MACHINE LEARNING OVER ENCRYPTED DATA WITH FULLY HOMOMORPHIC ENCRYPTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞEGÜL KAHYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

AUGUST 2022

Approval of the thesis:

MACHINE LEARNING OVER ENCRYPTED DATA WITH FULLY HOMOMORPHIC ENCRYPTION

submitted by **AYŞEGÜL KAHYA** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk-Kestel
Dean, Graduate School of **Applied Mathematics** _______________

Assoc. Prof. Dr. Oğuz Yayla
Head of Department, **Cryptography** _______________

Prof. Dr. Murat Cenk
Supervisor, **Cryptography, METU** _______________

Examining Committee Members:

Assoc. Prof. Dr. Oğuz Yayla
Cryptography, METU _______________

Prof. Dr. Murat Cenk
Cryptography, METU _______________

Assist. Prof. Dr. Eda Tekin
Mathematics, Karabük University _______________

Date: _______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:	AYŞEGÜL KAHYA
Signature :	

ABSTRACT

MACHINE LEARNING OVER ENCRYPTED DATA WITH FULLY HOMOMORPHIC ENCRYPTION

Kahya, Ayşegül
M.S., Department of Cryptography
Supervisor : Prof. Dr. Murat Cenk

August 2022, 40 pages

When machine learning algorithms train on a large data set, the result will be more realistic. Big data, distribution of big data, and the study of learning algorithms on distributed data are popular research topics of today. Encryption is a basic need, especially when storing data with a high degree of confidentiality, such as medical data. Classical encryption methods cannot meet this need because when texts encrypted with classical encryption methods are distributed, and the distributed data set is decrypted using the same key, the result is corrupted. Homomorphic encryption methods are suitable for this job because they allow polynomial operations on the ciphertext. The encryption key distribution to all these devices is a problem here, and the reliability of these devices' access to highly confidential plain text needs to be evaluated. Since homomorphic encryption allows polynomial operations and some machine learning algorithms are polynomial-based, training machine learning algorithms directly on the ciphertext could be a solution. Logistic regression is one of the polynomial-based machine learning algorithms. In this thesis, a logistic regression algorithm is trained on a data set containing various patient information. It was seen that the algorithm predicted with a 77.2 percent success rate whether people would be diagnosed with diabetes within five years or not. Afterward, the data set was encrypted using the CKKS fully homomorphic encryption method. While working logistic regression over the encrypted dataset, it is needed to use some approximations on the algorithm. We wanted to see the results of the applied approximation without any encryption first. And the learning algorithm was run again on the encrypted data set. And the successful prediction rate was 76.8. After the encryption, the algorithm predicted whether people would be diagnosed with diabetes in five years, and the correct prediction rate was 76.8 percent again. This showed us that machine learning with approximated logistic regression method could be run directly on a data set encrypted using the homomorphic encryption method.

Keywords: homomorphic encryption, fully homomorphic encryption, CKKS, machine learning, logistic regression.

ÖZ

HOMOMORFİK ŞİFRELEME İLE ŞİFRELENMİŞ VERİ ÜZERİNDE MAKİNE ÖĞRENMESİ UYGULAMALARI

Kahya, Ayşegül
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Murat Cenk

Ağustos 2022, 40 sayfa

Makine öğrenmesi algoritmaları ne kadar büyük bir veri seti üzerinden öğrenme sağlarsa alınan sonuçlar o kadar gerçekçi olur. Büyük veri, büyük verinin dağıtılması ve dağıtılan verinin üzerinde öğrenme algoritmalarının çalışması günümüzün popüler araştırma konularındandır. Özellikle sağlık verileri gibi gizlilik derecesi yüksek verileri saklarken şifreleme temel bir ihtiyaçtır. Klasik şifreleme yöntemleri bu ihtiyacı giderememektedir çünkü klasik şifreleme yöntemleri ile şifrelenmiş metinler dağıtıldığında ve dağıtılan veri seti aynı anahtar kullanılarak deşifre edildiğinde anlamlı bir sonuç elde edilemez. Homomorfik şifreleme yöntemleri bu iş uygundur çünkü şifreli metin üzerinde polinomsal işlemlerin yapılmasına izin verir. Bir sonraki adımda karşılaşılan sorunsa dağıtık veriyi işleyecek olan her bir cihazın şifreleme anahtarına erişiminin olması gerekmesidir. Hem şifreleme anahtarının tüm bu cihazlara dağıtımının nasıl olacağı bir sorundur hem de bu cihazların gizlikik derecesi yüksek açık veriye erişiminin güvenilirliğinin değerlendirilmesi gerekir. Homomorfik şifreleme polinom işlemlerine izin verdiği için ve bazı makine öğrenme algoritmaları polinom tabanlı olduğu için şifreli metin üzerinde doğrudan makine öğrenme algoritmalarının çalıştırılması buna bir çözüm olabilir. Lojistik regresyon polinom tabanlı makine öğrenme algoritmalarından biridir. Bu tezde öncelikle hasta bilgileri içeren bir veri setinde lojistik regresyon algoritması kullanılarak öğrenme sağlayan yazılımın yüzde 77.2 başarı oranıyla kişilerin beş sene içerisinde diyabet tanısı alıp almayacağını doğru tahmin ettiği görüldü. Sonrasında veri seti CKKS homomorfik şifreleme yöntemiyle şifrelendi. Şifrelenmiş veri seti üzerinde lojistik regresyon yapılırken algoritma üzerinde bazı yaklaşımların kullanılması gerekmektedir. Önce herhangi bir şifreleme olmadan uygulanan yaklaşımın sonuçlarını görmek istedik. Öğrenme algoritması, şifrelenmiş veri seti üzerinde tekrar çalıştırıldı. Başarılı tahmin oranının 76.8 olduğu görüldü. Şifrelemeden sonra algoritmanın doğru tahmin oranı yine yüzde 76.8 oldu. Bu bize, yaklaşık lojistik regresyon yöntemiyle makine öğrenmesinin, homomorfik şifreleme yöntemi kullanılarak şifrelenmiş bir veri seti üzerinde doğrudan çalıştırılabileceğini gösterdi.

Anahtar Kelimeler: homomorfik şifreleme, tamamen homomorfik şifreleme, CKKS, makine öğrenmesi, lojistik regresyon.

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. Murat Cenk for his motivation and support during the development and preparation of this thesis.

I would also like to thank my examining committee members Assoc. Prof. Dr. Oğuz Yayla and Assist. Prof. Dr. Eda Tekin for their time they spared for me.

I deliver my utmost thanks to my friends H. Bartu Yünüak, Giray Efe, Esra Günsay, Olha Khomlyak and F. Betül Tan for their friendship.

I deliver my appreciation to my fiancé Fatih Sırıt, for being with me whenever I need his love and support.

I would like to express my gratitude to all my family, especially my mother Nuran Kahya, my father Cezmi Kahya and my Fındık for their encourage and unconditional love.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
LWE	Learning With Errors
DLWE	Decisional Learning With Errors
RLWE	Ring Learning With Errors
DRLWE	Decisional Ring Learning With Errors
CKKS	Cheon-Kim-Kim-Song
SEAL	Simple Encrypted Arithmetic Library
PIDD	Pima Indians Diabetes Database
BMI	Body Mass Index
TSFT	Triceps Skin Fold Thickness
PGC	Plasma Glucose Concentration
OGTT	Oral Glucose Tolerance Test
DBP	Diastolic Blood Pressure
DPF	Diabetes Pedigree Function

CHAPTER 1

INTRODUCTION

One of the main purposes of cryptography is to prevent the formation of patterns in encrypted data to avoid statistical attacks. When a single element changes in plain text or cryptographic key, a large change is expected in the ciphertext, which reduces the possibility of pattern capture in the encrypted text. It was thought that machine learning algorithms would not work in encrypted data sets because machine learning algorithms tried to find patterns and cryptographic algorithms tried to avoid creating patterns. However, studies have shown that machine learning algorithms trained in the data set, which is encrypted with homomorphic encryption, can make correct inferences for other data sets.

