ANALYSIS OF ENCRYPTED MACHINE LEARNING MODEL USING FULLY HOMOMORPHIC ENCRYPTION AND CKKS SCHEME

BY

MD HABIBUR RAHMAN
ID: 191-15-12391

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By
Md Ismail Jabiullah
Professor
Department of CSE
Daffodil International University
Co-Supervised By
Sheak Rashed Haider Noori
Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
JANUARY 2023

APPROVAL

This project titled **"Analysis of Encrypted Machine Learning Model Using Fully Homomorphic Encryption and CKKS scheme"**, submitted by Md Habibur Rahman, ID No: 191-15-12391 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on *26th January 2023*.

BOARD OF EXAMINERS

Chairman

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Subhenur Latif
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Mohammad Monirul Islam
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

26-01-23

Dr. Dewan Md Farid
Professor
Department of Computer Science and Engineering
United International University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Md Ismail Jabiullah, Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Md Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Sheak Rashed Haider Noori
Professor
Department of CSE
Daffodil International University

Submitted by:

Habibur

Md Habibur Rahman
ID: 191-15-12391
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to Dr. **Md Ismail Jabiullah, Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Cryptography & Information Security*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr Touhid Bhuiyan, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

As encryption transmission was becoming a common phenomenon, it is important for everyone to ensure their own privacy and data protection. The capacity of Fully Homomorphic Encryption (FHE) to carry out calculations throughout the encoded domain has drawn more attention. Training a machine learning model can be properly outsourced by utilizing FHE. The primary goal of FHE is to make sure performing computation on encrypted files without decoding anything besides the result. CKKS scheme is used due to work with polynomials and it provides a good trade-off between security and efficiency as compared to standard computations on vectors. Network risk and hacking into system a major risk to credentials information because they make it possible for an unauthorized user to retrieve sensitive and important data after analyzing computation results performed on plain data. Thus, this study provides a solution and comparison to the issue of privacy protection incorporating with machine learning in data-driven applications. Our proposed takes a normal dataset as input and provides an encrypted form of the dataset and after that we can perform computation on encrypted form. The used machine learning algorithm is Logistic Regression (LR) to predict the encrypted data. The methodology may maintain accuracy and security of the earlier methods to produce the conclusion, independently of the distributed situation.

Keywords: Privacy Preserving Machine Learning (PPML), Logistic Regression, Fully Homomorphic Encryption, CKKS scheme, Cloud Security, Data Security.

TABLE OF CONTENTS

CONTENTS **PAGE**

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

In various fields of science, engineering, and finance, statistical learning methodologies have achieved unheard-of performance thanks to advancements in processing power and an abundance of data. Federal agencies are also implementing Artificial Intelligence (AI)-based techniques to make vital infrastructure safe, secure, robust [1] and essential infrastructure for the healthcare industry is not an exception. Trained ML models for prediction systems are extremely data dependent, and cannot be seamlessly transferred as in the case of image or dataset classification tasks. The necessity to safeguard the integrity of information has grown even more crucial as the use of Electronic Document Management (EDM) in public administration organizations and the private sector has increased. Only electronic documents, whose validity, temporality, and classification in respect to secrecy are supported by current law, have superseded documents on physical media (paper). The term "temporality" of electronic documents refers to their privacy and preservation as prescribed by law, which, depending on the type and content of the document, often ranges between 5 and 30 years. The need to examine a document after processing and filing it is extremely uncommon, even though the law mandates its possession, especially when more than a year has passed since its filing [2]. Sensitive information like a large number of individuals is handled online, rendering them vulnerable to multiple attacks from both internal and external adversaries. Cyber-attacks may result in data leaks and serious financial repercussions. For instance, Facebook experienced cyber security incidents costing tens of billions of dollars as a result of malware. For building a stable and adaptable system for detecting unauthorized access, additionally deals with more complex and active attacks, an increasing number of security techniques [3] examine the aspects of internet activity from a variety of angles [3]. Understanding the use and operation of Internet services is essential to people's lives. Network traffic is crucial for this operation.

Due to interconnection of data between devices and computers, the privacy and security of these data have grown to be a significant problem. Tracking applications that persistently distribute personal data to contacts are upsetting both common users and patients who have contracted viruses of COVID-19 [4]. So, these personal data leaks lead to crucial ending most of the time. Among the difficulties faced by Internet traffic applications are the massive traffic volumes, the speed of the transmission system, the proliferation of services and attacks, and the variety of data sources and measurement methods. As the complexity of the network increases, more observational elements may become available to researchers, making it possible to collect and analyze heterogeneous data. Effective methods for the online assessment of enormous flows of measurement approaches are required by this development, and it is difficult to create deployed and tailored solutions. Additionally, it is conceivable to create large historical databases for backward examination as storage prices decline [5].

1.2 Goal of this Study

To amplify the connectedness of individual personal, systems as well as “things”, the number of informative gadgets connected to the internet is growing. A recent prediction states by the Internet Data Center (IDC) that there will be 41.6 billion internet of things devices, producing 70.4 zettabytes (ZB) in 2025 [6]. Additionally, efforts are continually being made to increase the efficiency of data collecting from the internet connected devices [7]. Unexpectedly large numbers of datasets are created and kept, mostly on cloud service provider platforms. Because of the cloud's great efficiency, scalability, and reliable data centers, the majority of apps and smart city solutions will be stored there. Cloud services could be used by residents of smart cities and internet service providers to host, create, or deploy a variety of smart urban infrastructures and apps [8]. Big data, artificial intelligence, internet and the general economy are all seamlessly connected thanks to cloud technology, which is crucial for accelerating the development of the modern economic model. Cloud apps are still widely used as a whole. The proliferation of ubiquitous monitoring and mobile cloud computing technologies has led to a variety of problems with regard to the processing and storing of huge and varied data [9]. In context, technology like cloud computing can be leveraged to make a new breed of interconnected, scalable data mining algorithms with unlimited potential to offer better information for making dicision in crucial applications such as big data. To move analytic tool services into the cloud, however, is now being hampered by growing worries about uncertainty about cloud technology caused by the sensitive data confidentiality. Consequently, it is essential to develop secure data mining applications that can make use of cloud resources [10]. All these issues are the purpose of this study on encrypted data to ensure data protection and security.

1.3 Cloud Computing

A laptop, a communal workstation, and resources for mankind are regularly maintained via the cloud technology online platform. Currently, a lot of clients have uploaded their personal sensitive and non-sensitive to the cloud. A key component cloud technology is security that enables the production of customized customer information that is stored on the clouds in a secure manner. Other parties are unable to fully satisfy the requirement for information, hence user authentication becomes necessary. Depending on the area, users can register and log in. Starting with cloud space, customers have the option to upload those data, share with one another, and download the data while necessary. The two layers of security are parts at first step of security, after which it will have been stored on separate servers a small sized file that has attacked in a subsequent security level that will better align concerns in the cloud. The client can then log in and send the file. The user can locate and download the data by entering a passcode. If the demonstration is effective and the cloud components that were split are delayed, the proposal might enable for the downloading of innovative data from the cloud [11]. Cloud computing is the delivery of computer services over the Internet ("the cloud"), including servers, storage, databases, networking, software, analytics, and intelligence. It aims to provide rapid innovation, flexible resources, and scale economies.

1.4 Cloud Storage

Cloud technology is an online tool that lets users store and share data. Cloud storage has many advantages, including limitless capacity of storing data, simple, secure and efficient accessibility of file, remote backups, and a cheap price of use. The cloud infrastructure can be splitted into four groups based on how it is actually used: private cloud storage, public cloud storage, personal cloud storage, hybrid cloud storage, and shared cloud storage. Instead of setting up an architecture or maintaining their own architectures and computers, organizations using cloud services to outsource their storage of data according to need. Users who had permission could only access the data. The extensibility, scalability, cost benefit and other advantages of the public cloud have attracted a lot of small and medium-sized organizations. Personalized clouds, also known as portable clouds, are same to publicly accessible cloud ways but differ in that they provide consumers with public services based on cloud storage [12]. Businesses can now easily store costly and delicate information in the private clouds to store many other types information in the cloud platform additionally. The appeal of this storing model keeps growing. The financial and healthcare industries benefit greatly from the cloud context, a leatest breakthrough in cloud services. Cloud services can be obtained by a society's corporate sector from shared clouds. Typically, these businesses must work together on specific projects or have similar viewpoints. Participants in the public cloud can hire outside contractors to manage the servers or they can collaborate to design the architecture. Due to the nature of cloud services, difficulties with security and data privacy will inevitably arise during this process.

1.5 Cloud Security and Issues

Many cloud service providers were reluctant sharing their technology with the public for managing and developing a safe cloud landscape is a challenging process, it is imperative to evaluate the efficiency of security measures of cloud providers' too. When a platform is designed for a big audience, security concerns with cloud data can lead to financial losses as well as a bad reputation. These concerns are what have led to the widespread use of this new technology. Concerns about security and privacy are raised by the cloud service providers who run the services. Information stored in the cloud must be protected using appropriate security measures. The adoption of cloud computing is currently facing the most urgent security and confidentiality concern. Storage and security were intertwined, therefore improved security requires a suitable storage system. The supplier should set up security procedures and systems to safeguard the technology as well as the applications and data of their customers. The following issues that cloud storing systems to ensure data privacy and security face:

- the ability to manage fine-grained data access.
- Malicious cloud service providers may deliver false integrity audit results.
- Side channel attack

- Malicious cloud service providers do not comply with clients' requests to completely delete data from the cloud.
- Privacy-preserving.

Even while cloud storage has been around for a while, the Internet of Things, smart cities, and the digital economy all greatly depend on it. Access control, encryption, and security are three common studied approaches for privacy protection, however they are distributed and typically not systematic [13]. In context of all this, it is crucial to draw conclusions from recent research on a variety of technologies that support privacy and security in cloud computing. The majority of current privacy protection techniques have limited scalability and no dynamic overcurrent protection. Data security becomes crucial since cloud computing is networked and virtualized, making it impossible for users to immediately determine the memory location and data partitioning, etc. In cloud computing, data security is frequently preserved through identity verification or data encryption [14]. Now a days most common encryption algorithm is symmetric and asymmetric encryption. Among this two Fully Homomorphic Encryption is a prominent encryption algorithm [15].

1.6 Fully Homomorphic Encryption (FHE)

There are many types of homomorphic encryption, those are:

1. Fully Homomorphic Encryption (FHE)
2. Slightly Homomorphic Encryption (SHE)
3. Partially Homomorphic Encryption (PHE).

