Master thesis

Master's Programme in Network Forensics, 60 Credits

Malicious Activity Detection in Encrypted Network Traffic using A Fully Homomorphic Encryption Method

Thesis in digital networks, 15 credits.

Halmstad, 26.08.2022
Resmi Adiyodi Madhavan
Ann Zenna Sajan

HALMSTAD UNIVERSITY

Malicious Activity Detection in Encrypted Network Traffic using A Fully Homomorphic Encryption Method

Authors:

Resmi Adiyodi Madhavan

Ann Zenna Sajan

Supervisors:

Shooresh Sufiye –HMS Labs

Mohamed Eldefrawy – Halmstad University

Abstract

Everyone is in need for their own privacy and data protection, since encryption transmission was becoming common. Fully Homomorphic Encryption (FHE) has received increased attention because of its capability to execute calculations over the encoded domain. Through using FHE approach, model training can be properly outsourced. The goal of FHE is to enable computations on encrypted files without decoding aside from the end outcome. The CKKS scheme is used in FHE.Network threats are serious danger to credential information, which enable an unauthorised user to extract important and sensitive data by evaluating the information of computations done on raw data. Thus the study provided an efficient solution to the problem of privacy protection in data-driven applications using Machine Learning. The study used an encrypted NSL KDD dataset. Machine learning-based techniques have emerged as a significant trend for detecting malicious attack. Thus, Random Forest (RF) is proposed for the detection of malicious attacks on Homomorphic encrypted data in the cloud server. Logistic Regression (LR) machine learning model is used to predict encrypted data on cloud server. Regardless of the distributed setting, the technique may retain the accuracy and integrity of the previous methods to obtain the final results.

Keywords:

Malicious Activity Detection, Cloud Computing, Network Traffic, Fully Homomorphic Encryption (FHE), Machine Learning, Random Forest (RF), Logistic Regession (LR).

Preface

The thesis offered suggestions for securing the entire network using machine learning and an encryption mechanism. To begin with, they looked for proof of concept for a cutting-edge subdomain takeover detection strategy. So, with the guidance of our supervisors, we began searching for a hole in the currently used techniques for identifying subdomain attacks. We presented a novel high-level architecture as a remedy after reviewing the available market approaches and current research publications on this subject, and it was also prototyped as part of the thesis. Our solution has the benefit over other detection strategies currently in use because it offers data and network security for the entire network. Our research could serve as a starting point for future work by companies or cyber security experts to overcome subdomain attacks.

List of Abbreviations

HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
IDS	Intusion Detection System
RF	Random Forest
LR	Logistic Regression
CKKS	Cheon, Kim, Kim And Song
SHE	Slightly Homomorphic Encryption
PHE	Partially Homomorphic Encryption
HEANN	Homomorphic Encryption for Arithmetic of Approximate Numbers
BGV	Brakerski-Gentry-Vaikunathan
LWE	Learning With Errors
PSI	Private Set Intersection
SIMD	Single Instrcution Multiple Data
API	Application Programming Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ML	Machine Learning
DOS	Denial of Service
R2L	Remote-To-Local
U2R	User-To-Root
FHE-RN	Fhe Over Real Numbers
MSS	Managed Security Services
DMR-PDP	Dynamical Multi-Replica Provable Data Possession
CSP	Cloud Service Protocol
PIR	Private Information Retrieval
CIA	Confidentiality, Integrity, And Availability
SQL	Structure Query Language
IOT	Internet Of Things
KNN	K-Nearest Neighbour
SVM	Support Vecctor Machine
DDOS	Distributed Denial Of Service
TELNET	Terminal Network
FDECM	Frequency Division Embedded Component Method
SHES	Somewhat Homomorphic Encryption Scheme
OWASP	Open Web Application Security Project
DNN	Deep Neural Network
SYN	Synchronize Sequence Number
FSS	Feature Set Selection
AUC	Area Under the Curve

Summary

The study provided an efficient solution to the problem of privacy protection in data-driven applications using Machine Learning and Homomorphic Encrytion scheme. The goal of FHE is to enable computations on encrypted files without decoding aside from the end outcome. The CKKS scheme is used in FHE.The study used an encrypted NSL KDD dataset. Machine learning-based techniques have emerged as a significant trend for detecting malicious attack. Random Forest (RF) is proposed for the detection of malicious attacks on Homomorphic encrypted data in the cloud server. Logistic Regression (LR) machine learning model is used to predict encrypted data on cloud server. Regardless of the distributed setting, the technique may retain the accuracy and integrity of the previous methods to obtain the final results.

Keywords:

Malicious Activity Detection, Cloud Computing, Network Traffic, Fully Homomorphic Encryption (FHE), Machine Learning,Random Forest (RF), Logistic Regession (LR).

Table of contents

1. Introduction

Many individual’s sensitive information is processed over the online platform, making them susceptible to numerous attacks from both external and internal attackers. Malicious activities are increasing in number and variety as a result of the advancement in internet technologies. Data leaks and significant financial implications can result from cyber-attacks. For instance, owing to malware, Facebook incurred tens of billions of dollars in cyber security incidents. A growing number of security techniques (Aloqaily et al., 2019) evaluate the features of internet activity from many perspectives in order to create a reliable and flexible system for detecting intrusions in dealing with increasingly complicated and dynamic threats (Podgorelec et al., 2019). In order to determine whether such a computer system or otherwise computer network is under attack, the Intusion Detection System (IDS) gathers and examines data about various important locations in the system (M. Gao et al., 2020). It is crucial to people's lives to comprehend how Internet services were utilised as well as functioning. For this task, network traffic is critical.

Functionalities include giving users a glimpse of internet traffic, detecting abnormalities and undiscovered threats, and supplying data to frameworks in charge of utilization tracking as well as accounting. They gather the historic required data for network analysis as well as troubleshooting, assisting in the development of network technology and locating the source of issues. It is true to claim that internet traffic applications serve as a foundation for ensuring the consistency and smooth operation of the services that aid daily life. Analysing and interpreting internet traffic is a challenging endeavour. A few of the issues encountered by Internet Traffic applications are the enormous, speed of transmission system, traffic volumes, the organic growth of services and assaults, and the wide range of data source materials and techniques to obtain measurements. With additional observation aspects being available to investigators as the network's complexity rises, it may be possible to gather as well as analyse heterogeneous data. This development necessitates effective procedures for the online evaluation of massive flows of measurement techniques and it is challenging to develop customized solutions and deployed solutions. Additionally, as storage costs fall down, it does become possible to develop big historical databases for backward evaluation (D’Alconzo et al., 2019).

1.1 Purpose of the study

In order to enhance the interconnectivity of individuals, devices, as well as "things," a rising number of informational detecting devices are being linked to the Internet. In 2025, there would be 41.6 billion internets of things devices, as well as "things," producing 79.4 zettabytes (ZB), according to a recent forecast by Internet Data Centre (IDC)(Yang et al., 2020). Not just that, but individuals are still dedicated to enhancing the effectiveness of data collection from Internet-connected devices (H. Teng et al., 2019). Mostly on the provider of cloud services platforms, an unexpected amount of datasets is produced and stored. Most of the apps and solutions for smart urban will be housed in the cloud because of its high efficiency, scalability, and stable data canters. In order to host, develop, or launch various smart urban infrastructures and applications, citizens of smart cities and internet providers could depend on cloud services (Gharaibeh et al., 2017). Cloud technology is essential to expediting the evolution of the modern economic model and promotes the seamless connection of big data, the internet, artificial intelligence, as well as the overall economy. Collectively, cloud applications continue to be popular. Numerous issues in regard to the storing as well as processing of enormous and diverse data have emerged as a result of the overall increase in the number of pervasive observing and mobile cloud computing technologies (Inamdar & Tekeoglu, 2018). In this respect, cloud technology can be utilized to create a fresh generation of networked, scaled data mining methods with limitless possibilities to provide wiser information for decision-making in fundamental big data applications. Nevertheless, concerns over confidence issues in cloud technology due to the confidentiality of sensitive material have grown significantly, and this is seen as the main impediment to migrating analytic tools services into the cloud. Therefore, it is crucial to create safe data mining programs that can utilize cloud resources (Sun, 2019).

1.2 Cloud Computing

Cloud technology is an online platform that regularly maintains a laptop, a shared workspace, and resources for humanity. Individuals will use this technology with their laptops, workstations, Desktops, smartphones, etc. The combination of cloud internet, as well as mobile computation, is known as mobile cloud technology. Many customers have currently uploaded their data to the cloud. Therefore, security is a crucial factor in cloud technology that enables producing specific customer information that is placed in a secure way on the clouds. Information needs can't be fully met by other parties, so users' authentication becomes a mandated job. Customers can sign up as well as log in based on the space. Customers have the option to upload, share, and get the data starting with cloud space.

The two layers of security in order to better align concerns in the cloud are here admitted. Data or documents must be divided into far more over a few pieces for the first stage of security, after which it will be stored on various cloud web servers. The evaluation coupon is created for every file. Every chunk of a file that has been attacked in a successive level of security will be encoded before being stored in various locations. By receiving payment from the file holder, the general public has access to and may modify files in cloud storage. The purpose of filing is restricted to be adequate for self-examination. After that, the client may log in and forward the file. By using a passcode in, the user can locate then download the information. The plan can allow for the downloading of creative information from the cloud if the proof is successful as well as the cloud portions that were spilt are delayed (Gadekar et al., 2019).

1.3 Cloud Computing Functionalities

The term "cloud computing" denotes to internet computing that offers a distributed collection of information, resources, as well as software to individuals or devices on request and on a compensation basis. It relieves a user of worries about the provider's increasing technology knowledge. It enables end users and small businesses to utilize a range of computational services, including space, software, as well as the processing power offered by several other businesses.

Security, which contains data privacy concerns or data protection, is among the most important of the aforementioned challenges. Since the cloud vendor holds all of the data, there could be major issues with privacy protection if indeed the source abuses the info or the data. Additionally, any hacker or opponent with illegal access to cloud computing can mining the data as well as acquire a significant amount of sensitive data. Nowadays, a variety of data mining methods and algorithms are accessible that can be utilized effectively to mine important facts from enormous datasets by examining the behaviour and data in order. These tools for data mining are made available to consumers by several cloud services, and they can be utilized by an opponent (Mittal et al., 2014).

1.4 Cloud Storage

In essence, cloud technology is an online technology that enables the user to keep and exchange data. Limitless data storage capacity, easy, secure, and effective file availability, remote backups, as well as the low price of use, are all benefits of cloud storage. In terms of its practical use, the cloud infrastructure can be categorized into four groups: private cloud storage, public cloud storage, personal cloud storage,hybrid cloud storage, as well as communal cloud storage. In the cloud service, businesses outsource their storage of data needs to cloud collection rather than setting up and maintaining their own architectures and computers. Only users with permission could access data. Numerous small and medium-sized businesses are drawn to the public cloud because of its benefits, including extensibility, scalability, including cost benefits. Personalized clouds, often referred to as portable cloud storage, are similar to publicly available clouds but vary in that they offer public cloud-based storage services to customers (DesLauriers et al., 2021).

Businesses must set up private cloud technologies and hire qualified personnel to oversee and maintain equipment. This guarantees that the cloud infrastructure has stronger safety than the cloud platform while also ensuring that the organization itself has control over the information. However, the price goes up significantly. Large businesses with lots of pricey and sensitive files may benefit more from this storage approach. A cloud computing combines together private and public clouds, inheriting their respective benefits (Bai, 2022). In addition to storing other information in the cloud platform, businesses can store pricey and sensitive information in private clouds. This storing model's attractiveness keeps expanding. The cloud environment, a recent innovation in cloud services, is ideal for the financial and healthcare sectors. A society's corporate sector can get cloud services from communal clouds. Generally, these companies need to collaborate on certain initiatives or have similar thoughts. Public Cloud participants may work together to build the architecture and administer the servers, or they may contract these tasks out to outside parties. Because of the nature of cloud services, challenges with security as well as privacy of data are unavoidably created during this procedure.

1.5 Issues and Security of Cloud

The most pressing issue in cloud computing adoption today is security and confidentiality. The possibilities for penetration into the cloud environment were numerous and with larger rewards. The providers of cloud services that operate the services face privacy and security concerns. By establishing security processes and systems, the provider should make sure that the technology is protected and that their clients' applications and data were secured. Users' aversion to cloud computing is largely due to concerns about security. It is vital to assess the effectiveness of cloud providers' security methods since many providers were unwilling to disclose their technology to the public, and managing and creating a secure cloud environment is a difficult process. The privacy of cloud information is dependent on the use of adequate security procedures and methods . Storage with security were interrelated, therefore enhanced security necessitates an appropriate storage mechanism. Because fragmentation is important in effective storing, the type of fragmentation is determined by storage capacity as well as retrieving expense. Fragmentation is a process in which a document is separated into several uniform or fixed size parts known as divisions, with no consideration for the document's secrecy level. Security issues in cloud data can result in economic losses as well as a bad reputation if the platform is aimed toward a large audience and are the driving force behind the widespread acceptance of this new technology. Customers' data stored in the cloud shows the critical information. This is why having unauthorised person infringing on data is inappropriate. Users should exercise caution when keeping information in cloud services and every data should be encrypted before being transported to cloud services (Sajay et al., 2019).