In order for the machine teaching algorithm to give successful results, it should work on as large data sets as possible. This very large dataset is stored in the cloud. This huge data set is divided into smaller datasets as a training set for clients to run their learning algorithms. This large data set in the cloud is stored without encryption. If they were kept encrypted, clients would have pieces of encrypted data. When a client decrypts the encrypted piece, the client can't find the piece of original data. And also, key distribution is a problem here. All clients need to access the encryption key. There are also privacy concerns when the predictive analytics service providers can access the piece of data set without any encryption. Considering that using a standard cipher to encrypt data would result be corrupted, homomorphic encryption allows us to do it without disturbing the data, so homomorphic encryption is available for privacy-preserving outsourced storage and computation. There are some future works about storing encrypted data using fully homomorphic encryption. The storage is a serious problem here because even without any encryption, the data set is extremely large.

This thesis aims to compare the success of logistic regression algorithm in data sets without any encryption and encrypted with homomorphic encryption algorithms. First of all, the linear regression algorithm is trained on a data set containing various patient information. It was seen that the algorithm predicted with a 77.2 percent success rate whether people would be diagnosed with diabetes within five years or not. Afterward, the data set was encrypted using the CKKS fully homomorphic encryption method. While working logistic regression over the encrypted dataset, it is needed to use some approximations on the algorithm. We wanted to see the results of the applied approximation without any encryption first. The learning algorithm was run again on the encrypted data set. The successful prediction rate was 76.8. After the encryption, the algorithm predicted whether people would be diagnosed with diabetes in five years, and the correct prediction rate was 76.8 percent again. This showed us that machine learning with approximated logistic regression method could be run directly on a data set encrypted using the homomorphic encryption method.

CHAPTER 2

PRELIMINARY TO THE SUBJECT

In this chapter, some of the basic primitives will be introduced to provide the reader with sufficient theoretical background to provide the basics of cryptography and some machine learning background. There are four main sections in this chapter. Firstly some basics of cryptography will be summarized. Secondly, big data protocol will be defined, and then cloud computing systems will be presented. The third section will explain the necessary machine learning background due to its role in the next chapters. In the last section, logistic regression, one of the most popular machine learning algorithms, will be explained.

2.1 Basics of Cryptography

Cryptography is a field of study that focuses on writing secret messages with the intention of keeping data secret. The word "crypto" means "secret" or "hidden", and the suffix "graphy" means “writing”. When it comes to network security, cryptography is essential.

Modern cryptography has four main concerns such as confidentiality, integrity, non-repudiation, and authentication. First of all, confidentiality means that information is just accessible to the person for whom it was intended, and no one else has access to it. Secondly, integrity ensures the information cannot be altered while being stored or being transported between the sender and the intended recipient. The creator/sender cannot then refute that he or she intended to send data because of the non-repudiation promise. The last one, authentication, ensures that both the sender and the recipient are who they claim to be.

In cryptography, The operation of a secure cipher should have the properties confusion and diffusion which are identified by Claude Shannon in his 1945 classified report A Mathematical Theory of Cryptography [15].

Confusion means that each bit of the ciphertext must rely on many sections of the key to complicate the relationships between them. The ciphertext-key connection is hidden by the attribute of confusion. This aspect makes it hard to extract the key from the ciphertext because any change to a single bit in a key will alter the calculation of most or all of the bits in the ciphertext. Diffusion states that if one plaintext element is altered, about half of the ciphertext should also change. In a similar vein, if one ciphertext element is changed, approximately half of the plaintext bits should also change. Hence a single element changes in plain text or cryptographic key, a large change is expected in the ciphertext, which reduces ability of data distribution and the possibility of pattern capture in the encrypted text. This prevents the encrypted data distribution in cloud systems and also prevents learning algorithms from using the encrypted data set as training data. There are three main categories of cryptography such as private key cryptography, public key cryptography and hash functions.

2.1.1 Private Key Cryptography

It is also known as symmetric key cryptography. It entails the use of a single secret key, as well as encryption and decryption methods, to secure the message's contents [19] .

The number of key bits determines the strength of symmetric key encryption. It is relatively faster than public key cryptography. The key distribution problem occurs because the key must be sent from the sender to the recipient over a secure channel.

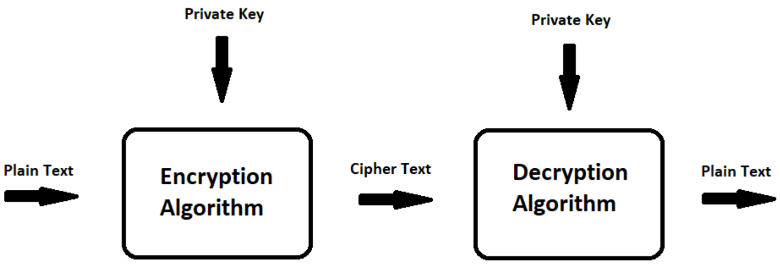

Figure 2.1: Private Key Cryptography

2.1.2 Public Key Cryptography

Another name for it is asymmetric key cryptography. In this system, encryption and decryption are performed using a pair of keys. The private key is used to decode data, whereas the public key is used to encrypt data. There is a public key along with secret key. Hence there is no the problem of key distribution here but it is very slow compared to symmetric key cryptography [17].

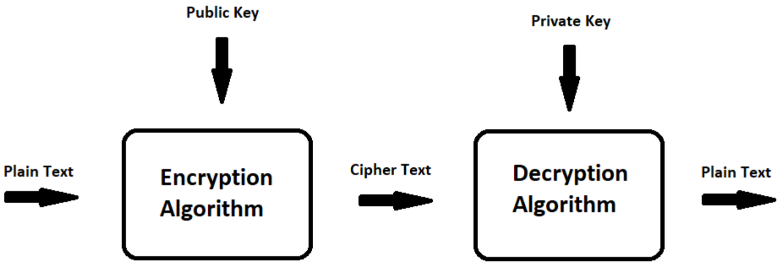

Figure 2.2: Public Key Cryptography

2.1.3 Hash Functions

Based on the plain text, a hash value with a fixed length is computed, rendering it challenging to reverse engineer the plain text's contents. Many operating systems encrypt passwords with hash algorithms [22].

2.2 Big Data Protocol and Cloud Computing Systems

2.2.1 Big Data Protocol

Big data is defined as data with increased diversity, coming in larger volumes with higher speed [24]. Basically, big data is the term used to describe larger, more intricate data sets, especially those obtained from new data sources.

As the training data set expands in machine learning applications, the result obtained from the application can be more realistic. Applications that work with larger data sets give more accurate results. With the increase in the work in the field of machine learning, more and more data has been produced and stored day by day.

Traditional data processing software is incapable of handling these massive data sets. Instead of storing and processing such extensive data by a physical device, keeping it in a cloud and distributing it to various devices when it is processed is one of the famous research topics of today.

2.2.2 Cloud Computing Systems

Cloud computing is the use of internet-based hosted services such as data storage, servers, databases, networking, and applications [23]. The data is kept on the physical servers that a cloud service provider manages. Users can store data on the cloud rather than on a storage device or hard drive. This method allows users to access files wherever they have an internet connection.