Recently, a significant technological improvement for protecting interests, has advanced to the point where it may be utilized in real-world applications is fully homomorphic encryption (FHE) [16]. The nGraph-HE design transforms neural networks into efficient FHE-based systems, enabling private reasoning [17] for instance. Fully homomorphic encryption (FHE) is a cryptographic technique that provides good security assurances, despite the fact that the server will only ever be able to view data encryption.

A homomorphic encryption technique allows for the execution of specific operations with inputs and outputs that can either be plaintext or cipher text. Machine learning problems range from linear and regression analysis to encrypted neural network prediction, which might be used to process private data. solutions using machine learning, such as shielded phishing email identification [18]. Bootstrapping which is a technique that can homomorphically assesses the encryption decryption circuitry (i.e., device which converts cipher text into plaintext), can be used to get around this restriction. There are some other schemes of homomorphic encryption:

1. CKKS (Cheon, Kim, Kim and Song) [19],
2. BGV (Brakerski-Gentry-Vaikunathan) [20] [21],
3. HEANN (Homomorphic Encryption for Arithmetic of Approximate Numbers) [22]

Cipher text noise exposure is reduced to a predefined lower threshold [23]. Modern systems used SIMD-style parallelism to further boost speed by make unreadable format hundreds of readable variables in a one cipher word. These APIs at the very least reveal the functioning of homomorphic addition together with key generation, encryption, and decryption. Although FHE libraries greatly improve the efficiency of developing FHE-based systems, they nevertheless necessitate a high level of understanding of the underlying scheme because they are still extremely basic cryptographic libraries. Without advancement tools, it is challenging to manually implement FHE-based computing including the required functions mathematically or using arbitrary specified arithmetic type framework; this requires substantial skill in both high-performance numerical integration and cryptography.

Processors commonly use FHE libraries to implement crucial homomorphic calculations, decryption, and encryption operations. HE is built on noise-level encryption techniques, where noise rises even as more calculations are done on data. Homomorphic encryption (HE) enables the secure outsourcing of computing to the clouds by allowing computation on encrypted information (ciphertexts). Covers for known frameworks are being established in an endeavor to take the first steps for ensuring compatibility, concurrent with efforts to standardize FHE method APIs [24]. Hardware multipliers have been recommended for FHE compute components in previous articles [25]. Near future in 2025, " at least 20% of businesses will be able to implement initiatives that use fully homomorphic encryption. " predicted by Gartner. For instance, machine learning technologies have shown to be useful and accessible while also providing innovative efficiency thanks to automatic optimizations which greatly outperform earlier hand-made approaches by experts. A "Learning with Errors" (LWE) method, where the majority of modern FHE algorithms rely on encrypted messages becoming jumbled with noise.

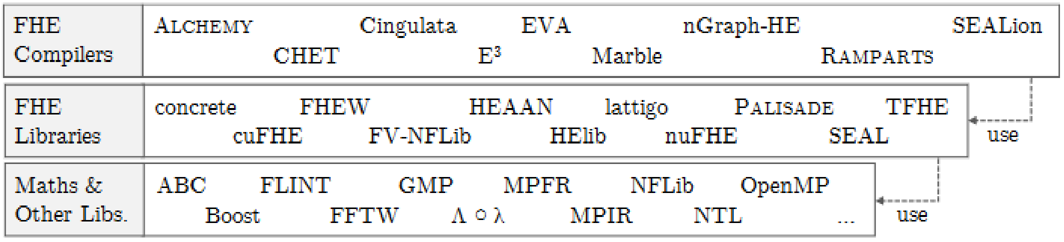


Figure 1.1: Tools of Fully Homomorphic Encryption [16]

FHE has been used in the medical field to ensure that genomic sequence confidentiality [26] applications over significant datasets. Higher-level tools—often started referring to as FHE compilers—have recently been employed to translate regular programs into FHE-based representations. In order to address some of the technological challenges in this field, researchers have observed a surge in the creation of technologies that aim to improve availability and reduce obstacles to participation in this discipline. Technologies like these are made to make FHE approachable to non-experts by gradually introducing complex advancements that were previously only available to professionals. As opposed to, FHE libraries frequently employ existing libraries for non-FHE-specific tasks including parallel processing, rapid numerical calculations, and others. During homomorphic processes, the encrypted text grows louder. For this reason, even merely a preset amount of sequential operations arithmetically (variable called as multiplicative depth) can be performed when decryption becomes impractical. Fortunately, during the past few years, FHE has greatly reduced its computing requirements due to significant improvements in hardware support, efficient algorithms, and low-level representations.

1.7 Problem Statement

Some of the most frequent and harmful cyber-security attacks have been documented against systems across numerous industries during the past few years. Security analysts predict that this year will set new records for both network intrusions and data security risks; organizations should be informed about potential assaults to ensure that their appropriate defenses are in place. Most frequently in security risk is 9th type [27].

1.8 Research Question

- What kind of encryption technique is employed throughout the encryption process?
- Why are credentials encoded using Fully Homomorphic Encryption (FHE)?
- How does machine learning identify harmful network threats?
- Why proposed system is the best one?

1.9 Objectives

Everyone needs their personal data to be protected and their privacy to be maintained, even though encrypted transmission has become commonplace in the online world. However, to guard against malicious and unauthorized data sent by attackers, traffic encryption is utilized. Fully Homomorphic Encryption (FHE), which allows for calculations to be performed over the encoded domain, has recently attracted a lot of attention as a result of the growing security requirements for data mining. The FHE technique can be used to effectively outsource model training to public cloud computing platforms that lack reliability but are nonetheless efficient. The main goals are

- Encrypted data publishing using FHE and processing encrypted results
- Preserving users' privacy with respect to efficient and secure content processing
- Without decoding dataset getting the end outcome
- Using machine learning for data driven application and offering effective solution with good model performance.

1.10 Key Contribution

- Encrypting data using FHE and building model
- Encrypted information predicting using LR
- Encrypted machine learning data model building

1.11 Thesis Layout

Chapter 1 Introduction – Goal of this Study – Cloud Computing – Cloud Storage – Cloud Security and Issues – Fully Homomorphic Encryption (FHE) – Problem Statement – Research Question – Objectives – Key Contribution – Thesis Layout

Chapter 2 Literature Review – Homomorphic Encryption Scheme – HE Applications and Advances – Machine Learning (ML) Classification Protocol

Chapter 3 Methodology – Proposed System Overview – Homomorphic Encryption (HE) – CKKS – Threat Technique – Encryption of the data – Dataset – Pre-processing – Feature Selection – Classification – Model building – CKKS scheme for encryption – CKKS parameter – Ciphertext modulus q – Ciphertext dimension N – CKKS keys – CKKS internal operations – Relinierization – Rescaling – TenSEAL CKKS context – Plain Tensor Creation – Prediction over encrypted data – Loss function – Parameters Update – Sigmoid Approximation (Sig(a))

Chapter 4 Implementation – Test Data Length – FHE with or without Network Analyzer – Implementation steps – Machine Learning – Our proposed model

Chapter 5 Results and Discussion – Results – Discussion

Chapter 6 Conclusion – Summary – Conclusion – Future Work

References

CHAPTER 2

LITERATURE REVIEW

2.1 Homomorphic Encryption Scheme

Using homomorphic encryption, user can compute precise functions on unreadable format values without being aware of the component of the values. This choice results from the circuit's encrypting characterization as nothing more than a team homomorphism that safeguards group procedures. The same calculation can be performed on encoded or plain values thanks to group homomorphism. When sensitive operations are delegated to unreliable parties in several implementations, this adaptability overcomes security problems. The properties and mechanics of the HE systems are discussed in this section.The development of schemes like HE can handle an infinite amount of homomorphic operations using random functions, or FHE, was sped up by the rising trend of cloud-based services. The first practical and workable FHE scheme is Gentry's [28]. It is founded on mathematical ideal lattices and serves as both a description of the scheme and a potent foundation for establishing FHE. However, it is not a realistic plan philosophically or practically. Particularly expensive in terms of computing is the bootstrapping step, which is the intermediate refreshing method of a processed ciphertext. As a result, numerous updates and fresh ideas were put forth in the years that followed.

The importance of third-party information security issues is increasing as cloud computing usage expands. However, standard cryptogram systems find it difficult to successfully extract encrypted data and carry out other tasks. Jian Li and colleagues [29], [30], [30] have developed a practical simple FHE approach that relies only on fundamental operations mathematically and is deduced from the cryptosystem invented by Gentry in order to guarantee privacy preserving cloud storage. This method successfully meets the need for collecting ciphertext as well as other computation on unsafe servers because encoded data could be operated openly without endangering the security of the encryption system. In order to address the issue of data security in cloud computing systems, Feng Zhao et al. [31] developed a unique form of data security remedies for insecurity of cloud computing systems and have created the instances for implementation. This cutting-edge security solution successfully paves the path to widespread scope of application, secure data transmission, and preservation of cloud computing by being fully capable of managing and recovering encoded data. Jung Hee

Cheon and colleagues [19] have put forth a way for developing an approximation arithmetic HE system that enables approximately multiplying and adding encrypted messages as well as a cutting-edge rescaling technique for managing plaintext size. This rounds the plaintext by reducing the modulus of the ciphertext to a lower value. Additionally recommended was a fresh batching technique for RLWE-based (Ring Learning with Errors) architecture.

2.2 HE application and Advances

Viand et al. [32] shows combination of IPFS and blockchain network to manage PHR data and metadata; Adoption of FHE techniques to reduce the demand for unencrypted data, supporting calculation on encrypted data; End-to-end encryption standardization to allow PHR data sharing and interoperability; Segregation of responsibilities regarding PHR to improve how individuals control personal data. The HE-friendly network, which was developed in previous works to redesign the machine learning model to be compatible with the HE scheme by swapping out the typical activation functions for straightforward nonlinear polynomials [33]–[37]. Though the CIFAR-10 dataset's greatest classification accuracy for the HE-friendly CNN with the straightforward polynomial activation function implemented by word-wise HE is 91.5% [20], a more effective PPML machine learning model has not yet been shown. Hamza et el. [38] presented an overview of privacy-preserving techniques in Big Data analytics with homomorphic encryption algorithms. Jassim et el. [39] have shown on the basis of data analysis that our project to store and dispense data quickly to the user for long term digital measurements as a vital patient over the course of his / her treatment that is longer than what can be obtained within the hospitals and even outside. Al Badawi et al. [36] introduced OpenFHE, a new open-source FHE software library that combines a number of fresh design principles and ideas with some design elements from earlier FHE projects like PALISADE, HElib, and HEAAN. The following succinctly describes the primary new design elements: (1) We assume from the outset that all implemented FHE schemes will support bootstrapping and scheme switching; (2) OpenFHE supports multiple hardware acceleration backends using a common Hardware Abstraction Layer (HAL); (3) OpenFHE includes both user-friendly modes, where all maintenance operations, like modulus switching, key switching, and bootstrapping are automatically invoked by the library, and compiler-friendly modes, where an external compiler makes the necessary modifications; (4) OpenFHE supports multiple hardware acceleration backends using a common Hardware.

