it makes sense to stage a situation were it is highly unlikely that even a single peer gets a lucky subset of data. Figure 5.9 demonstrates such a case. Again, there is not clear indication that averaging multiple models is better than simply publishing them individually and using them in ensemble classifiers.

5.2.2 Issues with cold start

Explain what cold start issues is. Discuss how this is very prominent in our sitaution, and the different ways we can deal with the cold start, and pros and cons of those approaches.

5.3 Potential Future applications

This section will discuss the potential application of a system based on our distributed machine learner.

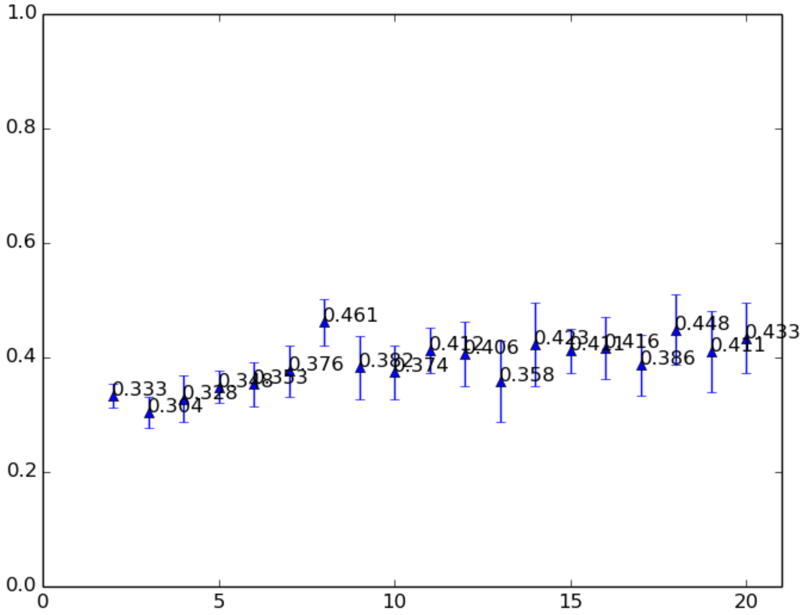


Figure 5.9: Spambase. $\epsilon = 1.0$, $\lambda = [1.0]$, 50 peers, aggregation sizes in range [2, 50], 20 samples per peer

5.3.1 Health

A growing worldwide market is the sale and usage of wearable sensors, such as environmental sensors, motion sensors, and health sensors. A IHS report [31] from 2014 estimates that the market for sensors in wearables will expand to 135 million units in 2019, up from 50 million in 2013. These wearables will evolve from being just a single purpose device such as a pedometer and grow into more multipurpose devices such as a smartwatch, which will consist of several sensors which can monitor several components within its area of use.

The wearable devices are implementing fitness and health monitoring by using a mixture of sensors, such as motion, pulse, hydration and skin temperature sensors. All of these wearables will therefore generate a massive amount of data about the person who are using them. This data can be considered as highly sensitive information, as it can unveil a lot about their user's health, and the manufacturers of these devices knows this. Dana Liebelson, a reporter for Huffington Post, queried several US-based fitness device companies about their privacy. One of the replies she got, was that ”the company does not sell information collected from the device that can identify individual users”, but that they were considering marketing aggregate information that cannot be linked back to an individual. As we saw in section 2.2 and 2.3, many of the popular methods for aggregating and anonymizing a dataset carries an inherent risk of a privacy breach.

This is where we see a potential application for our distributed framework, as it solves the problem faced by those who don't want to give up control of their data to a third party, but they still want to analyze their data as it can be highly useful.

-Future homes can potentially track you through you phone, or similar device.

-You don't want to send this data to someone else, but what you can learn from the data can be highly useful in your daily life if analyzed.

Kevin Fong and the England rugby team. Monitor heart rate, step balance, and a lot of other factors. Can pick up injuries and illnesses well before any doctor. This will trickle down into daily use over the next decade, and let people potentially discover illnesses before they even occur.

5.3.2 Private sharing of business data

A potentially interesting and lucrative market can be found in facilitating the sharing of data between businesses in a private manner. Our motivating example is found in the business of oil market analysis, where competing firms gather a lot of data about oil price, rig placements, supply ship availability, and more. They use this data to create analytical models which help them in their work, and they also sell this information to external clients. Often the firms would like to collaborate their models or their data with their competitors. This could be to validate that they are seeing the same trends or any other reason, but due to the sensitive nature of their data and their fear of losing a competitive edge, they cannot do this in a practical way.