The essential reason for cloud computing's rapid growth is the numerous advantages it provides. It reduces cost. It is only paid for cloud services used instead of expensive infrastructure. Cloud systems can enable businesses to be more flexible by scaling their storage to meet their needs. This helps them to quickly grow their user base from a few hundred to thousands. Employees can work from anywhere, at any time, because cloud data can be accessed directly through the internet.

On the other hand, security concerns are the most massive concern with cloud computing. There is always a risk while storing data in the cloud without any encryption, but especially when classical encryption methods are used, it becomes almost impossible to distribute data from the cloud. For this reason, data cannot be kept encrypted form in the cloud with standard encryption methods. New encryption methods such as homomorphic encryption may be employed in cloud systems, but there are still several issues that need to be improved.

2.3 Machine Learning Background

Giving computers the ability to learn without being expressly instructed to do so is the subject of the research of machine learning. Machine learning algorithms build a model from sample data, often called "training data," to produce predictions or judgments.

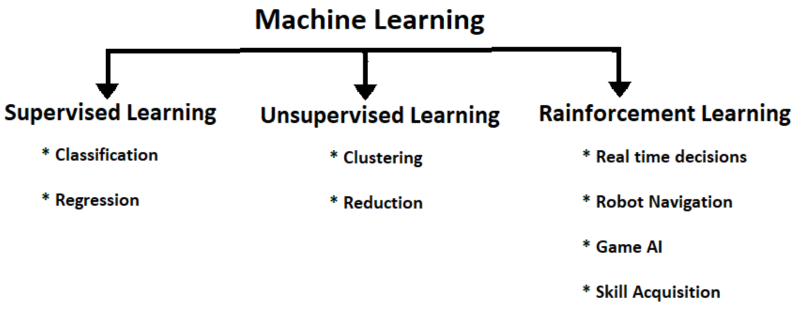

Figure 2.3: Machine Learning

Machine learning methods are basically of three types which are supervised learning, unsupervised learning and reinforcement learning [11]. In this part of the thesis, we will give short definitions and subtitles of them.

2.3.1 Supervised Learning

In this machine learning method, sample inputs and expected results are given to the algorithm. Here, learning a universal rule that connects inputs and outputs is the algorithm's aim. The model is trained repeatedly until it achieves the desired degree of accuracy in the training set of data [11]. The two main techniques of supervised learning are classification and regression.

Classification is the process of determining a function that divides a dataset into classes depending on several parameters [11].

Regression is the process of developing a model or function for converting data into continuous absolute values rather than using classes or discrete values.

2.3.2 Unsupervised Learning

The learning algorithm has no label to learn from and tries to find a structure in the input on its own. The main purpose of unsupervised learning is to discover hidden patterns in the sample data set. Clustering and dimensionality reduction are two main techniques of unsupervised learning [11].

Clustering involves dividing a population or set of data points into different groups so that the data points within a group are more similar to one another and different from the data points within other groups.

Dimensionality reduction is the technique of decreasing the quantity of random variables under consideration by generating a set of primary variables. It is basically the translation of converting data from a high-dimensional space to a low-dimensional space while preserving some of the pertinent characteristics of the original data, ideally close to its inherent dimension.

2.3.3 Reinforcement Learning

A learning strategy called reinforcement learning relies on rewarding and punishing principles. The program is trained according to the reward and punishment feedback while navigating the dynamic problem environment. The algorithm tries to find the best possible path in a particular situation to gain maximum reward.

2.4 Logistic Regression

Logistic regression is an instance of supervised learning. It estimates the probability that a binary event will happen. When a dependent variable is dichotomous, the proper regression analysis to use is logistic regression. One of the most crucial analytical techniques in the social and natural sciences is logistic regression[14]. The standard supervised machine learning approach for classification in natural language processing is logistic regression, which also closely resembles neural networks.

2.4.1 The Logistic Function

A logistic function is a function that expands exponentially at first, then plateaus and converges to its carrying capacity [14].

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}}$$

x_0 is the x value of curve's midpoint.
L is the curve's maximum value.
k is the logistic growth rate

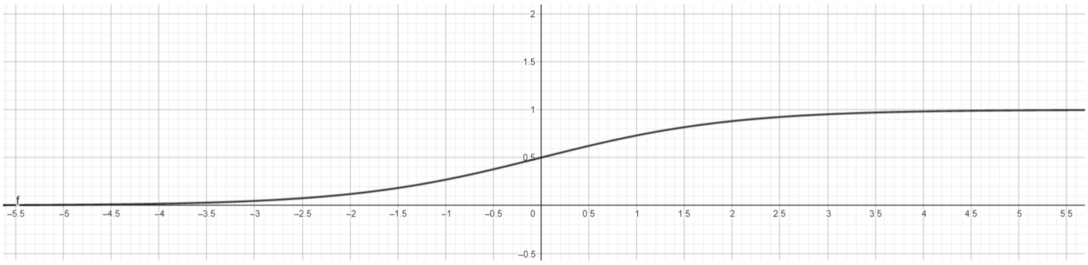

Figure 2.4: The Logistic Function, when $L = 1$ $k = 1$ $x_0 = 0$

When $L = 1, k = 1, x_0 = 0$, it is obtained the standard logistic function which is also called sigmoid or expit. The inverse of the logistic sigmoid function is the logit function.

2.4.2 Classification with Logistic Regression

Sigmoid maps real numbers into the range $[0, 1]$, and that is used for probability calculation in logistic regression. It is taken the instance x, then calculate the probability $P(y = \frac{1}{x})$ by sigmoid function.

$$decision(n) = \begin{cases} 1 & \text{if } P(y = \frac{1}{x}) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

2.4.3 The Sigmoid Function

The sigmoid function is a mathematical function to map a real number into the range [0,1]. It has a specific "S" shaped curve called "Sigmoid Curve".

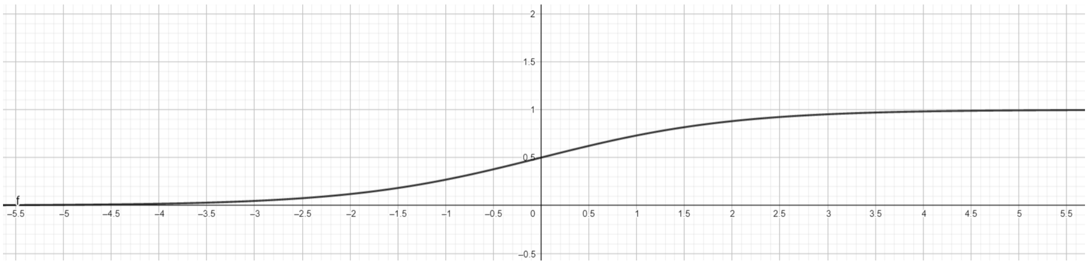

Figure 2.5: The Sigmoid Function

A wide range of sigmoid functions, such as the logistic and hyperbolic tangent functions, have been employed in a wide variety of fields. The most known sigmoid function type is the logistic function [21].

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} = 1 - \sigma(-x)$$

2.4.4 Learning in Logistic Regression

Logistic Model

A statistical model called the logistic model uses a logarithm of the event's odds that is a linear combination of one or more independent variables to represent the likelihood that one event will occur [14].