While customers appreciate the simplicity of cloud storage, cloud storage providers also acquired their individual identity, locations, and important information for the company. Privacy security techniques are utilised to ensure that these data remain hidden from curious opponents and hostile cloud service providers' staff. Furthermore, the likelihood of users' devices being subjected to a side channel assault is very significant. In conclusion, the following problems posed to data security and privacy in cloud storage systems:

- Control over fine-grained data access.
- False integrity audit findings may be returned by malicious cloud service providers.
- Attack from the side channel.
- Malicious providers of cloud-based services do not cooperate with clients' instructions to erase data from the cloud completely.
- Privacy-preserving.

Despite the fact that cloud storage has been around for a while, it is still incredibly significant in the smart cities, Internet of Things, and digital economy. There are numerous study methods for privacy protection involving access restriction, encrypting, and security, but they are dispersed and usually unsystematic (Cai et al., 2019). As a result, it is vital to draw conclusions from recent study findings of various technologies that help with privacy and security in cloud technology. A most present privacy protection technique lacks a dynamic overcurrent protection also have limited scalability. Since cloud computing is networked as well as virtualized, users cannot immediately detect the memory address and partition of data, etc., thus data security gets critical. Data security in cloud technology is often maintained through data encryption or identity authentication (Liu et al., 2018). At the moment, common encryption techniques are divided into two types: symmetric encryption techniques and public key encryption algorithms. Among these, Fully Homomorphic Encryption is a popular symmetric encryption algorithm (L. Teng et al., 2020).

1.6 Fully Homomorphic Encryprion (FHE)

The need for network operators to ensure the confidentiality and safety of their client's data grows as services like cloud computing are continuing to gain widespread acceptance. By performing computing on encrypted files, homomorphic encryption (HE) permits the secure outsourcing of computing to the clouds (ciphertexts). The foundation of HE lies in noise-level encryption methods, where noise increases even if more computations are performed on the information. Practical uses cannot take advantage of HE because of the restricted set of procedures that may be applied to the data. For FHE compute components, hardware multipliers have been suggested in earlier publications (S. Kim et al., 2022). Even though only data encryption will ever be accessible for the server, fully homomorphic encryption (FHE) is a cryptographic method which offers excellent security guarantees. FHE has not yet been widely used for two primary reasons: FHE takes substantial cryptographic knowledge to design implementations, as well as FHE is currently too computationally costly to be practicable. Fortunately, thanks to considerable advancements in hardware support, effective algorithms, and low-level representations over through the last several years, FHE has significantly lessened its computing requirements.

A homomorphic encryption method is one that allows for certain functions to be carried out along inputs and outcomes that be either plaintext or cipher text. There are several types of homomorphic encryption, including completely homomorphic encryption (FHE), slightly homomorphic encryption(SHE), as well as partially homomorphic encryption (PHE). There are also some other Homomorphic encryption schemes like CKKS (Cheon, Kim, Kim and Song), BGV (Brakerski-

Gentry-Vaikunathan), HEANN (Homomorphic Encryption for Arithmetic of Approximate Numbers) etc. A "Learning with Errors" (LWE) method, which depends on cipher texts becoming mixed with noise, is used by the majority of contemporary FHE algorithms. The right message could be derived from cipher text as much as that the noise is little enough. The cipher text becomes noisier during homomorphic procedures. The quantity of noise is greatly increased when two cipher texts are multiplied, but this effect is barely noticeable throughout the addition. As a reason, when decryption becomes impracticable, even just a predetermined number of consecutive arithmetic operations (a variable known as multiplicative depths) can be carried out. This restriction can be overcome by utilizing bootstrapping, a method that homomorphically evaluates the decryption circuits (i.e., the mechanism that turns cipher text in to plaintext) with such an encryption key as intake. This lowers the noise exposure of cipher text to a predetermined lower level (Gorantala et al., 2021).

Homomorphic Encryption (FHE), an important technological advancement for interests by ensuring, has recently developed to the point that it is useful in practical applications (Viand et al., 2021). FHE has transformed from a futuristic idea to a practical reality during the past ten years, including close to a five-order improvement in performance. For instance, it now takes below 15 seconds to multiply two cipher texts instead of 30 minutes. Although, this is approximately seven orders of magnitude slower than an IMUL operation on a contemporary CPU, but it is fast enough for allowingmultiple implementations realistic. In order to further increase speed, contemporary systems added SIMD-style parallelism, encoding hundreds of plaintext variables in a single cipher text. Those developments have opened the door for numerous applications in various sectors. This included mobile apps, wherein FHE has been employed to encryption the underside of a fitness tracker while maintaining privacy. The interface is still real-time. FHE has been applied in the healthcare profession to allow for the confidentiality of genome sequences (M. Kim et al., 2021) applications over significant datasets. FHE has typically been employed to solve a variety of really well challenges, such as Private Set Intersection (PSI) (Chen, Huang, et al., 2018), outpacing earlier approaches by a factor of 2. Machine learning challenges have included everything from linear as well as regression analysis to Encrypted Neural Network prediction, that could be utilized to operate private information ML-asa-Service solutions like protected phishing email identification (Chou et al., 2020). FHE has indeed been employed for these and other activities in the field.

As a result, FHE-based secured compute methods are becoming more popular. Around 2025, "a minimum 20% of firms will possess a capacity for initiatives that integrate completely homomorphic encryption," predicts Gartner. Developing safe and reliable FHE-based systems is still a difficult undertaking, regardless of these new advancements. This is mostly related to how the computational model used by FHE differs from the conventional programming paradigm and therefore presents particular difficulties. For instance, almost all commonly used programming paradigms, such as loop and also if expressions, relied on information splitting. FHE calculations, on the other side, must by necessity be information in order to adhere to the security obligations. Major engineering hurdles are also posed by operating with FHE in practice. The performance compromises provided by various techniques vary, and also the best options strongly rely on the purpose. Researchers have seen an increase in the development of technologies that aim to enhance availability and lower barriers to participation in this discipline in order to overcome a few of the technological issues in this area. Without development tools, it is difficult to realize FHE-based computation by manually incorporating the necessary mathematical functions or utilizing an arbitrarily defined arithmetic framework, requiring extensive experience including both high-performance numerical integration as well as cryptography.

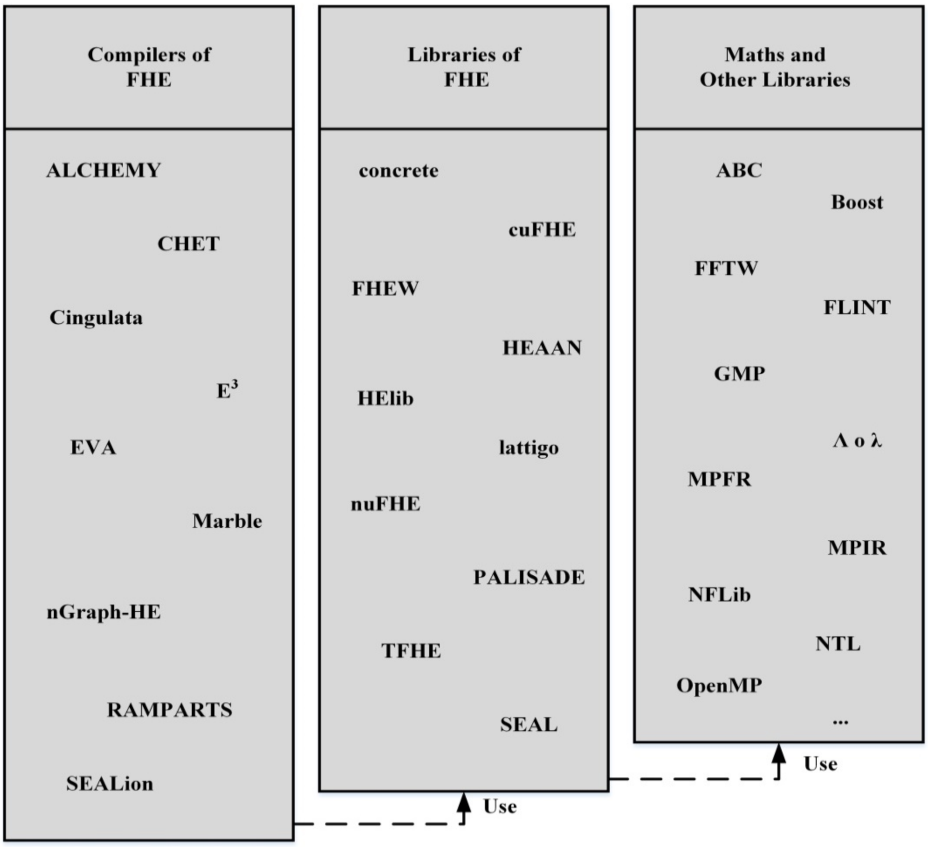


Figure1: Tools of FHE

These APIs disclose at minimum homomorphic addition operation in combination with key creation, decryption, as well as encryption. In actuality, though, libraries APIs frequently provide dozens of extra features for managing and manipulating ciphertext. Some APIs vary dramatically not only in terms of the process but also theoretically because strategies differ in terms of characteristics. In parallel with efforts to standardize FHE method APIs, coverings for established frameworks are already being developed in an effort to take the initial steps towards compatibility (Albrecht et al., 2021). However, working very closely with frameworks is still usually necessary to get competitive results. FHE libraries significantly increase the efficiency of creating FHE-based systems, but because they are still very minimal cryptographic libraries, they also call for a high level of knowledge and comprehension of the actual scheme.

In order to convert ordinary programs into FHE-based representations, higher-level tools often referred to as FHE compilers—have been recently developed. By continuing to enhance and gradually provided sophisticated improvements that were previously exclusive to professionals, these technologies are designed to make FHE approachable to non-experts. FHE libraries are typically used by processors to implement essential encryption, decryption, and homomorphic computations. In contrast, FHE libraries commonly use pre-existing libraries for parallel processing, quick numerical calculations, and other functions that aren't FHE-specific. Several FHE tools are shown in Figure 1 along with their placement inside this dependence structure. These technologies have greatly facilitated the process of creating FHE implementations, even though more work still must be done. For instance, technologies have shown to be accessible and usable in the machine learning field while simultaneously delivering cutting-edge efficiency because of automated optimizations that vastly surpass earlier hand-crafted approaches by specialists. Enabling private reasoning, neural networks are transformed into effective FHE-based systems using the nGraph-HE architecture (Boemer et al., 2019), for instance. Here, almost all FHE-related details have been typically extracted, and using TensorFlow in this method provides a largely similar experience for users.

1.7 Random Forest

An Ensemble Supervised Machine Learning method that has lately gained popularity is Random Forest. Inside the sector of data mining, machine learning algorithms are used. Descriptive data mining, as well as predictive data mining, are two broad categories of data mining. More emphasis is placed on explaining the data, classifying them, as well as evaluating them in descriptive analysis mining. Predictive data mining examines historical data to identify patterns and draw inferences about the future. The development of traditional statistical predictive models is the foundation of prediction data mining. The decision tree is the foundational classifier in Random Forest. Random Forest builds many decision trees; randomness occurs in two ways: first, randomly chosen gathering for bootstrap samples, as in bagging, as well as second, random selection of input characteristics for constructing separate base decision trees. The amplitude of individual decision trees as well as the connection between base trees is important factors in determining Random Forest classifier generalisation error (Kulkarni & Sinha, 2012).

For overuse, anomalous, and hybridization discovery, this research develops new methodological approaches that leverage the technology for network attack detection known as random forests. Among the most efficient data mining approaches is the random forests methodology, which uses ensemble classifications and extrapolation methodology. Figure 2 shows the structure of Random Forest. Adequate or even more acceptable big data architecture is required to handle large-scale information processing and storage. Today's world is data-centric, therefore need to process and analyse big data has taken centre stage for any significant organization. Cloud computing technology is to make it simple and quick to transfer computer resources. Companies can easily connect to a cloud-based system and employ the resources thereon in accordance with the appropriate utilization guidelines. With the ability to quickly respond to variations in storage capacity as well as variability, cloud computing technology is a method for big data analysis employing a shared stream of configurable computing resources. As a result, effective cryptographic algorithms that are resistant to malicious activities must be developed, in addition to the ability to execute computations using encrypted information without decryption. And over past few decades, cloud computing-based applications have grown in popularity. A Big data cloud is analysed and usable data is extracted using a cloud-based computing platform (Das, 2018).