It is in a situation like this that our distributed learner could be applied, and allow the sharing of data between competitors as our system could provide a privacy guarantee to all of the participants. The participants would never lose control of their data, as all they would need to install a program that allows them to connect as a peer in our network: No third party would ever need access to their data.

Chapter 6

Conclusion

6.1 Threats To Validity

6.1.1 Platform

A potential threat to the validity of our conclusion/work , is how we performed the setup of the Jade platform. Since we wanted to perform our experiments on over a range of parameters, we needed to find a way to reset the platform after a successive experiment and re-run it with a new set of configurations. We solved this by having a jade agent called CompletionAgent be responsible for waiting for every peer to message indicating their completion, which would trigger the CompletionAgent to deregister all the peers from the MainContainer and then reset the whole environment. The environment would then be set up again with new parameters.

Fix this sentence

What we see as a potential source for concern in this process is the possibility for error during the deregistration. During the implementation of this process we encountered some problems in making it work, as the CompletionAgent seemed to take an unreasonable amount of time in completing its purpose. Although we found a solution to this problems, there is still a risk that peers do not deregister as they should and carry through into the next iteration of testing. This could lead to false information being injected into our experiment, which would skew our results.

We have however minimized this risk by continuously developing unit test to verify new code additions, as well as using JADE's native GUI to supervise the behavior of the peers while running. We therefore conclude that the risk is negligible.

6.1.2 Resource Consumption

A clear weakness of our system is our lack of formal analysis of resource consumption and scalability. In

What have we done to to guarantee scalability? Is it enough to say it should scale well?

Due to time constraints, we've had to take certain shortcuts while implementing our systems. The most glaring liability for system scalability is our use of a single GroupFormingManager to handle allocating aggregation groups for the participating peers. This manager would quickly become a bottleneck in our system if we wanted to scale the amount of peers beyond just a small mass of users. Our solution to this predicament is found in section 6.2, where we provide a solution in the form of the Newscast algorithm.

6.1.3 Homomorphic encryption

As mentioned in Section 3.6, homomorphic encryption is still in an infantile stage of development and therefore cannot be called a well-proven technology. Our method for aggregating models from various peers is based on a homomorphic encryption scheme developed by Pathak et al. but due to time constraints we could not actually implement it and instead had to opt for simulating the results of applying this scheme. We therefore do not have real-world results that can validate the applicability of this scheme, nor do we know if applying this scheme would lead to increased run-time and resource consumption. While this remains an interesting area for future research, as it stands now it remains a possible threat to the validity of the results we've achieved and therefore also the conclusion we have drawn from them.

6.2 Future Work

Further develop and test the propagation of aggregated models. We experienced that when we shared the aggregated models globally in our network, we could decrease the SD in our classification error, as well as sometimes improving the classifier. Further research should go in expanding this behavior, as you could potentially propagate models only to peers in geographic and/or demographic vicinity. This could possibly lead to more specialized models, which could give better classification rate to a specialized subset of peers.

Another important area of research would be to further test the applicability of peers sharing data to create better aggregated models. Our original research questions was designed to explore the validity of our proposed method of doing differentially private machine learning, and our current research has been limited to testing on a small amount of datasets which is publicly available. In the future more research is needed on datasets with an uneven underlying distribution, which could potentially provide results highlighting the usefulness of sharing information between peers. An ideal dataset would be one where each peer only holds data which makes up only a part of the solution.

text

This last sentence is crap and I hate it

Implement the Newscast algorithm for selecting peers. The Newscast algorithm is a gossip protocol which facilitates a robust spread of information. The core of the protocol involves periodic and pairwise interaction between processes. Implementing this algorithm would allow our system to scale better when a big number of peers are added to the network. The biggest bottleneck of our system at the moment is the peer sampling during the group forming, as it requires a single agent to act as a manager for how groups are formed. The basic idea of the Newscast algorithm is that each node, or peer in our situation, has a partial view of the system. All nodes exchange their views periodically, which allows them to keep an up-to-date view locally and spread their information throughout the network. Further research into this algorithm would allow us to customize this algorithm so that peers in our network could form groups based on their partial views of the network.

Full data protection for each peer's data. This would involve dividing the epsilon by the biggest dataset size, as formalized by Dwork in . This is an ever tighter privacy guarantee, but it would potentially mean that the results would contain too much noise. To test this we would need a massive dataset, as we would need to test the correlation between dataset size, and amount of noise added to each peer. (More noise needs more data to smooth out.)

citation

Real world case which takes humans into account. Right now research in privacy is all about the technical details, and try to get it as close as possible to existing methods. Without some kind of popular support, the method will never see practice in real-world applications.