2.3 Machine Learning (ML) Classification Protocol

Arita and Nakasato et al. [40] developed the fully homomorphic encryption system FHE4GT, which, in the absence of the secret key, can homomorphically compute the encryption of the greater than bit that determines if x>x′ occurs. Then, using machine-learned parameters, we build the homomorphic classifier homClassify, which can homomorphically categorize supplied encrypted data without decrypting it. Yasumura et al. [41] thought about a situation where a company wants to offer a classification model and classification services via a cloud server as part of machine learning as a service, but they also want to protect the privacy of the classification model because it was trained using information that is just as private as the client's information and referred to this third party as a decryption server in this research. Kim et al. [42] and Li et al. [43] established a protocol for safe training and classification in which they introduce a third party who possesses the secret key of the system and is in charge of decrypting all ciphertext.The advantages of real-world applications using FHE or SHE are shown. The authors also examine the usefulness of applications in the fields of medicine, finance, and advertising and demonstrate their major computational cost limits are explained by Naehrig et al. [44] and Archer et al. [45].

CHAPTER 3
METHODOLOGY

3.1 Review of proposed system

The client-end data is encrypted before sent to the cloud platform using complete homomorphic encryption. Initially after loading the NSL-KDD dataset, the preprocessing has been done. Then the feature selection helped to filter unnecessary feature to select only the vital features to enlarge the classification rate. Finally, the encrypted data sent to the server and predicted by using logistic regression model classifier.

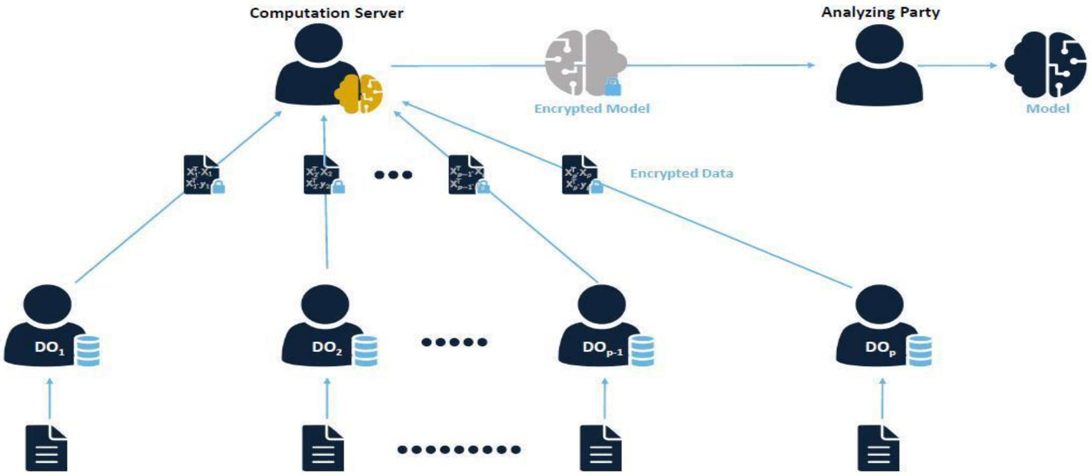


Figure 3.1: Overview of Proposed System

First multiple users create some data and transmitted to computational server using framework made by Cheon-Kim-Kim-Song known as CKKS which is a fully homomorphic encryption. The encrypted data will be predicted using the logistic regression model, and the user will be notified either it’s a blocked request or a normal request.

3.2 Homomorphic Encryption

To manipulate cipher messages solely homomorphic encryption is a type of encryption that allows one using available information publicly, notably without access to any secret keys. The term "homomorphic" in cryptography refers to this relationship because the field of both transmissions and cipher texts are frequently correlated with one another, and operations on cipher texts mirror operations on data being encrypted.

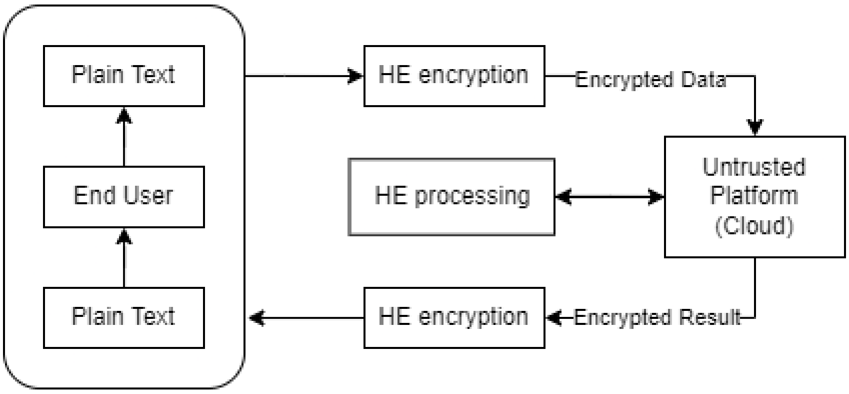


Figure 3.2: Homomorphic Encryption Process

Homomorphic encryption differs from other conventional methods in that it enables calculations on the encrypted files without requiring knowledge of the secret key used for encryption process. The outcome of these activities is similarly unreadable format, and need owner’s private key to decrypt data.

Let, x_1 is a factor, x_2 is addition and multiplication homomorphic

$$x_2 \Rightarrow y(z_1 + z_2) = y(z_1) \cdot y(z_2)$$

or

$$x_2 \Rightarrow y(z_1 \cdot z_2) = y(z_1) \cdot y(z_2)$$

The FHE system Gentry[28] put a method named forward for encryption that allows inconsequential addition and multiplications operations simultaneously on encrypted information. Considering a cryptosystem of probabilistic secret key is defined using multiple polynomial equation

$$f = ek * q + 2 * s + n$$

here,

n reflects bit as ‘f’ for encryption,

two arbitary integers r & q,

and ek represented as public key conditionally 2*s is lesser than ek /2.

If modulo operations has done twice in this way

$$n = f \bmod dk \bmod 2$$

where dk is the referring private key, the decryption is then retrieved.

The recommended strategy is splitted into four main sections, as follows:

a) **KeyGen (λ):** Using this function two random variables will be returned one is public and another private key represented as sk and pk. Result will be calculated according to safety variable which explains encryption key and encrypted data length.
b) **Encrypt (sk, n):** This component converts bit signal of 0 or 1 is encrypted (converted) into a huge number of the range of 7 bits that has same parity bit like the original binary values.
c) **Decrypt (pk, ci):** The input cipher text on the basis of useful secret key pk to decrypt the component to plain text.
d) **Evaluate (sk, C, *):** This component provides the cipher text outcome of the completed circuit to mask the encrypted values.

Building process of homomorphic encryption in detailed explained by OpenMinded [46] and a summarizing showed in Figure 3.3. The initial approach creates a clean, noise-free fresh cipher text by re-encrypting using public key to each bit of the cipher text. The unique cipher text can then be decoded using the specific secret key which has been encrypted after the inner surface of encryption has been removed. It is obvious that the actual plaintext was produced by twice decrypting the bootstrapped variables. By employing a bootstrapping process, the SHES (Somewhat Homomorphic Encryption System) builds an FHES (Fully Homomorphic Encryption System) that permits endless level of additions and multiplications. Rebooting, on the other hand, requires time since re-encryption often involves huge integer numbers.

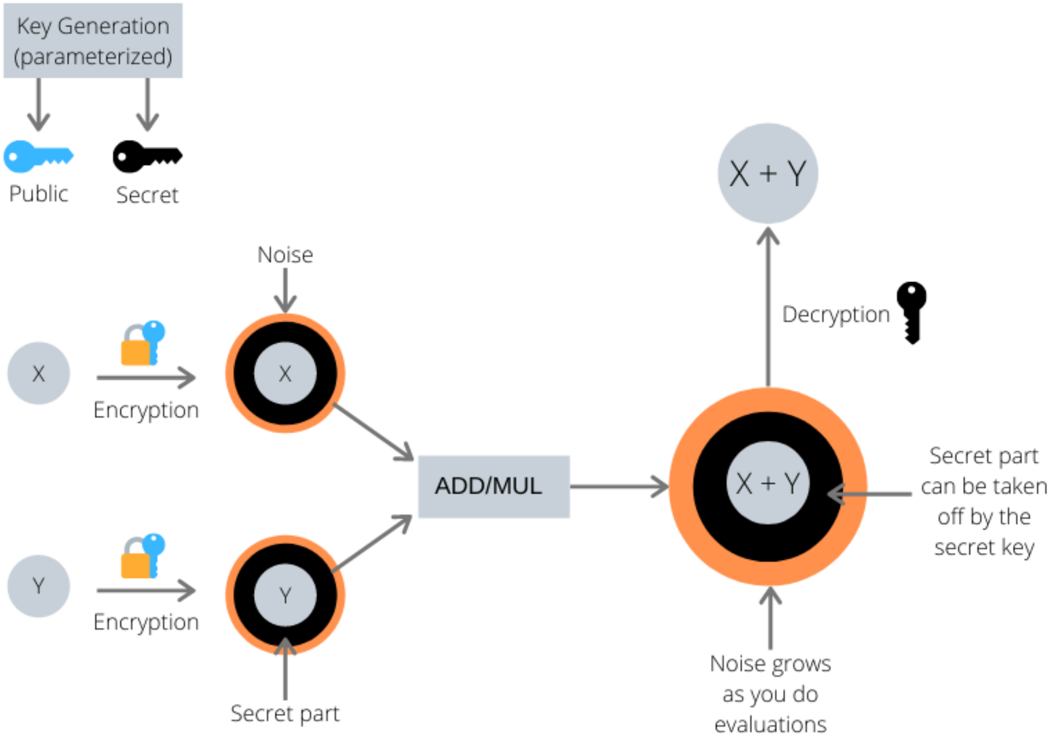

Key Generation (parameterized)
Public
Secret
Noise
X
Encryption
X
ADD/MUL
Y
Encryption
Y
Secret part
X + Y
Decryption
X + Y
Secret part can be taken off by the secret key
Noise grows as you do evaluations

Figure 3.3: Summarizing of HE using Graphics

3.2.1 Cheon-Ki-Kim-Song (CKKS) Homomorphic Encryption

Fully homomorphic encryption (FHE) approach known as CKKS is beneficial at processing real (or complex) numbers, which are the standard format of data for a variety of applications [19]. When using other FHE methods, such as the (B) FV and BGV methods, to deal with fixed-point actual numbers, the ciphertext length grows exponentially with respect to level. The most complex circuit that can be analyzed homomorphically without performing bootstrap process that determines the level of the ciphertext. In contrast, depending on CKKS method levels, the length of the ciphertext increases at quadratic rates.