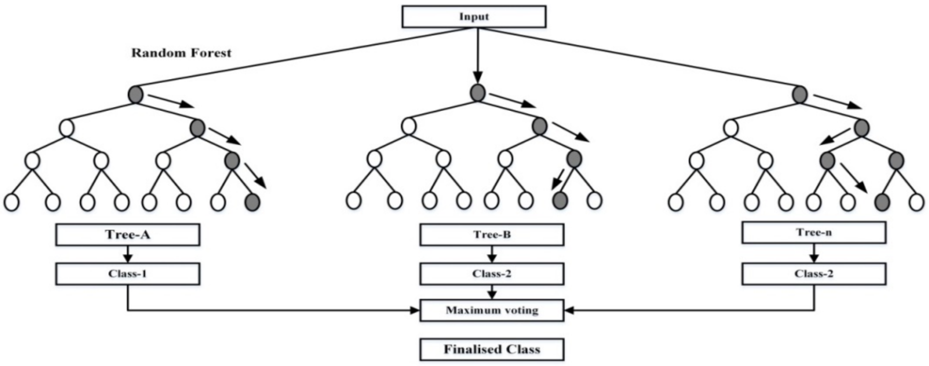


Figure 2: Structures of Random Forest

The privacy as well as securities of the informationretained in the cloud have been some of the key issues with cloud technology. Sending the data to the cloud securely is one option. Fully Homomorphic Encryption (FHE) is a method for enabling such calculations on encrypted data but still needs to perform usable computations on the encrypted files. While additional safe computation methods do exist, it should be noted that most of them call for information transmission between the various data providers. Given that FHE techniques utilize public key architectures, they are far more effective in situations when there are several data sets. Numerous systems have made substantial usage of the random forests technique. That has been used in predictions and probability evaluation, for example. The technique hasn't, therefore, been used for automated malware detection (Mangayarkarasi et al., 2020). The exploitation component of the system classifies intrusions using the random forests technique, while an anomaly component relies on the method's outlying detection technique. To provide a more precise and reliable forecast, random forest constructs many tree structures and combines them together.

1.8 Logistic Regression

Several real-world applications have made use of the popular classification technique known as logistic regression (LR) (Jiang et al., 2018).In terms of predictive accuracy, Logistic Regression is comparable with other algorithms and can be used to solve problems involving the challenge of estimating the probability that different events will occur. Data providers may desire to outsource a few of the labour-intensive processing that goes into training a logistic regression method in some real application settings because they have constrained resources for computers and storage. Recently, outsourcing data analysis has drawn a lot of attention since it permits data owners to train Logistic Regression models using the robust computing and storage capabilities of cloud-based service providers. Moreover, because of the improved sensitivity of training data, robust privacy protection must be implemented in order to carry out logistic regression evaluation securely and effectively without disclosing training data to unreliable cloud service providers.Data owners who wish to submit encrypted training data to service providers must first encrypt their training data. An encrypted training outcome from a logistic regression method that was trained using encrypted data for training is returned to the data owner by the service provider. The encrypted training outcome can be decrypted by the data owner for the unencrypted training result(Yu et al., 2022).

1.9 Problem Statement

Over the past few years, some of the most common and severe cyber-security assaults were frequently reported against systems in many industries. Security researchers anticipate more record-breaking year of network intrusions as well as data security dangers; businesses should stay up to date on the possible attacks to guarantee their adequate supporting are appropriate. The ninth assault type has the most frequency in security (Zhang et al., 2016). The adversary in a network attack needs be aware of active addresses, network structure, as well as facilities provided. Network scanners could detect unprotected TCP or UDP channels on a system, whereby shared services were associated with specific channels, and an intruder could transmit data to every channel. The attacker might determine whether a workstation is running Linux, Windows, or another system software. The information aids in the refinement of the attack as well as the search for vulnerabilities in specific information and infrastructure to get. Exhaust a server's capability by maintaining half-open interfaces, consuming traffic; an attacker must have a stronger relationship than the target to terminate each connection.

The network should be safeguarded against those assaults; effective malicious activity monitoring should be implemented before networking devices on the enterprise side. ML approaches are recently utilised to train detecting systems to collect malicious network threats (Alshammari & Aldribi, 2021). The primary idea behind ML-based malicious activity detection is to uncover patterns and generate a traffic detection using the data. To acquire enough, the study requires a genuine network traffic dataset including correct feature selection. As a result, a detection model using an ML framework is provided for detecting malicious traffic that relies on a database of network traffic properties to supply the detection mechanism. Machine learning-based solutions for identifying malicious attacks have emerged as a key trend. Thus, in this study, Random Forest (RF) is provided for detecting harmful attacks on Homomorphic encrypted data in the cloud server. An encrypted NSL-KDD dataset was utilized in this investigation. As a result, a safe RF approach to protect data privacy is created.

1.10 Research Questions

- What type of encryption method is used for encryption process?
- Why Fully Homomorphic Encryption (FHE) is utilized for encoding a credential data?
- How Machine Learning detects the malicious network threats?

1.11Objectives of the Study

Although encryption transmission has become prevalent in the online world, everyone requires their own data protection and privacy. Nonetheless, traffic encryption is used to protect against harmful and unauthorised data transmitted by intruders. Because of its capacity to run calculations over the encoded domain, Fully Homomorphic Encryption (FHE) has recently gained considerable attention due to the increasing security requirements in data mining. Model training can be adequately outsourced to unreliable but effective public cloud computing systems utilises the FHE technique. The study's goal is as follows:

- To publish the encrypted data content using FHE and process the encrypted results
- To improve the secure and efficient content processing while preserving the users privacy
- To enable computations on encrypted files without decoding aside from the end outcome
- To offer an effective solution for the issue of privacy protection in data-driven applications using Machine Learning
- To analyze the model performance, thus 4 features such as DoS, R2L, U2R and probe are selected from the NSL KDD dataset.
- To decrypt the results with secret key and send to the clients in an effective manner

1.12 Key Contribution

- FHE is used to encrypt data prior to it being sent via cloud computing.
- RF is utilized for the detection of malicious attacks on encrypted data in the cloud server and LR is used for predicting the encrypted information.
- Analyse the encrypted NSLKDD data set, which is observed by machine learning models.

1.13 Research Significance

- An efficient solution to the problem of privacy protection in data-driven applications is to use a Machine Learning Technique.
- The goal of FHE is to enable computations on encrypted files without decoding aside from the end outcome.

- Machine learning-based techniques have emerged as a significant trend for detecting malicious attack.

1.14 Organization of the thesis

Chapter 1: Introduction: Purpose of the study– Cloud Computing– Cloud Computing Functionalities– Cloud Storage– Issues and security of Cloud– Fully Homomorphic Encryption– Random Forest– Problem Statement– Research Question– Objectives of the study– Key Contribution– Research Significance–Organization of the thesis

Chapter 2: Literature Review: Homomorphic Encryption Schemes– HE applications and advances– Machine Learning (ML)-based classification protocol– ML for Malware detection–

Chapter 3: Methodology: Overview of the proposed system – Homomorphic Encryption – Datasets – Pre-processing of Data – Feature selection – Classification – CKKS scheme for encryption – Predicted Encrypted Data by Logistic Regression

Chapter 4: Results and Discussions: Influence of secret key - Influence of ‘n’ length - Input data length - FHE with and without Network Analyser – Accuracy evaluation of Random Forest classifier - Evaluation outcomes of proposed method

Chapter 5: Discussion

Chapter 6: Conclusion

2. Literature Review

To find a gap in the detection of malevolentbehaviours in the encrypted network traffic, a literature review was done. The review discussed the developed methods, effectiveness and its drawbacks. Based on the literature, methods employed for Homomorphic Encryption (HE) and Machine Learning (ML) in malicious activity detection has taken. A novel paradigm was suggested afterwards the problem statements were resolved.

2.1 Homomorphic Encryption Schemes

Performers can compute accurate functions on encoded values using homomorphic encryption without knowing the contents of the values. This option arises from the definition of the encrypting circuit as just a team homomorphism, which protects group processes. Group homomorphism enables the same computation to be done on encoded or plain values. This adaptability addresses security concerns in numerous implementations that assign sensitive operations to unreliable parties. In this section, the HE schemes characteristics and mechanisms are overviewed.

For both service providers and end users, the tendency to outsource information management and processing presents serious privacy-related risks. The research group has taken notice of these issues, and numerous strategies have been put forth to defend against hostile parties by supplying secure communication methods. However, the majority of the suggested methods call for the participation of a third-party, which in and of itself raises security issues. By employing a novel strategy that relies on data being organised, controlled, and kept in encoded form somewhere at remote systems, these security flaws can be prevented. The encryption cryptosystem should enable both multiplication and addition over encoded data in order to implement such a strategy.Perfectly symmetrical homomorphic cryptosystems enable homomorphic algorithms that enable encoded blind data processing without the requirement to decode it. In order to secure implementations, services, and routing mechanisms, Youssef Gahi et al. (Gahi et al., 2012)have developed a Fully Homomorphic Encryption (FHE) techniques. The researcher also created a number of circuits that enable blind data management and processing, preventing malevolent parties from accessing sensitive information. As they would be improved to enable additional calculations, the effectiveness of these methods continues to fall within the purview of the study interest. Lastly, the limited number of processes actually hurts the selected FHE, necessitating a re-encryption process to accommodate an unlimited number of processes.Third-party information security issues are becoming more and more important as cloud computing continues to grow. However, typical cryptogram systems struggle to retrieve encrypted data effectively and perform other functions. In order to assure privacy preservation in cloud storage, Jian Li et al. (J. Li et al., 2012)have introduced a pragmatic simple FHE method that employs only elementary mathematical operations and is inferred from the Gentry cryptosystem. Since encoded data could be functioned explicitly without compromising the encryption system's confidentiality, this scheme effectively addresses the requirement for ciphertext collection as well as other computation on unreliable servers. The suggested plan is considerably more workable and safe. Furthermore, the effectiveness and viability of the suggested SDC scheme were shown by the effectiveness assessment and security assessment. However, the encoded data efficiency is less, which requires more improvement. Feng Zhao et al. (Zhao et al., 2014)have presented a new type of data security resolutions for the cloud computing system's insecurity and have built the instances for this implementation for addressing the data security issue in cloud computing systems. This innovative security system is completely capable of handling and retrieving encoded data, successfully paving the way for broad applicability, secure data transport, and cloud computing retention. This system guarantees the security for transfering the dataamong the cloud and the user. Additionally, their information remains still secure in the cloud retention. Clients and the third-party provider can easily search for dates to rid of. However, the computational problem is high in this scheme and lack in users.

The widespread adoption of cloud computing has lead to a sharp progression in sharing and utilisation of data among many parties. Users' control lack over cloud systems was the fundamental obstacle preventing widespread adoption of cloud computing, which makes users' security and privacy concerns a significant challenge. Because of the users' complete control over their data, an appropriate FHE method makes sense as a solution for securing information throughout the data consumption lifespan in the cloud service. However, because to either a poor accuracy rates or unbearable latency, there hasn't been a successful FHE scheme built yet to satisfy practical objectives. Focusing on this issue, Keke Gai et al. (Gai et al., 2017)have suggested FHE over Real Numbers (FHE-RN), an improved FHE system optimised for handling real numbers. Experimental findings have demonstrated the excellent accuracy and effectiveness of this technique. Further, the execution time of this model is high. A strategy to create a HE scheme for approximate arithmetic has been proposed by Jung Hee Cheon et al(Cheon et al., 2017),which supports roughly adding and multiplying encrypted messages, as well as a novel rescaling method for controlling the size of plaintext. By doing this, a ciphertext gets reduced to a lower modulus, which rounds the plaintext. A fresh batching method for

RLWE-based architecture was also suggested. A complex number's message vector was transferred to a plaintext polynomial by using a complicated canonical embedding map, referred to as an isometric ring-homomorphism. This polynomial seems to be a member of a cyclotomic ring with characteristic zero. The outcome showed that, as a result of the rescaling technique, the ciphertext modulus bit size expands proportionally with the level of the circuit was examined.However, the level of a circuit limited the amount of precision loss that might occur during assessment.

Privacy is among the most important factors in relation to the offshoring of security problems. In order to deliver an efficient and dependable service, security surveillance and security operations in general need accessibility to as extensive data as feasible. It refers to the commonly-known contradiction between privacy and security, which was especially obvious in security monitoring technologies. A Managed Security Services (MSS) concept was examined by Luigi Coppolino et al. (Sgaglione et al., 2019)in attempt to offer a privacy-preserving method that would permit security management without infringing on privacy standards. The fundamental concept depends on the homomorphic encryption application. By utilising homomorphic encryption, MSS providers and cloud computing can process encoded data in a variety of ways without ever getting insight to its decryption.With this approach, data was kept private and safe not only while storage and transfer but also while being processed. Moreover, since the IDS lack the secret key, it is impossible to decode the ciphertext that results from the assessment on homomorphic domains.Data owners can offshore their information by securely preserving it in the internet, and they can take use of high-quality on-demand operations from a reconfigurable computational resources' shared pool. The outsourcing information may be at threat, nevertheless, as the cloud service could no long be completely recognized, as the data providers and the virtual servers mayn't reside in the identical trustable domain. Consequently, in a situation like this, data integrity, accessibility, and privacy are crucial. To preserve data secrecy, the data owner encodes the information before retrieving it in the internet. Prior multi copy verification systems both concentrated on static content or involved significant updating costs in a volatile file context to address such issues.The dynamical multi-replica provable data possession method (DMR-PDP) that (Mukundan et al., 2014)have suggested prohibits the CSP from scamming by retaining fewer duplicates than charged for or interfering with information while guaranteeing data secrecy. On replicas across the data centres, DMR-PDP also provides effective dynamic procedures including block change, inclusion, and removal. It was determined through security assessment and empirical outcomes that the suggested system is secure and outperforms certain other relevant concepts.These findings will offer the cloud several incentives to act appropriately, like conducting calculations in parallel, to give high performance. However, the employed data centres are less and the privacy and security concerns are not identified.