Work on a system that would work in a online setting. It could potentially improve the system, as you would have new data coming in which could replace old data with spent budgets, but it would also be potentially a big tradeoff as you won't have the same data history as you would have in a system without differential privacy. Dwork has written about this in her book, so we can take inspiration from there. (As well as our own paper)

Security mechanisms for stopping sabotage. In our current system we have assumed that the peers will be honest-but-curious when sharing their data, meaning that we have no way of detecting dishonest peers. In a real world system there would need to be safeguards against people which intend to either destroy the validity of the classifier created by feeding misinformation into the system, or people who tries to intercept and expose the data from other peers. Potential research ares would be intrusion detection in distributed systems, fraud detection, trust networks and reputation systems, and further research into encryption.

Notes

Bibliography

[1] , Feb. 2015. Administration Discussion Draft: Consumer Privacy Bill of Rights Act of 2015.
URL https://www.whitehouse.gov/sites/default/files/omb/legislative/letters/cpbr-act-of-2015-discussion-draft.pdf

[2] Abowd, J. M., Vilhuber, L., 2008. How protective are synthetic data? In: Privacy in Statistical Databases. Springer, pp. 239–246.

[3] Authorities, E. D. P., Sep. 2014. European results of the 2014 global privacy enforcement network sweep.
URL http://dataprotection.ie/docimages/GPEN_Summary_Global_Results_2014.pdf

[4] Barth-Jones, D. C., 2012. The're-identification'of governor william weld's medical information: a critical re-examination of health data identification risks and privacy protections, then and now. Then and Now (June 4, 2012).

[5] Bell, R. M., Koren, Y., Dec. 2007. Lessons from the netflix prize challenge. SIGKDD Explor. Newsl. 9 (2), 75–79.
URL http://doi.acm.org/10.1145/1345448.1345465

[6] Bonchi, F., Ferrari, E., 2010. Privacy-aware Knowledge Discovery: Novel Applications and New Techniques. CRC Press.

[7] Boutet, A., Kermarrec, A.-M., Frey, D., Guerraoui, R., Jegou, A., 2013. Privacy-Preserving Distributed Collaborative Filtering.
URL https://hal.inria.fr/hal-00799209/file/RR-8253.pdf

[8] Boutsis, I., Kalogeraki, V., Mar. 2013. Privacy preservation for participatory sensing data. In: 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, pp. 103–113.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6526720

[9] Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

[10] Chaudhuri, K., Monteleoni, C., 2009. Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems. pp. 289–296.
URL http://papers.nips.cc/paper/3486-privacy-preserving-logistic-reg

[11] Curtis, S., Oct. 2014. Sir Tim Berners-Lee calls for new model for privacy on the web.
URL http://www.telegraph.co.uk/technology/internet/11148584/Tim-Berners-Lee-calls-for-new-model-for-privacy-on-the-web html

[12] Dalenius, T., 1977. Towards a methodology for statistical disclosure control. Statistik Tidskrift 15 (429-444), 2–1.

[13] Dragland, Å., 2013. Big data–for better or worse. SINTEF, retrieved on July 22.

[14] Dwork, C., Roth, A., 2013. The algorithmic foundations of differential privacy. Theoretical Computer Science 9 (3-4), 211–407.

[15] Fung, B. C., Wang, K., Fu, A. W.-C., Yu, P. S., 2010. Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, 1st Edition. Chapman & Hall/CRC.

[16] Gentry, C., 2009. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing. STOC '09. ACM, pp. 169–178.
URL http://doi.acm.org/10.1145/1536414.1536440

[17] Haeberlen, A., Pierce, B. C., Narayan, A., 2011. Differential privacy under fire. In: Proceedings of the 20th USENIX Conference on Security. SEC'11. USENIX Association, Berkeley, CA, USA, pp. 33–48.
URL http://dl.acm.org/citation.cfm?id=2028067.2028100

[18] Han, S., Topcu, U., Pappas, G. J., Nov. 2014. Differentially Private Distributed Constrained Optimization.
URL http://arxiv.org/abs/1411.4105

[19] Hern, A., Oct. 2014. Sir Tim Berners-Lee speaks out on data ownership.
URL http://www.theguardian.com/technology/2014/oct/08/sir-tim-berners-lee-speaks-out-on-data-ownership

[20] Hopkins, M., Reeber, E., Suermondt, J., 1999. UCI machine learning repository, spambase data set.
URL https://archive.ics.uci.edu/ml/datasets/Spambase

[21] House, W., 2014. Big data: Seizing opportunities, preserving values.