Due to some mistakes in its encrypted data, the CKKS system presents a compromise between correctness and efficiency. Faults in encrypted data are spread as well as combined together by homomorphic activities.

As operations on encrypted data are carried out, the upper constraint on errors in encrypted data becomes weak in some cases and inefficient bound. Additionally, reducing errors is eventually much more crucial while using CKKS method to decrease the risk of attack because CKKS has recently been the focus of attacks. Therefore, a novel approach that can successfully manage errors in encrypted data is needed. Once the error has been effectively managed, this recommended technique can be applied for reducing the computation time and error.

3.3 Threat Techniques

It is essential to know the threat approach for a structured description of each piece of information that affects how well an application is protected. Attacks by injecting some codes are the main area of the threat model focusing for this study, which is in 3 out of top 10 for 2010 publication based on the OWASP [47]. Injection is a process of taking advantage of a computer vulnerability caused by handling false data. A computer programmer that is vulnerable gets "injected" with information via injection. Then it will function opposite to the way it should work or produce distinct results. A successful injection of code can have negative consequences sometimes.

3.4 The data Encryption Process

The data is encrypted using HE method, is reliant on matrix format operations. The goal of the encryption process is to create highly accurate machine or deep learning methods, using encrypted data for training and testing purpose, and prevent the developers of the methods from continually granting access to the plain text. Homomorphic encryption can safeguard a variety of electrical and electronic applications in the power infrastructure by providing the following features for protecting data stored on a cloud platform:

- Providing controlled utility corporations with access to data analytics.
- After the data has reached to webserver, HE secures the devices to prevent the attempts of eavesdropping.
- HE is able to stop unauthorized data exchange
- HE may render data breaches from attacks like man-in-the-middle or eavesdropping incomprehensible for the hackers.

3.5 Dataset

The DARPA 98 [48] and KDDcup 99 [49] were mostly used datasets in the previous to examine various topics, but the dataset's evaluation of anomaly detection was mediocre due to statistical degeneration. The recently released NSL KDD datasets, which are detailed in, are a result of the inherent problems with the KDD dataset. Even though it is very difficult to reflect the system in real-world, it can nevertheless be used as a evaluate the data set by researchers comparing different detection strategies.

Analytical research revealed that there may be substantial faults in the data set, which could have a negative impact on the effectiveness of the systems and result in a very incorrect evaluation of methods for anomalies detection. The proposed answer to address these issues is a distinct set of data named NSL-KDD, that is composed of selected necessary features from the whole KDD dataset.

- The extractor will not produce any biased result, so training set is unnecessary.
- The testing set has no consolidate files, which makes efficiency levels higher.
- The number of documents picked from every category of records with high level difficulty which is negatively correlated with proportion of docs in the original data set for KDD.

21 out of 37 threats in the set of validation in the dataset that are present in the training set are also present. The sample test set incorporates additional assaults that are not provided in training dataset while training dataset only contains the well-known type of attacks. Four main categories into which the assault methods are separated are DoS, Probe, U2R, and R2L.

Targeted broadcast is how attacks swarm a trafficked target. Attacker normally sends a ping packet to the network address using network and the Internet called surf amplifier, that receives and responds to targeted broadcasts. Attacks in the training phase total 45927, whereas those in the testing phase number 7456. Only the testing phase detects attacks like mailbomb, apache, and process table.

3.6 Data Pre-processing

Machine learning algorithms need to be trained on a massive amount of data before they can produce good results. Since the data is frequently stored in storage units in the cloud as files, databases, etc., it cannot always be used the right away for training a model. The study must pre-process or improve the data before sending it to the machine learning model for training to yield better results. Because training examples help algorithm like machine learning understand how presented value relates with the class, it may quickly understand the training data and generate better results. Pre-processing information is a phase that involves multiple steps. The procedure involves several processes, including caching the data into techniques of machine learning, resolving the neglected variables of the datasets, then the scaling process of the data with the aid of normalization and standardization. Lastly splitting the sample into training and testing datasets. ln order for the research to use test set assessing the effectiveness of the classifiers automation for learning and to give learning classifier the training set for both training and testing cases.

3.7 Features Selection

Feature selection is a typical pre-processing step to extract data from main dataset (FSS). In order to increase accuracy, it minimizes dimensions and gets rid of extraneous components. It makes reference to the problem of identifying the qualities that are necessary for class prediction. Filter, wrapper, and embedding techniques are the three different categories of feature selection approaches. According to this study, filter techniques are used to assess the significance of features in relation to the dependent variable. Classifiers are significantly quicker than wrapper approaches because they don't require a training model.

3.8 Classification

A machine learning approach called logistic regression is used to develop classification models. In order to keep presentation of this work as simple as possible, we mostly used the binary classification. Logistic regression considering the model below:

$$log\, log \left[\frac{P_r\,(Y=0 \mid X=x)}{P_r\,(Y=1 \mid X=x)}\right] = \langle w, (1, \quad x)\rangle$$

$$[(1, \quad x)\; for\ a\ vector\ extended\ with\ 1\ from\ x\ ]$$

Here,

$$p_r(Y=1 \mid X=x) = \frac{1}{1+e^{-\langle w,\ (1,x)\rangle}}$$

$$p_r(Y=0 \mid X=x) = \frac{e^{-\langle w,(1,x)\rangle}}{1+e^{-\langle w,(1,x)\rangle}}$$

for X as vectorized input of n features available in dataset
class Y, a weighted vector w ∈ R^{n+1}.

The goal of the logistic regression training, given n samples $\{(x_i, y_i)\}_n$, is to find weighted vector w which minimizes the negative log related function.

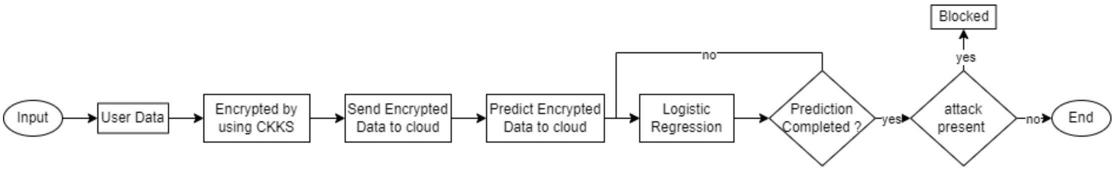


Figure 3.4: Flowchart of the Proposed System.

3.8.1 Model Building

PyTorch-like model that can train an encrypted logistic regression model on encrypted data while also forwarding encrypted data and back propertying to update the weights.

3.8.2 CKKS scheme for encryption

TenSEAL package developedboth BFV and CKKS techniques, is one of the newest homomorphic strategies. A conversion of TenSEAL package created in C++ named latigo library, provides access to BFV and CKKS in the Golang language. The ability to write in Golang creates new possibilities for homomorphic functionality in web services. The assessments of the official procedure show that the two languages operate rather similarly. CKKS activation in response extends up to complex values, despite BFV only allowing actions on integers. The Leveled Homomorphic Encryption method Cheon-Kim-Kim-Song (CKKS) allows approximation across complex quantities (real number). Text encryption by using CKKS

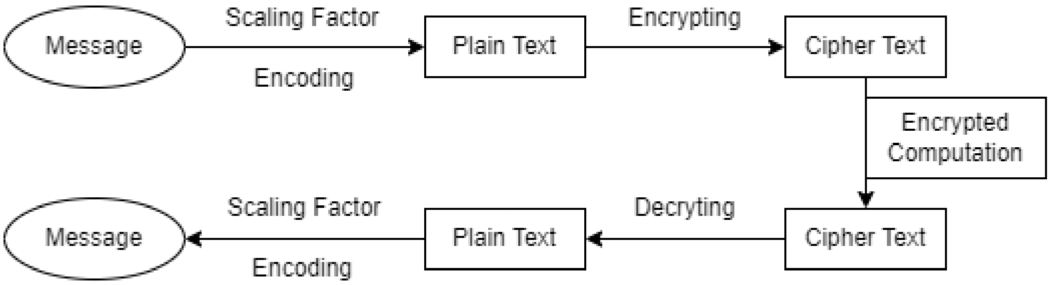


Figure 3.5: CKKS for Text Encryption

3.8.3 CKKS Parameter

The CKKS method is used to convert a vector of real numbers into polynomials in plaintext.

Scaling factor (Sf): Scaling factor determines binary description of the precision of the number encoded. Integers are created in the conversion real numbers.

$$\text{Scaling factor (Sf)} = 2^{p}$$

3.8.4 Ciphertext modulus q

Functional parameter that determines most of the calculations that are allowed (tolerated noise quantity). A user-specified multiplicative depth value is frequently used to set implicit values.

3.8.5 Ciphertext dimension N

A lowest value is computed based on the security level that was chosen and the cypher text modulus q. In addition, the vector used to encrypt actual values is N = 2n times larger.

3.8.6 CKKS keys

In the CKKS public key encryption system, both secret and public key are generated. While public key can be shared, private key is used for encryption and must be kept secret while decrypting data*. A second class of public keys is required for this operation, called key relinearization created by the owner of secret key. Galois keys are an additional type of public key needed to operate operations on batched ciphertexts, such as rotation of the encryption vector. One use for vector rotation is the summing of encrypted batched vectors.

3.8.7 Internal Operations

3.8.7.1 Linearization

TenSEAL library does the procedure instantly following each encrypted multiplication. This makes it possible to utilize a polynomial pair rather than a triple, which multiplies two fundamental plaintexts when they are decrypted using conventional encrypted circuits, that only needs the secret key not the whole square. Ciphertexts size will always be similar length and employ the identical circuit for decrypting if reshuffling is carried out after each multiplication between ciphertext-ciphertext.