Big data's advent and the cloud computing implementations ongoing expansion raised significant concerns about privacy and security. Due to this, various scholars and cybersecurity professionals have started a project to broaden encryption of data in cloud computing implementations and big data frameworks. Confidentiality seems to be a highly difficult issue as several people utilise open cloud services. Users of public clouds who save their data there are always searching for solutions to the privacy problem. Homomorphic encryption was used to protect client data in the cloud, allowing for some recovery and manipulation operations without the necessity of proper decryption.A comprehensive review of studies that have been conducted in the area of homomorphic encryption has been offered by Mohamed Alloghani et al. (Alloghani et al., 2019). In order to evaluate studies gathered from diverse resources, this research uses the PRISMA criteria in addition to a few components of the Cochrane Quality Evaluation. It was clear from the publications the research evaluated that cloud security and big data had gotten the most emphasis. Although additional possible issues have been found by the thematic assessment, homomorphic encryption was recommended in the majority of studies. The explicit declaration of research aims, acknowledgment of the technique, and forms of financing employed in the study were three criteria that 38% of the papers did not match.Additionally, the paper included a comprehensive textual examination of the various homomorphic encryption methods, their uses, and potential future research directions. The preponderance of research papers explored the possible use and HE application as a response to the increasing needs of big data as well as the lack of privacy and security safeguards therein, according to the findings of the assessment through Cochrane program and PRISMA. However, the HE’s derivations and theoretical representation’s qualitative assessment takes more time. HE, one of numerous cryptographies, has drawn significant attention from academics for its unique capabilities. While HE allows calculations on encrypted data, conventional cryptography does not. As a consequence, the operations' outputs are also instantly encrypted. In the areas of safe multi-party calculation, cypher text searching, electronic voting, encrypted mail filtration, and mobile cypher, HE has a broad and promising application future.An improved ElGamal encryption has been introduced after the related technology has been described by Guangli Xiang et al. (Xiang et al., 2012). The updated encryption can improve security and satisfy both multiplication and cumulative homomorphism. Lastly, security analysis has been put to the test, and further study directions are suggested. The application potential for the algebra HE system is promising. Based on this, more work should be put into developing robust, effective security analyses and cryptosystems. However, the effectiveness of this algorithm is less that requires more enhancements.

Numerous uses for the Fully HE (FHE) method exist, particularly in cloud computing. FHE has lately been the subject of in-depth research; however most of the emphasis has focused on the strategy's implementation rather than its design or effectiveness. A new approach for retrieving encrypted data that combines FHE and ABE (Attribute based Encryption) has been presented by Jing-Li Han et al. (Han et al., 2012). Everyone can search the information using this technique, including those without access to the encoded data's secret key. The study concluded by discussing the use of FHE on computes outsourcing and presenting two distinct systems that fulfilled the various requirements.This plan also preserves the confidentiality of the consumer's inputs and outcomes. Efficiency did increase to some extent, but the researchers warn that there are still no completely workable solutions.Notwithstanding all of its advantages, maintaining information privacy and secrecy is a difficult task for cloud computing. People are hesitant to use it because of the various threats and security flaws it has experienced. With a main attention on the cloud computing safety area, its key risks, and the defence against each of them, Saja and Dujan (Mohammed & Taha, 2021)have supplied thorough research on the cloud computing principles. Data confidentiality and protection in the cloud context are also covered, and HE has been mentioned as a well-liked method for protecting the sensitive information privacy in several cloud computing implementations.For people new to the area who aren't yet completely prepared to comprehend the intricate and difficult technical components of cloud computing, as well as for practitioners as well as researchers currently engaged in the field, this study sought to provide a sufficient introduction of cloud computing safety. However, the in-depth research was not conducted in this article.

Data security was a major obstacle to cloud computing adoption. Data security for retention and transfer is provided by conventional standard encryption techniques. However, in the processing stage, decrypting the data is necessary before conducting actions on it. Data is currently accessible to cloud service providers. Therefore, using typical encryption techniques alone will not totally protect data. In order to secure information while it is processed, Kamal Kumar Chauhan et al. (Chauhan et al., 2015) have explored HE approaches and their usage in cloud computing. HE enables direct operations on encrypted information without the need for decryption. The result indicated that the HE schemesprevent the data security problems in cloud computing. However, both partial as well as fully HE schemes aren’t feasible and it is difficult to implement. As usage of cloud storage soars, there are numerous security issues. The security of user information stored in the current cloud has to be improved owing to hacking and malware attacks. HE is a newly developed cryptographic method that permits cloud-based data change without sending modified information back to the computing node. Jithin Raj et al. (Raj et al., 2015)explored HE with privacy barrier using identity-based identification and hypervisor-based invasion recognition for increased safety in the cloud, allowing the manipulation of encrypted information within the cloud by authorised users. The benefit of the suggested method is that it effectively prevents any hostile nodes from gaining unauthorised access to cloud networks and internet services.When compared to the current authentication methods, identity-based anonymity identification demonstrated a generally higher level.But, the processing time of this approach is higher.

2.2. HE applications and advances

The majority of the papers that were evaluated explored HE in relation to cloud computing and big data, and as a result, the majority of developments and advancements are oriented in this direction. The research examined the use of a revolutionary technique to drone safety, and Cheon and Kim (Jung Hee Cheon & Jinsu Kim, 2015)provide among the most original uses of HE. To eliminate monitoring and forging threats, the authors specifically developed a linearly homomorphic verified encryption architecture whichfacilitates and safeguards ground-regulated multirotor drones(Acar et al., 2019).Additionally, Cheon and Kim (Jung Hee Cheon & Jinsu Kim, 2015)developed a combined public-key encrypting approach that lessens the amount of storage needed for somewhat HE (SHE)(Yasuda et al., 2014).

The Somewhat Homomorphic Encryption Scheme (SHES) technique seeks to grant a small lot of purposes over the information that is being protected. This criterion is mostly predicated on the requirement that the noise of the scheme, commonly known as the ciphertext ci mod sk, must be less than sk/2. When the noise value surpasses the sk/2 threshold after a finite number of processes, which doubles after every addition as well as squares after every multiplier, the accuracy of the decryption is no longer offered. Gentry have proposed a scaling strategy to eliminate the noise as well as deliver a larger number of logical computations.

With an emphasis on complex mathematical circles, the suggested approach integrates the computational capabilities and methods employed in FHE along with the additive functions(Barkataki & Zeineddine, 2015). A closer examination of the past reveals that exporting of storage and processing is among the main applications of HE. Moreover, HE may be used by businesses that generate big data and contract out its processing and storage. In this regard, HE is valuable since it enables businesses to outsource certain functions in a safe manner without disclosing any potentially sensitive data(Dugan & Zou, 2017).

The technology is therefore perfect for use in organisations of various sizes. For instance, a small business might use HE to protect its important data while transitioning to the online. In the lack of HE, such a situation would leave the organisation at a loss because it might be forced to release the private data without any way to secure it. HE can therefore be a workable solution. Private queries processing and Private Information Retrieval(PIR) seems to be another crucial implementation of HE. Users may utilise HE to allow private searches to a particular database or search engine; PIR seems to be an excellent instance of such an implementation.The PIR queries end-users typically want to retrieve a single information from a service that has a big database of entries. The operator can apply HE to the indices of the relevant record to obtain it in an encoded manner, even though obtaining this information from mass records may leave the user open to attacks. This maintains the confidentiality, integrity, and availability (CIA) triad while making PIR quick, safe, and secret. A similar implementation might also include larger, more intricate SQL database queries.

2.3. Machine Learning (ML)-based classification protocol

On the security categorization of encoded data employing FHE, numerous investigations have been done. Studies like (Bost et al., 2015; Khedr et al., 2016; Park et al., 2018; Wood et al., 2018)suggested secure categorization techniques for various circumstances and system models. However, these efforts fall short of our desired outcome, which is to entrust the categorization duties to a cloud server whereas maintaining the user's data confidentiality, cloud-derived results privacy, and categorization model confidentiality. There have also been research that concentrate on FHE-based categorization and training over encoded data.Studies like (Aslett et al., 2015; Chabanne et al., 2017; Chen, Gilad-Bachrach, et al., 2018)offered techniques where the classifier parameters are estimated whereas FHE could be employed for training. Even though categorization phase is a priority of these research as well, either real-world system architecture or technique is suggested by them.

In order to prevent the cloud and user from being able to decode each other's information, Kim et al. (S. Kim et al., 2018)and Li et al. (P. Li et al., 2018)established a method for secure categorization and training wherein they add a third entity who has the platform's private key and operates in charge of decrypting all ciphertext. However, in Kim et al. work's some data from the user and information about the categorization method are disclosed to a third entity via classification's intermediary findings. In Li et al. research's ciphertexts are re-encrypted via an intensive proxy re-encryption.Their gateway re-encryption is predicated on Gentry's (Gentry, 2009)bootstrapping method, which requires laborious calculation that may take several seconds to several minutes and may result in a bottleneck. In actuality, the method can be implemented without such a proxy re-encryption. Furthermore, despite its shortcomings, the suggested protocol by Li et al. doesn't specify any specific classification model utilising FHE since it is generic. This is a challenge when implementing FHE to any implementation.

For hyper plane options, decision trees, and naïve Bayes, Bost et al. (Bost et al., 2015)provided classification methods. Their methods perform categorization using FHE and two additional HE systems, which only permit addition procedures. In their research, it is presumed that the classification method was trained on plaintext before being encoded and kept on a cloud platform. The user receives the encoded model at categorization and calculates the categorization probabilities before communicating with the clouds to retrieve the categorization outcome. Nevertheless, their methodology has the drawback that the user calculates the categorization probabilities and connects with the cloud numerous times, leading to a significant computing strain at the user. Park et al. (Park et al., 2018) two different categorization methods for the naïve Bayes classifier, an information provider who trains the system on plaintext, a cloud platform, and numerous users are all present. The first method, which really is server-centric, outsources categorization to a cloud platform, but the prototype is saved in the internet as plaintext. When users outsource categorization to the cloud, they each have a unique set of encryption keys that are used to protect their data. The strategy cannot be saved in the internet as ciphertext since each user must encode the prototype in order to categorize the data.In the second method, which would be user-centric, the categorization is done at the user, but the categorization algorithm is encoded. The system only uses one set of keys, which the cloud manages. The data supplier encodes the prototype before sending it straight to the user rather than keeping it in the cloud. The categorization is carried out by the client, who then blinds the outcome before sending it to the internet for decoding. The client must deactivate the blinded outcome from the cloud after it has been decoded in order to retrieve the categorization outcome.In conclusion, there has been a trade-off among the two methods: the user's computing load and the secrecy of the categorization model.

2.4. ML for malware detection

ML is now quite useful in the categorization and grouping of malware. The classification of benign and malicious data has been the subject of extensive research in the literature. By taking into account additional attributes of both malicious software and benign instances, ML techniques give developers greater flexibility and room to build more effective models. Malicious URL identification, intrusion identification, and malware identification are only a few applications of ML in the realm of computer safety. A comparative study for categorising network traffic was suggested by Xianwei Gao et al. (X. Gao et al., 2019). ML classifiers were utilised for attack detection. KDD and CICIDS are the datasets that were obtained from the UCI library. When compared to other methods, they thought support vector machine (SVM) was among the best. For malware detection, Tongtong Su et al. (Su et al., 2020a). developed adaptive learning. They made use of the KDD database that was available online. R-forest, Dtree, and KNN classifiers seem to be the models in question. The authors of this study discovered that ensemble and Dtree algorithms produce good classification outcomes.There are numerous security concerns as a result of the Internet of Things' (IoT) rapid adoption. After the Mirai-based DDoS assault in 2016 that corrupted IoT gadgets, a tonne of new spyware that targets IoT devices and uses the Mirai core code has emerged. These malware have been harder to stop using current techniques like firewalls since they use software flaws to attack IoT gadgets rather than open TELNET connections (like Mirai). EDIMA, a distributed modular solution developed by Ayush Kumar and Teng Joon Lim(Kumar & Lim, 2019), can be utilized to identify IoT malicious network activity in large-scale systems during the monitoring phase as opposed to during an assault.For the traffic categorization of edge gadgets, EDIMA uses ML methods. It also has a policy component, a network traffic characteristic vector databases, and an additional packet sub-sampling subsystem. Additionally, testbed experiments were used to assess EDIMA's categorization effectiveness, and the results indicated higher performance.An idea for a video steganalysis botnet was put forth by Kwak et al.(Kwak & Cho, 2021). Additionally, they intend to deploy alternative video steganography technique predicated on the payload approach (DECM: Frequency Division Embedded Component mechanism) that could incorporate data with substantially more capabilities than current tools using two open programmes, Stegano, and VirtualDub. The interpretability and effectiveness of the DECM and proposed model were compared to those of the present picture steganography-based botnets and methodologies, and it was demonstrated that the suggested frameworkcould be applied in the Telegram SNS messenger.