The logistic function $P(x)$:

$$P(x) = \frac{1}{1+e^{-(x-\mu)/s}}$$

μ is a location parameter, where$P(\mu) = 1/2$

s is a scale parameter

$$y = \beta_0 + \beta_1 x$$

$$P(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)/s}}$$

$\beta_0 = -\mu/s$ which is the vertical intercept

$\beta_1 = 1/s$ which is the inverse scale parameter

Hence, $\mu = -\beta_0/\beta_1$

$$s = 1/\beta_1$$

Model Fitting

When x_k and y_k are given, P_k is the possibility that the associated y will be 1 and $1 - P_k$ is the possibility that it will be 0 by Bernoulli distribution [8].

$$p_k = P(x_k)$$

It is aimed to find β_0 and β_1 for best fitting for the data set.

$$\text{The log loss for the } k\text{'th point} = \begin{cases} -\ln p_k & \text{if } y_k = 1 \\ -\ln(1-p_k) & y_k = 0 \end{cases}$$

The goal is minimizing the negative log-likelihood $-\ell$ to find the best candidate of β_0 and β_1.

$$\ell = \sum_{k:y_k=1} \ln(p_k) + \sum_{k:y_k=0} \ln(1-p_k)$$

$$\ell = \sum_{k=1}^{K} (y_k \ln(p_k) + (1-y_k)\ln(1-p_k))$$

So, the maximum likelihood estimation is:

$$L = \prod_{k:y_k=1} p_k + \prod_{k:y_k=0} (1-p_k)$$

Parameter Estimation

While ℓ is nonlinear in β_0 and β_1, it is needed to find numerical methods to determine best values for β_0 and β_1 for best fitting for the data set.

In order to maximize ℓ, the derivatives of ℓ with respect to β_0 and β_1 should be zero. P_k is the possibility that the associated y will be 1 and $1 - P_k$ is the possibility that it will be 0 by Bernoulli distribution [8].

$$0 = \frac{\partial \ell}{\partial_{\beta_0}} = \sum_{k=1}^{K} (y_k - p_k)$$

$$0 = \frac{\partial \ell}{\partial_{\beta_1}} = \sum_{k=1}^{K} (y_k - p_k) x_k$$

Making Predictions

β_0 and β_1 coefficients are found at the parameter estimation section.

p is the probability for the value x.

$$t = \beta_0 + \beta_1 x$$

$$p = \frac{1}{1 + e^{-t}}$$

$$decision(n) = \begin{cases} 1 & \text{if } p > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

CHAPTER 3

HOMOMORPHIC ENCRYPTION SCHEME

Definition 3.0.1. A map that respects the structure between two algebraic structures of the same kind, such as two groups, two rings, or two vector spaces, is called a homomorphism.

This means a map $f : A \to B,\ f : A \to B$ between two sets A, B equipped with the same structure such that

$$F(a.b) = F(a).F(b)$$

for every pair a, b of elements of A.

In homomorphic encryption, it is possible to think of the encryption and decryption functions as homomorphisms between the plain text and cipher text spaces. The term "homomorphic" alludes to homomorphism in algebra.

Homomorphic encryption is a type of encryption that allows users to work on encrypted data without having to decrypt it first [20].

$$Enc(a + b) = Enc(a) + Enc(b)$$

$$Enc(a * b) = Enc(a) * Enc(b)$$

Homomorphic encryption can be utilized to protect privacy in outsourced storage and computing. This makes it possible to encrypt data and send it to commercial cloud computing environments for computation. Homomorphic encryption would be applied to enable innovative services in highly regulated areas like health care by reducing privacy obstacles that prevent data sharing. Because of worries about the privacy of medical data, using predictive analytics in the healthcare industry, for example, can be challenging. However, if the predictive analytics service provider can operate with encrypted data instead, these risks are reduced.

Homomorphic encryption schemes has both symmetric-key and public-key encryption algorithms. There are three main types of homomorphic encryption such as, partially homomorphic, somewhat homomorphic and fully homomorphic encryption.

- Partially Homomorphic Encryption: On the encrypted message, only one type of mathematical operation is permitted such as addition or multiplication, with an arbitrary number of repetitions.
- Somewhat Homomorphic Encryption: On the encrypted message, both addition and multiplication operations are permitted, but there is limit to use.
- Fully Homomorphic Encryption: On the encrypted message, it is possible to use various types of evaluation operations on the encrypted message an arbitrary number of times.

There are four sections in this chapter. The preliminaries of homomorphic encryption will be briefly summarized. Then a fully homomorphic encryption scheme Cheon-Kim-Kim-Song (CKKS) will be explained.

3.1 The Preliminaries for Fully Homomorphic Encryption

Many partially homomorphic schemes could be created using LWE and its Ring variation. Therefore, unlimited addition and multiplication operations on encrypted data are not permitted by such systems. The encrypted ciphertext contains "error" as a result of each operation. After sufficient encrypted operations, especially multiplication, this mistake ultimately becomes too enormous to decrypt correctly [10]. An approach called bootstrapping could be used to fix this. Bootstrapping, on the other hand, is highly costly and slow [5].

3.1.1 Lattices

Definition 3.1.1. Let $v_1, ..., v_n \in \mathbb{R}^m$ be linearly independent set of vectors ($m \geq n$). The set of linear combinations of v_i is called L generated by $v_1, ..., v_n$.

$$L = \left\{ \sum_{i=1}^{n} a_i v_i : a_i \in \mathbb{Z} \right\}$$

The vectors $v_1, ..., v_n$ are the basis of the lattice. The lattice rank is n and the lattice dimension is m. If $n = m$ then L called full rank lattice.

3.1.2 Learning With Errors Problem

The Learning with Errors (LWE) problem provides an incredibly flexible foundation for cryptographic constructs [20] [16]. Since all cryptographic constructs relying on it are secure under the supposition that worst-case lattice problems are hard, it is as hard as worst-case lattice problems.

The LWE problem tries to obtain the secret $s \in {\mathbb{Z}_q}^n$ from a sequence of random linear equations on s. Finding s would be relatively easy if the error didn't exist. Gaussian elimination allows us to retrieve s in polynomial time after about n equations. There is a noticeable increase in difficulty of the problem after the error is introduced. The Gaussian elimination algorithm uses linear combinations of n equations, amplifying the error to uncontrollable levels and effectively removing any information from the equations that result [12].

Definition 3.1.2. Let q be a prime number, χ is a probability distribution which outputs. It is said that $LWE(q, \chi, n)$ is a set of pairs of the form

$$\{(a_i, < a_i, s > +e_i) \, | s \leftarrow {\mathbb{Z}_q}^n, e_i \leftarrow \chi, a_i \leftarrow {\mathbb{Z}_q}^n\}$$

3.1.3 Ring Learning With Errors Problem

Typically, key sizes for cryptographic techniques based on the LWE concerns must be on the order of n^2 [20, 16, 18]. This is due to the fact that key sizes of order $n2$ are often required for cryptographic applications, which normally require at least n vectors $a_1, ..., a_n \in \mathbb{Z}_q{}^n$. This is due to the requirement that at least n vectors leading to keys of order n^2 be provided for cryptographic applications. In addition to being highly desirable from a practical standpoint, as we shall see below, reducing this to almost linear dimensions may also result in intriguing theoretical advancements.

3.1.4 Bootstrapping

It is possible to reduce the error. Bootstrapping is a method that uses the encrypted secret key and homomorphically executes the decryption process without disclosing the message [5]. Decryption is achieved using a sequence of XOR and AND gates or additions and multiplications. Binary messages serve as the homomorphic encryption scheme's input. It is necessary to have an ordered collection of encryptions of its bits in order to obtain the encryption of the secret key. With these two components, homomorphic decryption can be completed while minimizing the noise left over from earlier processes.