3.8.7.2 Rescaling

In CKKS, modulus switching is referred to as rescaling since the underlying encryption plaintext rescales in essence and eliminates a set number of the least significant bits from the communication. Regardless of whether the job is encrypted or plain, TenSEAL completes it on its own after each multiplication. The prediction error dramatically increases as the number of homomorphic multiplications increases. For solving this, the bulk of HE techniques typically use a modulus-switching approach. In the context of CKKS, the modulus-switching procedure is referred to as rescaling. The prediction error will increases linearly instead of exponentially following a homomorphic multiplication and the rescaling method.

3.8.7.3 TenSEAL CKKS Context

The primary structural component of the library is the TenSEAL context. It generates and stores the necessary keys for a calculation that is encrypted. The mechanism generates public key to encrypt and Galois keys for rotation, the relinearization keys for relining the ciphertexts, a secret key for decoding, and the Galois keys for rotation. The same class will also manage the thread-pool, that controls the number of tasks should run simultaneously when carrying out parallelized activities. Furthermore, while ciphertext is being computed, the context may be configured to relinearize and scale it automatically.

3.8.7.4 Plain Tensor Creation

The PlainTensor class converts fundamental tensor forms into encryption forms offered by the library TenSEAL. First step is very essential for using TenSEAL to build an encryption tensor. This conversion is also carried out independently by the encryption tensor constructors.

3.9 Prediction Over Encrypted Data

Logistic Regression can be used to train as a single node, neural network of one layer (without any coding). It employs this strategy to contrast encrypting learning and evaluation. The primary goal of the study is evaluating logistic regression model with plain variables on encrypted testing set (or encrypted variables) to create a TenSEAL context that details about variables and proposed method. Here, the study opts for safe, controllable variable that enables to perform a single multiplication of the training model and dataset. That is adequate evaluating the proposed model of logistic regression, but the study will look at whether learning from encrypted information requires the use of more detailed variables.

Algorithm for TenSEAL:

Input: Mini-batches of training data $\{Z_i\}$ where $Z_i \in R^{m\times n}$ (i.e., the mini-batch size is m), parameters γ and η, the number of iterations K and a polynomial approximation of sigmoid σ'.

Steps of the Algorithm: Weighted vectors w, v $\in R^n$

```
Step 1: Initializing weighted vector:  w, v ← 0
Step 2: for k in [1..K ]  do
Step 3: select mini-batch $Z_i$ (orderly or randomly)
Step 4: $a = Z_{i.}\ v$
Step 5: for j   in [1..m]  do
Step 6: $b_j = \sigma'(a_j)$
Step 7: end for
Step 8: $\Delta = \sum_{j=0}^{m-1} b_j \cdot Z_i[j]$
Step 9: $w^+ = v - \gamma \cdot \Delta$
Step 10: $v^+ = (1 - \eta) \cdot w^+ + \eta \cdot w$
Step 11: $w = w^+, v = v^+$
Step 12: end for
```

Output: Encrypted form of the whole dataset.

3.9.1 Loss Function

The cross entropy loss function is in binary mode with regularization where $y^{(i)}$ is i'th expected label, $y^{(i)}$ is i'th logistic regression model output and θ is a weighted vector of n-size.

$$Loss(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(\hat{y}^{(i)}) + (1 - y^{(i)})log(1 - \hat{y}^{(i)})] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

3.9.2 Parameters Update

To update the parameter, the regular rule is used where $x^{(i)}$ is the i'th number of input data:

$$\theta_j = \theta_j - \alpha \, [\frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x^{(i)} + \frac{\lambda}{m} \theta_j]$$

We choose to utilize an $\alpha = 1$ to decrease a multiplication and set $\frac{\lambda}{m} = 0.05$ instead due to the homomorphic encryption constraint, which leads to the update rule that follows:

$$\theta_j = \theta_j - [\frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x^{(i)} + 0.05\theta_j]$$

3.9.3 Sigmoid Approximation (Sig(a))

The lower the degree of the polynomial, the better, as we seek to execute as few multiplications as feasible, to be able to employ fewer parameters and so optimize computation. Since we cannot just compute sigmoid on encrypted data using degree 3 polynomial, we must approximate it [50]. Sigmoid function range [-5, 5].

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

The logistic function in linear regression is a type of sigmoid, or a collection of functions having related but distinctive features. Using the mathematical function sigmoid, any real value can be transformed into a likelihood between 1 and 0.

$$Sig(a) = \left(\frac{1}{1 + e^{-a}}\right)$$

To reach the final part, this includes training a secured logistic regression method using secured input. The weight for the model must need to be decrypted and then re-encrypted for each epoch. After upgrading weights, the study cannot use it to perform the necessary multiplications and ought to go back to the initial level of the ciphertext. In reality, this would mean giving weights back to the secret key owner so they may be decrypted and re-encrypted. The number of connections each epoch will be restricted to a few Kilobytes in such a case. Logistic regression framework to predict information in figure 3.6

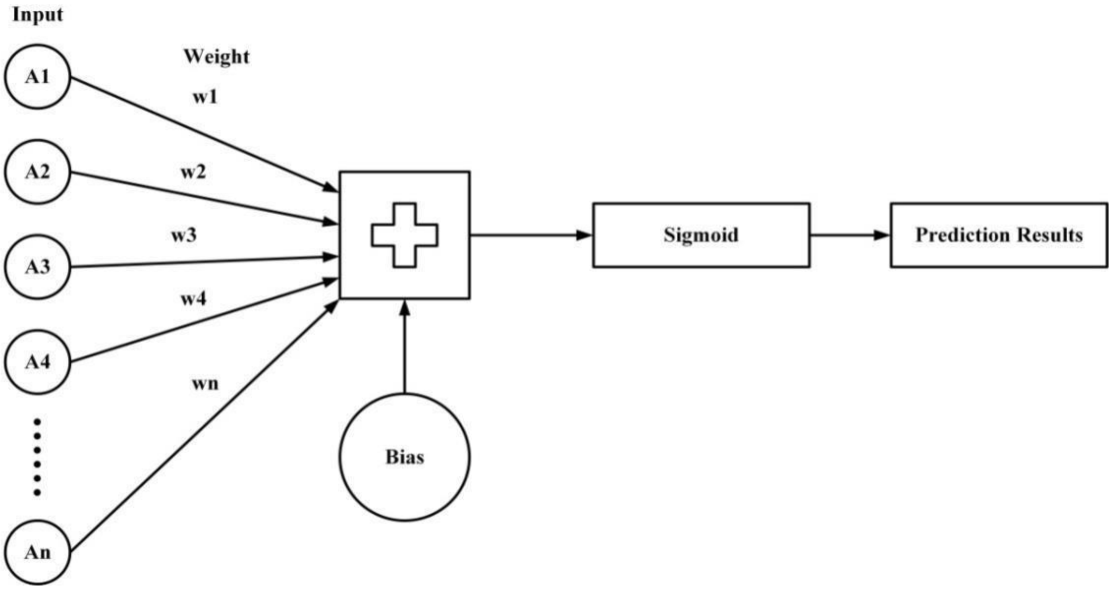


Figure 3.6: Logistic Regression to Predict Encrypted Data

HE will be utilized to generate an encrypted method and labels after inference and training. The only person who can decipher the results and receive either the tags or classifiers for categorization is the user who has access to HE's secret key. There is a possibility for the cloud may run homomorphically in encrypted files learning model to create encrypted logistic regression model. The data owner could then receive this technique for decryption. This allowed the data owner to outsource training process without providing whole cloud access to private data or trained model.

CHAPTER 4
IMPLEMENTATION

4.1 Test data length

The total evaluated test set has 5038 entries which is evaluated in 37 seconds. It provides the accuracy in 4648 tests set among the whole entry set. Due to use of random choosing the accuracy fluctuate according to the dataset. The more data the model will get to know the more accuracy will increase.

4.2 FHE with or without Network Analyzer

Figure 4.1 illustrates how to create a Fully Homomorphic Encryption to safeguard data with feature sets to achieve the desired goal. It can be done with or without a network analyzer. The host process's network activity is monitored by the network analyzer. Some of these tasks include replication procedure and contact with the centralized command and control server. Although some of these signals are repeated and arranged in a different way to improve the accuracy in identifying legitimate network traffic. In fact, these qualities are necessary to achieve high detection accuracy. In order to improve the detection of encrypted data, new features are also proposed.

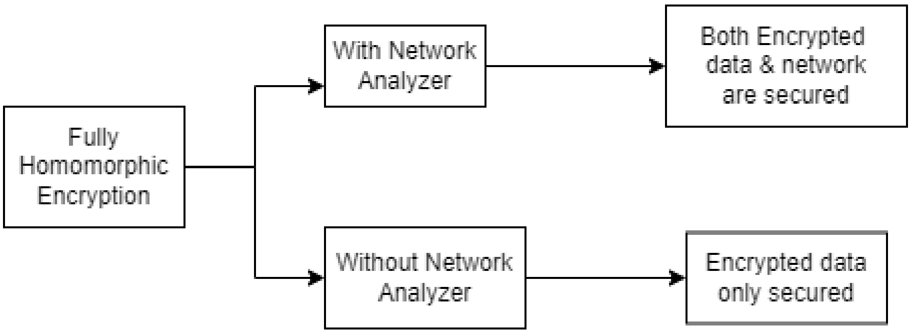

Fully Homomorphic Encryption
With Network Analyzer
Both Encrypted data & network are secured
Without Network Analyzer
Encrypted data only secured

Figure 4.1: FHE With or Without Network Analyzer

Without a network analyzer, Fully Homomorphic Encryption just secures the encrypted data. When Fully Homomorphic Encryption and a network analyzer are used, the network and encrypted data are both secure. We might conclude from this that network and data protection is quite precise. We may safely save our data and sensitive information utilizing this technique.

4.3 Implementation Steps

The efficiency of the proposed method for authenticating data encryption during aggregation is assessed. Here, the difficulty of computing the key needed encrypting the message conveyed between models is crucial. After determining the effects of a secret key, the effect of n is assessed. In this study, Encryption time, Decryption time, 2-Encryption time, and 2-Decryption time are used to calculate the operating time of the proposed approach. While 2-Encryption and 2-Decrypiton build a second layer of encryption, Encryption and Decryption specify a one layer of encryption.