3. Methodology

3.1 Overview of the proposed system

Complete homomorphic encryption is utilized for encrypting the user-side data before it is transferred to the cloud platform. The pre-processing is done after loading the NSL-KDD information. The feature selection process is carried out utilising the filter approach to select only the important features in order to further enhance the identification rate. Furthermore, the random forest (RF) classification is used to categorise the anomalies. It recognises all assault kinds during the prediction stage. The encrypted data on the cloud server is then predicted by the logistic regression model classifier. The suggested system's overall process is represented in Figure 3.

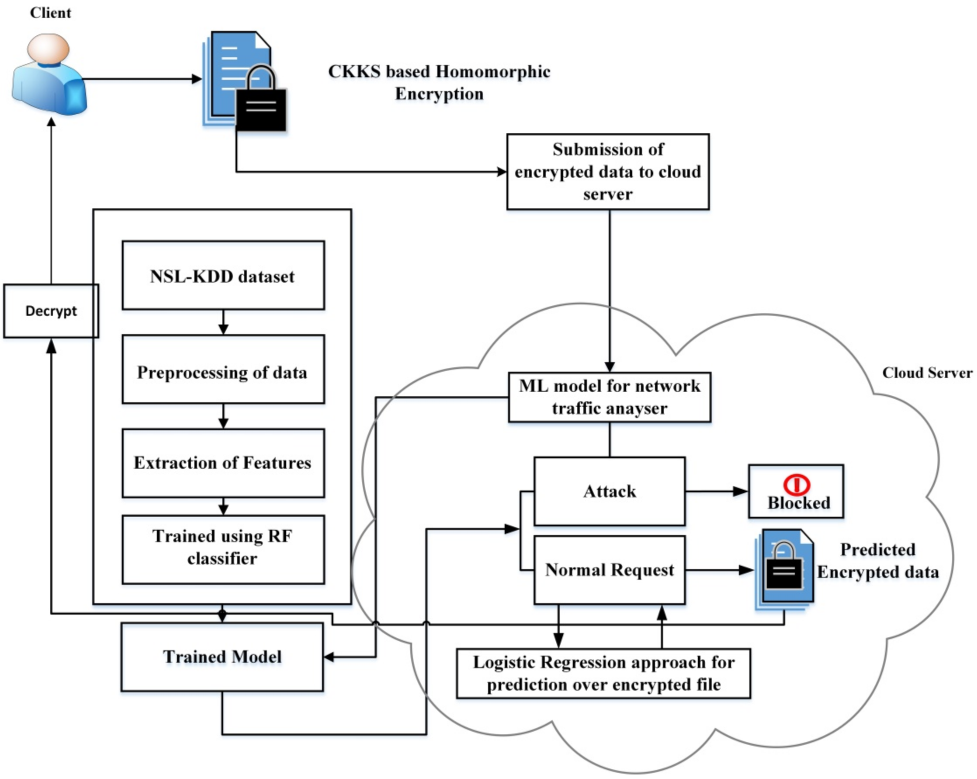

Client
CKKS based Homomorphic Encryption
Submission of encrypted data to cloud server
Decrypt
NSL-KDD dataset
Preprocessing of data
Extraction of Features
Trained using RF classifier
Trained Model
Cloud Server
ML model for network traffic anayser
Attack
Blocked
Normal Request
Predicted Encrypted data
Logistic Regression approach for prediction over encrypted file

Figure 3: Overall workflow of the proposed method

First, the user data is encrypted using Cheon-Kim-Kim-Song (CKKS) based on a fully homomorphic encryption framework. This encrypted data is transmitted over the network to the cloud server. For the training process, NSL-KDD datasets are used for pre-processing and specific feature extraction, which is trained for random forest classifiers. It is used to predict whether the data is affected by threats or not. If there are threats presented in the intercepted data it has been blocked directly otherwise it will pass to predict the encrypted data. Finally, the logistic regression model is used to predict the encrypted data and it will be notified to the user side (blocked/normal request).

3.2 Homomorphic Encryption

Homomorphic encryption is an encryption mechanismwhich enables one to manipulate cypher messages exclusively using information that is readily accessible to the public, and in especially not getting access to any private key. Because there is, generally, a correlation between the domain of the transmissions and the domain of the cypher texts, actions on the cypher texts somehow represent activities on the information being encrypted, this relation is the basis for the term "homomorphic" in cryptography.

For privacy reasons, an error term is often inserted and during encryption process in the majority of homomorphic encryption algorithms that are generally recognized. This is due to the fact that many encryption algorithms depend on how difficult it is to solve "background noise" issues, or difficulties where the connections are approximate but are disrupted by a little amount of mistake. By merging the lots of noise as well as the cipher texts when combining multiple ciphertexts using homomorphic processes, the size of the fault in the final encryption is increased. Validity is lost when the defect rises above a particular point, which prevents the decryption process from producing the desired outcome. If the encryption method can assess a predetermined number of homomorphic functions before the error becomes too great to establish the accuracy of the measurement, it is considered to be relatively homomorphic.

The selection of variables is one of the important aspects in providing the required an encryption algorithm. This is typically a situation where performance must be compromised for privacy, and it is one of the possible weaknesses for purportedly safe systems. For homomorphic encryption methods, this is also true, but in this situation, the issue of locating variables is considerably more crucial. The approach below, the noise must be maintained to ensure a successful decryption is defined by variables like the size of the factor or the difference of noise elements. These factors thereby limit the iteration of homomorphic processes which is needed to be performed. If a set of variables exist for any cumulative depths L determined a priori, so that the encryption method implemented with those variables can analyze any circuit with multiplication depth equal to L, then the encryption method is levelled homomorphic. The restriction on the total number of actions in this setting must be specified at processing times, or when establishing the variables. This indicates that rather than being a general tool for evaluating any value, the method will typically be adapted for a particular function.

Traditional asymmetric and symmetric encryption techniques (such AES, RSA, DES, etc.) do not permit operations on encrypted files without decrypting it, hence they cannot be used as approaches to allow privacy-preserving tracking. Homomorphic Encryption (HE), on the other hand, is a form of encryption, whichperforms processing on ciphertexts, creating an encrypted version that, while decrypted, equals the outcomes of the functions as if they had been accomplished on the plaintext. It uses an approach akin to an asymmetric method, requiring key pair: a public key for processing as well as a secret key for encrypting and decrypting the data both before and after operation. The resource (untrusted) providers examines the final user's homomorphic private key-encrypted data that has been obtained, applies procedures to this information using the final user's public key, and then transmits the encrypted outcome to the last customer, who is the only one capable to decode the data . The mechanisms used in homomorphic encryption are shown in Figure 4.

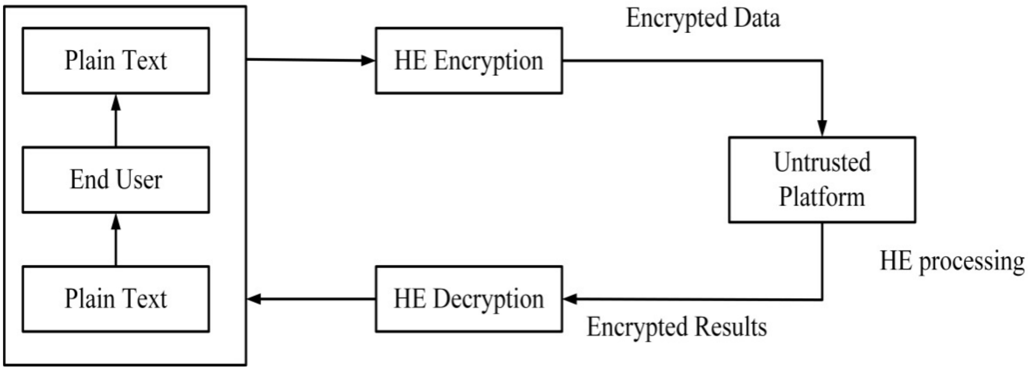


Figure 4: Homomorphic Encryption processes

Partially Homomorphic Encryption (PHE), the very first HE method, could only perform one specific kind of function (e.g., multiplication or addition). It is obvious that the use of HE in real world applications was impeded by the restriction on the type of usable calculations. a Homomorphic Encryption (FHE) system's initial application. The noise that is added to the ciphered text determines how secure the method is. The ciphertext is indecipherable once the noise reaches a certain level. In order to enhance its efficiency, the study also offers a concurrent implementation of the FHE for a public cloud. A Somewhat Homomorphic Encryption (SHE) over the digits has been offered as an approach to streamline the technique. The few possible arithmetic that can be utilized in SHE indicate the cost that must be paid. However, in so many practical uses (such as financial and medical), that does seem logical given that most of the needed assessments, or one-time perform mathematical, fit nicely with SHE restrictions as analysis reports. During the Crypto Standards Workshop, an exact list of prospective real-world use applications that inspires further advancements in homomorphic encryption was compiled.

Homomorphic encryption is an unique from other traditional approaches in that it makes it possible to do calculations on the encrypted files deprived of having to know the hidden encryption key. The result of these operations is encrypted similarly, and the data owner's private key can be utilized to decrypt it. X1 is a factor of!,if X2 implies that y (z1 + z2) = y (z1) y (z2) and y (z1 • z2) = y (z1) y (z2), correspondingly, and where and are the procedures in X2, then X2 is said to be addition and multiplication homomorphic.

Eval is an additional method attribute of a homomorphic encryption approach that can be utilized to perform or calculate across encryption. Without needing the private key that was used to encrypt the information at first, anyone can perform Eval on the encrypted information. On contrary, the security of the original data in the encrypted data is maintained because the ciphertext does not need to be decrypted in order for calculations to be performed on it in the optimization technique.

There are primarily two processes involved in the homomorphic optimization of a deep neural network using discretized weights as well as inputs. The following are the steps listed:

- Calculation of the multisum between the discretized values at each artificial neuron and the encoded signals that are sent to the neurons; the calculations of the multisum utilizes homomorphic addition as its fundamental function.
- Determining the output value of the each neuron.

Each neuron in the surface performs the bootstrapping function to effeciently encrypt the output indication and utilize it in subsequent calculations to the following layers of the network in order to create the neural network suitable regarding the several layers. In the categorization case study employ the orthogonal matrices transformations-based homomorphic encryption for machine learning that protects privacy. The invertible matrix (U1) of ’m’size, where m is equal to total quantity of documents in the training examples, serves as the private key for the suggested matrix transformations-based HE technique.The Somewhat Homomorphic Encryption Schemes

The FHE system Gentry put forward was a method of encryption that permits simultaneous execution of arbitrary addition and multiplications on encrypted information. A probabilistic secret key cryptosystem is defined using the polynomial equation $f = rc * q + 2 * s + n$, where n represents bit value to encrypt as f, two random integers are called as r and q, and rc denotes as public key with the condition that $2*s$ is lesser than $rc /2$. By doing two modulo operations in a way that makes $n = f \bmod dc \bmod 2$, where dc is the corresponding private key, the decryption is then recovered. The suggested plan is divided into four primary parts, as follows:

a.KeyGen(λ): The public and private keys, sk and pk, are returned as two random variables by this function. These results are based on the safety variable, which details the size of the encryption keys as well as the encrypted data.

b. Encrypt (sk, n): A bit signal of 0 or 1 is encrypted (transformed) into a large number of the range of 7 bits that has the same parity as the original binary values using this component.

c. Decrypt (pk, ci): This component decrypts the input cipher text on the basis of suitable secret key pk.

d. Evaluate (sk, C, *): This component offers the cipher text outcome of the accomplished circuit Cover the encrypted values.

The start-up method re-encrypts each bit of the cipher text using a public key to generate a clean, noise-free fresh cipher text. The inner surface of encryption is then removed, allowing the original cipher text to be extracted using a secret key that has been encrypted. It is clear that the bootstrapped variables were doubly decrypted to create the actual plaintext. The SHES creates an FHES that enables an infinite number of multiplications and additions by using a bootstrapping procedure. However, it is important to note that rebooting takes time because re-encryption is typically done across large integer numbers.

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

The CKKS technique is effective at handling realistic (or complicated) numbers—the typical data format for several purposes, including machine learning one of the emphasised fully homomorphic encryption (FHE) techniques(Cheon et al., 2017). The length of the ciphertext increases exponentially with level whenever dealing with fixed-point actual numbers utilising various FHE methods, including such (B) FV and BGV methods. The level of the ciphertext is determined by the deepest circuitry that may be homomorphically analyzed without bootstrapping. The ciphertext length, on the other hand, grows at quadratic rates depending on the CKKS technique levels.

The CKKS system offers a trade-off between message accuracy and efficiency because its encrypted data contains errors. Homomorphic actions are propagated as well as added together including faults in encrypted data. The top bound of mistakes in encrypted data will become a weak as well as ineffective bound as processes on encrypted data are performed. Additionally, as CKKS has lately been the target of attacks, minimizing errors is even more important when utilizing the CKKS method to lessen the danger of an attack. Consequently, a novel method is required that can effectively handle faults in encrypted information. This suggested strategy can be used to decrease error and computing time once the mistake has been properly managed.