[22] Ji, Z., Jiang, X., Wang, S., Xiong, L., Ohno-Machado, L., Jan. 2014. Differentially private distributed logistic regression using private and public data. BMC medical genomics 7 Suppl 1 (Suppl 1), S14.
URL `http://www.biomedcentral.com/1755-8794/7/S1/S14`

[23] Kaplan, B., 2014. Patient health data privacy. Yale University Institute for Social and Policy Studies Working Paper, 14–028.

[24] Kumar, R. K., Poonkuzhali, G., Sudhakar, P., 2012. Comparative study on email spam classifier using data mining techniques. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. pp. 14–16.

[25] Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M., 2007. l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1), 3.

[26] McSherry, F., June 2009. Privacy integrated queries. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD). Association for Computing Machinery, Inc., for more information, visit the project page: http://research.microsoft.com/PINQ.
URL `http://research.microsoft.com/apps/pubs/default.aspx?id=80218`

[27] McSherry, F., Talwar, K., Oct 2007. Mechanism design via differential privacy. In: Foundations of Computer Science, 2007. FOCS '07. 48th Annual IEEE Symposium on. pp. 94–103.

[28] Micciancio, D., Mar. 2010. Technical Perspective: A First Glimpse of Cryptography's Holy Grail.
URL `http://cacm.acm.org/magazines/2010/3/76275-technical-perspective-a-first-glimpse-of-cryptographys-holy-g`
`fulltext`

[29] Narayanan, A., Shmatikov, V., 2008. Robust de-anonymization of large sparse datasets. In: Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, pp. 111–125.

[30] News, E. P., Mar. 2013. Q&A on EU data protection reform.
URL `http://www.europarl.europa.eu/news/en/news-room/content/20130502BKG07917/html/QA-on-EU-data-protection-reform`

[31] Nissil, S., Bouchaud, J., Boustany, M., Oct. 2014. White paper: Mems & sensors for wearables report. Tech. rep., IHS Technology.
URL `https://technology.ihs.com/496122/mems-sensors-for-wearables-2014`

[32] Pandurangan, V., Jun. 2014. On taxis and rainbows: Lessons from nycs improperly anonymized taxi logs. Visited: 2016-03-06.
URL `https://medium.com/@vijayp/of-taxis-and-rainbows-f6bc289679a1`

[33] Parliament, E., of the European Union, C., Apr. 2006. Directive 2006/24/ec.
URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF

[34] Pathak, M., Rane, S., Raj, B., 2010. Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers. In: Advances in Neural Information Processing Systems. pp. 1876–1884.
URL http://papers.nips.cc/paper/4034-multiparty-differential-privacy

[35] Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., Feinstein, A. R., 1996. A simulation study of the number of events per variable in logistic regression analysis. Journal of clinical epidemiology 49 (12), 1373–1379.

[36] Rivest, R. L., Adleman, L., Dertouzos, M. L., 1978. On data banks and privacy homomorphisms.

[37] Roy, I., Setty, S. T. V., Kilzer, A., Shmatikov, V., Witchel, E., 2010. Airavat: Security and privacy for mapreduce. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation. NSDI'10. USENIX Association, Berkeley, CA, USA, pp. 20–20.
URL http://dl.acm.org/citation.cfm?id=1855711.1855731

[38] Sarathy, R., Muralidhar, K., Apr. 2011. Evaluating laplace noise addition to satisfy differential privacy for numeric data. Trans. Data Privacy 4 (1), 1–17.
URL http://dl.acm.org/citation.cfm?id=2019312.2019313

[39] Schneier, B., Jul. 2009. Networks (2nd ed.).
URL https://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html

[40] Sharma, S., Arora, A., 2013. Adaptive approach for spam detection. IJCSI International Journal of Computer Science Issues 10 (4), 23–26.

[41] Sweeney, L., 2002. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (05), 557–570.

[42] Tockar, A., Sep. 2014. Riding with the stars: Passenger privacy in the nyc taxicab dataset. Visited:2016-03-06.
URL http://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/

[43] Wilson, D. R., Martinez, T. R., Dec. 2003. The general inefficiency of batch training for gradient descent learning. Neural Netw. 16 (10), 1429–1451.
URL http://dx.doi.org/10.1016/S0893-6080(03)00138-2

Appendix

Write your appendix here...