3.2 Public Key Encryption Scheme Based on LWE

The message space is $\{0, 1\}$. Let q be a prime, $n, m \in \mathbb{Z}$, and χ be a noise distribution where for any $e \leftarrow \chi$. It is highly possible that $||e|| \leq q/4m$.

Key Generation:

The security parameter is n and $A \leftarrow \mathbb{Z}_q{}^{nm}$, $e \leftarrow \chi^m$.

$$\text{The secret key } sk = s \leftarrow \mathbb{Z}_{||}{}^n.$$

$$\text{The public key } pk = \left(A, s^T A + e^T\right).$$

Encryption:

The public key pk is $\big(A, b^T\big)$.

The message $x \in \{0, 1\}$.

r is random bit string $r \leftarrow \{0, 1\}^m$.

$$Enc(pk, x) = (Ar, b^T + x.\lceil q/2 \rceil)$$

Decryption:

The secret key is s and the cipher text is (u, v).

$$Dec(sk, (u, v)) = \begin{cases} 0 & \text{if } ||v - s^T u|| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness:

The cipher text is (u, v).

$$v - s^T u = b^T r + x.\lceil q/2 \rceil - s^T Ar$$

$$= e^T r + x.\lceil q/2 \rceil$$

If x is 1 the output is 1 obviously, else if x is 0 then by the requirement for χ,

$$||e^T r|| \leq m.\frac{q}{4m} = \frac{q}{4} \text{ with high probability.}$$

Hence the output will be 0 with a high probability [7].

CHAPTER 4

CKKS SCHEME

4.1 Cheon-Kim-Kim-Song (CKKS) Scheme

The homomorphic encryption scheme known as CKKS was designed by Cheon, Kim, Kim, and Song. CKKS is a RLWE based fully homomorphic public key cryptosystem.

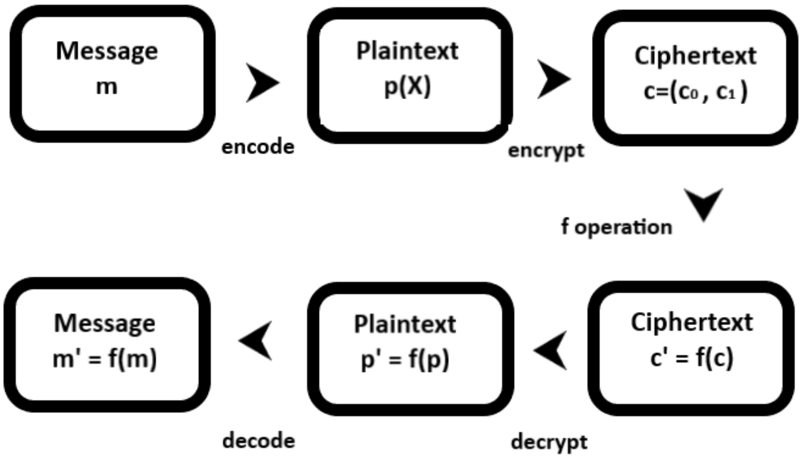

Figure 4.1: CKKS Fully Homomorphic Public Key Cryptosystem

Up until CKKS, homomorphic calculations could only be performed on integers. Now it is possible to use homomorphic computations on real/complex numbers. With the previous techniques, homomorphic calculations perform only integer-based encrypted inputs. Using real numbers for approximate arithmetic creation was not possible. Every existing scheme shares the characteristic that while encrypting data, a small amount of "noise" is added. These noises will increase when ciphertexts are computed homomorphically, eventually growing so huge that decryption is impossible. The noise in a ciphertext can then be reduced using Gentry's bootstrapping method to a set level that depends on the complexity of the decryption circuit. One of the biggest challenges in creating effective schemes has been the noise growth brought on by homomorphic multiplication. In the first generation of schemes, the sounds themselves multiplied with each homomorphic multiplication, which caused the depth of the circuit to increase at a doubly exponential rate. A naive method is scaling using a huge integer to multiply the input and produce a real number, doing homomorphic calculations with the scaled number's floor, and then dividing the result by the large integer. The issue, though, is that after encrypted multiplications, the scaling factor itself grows exponentially. CKKS provides a solution and gives a method to achieve homomorphically rescale.

4.1.1 CKKS Encoding and Decoding Procedure

The fact that the encryption, decryption, and other methods rely on polynomial rings requires the encoder-decoder step. Therefore there is a requirement for the method to convert our vectors of real values into polynomials. First of all, it is required to develop an encoder and a decoder to transform complex vectors into polynomials. The encoder takes the input $z \in \mathcal{C}^N$ and transforms it into the polynomial $m(X)$.

$$m(X) \in \mathbb{Z}[X]/(X^N + 1)$$

The polynomial degree modulus is denoted by N, while N is power of 2.

$$\text{let } M = 2N$$

$$\Phi_M(X) = X^N + 1$$

The plain text space is the polynomial ring $\mathfrak{R} = \mathbb{Z}[X]/(X^N+1)$, and ξ_M is M'th root of unity.

$$\xi_M = e^{2i\pi/M}$$

$z \in \mathcal{C}^N$ will be transformed into $m(X) \in \mathbb{Z}[X]/(X^N + 1)$ To encode and decode the vectors the canonical embedding will be used.

$$\sigma(R) \subseteq H = z \in C^N$$

$$\sigma : R = Z[x]/[X^N + 1] \rightarrow \sigma(R) \subseteq H$$

$$\sigma : \mathcal{C}[X]/(X^N + 1) \rightarrow \mathcal{C}^N$$

The N roots of cyclotomic polynomial $\Phi_M(X) = X^N + 1$ are $\xi_1, \xi_3, ..., \xi_{2N-1}$

$$\sigma(m) = (m(\xi), m(\xi^3), ..., m(\xi^{2N-1})) \in \mathcal{C}^N$$

σ describe an isomorphism, in other words it is a bijective homomorphism. Hence, every vector will have a unique encoded representation in the related polynomial, and vice versa.

Begin with expand $z \in \mathcal{C}^N$ by π^{-1}, After that it is obtained $\pi^{-1}(z) \in H$ It is needed to compute the inverse σ^{-1}. The challenging part is to find a polynomail $m(X)$ as

$$m(X) = \sum_{i=0}^{N-1} \sigma_i X^i \in \mathcal{C}[x]/S(X^N + 1)$$

given a vector $z \in \mathcal{C}^N$, such that:

$$\sigma(m) = (m(\xi), m(\xi^3), ..., m(\xi^{2N-1})) = (Z_1, ..., Z_N)$$

Thus, It is obtained the following equation:

$$\sum_{j=0}^{N-1} \sigma_j(\xi^{2i-1}) = z_i, i = 1, 2, ..., N$$

This could be thought of as a linear equation $A\sigma = z$, with A the Vandermonde matrix of the $(\xi^{2i-1})_{i=1,2,...,N}$ where z is the encoded vector and σ is the polynomial coefficients vector.

Therefore we get

$$\sigma = A^{-1}z$$

$$\sigma^{-1}(z) = \sum_{i=0}^{N-1} \sigma_i X^i \in \mathcal{C}[x]/(X^N + 1).$$

When the result is rounded, it could make changes on some significant numbers. Because of that it is necessary to multiply by $\triangle$ while $\triangle > 0$ in the course of encoding procedure and dividing by during decoding part. So, the order of encoding operations is as follows:

1. Expand the element $z \in \mathcal{C}^N$ by π^{-1}, while $\pi^{-1}(z) \in H$.

2. Multiply by $\triangle$.

3. $\sigma(R) : \lfloor \triangle.\pi^{-1}(z) \rceil_{\sigma(R)} \in \sigma(R)$.