3.2.2 Threat technique

In essence, a risk approach is a organized description of every data that influences an application's protection. Based on the OWASP Top Ten for 2010 publication, Injection Attacks are the primary emphasis of the risk model for this study. Exploiting a computer flaw brought on by handling erroneous data is called injection. An attacker uses injection to "inject" information into a computer programme that is weak so that it will run differently or have different effects. A effective code injection can also have severe effects.

3.2.3 Intrusion Detection System (IDS)

An intrusion detection system (IDS) is a software or a physical curriculum whichobserves a system or a network for illegal behavior or regulatory activities. An IDS monitor’s significant moment by looking for security flaws, checking the validity of data, and analysing trends based on previous known assaults.

3.2.4 Encryption of the Data

The HE method, which is dependent on matrix operations, is used to encrypt the information. Encryption process aims to produce high accuracy machine learning or deep learning approaches and utilize the encrypted information for training or testing, protecting method producers from constantly allowing permission to the plain text.

Two pairs of calculations have been run in the initial case study for error detection. The DNN is built in the initial set using basic training examples that the concept creator has access to. There is no private information methods used in this situation. In the next series of trials, the fault identification and segmentation information is initially encrypted on the source side using holomorphic encryption. The design creators receive the encrypted version of the information. As a result, the data cannot include any information that has released. The DNN model is constructed utilizing encrypted information, and the encrypted forecasts are then sent to the original side. Once the encrypted forecasts have been decrypted, the data holder can use the actual projections. The accuracy of the created systems is determined using these forecasts and grounded truth values.

In the second peak load predicting case study, in which the system creators have immediate access to the raw data, a machine learning framework is created based on the plain information. The learned information, including the model's variables, can be applied in situations where future load projections are necessary. Since it is encrypted during in the training stage, access to the plain data is not necessary at these moments. A paillier encryption algorithm is used in this research, and a private and public key combination is formed. The estimations are produced using the frameworks on the encrypted test data once the testing data is encrypted through a shared key. The secret key, which is only accessible to the data holder, is required to decrypt the encrypted prediction accuracy numbers.

By offering the following features, homomorphic encryption could protect a wide range of electrical applications in power grid: securing information kept on a cloud platform.

- Making data analytics accessible to controlled utility companies.
- After the information has arrived at the webserver, HE safeguards the devices against eavesdropping attempts.

- The HE might make data leaks from man-in-the-middle or eavesdropping assaults unintelligible to hackers.
- HE is able to prevent data exchange that is not authorised.

3.3 Datasets

The DARPA 98 and KDDcup 99 datasets were utilised in the past to study involved in different, but because of statistical degeneration, the dataset's assessment of anomaly detection was subpar. The KDD dataset's intrinsic issues have given rise to the newly released NSL KDD datasets that are discussed in. Although it is highly challenging to represent existing actual systems, it can nonetheless be used as a useful benchmark data set for scientists to examine various intrusion detection approaches.

The statistical analysis showed that perhaps the data set has critical flaws that have a significant impact on the systems' efficiency and lead to a very inaccurate assessment of methods for detecting anomalies. A separate data set called NSL-KDD, which is made up of chosen features from the entire KDD dataset, is suggested as a solution to all these problems.

- The train set has no unnecessary information, thus the extractor won't generate any biased results.
- The test set contains no duplicate files, which results in higher efficiency levels.
- The percentage of documents in the initial data set for KDD is negatively associated with the number of documents chosen from each category of records with a high level of difficulty.

Out of the 37 threats that are contained in the validation set, 21 are included in the training sample. The training dataset includes the known kinds of attacks, while the test sample contains additional assaults that are not part of the training datasets. DoS, Probe, U2R, and R2L are the four major types into which the assault methods are divided.

In a DOS attack, the attacker exploits storage or processing power that is overloaded and unable to accommodate genuine user requests. By delivering a TCP message asking to start a Tcp segment, the Neptune operations might overload a target's main memory. To establishing a TCP link between two domains, this packet is a necessary component of a three-way interaction. This packet's SYN flag has been set to signify that a new connection is about to be formed. A common DoS packet flood is the Smurf assault, also known as a targeted broadcast operation. Attacks depend on targeted broadcast to swamp a target with traffic. The attacker utilizes a network on the Inter web the Smurf amplifier, which will receive and reply to targeted transmissions, to transmit a ping packet to the network address. Training assaults amount 45927, testing attacks total 7456, and attacks like mailbomb, apache, and processtable are only detected during the testing stage.

3.4 Pre-processing of Data

Before producing good outcomes, machine learning techniques must be trained on huge amount of data. The majority of the time, the information is kept in storage devices like files, databases, etc.; therefore, it can't be used immediately for training. Before passing the data to the model of machine learning for training, the study must pre-process or enhance it to produce greater benefits. The machine learning algorithm can comprehend how provided values connect to the class thanks to training examples,in order for the machine learning mechanism to comprehend the training data quickly and produce superior outcomes. The phase of pre-processing information involves several procedures. Starting with loading the information into the machine learning techniques, addressing the dataset's omitted variable, scaling the information with the help of normalisation and standards, and dividing the sample into training and testing datasets are all steps in the process. So that the research can utilise the test set to evaluate the efficiency of the automated learning classifiers and provide the training set to the learning classifier for training and testing purposes.

3.5 Feature selection

A frequent pre-processing step in data extraction is feature selection (FSS). It reduces dimensions and eliminates superfluous elements to improve accurateness. It alludes to the issue of figuring out which characteristics are essential to class prediction. Three types of feature selection techniques exist: filter techniques, wrapper methods, and embedding techniques. The importance of features is evaluated in this research using filter approaches based on their connection with the dependent variable. Because they do not require training model, classifiers are substantially faster than wrapper approach**.**

3.6 Classification

Numerous false alarm rates (false both negative and positive) and a dearth of real-time responses are the main problems IDS encounter. These problems can be solved using machine learning techniques. Intelligent IDS that can identify known and unidentified intrusions with high speed, high efficiency, and minimal false alarm rates can be constructed using machine learning algorithms. Thus, the IDS can be strengthened with improved capabilities using machine learning methods.

Breiman introduces the concept of a forest and an election with the random forest method. In a forest, every tree registers to participate. The threshold for selecting the final choice is the collection of percentages and opinions. For the purpose of creating a random training dataset, the RF technique uses the bagged approach for each tree. Dividing features are chosen by RF in a semi-random manner. The potential feature space splitting provides a random sample of a certain ratio. The suggested system's overall workflow diagram is represented in Figure 5.

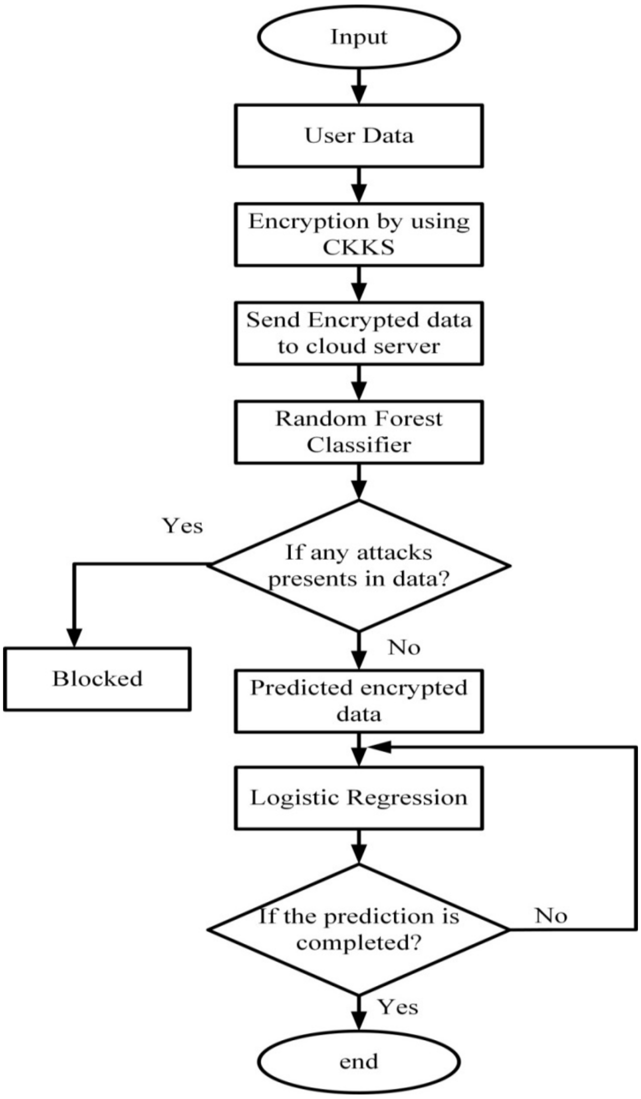


Figure 5:Overall flowchart of the proposed methodology

3.6.1 IDS based on enhanced random forest

Numerous theoretical and practical investigations have concentrated on the usage of these techniques because they have reasonably high identification accuracy compared to other classification techniques and are more tolerance of noisy information. As a multifunctional classification model, the cloaked enhancement of the examples used for training may be accomplished by studying the fundamentals of the bagging method to get N bootstrap training samples with a put-back sampling of the entire data. This strategy successfully lowers the likelihood of classifier. A tree based framework is trained utilizing the dataset generated from the previous operation, and the resulting model is then integrated to produce a forest classification model. The results are predicted by the algorithm using a clear majority.

The conventional random forest algorithm has a lot of opportunity for development. The classification abilities of the every decision trees in the forest have been improved, and the relationship between decision trees in the integrated forest design has also been further optimised. The voting technique used to determine the results has also been optimised. A decent multimodal model must have the following qualities: little interaction between classifiers and effective decision-making inside detectors. The random forest was enhanced in this paper in the forthcomingmanner. The area under the curve (AUC) index was utilized to pick the classification model with the right decision achievement. Next, inter-tree efficiency was done by determining the resemblance between the decision trees. Finally, the outcome correction method was performed by comparing the resemblance between the cyber-attack outcomes. The different attack predictions made using random forest are shown in Figure 6 for cloud servers.

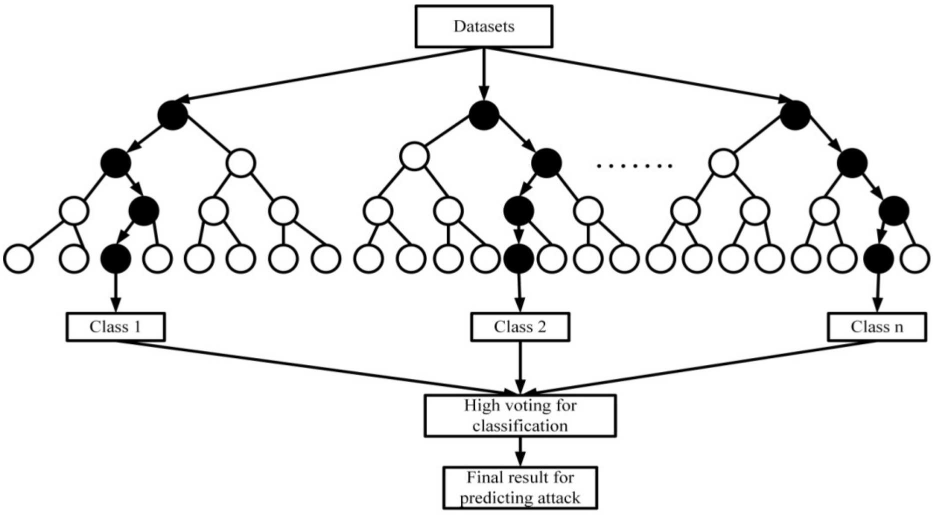

Datasets
....
Class 1
Class 2
Class n
High voting for classification
Final result for predicting attack

Figure 6: Using random forests, various assault predictions

3.6.2 Model Building

The virtualization platform's training data for the Linux virtual machine is used to construct the machine learning technique. For all the benign and malicious examples present in the repository, the activity list data model has been extracted. The machine learning server requires the data that is located in the database server from the virtualization platform to do in memory analyses by using Random Forest technique. The proposed methodology will be kept in the database and used to make predictions.

Iterative approach construction proceeds until high precision is attained. To get the best accuracy and processing speed for the massive amount of data, several random forest method variables, such as the no. of trees (ntree) and the range of variables attempted at every split (mtry), will be modified.

3.7 CKKS scheme for encryption

Among the most recent homomorphic techniques, along with BFV, is the CKKS technique that was modified from the TenSEAL package. CKKS and BFV are accessible in the Golang language through the latigo library that is a transformation of the C++-written TenSEAL package. The ability to write in Golang opens up fresh options for web services homomorphic functionalities. The evaluations on the official process demonstrate that these two languages function pretty similarly. Though BFV only permits actions on integers, CKKS activation in response up to complex numbers.

Cheon-Kim-Kim-Song (CKKS) is defined as a Leveled Homomorphic Encryption technique,which permits approximation across complex numbers (hence, real numbers). Text encyrption using CKKS approach is shown in figure 7.