4.3.1 Machine Learning

Traditional Machine Learning enables two specific tasks known as regression and the classification part of a dataset. Traditional machine learning or deep learning model we directly provide an explicit computer program to compute the feature and mapping accordingly. After all the necessary feature selection model provides the necessary results on test set.

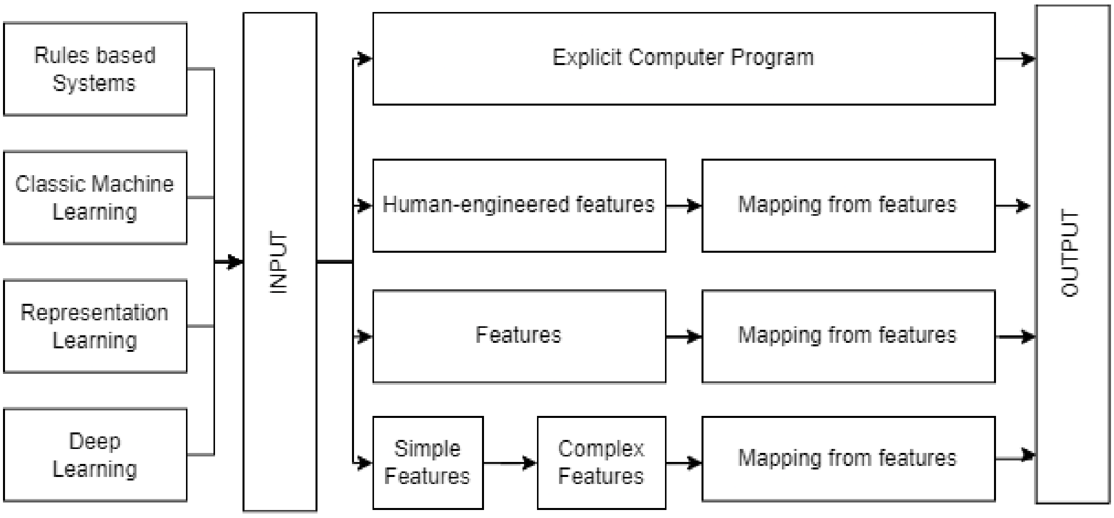


Figure 4.2: Various Type of learning ([51])

4.3.2 Our Proposed Model

In our model all the model will be encrypted using CKKS scheme as well as the test system is encrypted too while training the model and the test case evaluation.

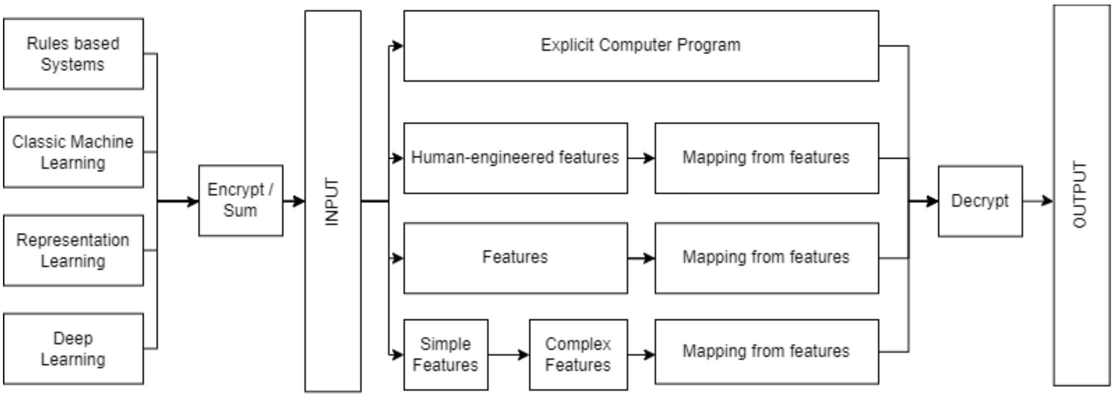
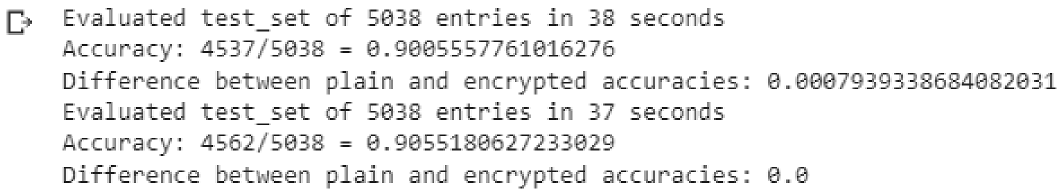


Figure 4.3: Our Proposed System Input & Output

CHAPTER 5
RESULTS AND DISCUSSION

5.1 Results

According to our proposed CKKS scheme we used polynomial modulus of 8129 and the saved key was Galois key, is a method of action for cryptographic block cipher. It is highly used for its computational cost, time and performance. Modern high speed communication channels can be used to achieve high throughput rates using cheap hardware resources [52]. Block ciphers are used in GCM with 128-bit block size. The coefficient modulus size used [40, 21, 21, 21, 21, 21, 21, 40] to use the vector of small modulus objects, which must be distinct prime numbers and at most 60 bits in size [53]. Our proposed model provided result as follows:



```
Evaluated test_set of 5038 entries in 38 seconds
Accuracy: 4537/5038 = 0.9005557761016276
Difference between plain and encrypted accuracies: 0.0007939338684082031
Evaluated test_set of 5038 entries in 37 seconds
Accuracy: 4562/5038 = 0.9055180627233029
Difference between plain and encrypted accuracies: 0.0
```

Show that testing on encrypted data has a minimal effect on the outcome by comparing accuracy on encrypted data to the accuracy attained up to the mark. Testing on encrypted test set has no discernible effect on accuracy. The evaluation worked much better when it was encrypted in a few instances.

The model encryption time is 0 second
Test set encryption time 21 seconds and evaluation time 37 seconds over 5038 test cases. The accuracy difference in 0.

5.2 Discussion

It is important to know about the security of data. In our proposed model we used asymmetric encryption where serialized context size 86.88 MB (according to tenSEAL). It creates a serialized ciphertext like 427.15 KB and the encryption increased 48.52% which indicates the encryption is going towards non decrypted form showed in Table 1.

Table 5.1: Context & Ciphertext Serialized Size

Encryption Type	Context Serialized Size	Ciphertext Serialized Size	Encryption increase ratio
Symmetric	86.87 MB	-	-
Asymmetric	86.88 MB	427.15 KB	48.52

According to our dataset and proposed algorithm we use 2**21 precision to encrypt it. If we want to decrypt it make the ciphertext hard to break without the decryption key in Table 5.2.

Table 5.2: Cipher Text Precision Impact

Value Range	Precision	Operation	Status
$2^{-1} - 2^{0} *$	2**21	encrypt	decryption 2 ** -11
$2^{-1} - 2^{0} *$	2**21	sum	decryption 2 ** -10
$2^{21} - 2^{22}$	2**40	encrypt	decryption 2 ** -31
$2^{21} - 2^{22}$	2**40	sum	decryption 2 ** -30

Compared to other proposed model our model preserves the privacy of data as well as ensure data security keeping the accuracy nearabout same according to results. Deep learning, other supervised and unsupervised machine learning algorithm in some cases provides better accuracy but that’s not secure.

Table 5.3: Comparison with Other Model

Author & year	Method	Accuracy (%)	Valued
[51]	Deep learning	87.20	Accuracy
[54]	Supervised ML	97.40	Accuracy
[55]	ML	87.05	Accuracy
	Our proposed model	90.05	Accuracy, Security & Privacy

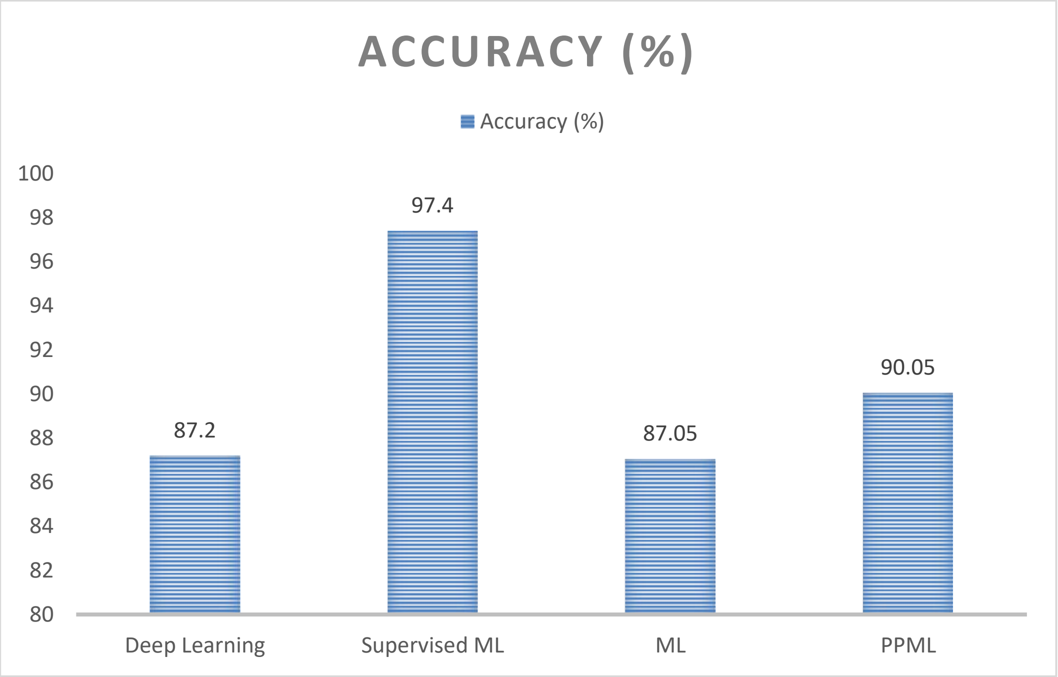


Figure 5.1: Evaluation Outcome of Proposed System

Because the traditional Logistic Regression algorithm contains a number of homomorphic Encryption-incompatible steps and operations, this research identified an FHE-enhanced variant of the algorithm. Logistic regression on encrypted files will become increasingly important to protect both model knowledge and data against multiple attackers. The homomorphic encryption (HE) for such arithmetic of approximation numbers approach enables efficient arithmetic assessments of encrypted real figure data and encourages the development of privacy-preserving machine learning approaches and algorithms.