4. Encode $\sigma : m(X) = \sigma^{-1}(\lfloor \triangle.\pi^{-1}(z) \rceil) \in R$.

The decoding procedure is much more easier, it will be obtained z from the polynomial $m(X)$.

$$z = \pi.\sigma(\triangle^{-1}.m)$$

4.1.2 CKKS Encryption and Decryption Procedure

A secret key and a public key are generated in the CKKS public key encryption system. The private key is required for decryption and must be kept secret, whereas the public key is needed for encryption and can be shared. The security of CKKS and many other homomorphic encryption schemes, is based on the Learning With Error (LWE) problem. The form of LWE is $(A, A.s + e)$ when s is the secret key $s \in {Z_q}^n$ and $e \in {Z_q}^n$ are small noises. In the absence of the e, it could be solved the linear system using Gaussian elimination, making the problem much simpler to solve.

$$\text{The secret key } s \in {Z_q}^n.$$

$$\text{The public key } p = (A.s + e, A).$$

When $A \in {Z_q}^{nxn}$ and $e \in {Z_q}^n$ by the RLWE problem it is hard to get secret s from p.

Encryption

The message is $\mu \in {Z_q}^n$.

$$Enc(\mu, p) = c = (\mu, 0) + p$$

$$c = (\mu, 0) + p = (\mu - A.s + e, A)$$

$$c = (c_0, c_1)$$

Decryption

$$Dec(c, s) = \tilde{\mu}$$

$$c_0 + c_1.S = \mu - A.s + e + A.s = \mu + e$$

$$\mu + e \approx \mu$$

because e is negligible.

4.1.3 Addition and Ciphertext-Plaintext Multiplication Procedure

With the use of addition and multiplication implementations, it is demonstrated that it is possible to encrypt certain data and execute operations on it so that the outcome is the same after decryption as it would have been had the actions taken with regard to unencrypted data [7].

Addition Procedure

There are two messages μ and $\mu^{'}$. After encryption

$$Enc(\mu) = c = (c_0, c_1)$$

$$Enc(\mu^{'}) = c^{'} = (c_0^{'}, c_1^{'})$$

Addition:

$$c + c^{'} = c_{add} = (c_0^{'} + c_0, c_1^{'} + c_1) = (c_{add_0}, c_{add_1})$$

When $c_{add} = (c_{add_0}, c_{add_1})$ is decrypted, the result should be $\mu + \mu^{'}$ to cover homomorphic encryption properties.

$$Dec(c_{add}) = c_{add_0} + c_{add_1}.s = c_0 + c_0^{'} + (c_1 + c_2^{'}).s$$

$$= \mu + \mu^{'} + 2e \approx \mu + \mu^{'}$$

because e is negligible.

Ciphertext-Plaintext Multiplication Procedure

There is a message μ and a plaintext k.

$$Enc(\mu) = c = (c_0, c_1)$$

Multiplication:

$$k.c = c_{mult} = (kc_0, kc_1))$$

When $c_{mult} = (c_{mult_0}, c_{mult_1})$ is decrypted, the result should be $k\mu$ to cover homomorphic encryption properties.

$$k.c_0 + k.c_1.s = k(c_0 + c_1.s)$$

$$= k(\mu + e)$$

$$= k.\mu + k.e$$

$$\approx \mu.k$$

4.1.4 Ciphertext-Ciphertext Multiplication and Relinearization Procedure

Ciphertext-Ciphertext Multiplication Procedure:

There are two messages μ and $\mu^{'}$.

$$Enc(\mu) = c = (c_0, c_1)$$

$$Enc(\mu^{'}) = c^{'} = (c_0^{'}, c_1^{'})$$

The multiplication of c_0 and $c^{'}$ is $c_{mult}(c, c^{'})$, when $c_{mult} = (c_{mult_0}, c_{mult_1})$ is decrypted, the result should be $\mu.\mu^{'}$ to cover homomorphic encryption properties.

$$Dec(c_{mult}, s) = Dec(c, s).Dec(c^{'}, s)$$

$$Dec(c, s).Dec(c^{'}, s) = (c_0 + c_1.s)(c_0^{'} + c_1^{'}.s)$$

$$Dec(c, s).Dec(c^{'}, s) = (c_0.c_0^{'} + (c_0.c_1^{'} + c_1.c_0^{'}).s + (c_1.c_1^{'}).s^2)$$

$$\text{Let, } d_0 = c_0.c_0^{'}$$

$$d_1 = c_0.c_1^{'} + c_1.c_0^{'}$$

$$d_0 = c_1.c_1^{'}$$

$$Dec(c_{mult}, s) = d_0 + d_1.s + d_2.s^2$$

The problem here is that the size of the ciphertext increases exponentially after each multiplication operation [7]. Relinearization is used for doing multiplication operation without increasing the ciphertext size at each step.

Relinearization Procedure

$$Dec(c_{mult}, s) = d_0 + d_1.s + d_2.s^2$$

$$= Dec(c, s).Dec(c^{'}, s)$$

If it is found the couple of polynomials $(d_0^{'}, d_1^{'})$ such that:

$$Dec((d_0^{'}, d_1^{'}), s) = Dec(c, s).Dec(c^{'}, s)$$

$$Dec((d_0^{'}, d_1^{'}), s) = d_0^{'} + d_1^{'}.s$$

$$= d_0 + d_1.s + d_2.s^2$$

So relinearization function is used for finding the polynomials $(d_0^{'}, d_1^{'})$ such that:

$$(d_0^{'}, d_1^{'}) = (d_0, d_1 + P) \text{ while } Dec(P, s) = d_2.s^2$$

To compute P it is needed a evaluation key:

$$evk = (-a_0.s + e_0 + s^2, a_0)$$

Let e_0 is a very small randomly chosen polynomial and a_0 is an uniformly sampled polynomial on R_q .

$$Dec(evk, s) = e_0 + s^2 \approx s^2$$

The evaluation key evk can be publicly exposed, by the RLWE problem it is very hard to find the secret from it.

$$P = d_2.evk = (d_2.(-a_0 + e_0 + s^2), d_2, a_0)$$

$$Dec(P, s) = d_2.s^2 + d_2.e_0$$

$$d_2.s^2 + d_2.e_0 \approx d_2.s^2$$

It is known that e_0 is very small, but it is not sure d_2 is small or not. The problem here can be solved by changing evaluation key a little bit.

While p is big integer and a_0 is an uniformly sampled polynomial on $R_{p.q}$, let the evaluation key:

$$evk = (-a_0.s + e_0 + p.s^2, a_0)(\mod p.q)$$

$$P = \lfloor p^{-1}.d_2.evk \rceil (\mod q)$$

Hence, relinearize function is

$$Rel((d_0, d_1, d_2), evk) = (d_0, d_1) + \lfloor p^{-1}.d_2.evk \rceil.$$

While the messages are μ and $\mu^{'}$, the multiplication of them is μ_{mult} and the corresponding cipher texts are $(c, c^{'})$,the multiplication of them is c_{mult} [7]. If one or more multiplication operations are to be performed on cipher text, the order of operations is as follows:

1. Multiplication $c_{mult}(c, c^{'}) = (d_0, d_1, d_2)$
2. Relinearize $Rel((d_0, d_1, d_2), evl = c_{rel})$
3. Decryption $Dec(c_{rel}, s) \approx \mu_{mult}$

As a result, after multiplying two ciphertexts, their size is kept constant by relinearization procedure.