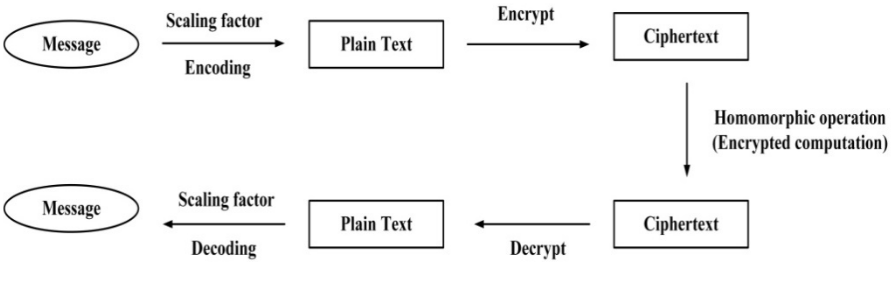


Figure 7: Text encryption using CKKS approach

3.7.1 CKKS Parameters

A vector of actual numbers is encoded using the CKKS technique into polynomials in plaintext.

Scaling factor (Sf): The binary description of the number's encoded accuracy is determined by the scaling factor. Actual numbers are transformed into integers.

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

3.7.2 Ciphertext modulus q

Operational parameter that establishes the maximum number of calculations permitted (how much noise can be tolerated). Frequently set implicitly utilising user-specified multiplicative depth value

3.7.3 Ciphertext dimension N

Depending on the selected level of security and cypher text modulus q, a lowest value is calculated. Additionally, it is N = 2n times larger than the vector of encrypting actual values.

3.7.4 CKKS Keys

A secret key and a public key are created in the public key encryption system known as CKKS. The private key is utilized for decryption and should be kept secret, whereas the public key is utilized for encryption and may be shared. Relinearization keys generated by the owner of the secret key are a different class of public keys needed for this process. Another kind of public keys required to carry out encryption vector rotation operations on batched ciphertexts are galois keys. The encrypting batched vector's summation is one use for vector rotational.

3.7.5 CKKS internal operations

3.7.5.1 Relinearization

TenSEAL automatically performs the operation after each encrypted multiplying. This enables the use of a polynomial pair instead of a triple, resulting in the multiplication of two basic plaintexts when decrypted using standard encryption circuits, which require only the secret key and not its square. Consequently, if reordering is performed after each ciphertext-ciphertext multiplication, the ciphertexts will always be of the same size and use the same decryption circuit.

3.7.5.2 Rescaling

As it essentially rescales the underpinning encryption plaintext and removes a specific amount of the least important bits from the communication, modulus switching is referred to as rescaling in CKKS. TenSEAL autonomously performs the task after every multiplication, whether it is encryption or plain. The amount of homomorphic multiplications causes the prediction error to grow massively. The majority of HE methods typically employ a modulus-switching strategy to address this issue. The modulus-switching process is known as rescaling in the context of CKKS. After a homomorphic multiplication and the rescaling technique, the prediction error increases linearly rather than exponentially.

3.7.5.3 TenSEAL CKKS Context

The TenSEAL context serves as the library's main structural element. It creates and saves the appropriate keys for an encrypted calculation. The contextual produces the secret key for deciphering, the public key for encrypting, the Galois keys for rotational, and the relinearization keys for relining ciphertexts. The thread-pool, which regulates how many tasks should be executed concurrently while carrying out parallelized processes, will likewise be handled by the same class. Additionally, the environment may be set up to automatically relinearize and scale ciphertext while it is being computed.

3.7.5.4 Plain tensor creation

Basic tensor forms are translated by the PlainTensor class into the encryption forms provided by TenSEAL. It's the initial step necessary to use TenSEAL to create an encryption tensor. The encryption tensor builders also autonomously do this conversion.

3.8 Predicted Encrypted Data by Logistic Regression

It may be utilised as a single node, one-layer neural network to learn a logistic regression model (without any coding). In order to compare encrypting learning and evaluation, it makes use of this approach. The study's main objective is to assess the logistic regression model using plain variables on an encrypted testing set (alternatively using encrypted variables). To build a TenSEALContext to describe the variables and the strategy that will be used. Here, the study chooses for manageable and safe variables that let us do a solitary multiplication. That is sufficient to assess a logistic regression model, but the research will examine the demand for more extensive variables while learning on encrypted files.

The EncryptedLR class, the study doesn't compute the sigmoid function on the encrypted output of the linear layer, simply because it's not needed, and computing sigmoid over encrypted data will increase the computation time and require larger encryption parameters. However, the study will use sigmoid for the encrypted training part. It proceeds with the evaluation of the encrypted test set and compares the accuracy to the one on the plain test set. The study will redefine a Logistic Regression model that can both forward encrypted data, as well as backpropagate to update the weights and thus train the encrypted logistic regression framework on encrypted data. Following are more details about the training.

3.8.1 Loss Function

The research uses the binary cross entropy loss function with regularisation, where X(i) indicates the i'th predicted label, $X'(i)$ indicates the i'th outcome of the logistic regression approach, and represents the n-sized weighted vector.

$$Loss(\phi) = -\frac{1}{n}\sum_{i=1}^{n}\left(X(i)\log X'(i)\right) + \left(1 - X(i)\right)\log\left(1 - X'(i)\right) + \frac{\lambda}{2n}\sum_{k=1}^{m}\phi_k{}^{2}$$

3.8.2 Parameters Update

The typical rule is as following, where Y(i) is the i'th primary input, to update the variable: $\phi_i = \phi_i - \beta\left(\frac{1}{n}\sum_{i=1}^{n}\left(X'(i) - X(i)\right)Y(i) + \frac{\lambda}{n}\phi_i\right)$

3.8.3 Sigmoid Approximation (Sig(a))

A form of sigmoid, or a group of functions with similar distinct characteristics, is the logistic function in linear regression. Every real value may be converted to likelihood among 1 and 0 using the mathematical function sigmoid.

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

This involves training an encrypted logistic regression approach using encrypted input to get at the last portion. Each epoch requires the weight to be decrypted and then re-encrypted, which is required since, after upgrading the weights, the research is no longer able to utilise it to conduct appropriate multiplications and must return to the starting ciphertext level. In practise, this would entail returning the weights to the owner of the secret key for decryption and re-encryption. In such situation, the amount of connection every epoch will be limited to a few Kilobytes. The Logistic regression framework for predicting encrypted information is represented in figure 8.

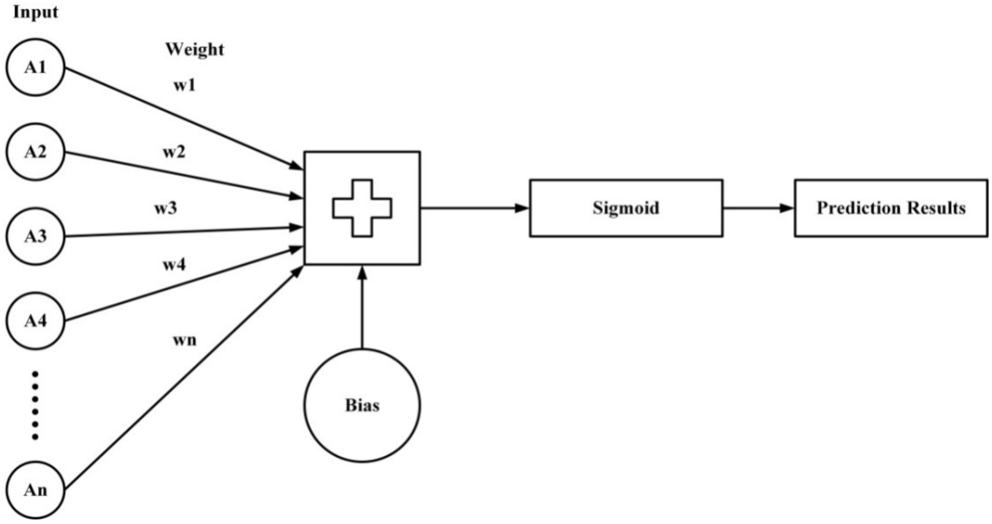


Figure 8: Logistic regression mechanism for predicting encrypted data

Following inference and training HE will be used to produce an encrypted approach and labels. Only the user with access to HE's secret key may decipher the outcomes and obtain either the classifiers or the tags for the categorization. In order to create an encrypted logistic regression model, the cloud may homomorphically execute a learning algorithm to the encrypted files. This approach could then be transmitted towards the data owner for decryption. By doing this, the owner of the data was able to effectively outsourcing the training phase without giving the cloud access to either their sensitive information or the trained model.

4. Results

The effectiveness of the suggested solution for data encryption authenticating during aggregating is evaluated. Here, the computational burden of creating a key to encrypt the message sent between the models is of utmost importance. The impact of n is evaluated after first estimating the influence of a secret key.In this research, the offered method's operating time is calculated using the following units: En time, De time, 2-En, and 2-De. Encryption and Decryption define a single layer of encryption, whereas 2-En and 2-De establish a second layer.

4.1 Influence of secret key

In this, the Fully Homomorphic Encryption method's effect on the length of the secret key is calculated. The secret key bit length is determined by adding 100 to the range [100,500]. Figure. 9 displays the outcome of the suggested technique. It can be demonstrated that using a secret key has no impact on encryption, and it is depicted in table 1. Meanwhile the computation time grows with the size of the secret key, the other methods must do an exponential function over it. It is well established that perhaps the size of exponentiations has an impact on how effective the suggested one is.Consequently, the effects of all other measurement data on costs can be mitigated if random number length is defined before encryption. Because the longer secret key requires more computations to encrypt and decrypt data from the original message, it is discovered that the longer secret key has significantly raised the computational cost. The length of the key increases with network density, and when key size is decreased for big dense networks, privacy may tend to be reduced. The operational costs of computation constantly rise as the key length does as well, reducing the speed at which encrypted information may be sent. To prevent overhead as well as complexities, the key size must be adjusted at a specified pace.

Table 1: Effect of secret key

Key bit length	Encryptiontime (ms)	Decryptiontime (ms)	2-encryptiontime (ms)	2-decryptiontime (ms)
100	6.5	9.2	14.3	3.2
200	6.4	9.3	14.5	3.2

300	9.6	11.6	14.4	4.2
400	12.7	15.2	14.4	6.0
500	14.2	16.3	13.9	7.1

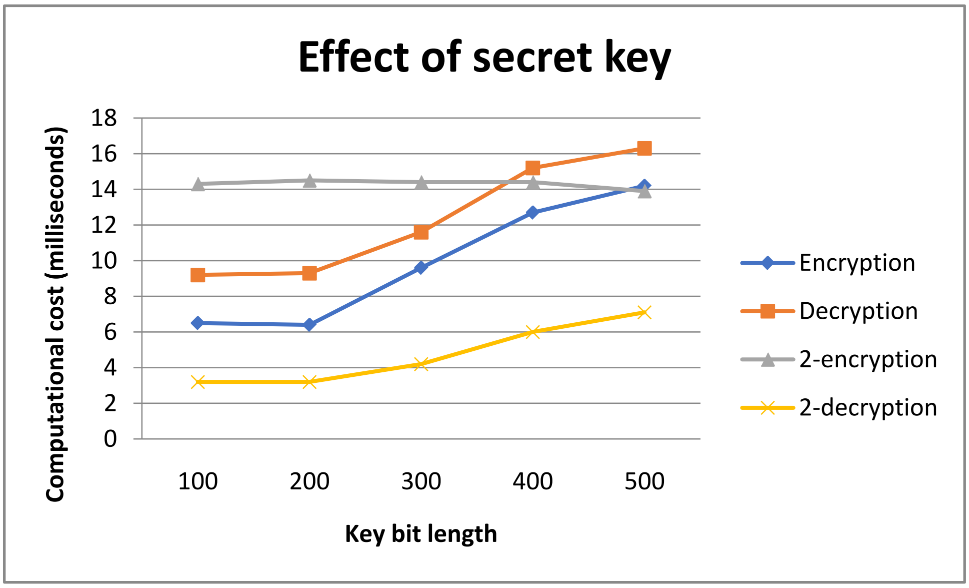


Figure 9: Influence of secret key

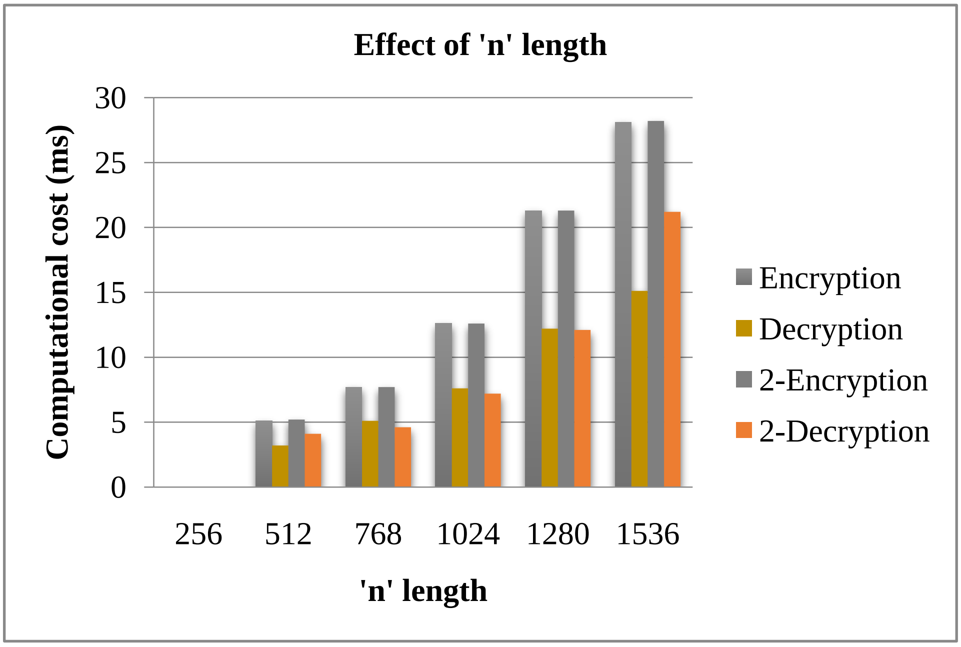
4.2 Influence of ‘n’ length

This is implemented utilizing modules in sizes ranging from [256–1536] in multiple sets of 2. The outcomes are displayed in table 2 and Figure10. The effectiveness of malicious action recognition has been significantly impacted by the length of 'n'. It is clear that perhaps the computing time of the suggested technique grows as 'n' length rises, but as the modular size grows, so does the adequate security. The computational cost of the suggested approach has been lowered by 10 milliseconds for 1024 bits, which can be observed. Provided that decryption is more effective than other related procedures, which is acceptable for the "u" with really limited resources, this could be applied generally.Further, it can be noticed that the proposed technique's validity is demonstrated through superior development and evaluation.