Fully homomorphic encryption utilized in the deductive phase on the encrypted form enables real-time predictions. To forecast encrypted data, a logistic regression with fully homomorphic encryption is used. Therefore, our findings support the development of new functionalities as well as logistic regression that protects privacy. The experimental results showed that our strategy was more effective at classifying data from the NSL-KDD dataset and the bulk of real datasets. Additionally, the predicted time for iteration was effective because our method does not require any non-polynomial approximation operations during the training phase. The logistic regression procedure can be used with FHE from training to inference, according to research.

CHAPTER 6
CONCLUSION

6.1 Summary

Traditional machine learning doesn’t provide safety and security of data driven applications. This study offered a way combining machine learning and fully homomorphic encryption technique to effectively address the difficulty of protecting privacy in data-driven applications. FHE enables computation on encrypted files without decoding for any purpose other than the final result. As BFV is better for arithmetic operation on integers, CKKS is preferable for arithmetic on real number. Our proposed model built in real numbers which is the reason of using this for approximate but close result. Initially, in NSL-KDD dataset we perform usual logistic regression after handling the outlier, we split dataset in two different set train and test. Then we scalarization to reduce multi object to single object optimization. Finally, we applied the logistic regression model to get the accuracy.

After that, we enhanced our encrypted model according to the algorithm provided on the methodology and get the expected model accuracy. Then we showed up the total accuracy on test set compared to others. Our model is providing privacy, security and integrity of the data which is not provided by others. The total encryption time is pretty much reasonable and breaking the security key is harder than normal machine learning model. The main importance to this research is follows:

- For identifying malicious attacks, machine learning-based techniques have become a key trend for any dataset.
- FHE enables calculations on encrypted files without decoding for any purpose other than the final result.
- Using a machine learning technique is an effective way to address the issue of privacy protection in data-driven applications.

6.2 Conclusion

Machine learning (ML) categorization has several applications. In the big data era, a user may decide to assign classification tasks to others in order to reduce their own computing workload. In the interim, a company might desire to supply their customer a classification model service. However, certain tasks, such as network traffic, call for sensitive data that either side might not want to provide. Particularly, the increased demand for cybercrime services and the deployment of new technologies by hackers have resulted in even more advanced hazards. Organizations and businesses are becoming increasingly worried about security challenges due to the rise in these and other illicit activities. The rising costs of a personnel and environment for successful IT safety also pose a significant issue. Fully homomorphic encryption (FHE) does not need any kind of decoding, enables secure calculations over encoded data. Without revealing any data, FHE enables categorization to be outsourced to the cloud. As a result, our research offered an efficient ML-based solution to the privacy protection issue.

NSL KDD dataset includes attacks like r2l, u2r, and DoS, is used in this study. Data pre-processing, one of the key steps in the data mining process, entails preparing and manipulating the initial dataset. A variety of processes, including feature reduction, data cleansing, and feature building, are comprised in data pre-processing. The processes of feature selection and extraction are also in feature reduction work. Feature retrieval, construction, and selection are all distinct strategies in data pre-processing. Depending on the analysis of the circumstance, feature development and selection can be coupled as well, as can feature extraction and selection. The feature extraction method is used in this study to transform high-dimensional data into low-dimensional information.

This study suggests using an FHE technique to encrypt user data. In CKKS encoding system a barely unbreakable secret key and a public key are produced, which are also utilized in FHE. While for encoding a public key has been required and after encoding it can be released, the secret key is necessary for decoding and it should be kept top secret. Additionally, ML techniques like Logistic Regression (LR) are used in this study. Additionally, the encrypted information on the server is anticipated using the LR approach, and the predicted information is sent to the user. It is investigated how adding FHE to ML prediction results in a cloud server that provides safe predictions over encoded data while respecting data privacy of the customers. The user can decrease their own computing load and safely offshore their data prediction job in this way. To further assess the efficacy of the created ML methodology, the success possibility of the offered framework is verified and compared to other available pre implemented methods. The outcomes showed that given ML model had greater accuracy enhancement with privacy.

6.3 Future Work

When employing the homomorphic encryption algorithm, there are two extra factors to consider additional to the known computing costs. Lack of multi-party support in the majority of widely used homomorphic encryption methods. Although we assumed in our theoretical research that all data vendors would use the same key to encrypt data, in reality this is a very rare case when there are many sources of data and users too. In this case, it may or may not be possible to trust different data owners. The owners of the data do not interest to train a system using encrypted data collaboratively.

Practically, this problem can be solved by using a multi key homomorphic encryption algorithm because it ensures that the data will be encrypted with a variety of non-dependable keys and makes it possible to retrieve the training data from the multiple data holders, each of them have a different key.

There is a drawback is that extensive structural changes, unique client-server programs, and unique client-server programs are required for homomorphic encryption to work correctly. For scientific analysis, commercial companies are not allowed to utilize this technology without first obtaining user consent. This may result in higher overall expenditures and a continued push on substitute alternatives that are more effective at analytical activities for encrypted data and couldn’t care less about privacy issues.

Unlike more traditional encryption methods, fully homomorphic encryption does not ensure data integrity. Unlike traditional encryption methods, homomorphic encryption systems do not ensure data security. Corruption of data problem or integrity losses in the system of homomorphic encryption may restrict the accuracy and usefulness of any Big Data architecture for privacy preservation of data. Additionally, even though federated framework model proposed as a way to support training cooperatively, the concern of privacy has increased due to the potential for collaborative data to be leaked by combining many databases, especially in wide range of privacy exploits, such as the privacy risk from connectivity among databases. Serious privacy issue is a reason with the method of federated learning, it is necessary for researchers to look into specific security solutions for the aggregator of supervised learning to distinguish malicious participants relying on their updates. Attackers with bad intent may eventually get access to the computational infrastructure.

Completely homomorphic encryption is currently advantageous in many applications, however some of them cannot use it which is the reason of the limitations of this technique. The usage of homomorphic encryption, that are often scenario of client-server in which both the data and the algorithm need be kept under wraps, are also not actually covered by much current research. A choice is made in this case depending on the particular circumstances of the application.

Change the decryption process such that it generates no output but it is just the independent of the secret key and encryption randomness independent for a natural defense against cyber-attacks on the CKKS system, which effectively returns the interface of cipher text's encryption. Limiting the impact of the assaults outlined in the study, only straightforward remedies are required.

The results of the tests showed that our algorithm was more effective at datasets classifying data. Additionally, the average duration of iterative process was successful thanks to our technique throughout that period. Our training method can be combined with lowering FHE inference techniques for logistic regression. Thus, our discovery enables FHE usage throughout a logistic regression approach, including both inference and training.

REFERENCES

[1] E. Sarkar, E. Chielle, G. Gursoy, L. Chen, M. Gerstein, and M. Maniatakos, “Scalable privacy-preserving cancer type prediction with homomorphic encryption,” *ArXiv Prepr. ArXiv220405496*, 2022.

[2] M. Al-Qaraghuli, S. Ahmed, and M. Ilyas, “Encrypted vehicular communication using wireless controller area network,” *Iraqi J. Electr. Electron. Eng. Sceeer*, pp. 17–24, 2020.

[3] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” *Ad Hoc Netw.*, vol. 90, p. 101842, 2019.

[4] M. M. Arifeen, A. A. Mamun, M. S. Kaiser, and M. Mahmud, “Blockchain-enable Contact Tracing for Preserving User Privacy During COVID-19 Outbreak.” Preprints, Jul. 22, 2020. doi: 10.20944/preprints202007.0502.v1.

[5] A. D’Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, “A survey on big data for network traffic monitoring and analysis,” *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 3, pp. 800–813, 2019.

[6] P. Yang, N. Xiong, and J. Ren, “Data security and privacy protection for cloud storage: A survey,” *IEEE Access*, vol. 8, pp. 131723–131740, 2020.

[7] H. Teng *et al.*, “A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities,” *Future Gener. Comput. Syst.*, vol. 94, pp. 351–367, 2019.

[8] A. Gharaibeh *et al.*, “Smart cities: A survey on data management, security, and enabling technologies,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2456–2501, 2017.

[9] M. S. Inamdar and A. Tekeoglu, “Security analysis of open source network access control in virtual networks,” in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2018, pp. 475–480.

[10] P. J. Sun, “Privacy protection and data security in cloud computing: a survey, challenges, and solutions,” *IEEE Access*, vol. 7, pp. 147420–147452, 2019.

[11] D. P. Gadekar, N. P. Sable, and A. H. Raut, “Exploring Data Security Scheme into Cloud Using Encryption Algorithms,” *Int. J. Recent Technol. Eng. IJRTE Publ. Blue Eyes Intell. Eng. Sci. Publ. ISSN*, pp. 2277–3878.

[12] J. DesLauriers *et al.*, “Cloud apps to-go: cloud portability with TOSCA and MiCADO,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 19, p. e6093, 2021.

[13] F. Cai, N. Zhu, J. He, P. Mu, W. Li, and Y. Yu, “Survey of access control models and technologies for cloud computing,” *Clust. Comput.*, vol. 22, no. 3, pp. 6111–6122, 2019.

[14] L. Liu, Z. Cao, and C. Mao, “A note on one outsourcing scheme for big data access control in cloud,” *Int. J. Electron. Inf. Eng.*, vol. 9, no. 1, pp. 29–35, 2018.

[15] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, “Comprehensive study of symmetric key and asymmetric key encryption algorithms,” in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–7. doi: 10.1109/ICEngTechnol.2017.8308215.

[16] A. Viand, P. Jattke, and A. Hithnawi, “Sok: Fully homomorphic encryption compilers,” in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1092–1108.

[17] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, “nGraph-HE2: A high-throughput framework for neural network inference on encrypted data,” in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2019, pp. 45–56.

[18] E. J. Chou, A. Gururajan, K. Laine, N. K. Goel, A. Bertiger, and J. W. Stokes, “Privacy-preserving phishing web page classification via fully homomorphic encryption,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2792–2796.

[19] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *International conference on the theory and application of cryptology and information security*, 2017, pp. 409–437.

[20] J. Huang and D. Wu, “Cloud Storage Model Based on the BGV Fully Homomorphic Encryption in the Blockchain Environment,” *Secur. Commun. Netw.*, vol. 2022, 2022.

[21] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart, “Ring switching in BGV-style homomorphic encryption,” in *International Conference on Security and Cryptography for Networks*, 2012, pp. 19–37.

[22] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A survey on homomorphic encryption schemes: Theory and implementation,” *ACM Comput. Surv. Csur*, vol. 51, no. 4, pp. 1–35, 2018.

[23] S. Gorantala *et al.*, “A general purpose transpiler for fully homomorphic encryption,” *ArXiv Prepr. ArXiv210607893*, 2021.