4.1.5 Rescaling Procedure

Only integer-based encrypted inputs can be used for homomorphic calculations using the prior methods. It was impossible to perform approximation arithmetic using real numbers. A simple technique for scaling a real number involves multiplying the input by a big integer, using the floor of the resulting scaled number to perform homomorphic operations, and then dividing the outcome by the large integer. Unfortunately, after encrypted multiplications, the scaling factor itself expands exponentially [7, 20]. A solution and a mechanism to achieve homomorphically rescaling are provided by CKKS. In the encoding part there is a multiplication by $\triangle$ to keep some level of precision. When c and $c^{'}$ are multiplied, the result covers $z.z^{'}.\triangle^2$ The square of the scale includes at each multiplication, because the scale could increase exponentially, which could cause overflow after a few multiplications. As a result, keeping the scale constant is the aim of the rescaling operation. First of all it is necessary to define the parameter q while the polynomial ring

$$R_q = \mathbb{Z}_q[X]/X^N + 1.$$

If there are L multiplications, with a scale $\triangle$,while $q_0 \geq \triangle$

$$q = \triangle^L.q_0$$

Let q_l is the modulo for given l to do L multiplication operation. The rescaling operation from level l to $(l - 1)$ as:

$$Res_{l \to l-1}(c) = \lfloor \frac{q_l - 1}{q_l}.c \rceil \mod (q_l - 1)$$

$$\lfloor \triangle^{-1}.c \rceil \mod (q_l - 1)$$

because of $q_l = \triangle^l.q_0$.

CHAPTER 5

APPLICATIONS AND TOOLS

As mentioned in Chapter 3, homomorphic encryption allows the addition and multiplication of the ciphertext. This means that polynomial operations can be performed on ciphertexts with homomorphic encryption applied. In this thesis, it was aimed to run a logistic regression algorithm on the ciphertext. Since homomorphic cryptography only allows polynomial operations, approximate polynomial functions had to be used instead of logistic regression operations. In the first part, it will explain the approximation method we used and the approximating sigmoid function. The second part will introduce the data model we used. Microsoft SEAL Homomorphic Encryption Library [13] is used to encrypt the data set. It is given brief information about Microsoft SEAL in the third part. And lastly, it will be summarized the results obtained.

5.1 Approximating The Sigmoid Function

Multiple techniques can be used for obtaining an approximate polynomial for a given function [6, 9]. We used the same approach as [4], called minimax approximation.

For the interval [5, 5] and degrees 1 and 3, the algorithm gives following results:

$$\sigma_1(x) = 0.5 + 0.125x$$

$$\sigma_3(x) = 0.5 + 0.197x - 0.004x^3$$

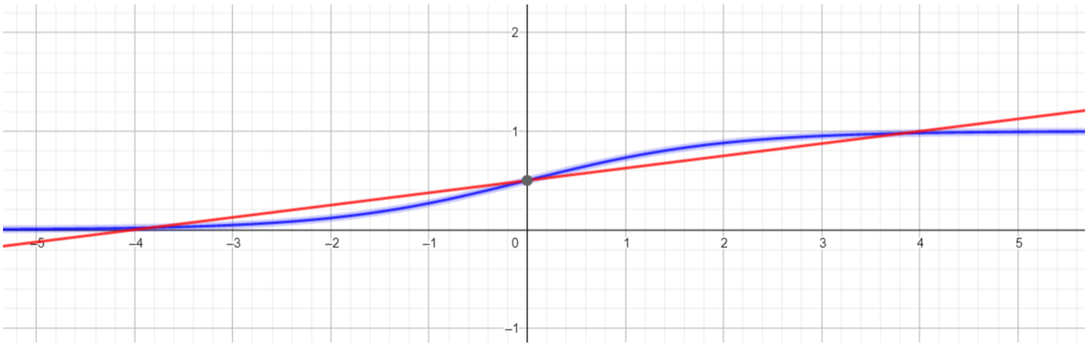

Figure 5.1: $\sigma_1(x) = 0.5 + 0.125x$

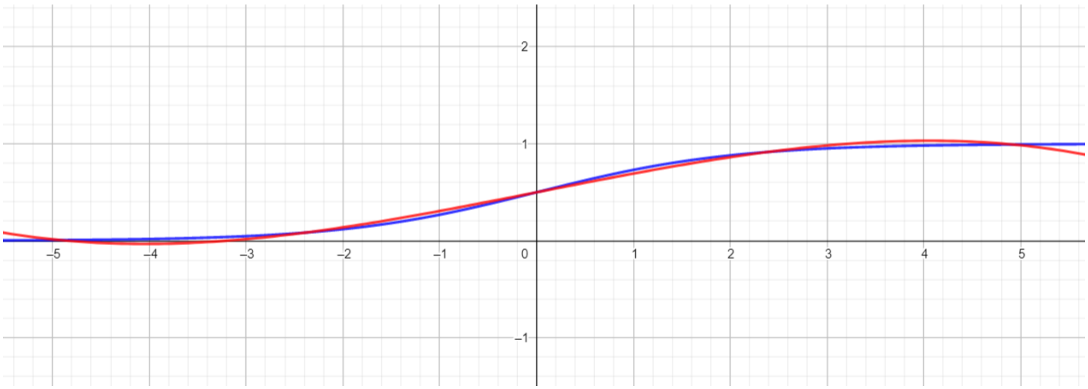

Figure 5.2: $\sigma_3(x) = 0.5 + 0.197x - 0.004x^3$

5.2 The Data Model

Pima Indians Diabetes Database The National Institute of Diabetes and Digestive and Kidney Diseases is the original source of this dataset [3, 2]. The condition known as diabetes mellitus occurs when the body is unable to create, utilize, or store glucose. Blood glucose, sometimes known as "sugar," rises too high as glucose builds up in the bloodstream. A variety of long-term consequences, including heart attacks, strokes, blindness, kidney failure, and blood vessel damage, can result from poorly controlled diabetes. Based on specific diagnostic metrics present in the dataset, the dataset's goal is to diagnostically forecast whether a patient has diabetes or not in 5 years. Remarkably, all patients at this facility are Pima Indian women at least 21 years old. There are 768 cases present in PIDD with nine attributes such as DPF, the number of times pregnancy, the DBP (mm Hg), PGC of two hours in an OGTT, TSFT (mm), two hours of serum insulin (mu U/ml), age (years), BMI, and the class variable (0 or 1).

5.3 Simple Encrypted Arithmetic Library

The Microsoft Cryptography and Privacy Research Group designed Microsoft SEAL, an intuitive open-source homomorphic encryption library with an MIT license [13, 1]. Microsoft SEAL is written in contemporary standard C++, making it simple to compile and execute in a variety of environments. In order to enable computations to be made directly on encrypted data, Microsoft SEAL provides a collection of encryption libraries based on open-source homomorphic encryption technology. With the advent of end-to-end encryption, software developers may now offer compute and data storage services without ever needing the customer to divulge their key.

Although homomorphic encryption has a strong theoretical foundation, its application has long lagged behind. The situation has recently begun to improve thanks to new implementations, data encoding methods, and applications, but much work still needs to be done. The Simple Encrypted Arithmetic Library (SEAL) was developed specifically to meet the need for a homomorphic encryption library that was well-engineered, well-documented, free of external dependencies, and simple to use for both experts and non-experts with little to no prior knowledge of cryptography. The Simple Encrypted Arithmetic Library (SEAL) first version was released in 2015 [13]. We used Microsoft SEAL version 4.0 to encrypt our data set.