[24] M. Albrecht *et al.*, “Homomorphic encryption standard,” in *Protecting Privacy through Homomorphic Encryption*, Springer, 2021, pp. 31–62.

[25] S. Kim *et al.*, “Bts: An accelerator for bootstrappable fully homomorphic encryption,” in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 711–725.

[26] M. Kim, X. Jiang, K. Lauter, E. Ismayilzada, and S. Shams, “Secure human action recognition by encrypted neural network inference,” *Nat. Commun.*, vol. 13, no. 1, pp. 1–13, 2022.

[27] L. Zhang, Z. Cai, and X. Wang, “Fakemask: A novel privacy preserving approach for smartphones,” *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 2, pp. 335–348, 2016.

[28] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[29] Y. Zhang, J. Li, D. Zheng, P. Li, and Y. Tian, “Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice,” *J. Netw. Comput. Appl.*, vol. 122, pp. 50–60, 2018.

[30] B. Wang, B. Li, and H. Li, “Knox: privacy-preserving auditing for shared data with large groups in the cloud,” in *International conference on applied cryptography and network security*, 2012, pp. 507–525.

[31] F. Zhao, C. Li, and C. F. Liu, “A cloud computing security solution based on fully homomorphic encryption,” in *16th international conference on advanced communication technology*, 2014, pp. 485–488.

[32] A. Viand, P. Jattke, M. Haller, and A. Hithnawi, “HECO: Automatic Code Optimizations for Efficient Fully Homomorphic Encryption.” arXiv, Feb. 04, 2022. doi: 10.48550/arXiv.2202.01649.

[33] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy,” in *International conference on machine learning*, 2016, pp. 201–210.

[34] E. Chou, J. Beal, D. Levy, S. Yeung, A. Haque, and L. Fei-Fei, “Faster cryptonets: Leveraging sparsity for real-world encrypted inference,” *ArXiv Prepr. ArXiv181109953*, 2018.

[35] T. van Elsloo, G. Patrini, and H. Ivey-Law, “SEALion: A framework for neural network inference on encrypted data,” *ArXiv Prepr. ArXiv190412840*, 2019.

[36] A. Al Badawi *et al.*, “Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus,” *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 3, pp. 1330–1343, 2020.

[37] E. Hesamifard, H. Takabi, and M. Ghasemi, “Cryptodl: Deep neural networks over encrypted data,” *ArXiv Prepr. ArXiv171105189*, 2017.

[38] R. Hamza *et al.*, “Towards Secure Big Data Analysis via Fully Homomorphic Encryption Algorithms,” *Entropy*, vol. 24, no. 4, Art. no. 4, Apr. 2022, doi: 10.3390/e24040519.

[39] “Data Security Using Homomorphic Encryption for Cloud Based Medical Record Management System – MJPS.” https://muthjps.mu.edu.iq/data-security-using-homomorphic-encryption-for-cloud-based-medical-record-management-system/ (accessed Jan. 03, 2023).

[40] S. Arita and S. Nakasato, “Fully homomorphic encryption for classification in machine learning,” in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–4.

[41] Y. Yasumura, Y. Ishimaki, and H. Yamana, “Secure Naïve Bayes Classification Protocol over Encrypted Data Using Fully Homomorphic Encryption,” in *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, New York, NY, USA, Feb. 2020, pp. 45–54. doi: 10.1145/3366030.3366056.

[42] S. Kim, M. Omori, T. Hayashi, T. Omori, L. Wang, and S. Ozawa, “Privacy-preserving naive bayes classification using fully homomorphic encryption,” in *International Conference on Neural Information Processing*, 2018, pp. 349–358.

[43] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, “Privacy-preserving outsourced classification in cloud computing,” *Clust. Comput.*, vol. 21, no. 1, pp. 277–286, 2018.

[44] M. Naehrig, K. Lauter, and V. Vaikuntanathan, “Can homomorphic encryption be practical?,” in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, New York, NY, USA, Oct. 2011, pp. 113–124. doi: 10.1145/2046660.2046682.

[45] D. Archer *et al.*, *APPLICATIONS OF HOMOMORPHIC ENCRYPTION*. 2017.

[46] “Build an Homomorphic Encryption Scheme from Scratch with Python,” *OpenMined Blog*, Apr. 27, 2020. https://blog.openmined.org/build-an-homomorphic-encryption-scheme-from-scratch-with-python/ (accessed Jan. 02, 2023).

[47] “OWASP Top Ten | OWASP Foundation.” https://owasp.org/www-project-top-ten/ (accessed Jan. 02, 2023).

[48] “1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory.” https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed Dec. 28, 2022).

[49] “KDD Cup 1999 Data.” https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data (accessed Dec. 28, 2022).

[50] H. Chen *et al.*, “Logistic regression over encrypted data from fully homomorphic encryption.” Accessed: Dec. 28, 2022. [Online]. Available: https://eprint.iacr.org/undefined/undefined

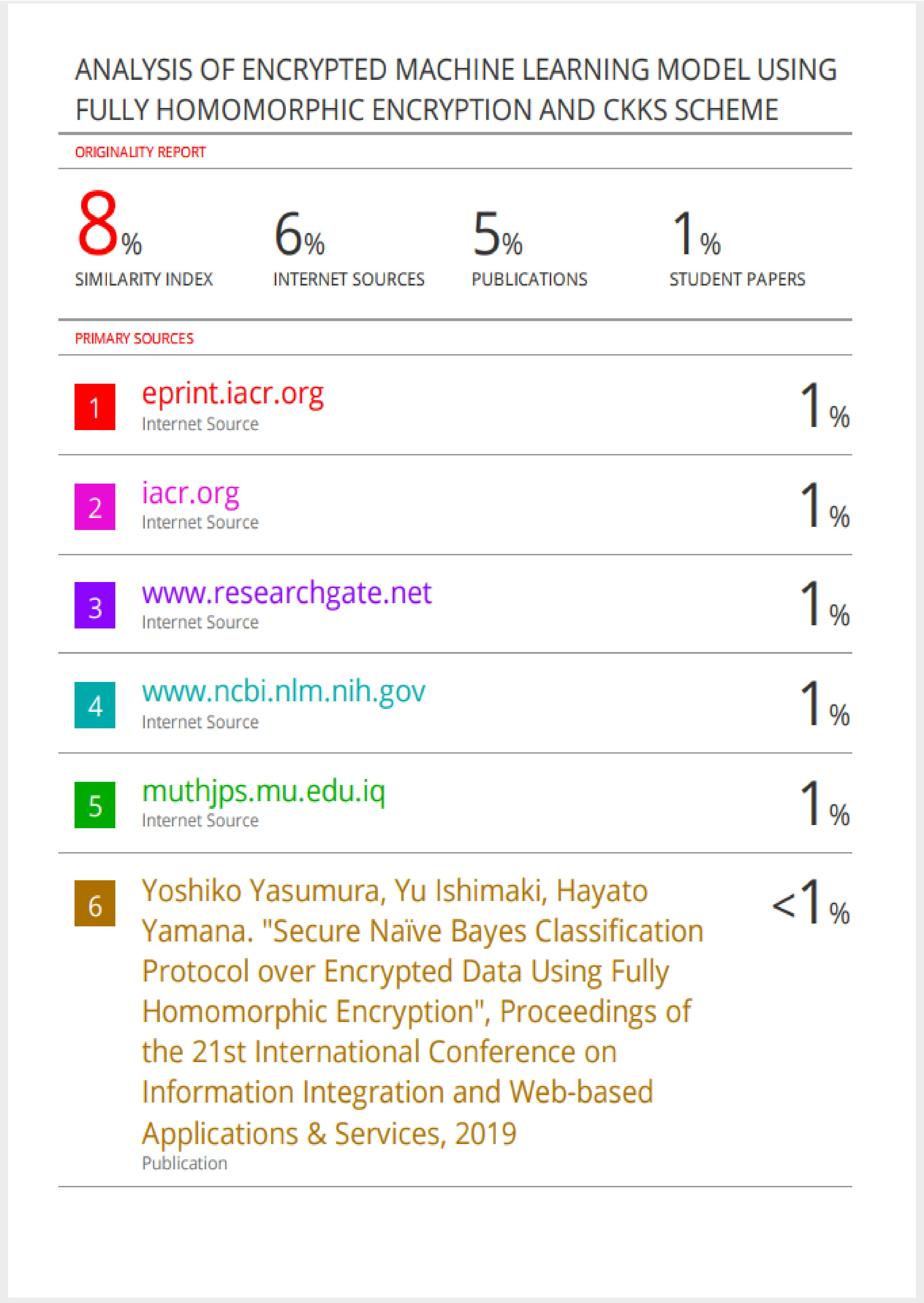
[51] S. Gurung, M. K. Ghose, and A. Subedi, “Deep learning approach on network intrusion detection system using NSL-KDD dataset,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 8–14, 2019.

[52] S. Lemsitzer, J. Wolkerstorfer, N. Felber, and M. Braendli, “Multi-gigabit GCM-AES Architecture Optimized for FPGAs,” in *Cryptographic Hardware and Embedded Systems - CHES 2007*, Berlin, Heidelberg, 2007, pp. 227–238. doi: 10.1007/978-3-540-74735-2_16.

[53] H. Chen, K. Han, Z. Huang, A. Jalali, and K. Laine, “Simple Encrypted Arithmetic Library v2.3.0”.

[54] R. Devi and M. Abualkibash, “Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 11, pp. 65–80, Jun. 2019, doi: 10.5121/ijcsit.2019.11306.

[55] I. Abrar, Z. Ayub, F. Masoodi, and A. M. Bamhdi, “A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, Sep. 2020, pp. 919–924. doi: 10.1109/ICOSEC49089.2020.9215232.



ANALYSIS OF ENCRYPTED MACHINE LEARNING MODEL USING FULLY HOMOMORPHIC ENCRYPTION AND CKKS SCHEME

ORIGINALITY REPORT

8%	6%	5%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprint.iacr.org Internet Source	1%
2	iacr.org Internet Source	1%
3	www.researchgate.net Internet Source	1%
4	www.ncbi.nlm.nih.gov Internet Source	1%
5	muthjps.mu.edu.iq Internet Source	1%
6	Yoshiko Yasumura, Yu Ishimaki, Hayato Yamana. "Secure Naïve Bayes Classification Protocol over Encrypted Data Using Fully Homomorphic Encryption", Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, 2019 Publication	<1%