5.4 Results

In the data set we used, the total number of instances is 768. It aims to compare the successful prediction rate of the logistic regression algorithm over the data set without encryption, approximated logistic regression algorithm over data sets without encryption, and encrypted with CKKS Scheme.

For the first case, without any encryption and approximation:

1. Correctly classified instances is 593.

2. Incorrectly classified instances is 175.

So, without any encryption and approximation, the algorithm predicted with a 77.2 percent success rate whether people would be diagnosed with diabetes within five years or not. Without any encryption, the size of the data set is 24 KB. It took 0.16 seconds to process the data with logistic regression.

For the second case, without any encryption, the results of degree-3 minimax approximation over logistic regression are seen on the Pima Indians Diabetes Database.

1. Correctly classified instances is 590.

2. Incorrectly classified instances is 178.

So, without any encryption, while using degree-3 minimax approximation over logistic regression, the algorithm predicted with a 76.8 percent success rate whether people would be diagnosed with diabetes within five years or not. Without any encryption, the size of the data set is 24 KB. It took 0.14 seconds to process the data with approximated logistic regression.

For the third case, After the CKKS encryption scheme is applied to the database, the results of degree-3 minimax approximation over logistic regression are seen on the Pima Indians Diabetes Database.

1. Correctly classified instances is 590.

2. Incorrectly classified instances is 178.

So, with fully homomorphic encryption, any encryption, and approximation, the algorithm predicted with a 76.8 percent success again rate whether people would be diagnosed with diabetes within five years or not. The size of the encrypted data set is 6540 KB. It took 681,06 seconds to process the encrypted data with approximated logistic regression.

Table 5.1: Table of comparison for each step

Results			
	Without Approximation Without Encryption	Approximation	Encryption Approximation
Total instance	768	768	768
Correctly classified	593	590	178
Incorrectly classified	175	590	178
Success rate	77.2 %	76.8 %	76.8 %
Data size	24 KB	24 KB	6540 KB
Process time	0.16 s	0.14 s	681.06 s

This demonstrated that a data set that has been encrypted using the fully homomorphic encryption approach might be directly used for machine learning utilizing the approximation over logistic regression method. However, after encryption, the size of the input data set has increased to 272.5 times, and the processing of the data has increased to approximately 4865 times. Hence, working on encrypted data sets severely reduces machine algorithms' efficiency in terms of both memory usage and time consumption.

CHAPTER 6

CONCLUSION

Logistic regression is one of the most popular machine learning algorithms. While working logistic regression over the encrypted dataset, it is needed to use some approximations on the algorithm. For this thesis, it is implemented logistic regression and approximated logistic regression algorithms. Then we encrypted a data set with a fully homomorphic encryption scheme CKKS by using Microsoft SEAL Homomorphic Encryption Library. As a result, it is compared the successful prediction rate of the logistic regression algorithm over the data set without any encryption, approximated logistic regression algorithm over the data sets without any encryption, and encrypted with CKKS Scheme. Homomorphic encryption is available for privacy-preserving outsourced storage and computation because it enables us to perform any operations on data encrypted using a traditional cipher without causing the data to become corrupted. Future research will focus on employing completely homomorphic encryption to store encrypted data. Even without any encryption, the data collection size presents a severe storage issue.

We used Pima Indians Diabetes Database to run the machine learning algorithms and Microsoft Simple Encrypted Arithmetic Library to encrypt the data set. First, a data set containing diverse patient data is used to train the linear regression method. It was discovered that the algorithm correctly predicted whether or not participants would receive a diabetes diagnosis within five years, with a success rate of 77.2. The CKKS fully homomorphic encryption technique was used to encrypt the data set. There are various estimates on the algorithm that must be used while running logistic regression over the encrypted dataset. We wanted to see how the applied approximation performed without any encryption. The percentage of correct predictions was 76.8. Following encryption, the system once again had a 76.8 accuracy rate when predicting whether or not people would receive a diabetes diagnosis in the next five years. However, after encryption, the size of the input data set has increased to 272.5 times, and the processing of the data has increased to approximately 4865 times. This demonstrated to us that a data set encrypted using the homomorphic encryption approach might be directly used for machine learning with the approximated logistic regression method. However, custom works are needed for machine learning models to be adapted to the FHE environment. In addition, working on encrypted data sets severely reduces machine algorithms' efficiency in terms of both memory usage and time consumption.

REFERENCES

[1] Microsoft seal, https://www.microsoft.com/en-us/research/project/microsoft-seal, accessed on 24 August 2022.

[2] Pidd, http://www.ics.uci.edu/ mlearn/mlsummary.html, accessed on 24 August 2022.

[3] J. L. Breault, Data mining diabetic databases: are rough sets a useful addition, Computing science and statistics, 34, p. 404, 2001.

[4] H. Chen, R. Gilad-Bachrach, K. Han, Z. Huang, A. Jalali, K. Laine, and K. Lauter, Logistic regression over encrypted data from fully homomorphic encryption, BMC medical genomics, 11(4), pp. 3–12, 2018.

[5] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, Bootstrapping for approximate homomorphic encryption, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 360–384, Springer, 2018.

[6] J. H. Cheon, J. Jeong, J. Lee, and K. Lee, Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form, in *International Conference on Financial Cryptography and Data Security*, pp. 53–74, Springer, 2017.

[7] J. Fan and F. Vercauteren, Somewhat practical fully homomorphic encryption, Cryptology ePrint Archive, 2012.

[8] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*, volume 4, Wiley New York, 2011.

[9] W. Fraser, A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable, Journal of the ACM (JACM), 12(3), pp. 295–314, 1965.

[10] C. Gentry, *A fully homomorphic encryption scheme*, Stanford university, 2009.

[11] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, " O'Reilly Media, Inc.", 2019.

[12] S. Gharib, S. R. Ali, R. Khan, N. Munir, et al., System of linear equations, guassian elimination, Global Journal of Computer Science and Technology, 2015.

[13] K. Laine, Simple encrypted arithmetic library 2.3. 1, Microsoft Research https://www. microsoft. com/en-us/research/uploads/prod/2017/11/sealmanual-2-3-1. pdf, 2017.

[14] D. J. . J. H. Martin, Speech and language processing, December 2021.

[15] D. F. Masood, Network security, November 2019.

[16] J. McIvor, Ring theory (math 113), summer 2014, University of California, Berkeley, 2014.

[17] R. A. Mollin, *RSA and public-key cryptography*, Chapman and Hall/CRC, 2002.

[18] S. Noether, A. Mackenzie, et al., Ring confidential transactions, Ledger, 1, pp. 1–18, 2016.

[19] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*, Springer Science & Business Media, 2009.

[20] V. Pathak, Lattices, homomorphic encryption, and ckks, arXiv preprint arXiv:2205.03511, 2022.

[21] H. Pratiwi, A. P. Windarto, S. Susliansyah, R. R. Aria, S. Susilowati, L. K. Rahayu, Y. Fitriani, A. Merdekawati, and I. R. Rahadjeng, Sigmoid activation function in selecting the best model of artificial neural networks, in *Journal of Physics: Conference Series*, volume 1471, p. 012010, IOP Publishing, 2020.

[22] B. Preneel, Cryptographic hash functions, European Transactions on Telecommunications, 5(4), pp. 431–448, 1994.

[23] R. J. Rafaels, Cloud computing: From beginning, 04 2015, ISBN 1511404582.

[24] E. P. Reznor, Big data: A beginner's guide to using data science for business, 08 2017, ISBN 1975699734.