Table 2: Effect of ‘n’ length

Key bit length	Encryptiontime (ms)	Decryptiontime (ms)	2-Encryptiontime (ms)	2-Decryptiontime (ms)
256	0	0	0	0
512	5.1	3.2	5.2	4.1
768	7.7	5.1	7.7	4.6
1024	12.6	7.6	12.6	7.2
1280	21.3	12.2	21.3	12.1
1536	28.12	15.11	28.2	21.2

Figure



10: Effect of ‘n’ length

4.3 Input data length

Utilizing the input data, which is configured with various values of the range [50, 250] bits in multiple sets of 2, the suggested technique is analyzed for its adaptability for various lengths. As can be observed from table 3 and Figure 11, the algorithm's computing time differs less than the actual input data. This may suggest that the suggested solution may handle a variety of messages while providing more privacy and security. The approach does not use any kind of compressing to reduce the size of the larger message; hence system overhead tends to rise. Additionally, as system inefficiency rises, the system's computing performance is significantly decreased. Moreover, whenever a message is transmitted during encryption at a larger measurable rate, message loss is more likely to occur. The number of transmitted messages, in addition to the message's length, its quantity of bits, or even its secret key, is another consideration.

Table 3: Effect of input data length

Key bit length	**Encryption-time (ms)**	**Decryption-time (ms)**	**2-Encryption-time (ms)**	**2-Decryption-time (ms)**
50	13.37	8.81	13.22	6.2
100	13.37	7.22	13.22	7.4
150	13.5	7.22	13.33	6.6
200	13.76	7.22	13.72	6.8
250	13.36	7.51	13.34	6.6

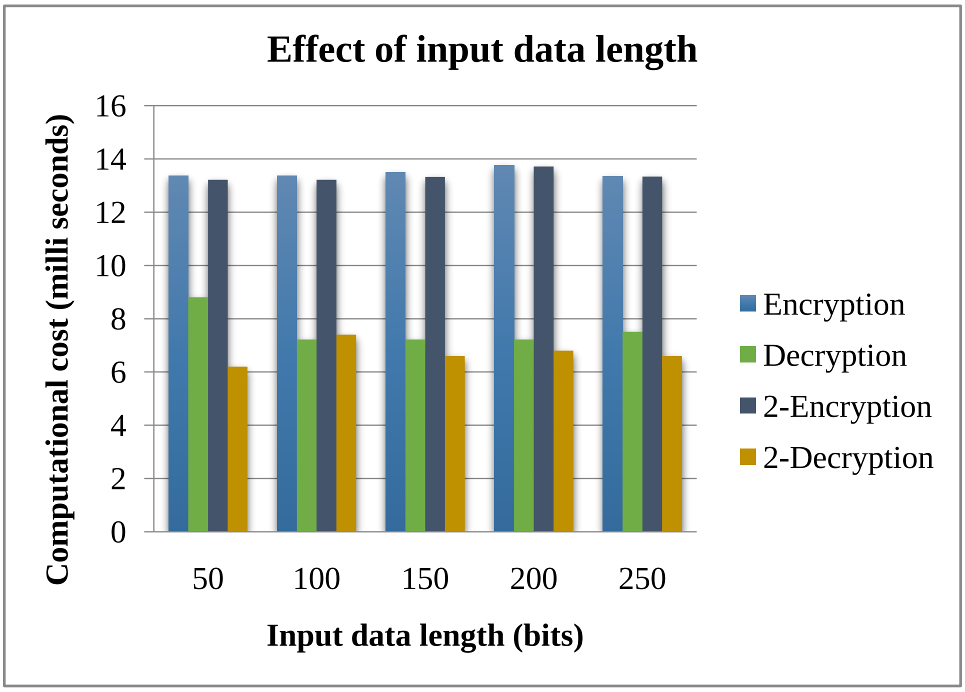


Figure 11: Effect of input data length

4.4 FHE with and without Network Analyser Influence

Here, the creation of a Fully Homomorphic Encryption with or without a network analyser is shown in figure 12 to protect the data with feature sets in order to accomplish purpose.The network analyser keeps track of the host process's network behaviour.Propagation operations and communication with the centralized command as well as control server are some of these tasks. Despite the fact that some of these signals are repeated and organized differently to increase the precision in identifying valid network traffic. In actuality, these characteristics are essential for attaining high detection accuracy. New features are also suggested to improve the detection of encrypted data.

Accuracy on plain test set: 0.982421875

Final encrypted accuracy is 0.9453125

Difference between plain accuracy and encrypted accuracies: 0.037109375

Comparing accuracies over encrypted data versus the plain accuracy obtained above, demonstrate that training on encrypted data has slight impact on the outcome. The accuracy is not significantly impacted by testing on the encrypted test set. In several cases, the evaluation performed even better when it was encrypted.

In Fully Homomorphic Encryption, that is without combining network analyser the encrypted data is only secured. The network and encrypted data are both secure when Fully Homomorphic Encryption and network analyser are used together.This, we may get the conclusion that the network and data protection is very precise. By using this method, we may securely save our data as well as confidential material.

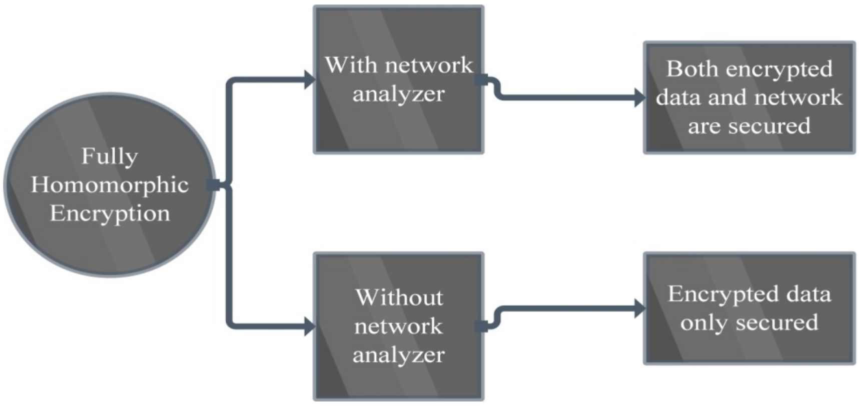
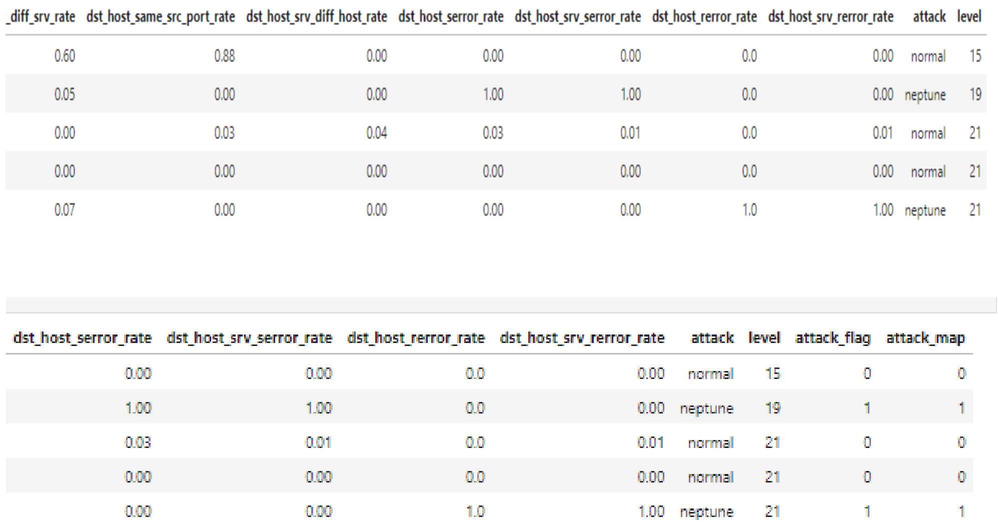

Fully Homomorphic Encryption
With network analyzer
Both encrypted data and network are secured
Without network analyzer
Encrypted data only secured

Figure 12: FHE with or without Network Analyser

After each epoch, the weights are decrypted and then re-encrypted because they need to be updated after each epoch. We must restore them to the starting cipher text level because we can no longer utilize them to accomplish enough multiplications. In practice, this would entail returning the weights to the owner of the secret key for decryption as well as re-encryption. In that case, the amount of communication every epoch will be limited to a few Kilobytes.

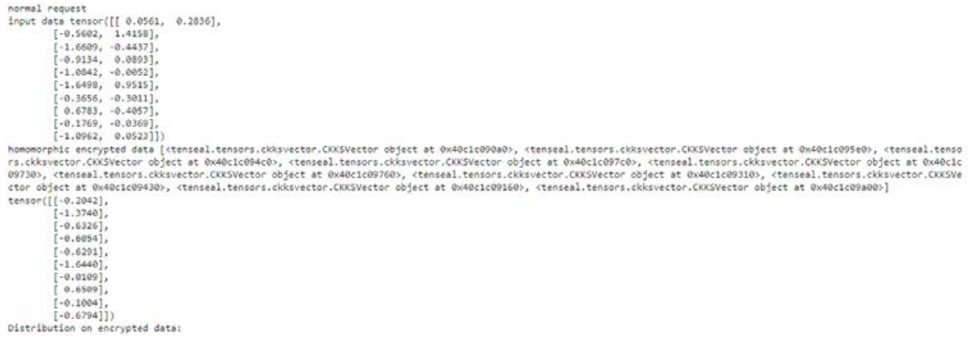


_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	attack	level
0.60	0.88	0.00	0.00	0.00	0.0	0.00	normal	15
0.05	0.00	0.00	1.00	1.00	0.0	0.00	neptune	19
0.00	0.03	0.04	0.03	0.01	0.0	0.01	normal	21
0.00	0.00	0.00	0.00	0.00	0.0	0.00	normal	21
0.07	0.00	0.00	0.00	0.00	1.0	1.00	neptune	21

dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	attack	level	attack_flag	attack_map
0.00	0.00	0.0	0.00	normal	15	0	0
1.00	1.00	0.0	0.00	neptune	19	1	1
0.03	0.01	0.0	0.01	normal	21	0	0
0.00	0.00	0.0	0.00	normal	21	0	0
0.00	0.00	1.0	1.00	neptune	21	1	1

Figure 13: Attack description

The attack result shows in figure 13, in which 0 indicates that there has been no attack and 1 indicates that an attack has occurred. This concerns using encrypted data to build a logistic regression model.



```
normal request
input data tensor([[ 0.0561,  0.2836],
        [-0.5602,  1.4158],
        [-1.6609, -0.4437],
        [-0.9134,  0.0893],
        [-1.0842, -0.0052],
        [-1.6498,  0.9515],
        [-0.3656, -0.3011],
        [ 0.6783, -0.4057],
        [-0.1769, -0.0369],
        [-1.0962,  0.0523]])
homomorphic encrypted data [<tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c090a0>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c095e0>, <tenseal.tenso
rs.ckksvector.CKKSVector object at 0x40c1c094c0>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c097c0>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c
09730>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c09760>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c09310>, <tenseal.tensors.ckksvector.CKKSVe
ctor object at 0x40c1c09430>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c09160>, <tenseal.tensors.ckksvector.CKKSVector object at 0x40c1c09a00>]
tensor([[-0.2042],
        [-1.3740],
        [-0.6326],
        [-0.6054],
        [-0.6291],
        [-1.6440],
        [-0.0109],
        [ 0.6509],
        [-0.1004],
        [-0.6794]])
Distribution on encrypted data:
```

Figure 14: The outcome when it is detected that there is no attack

The outcome when it is detected that there is no attack is depicted in figure 14, which shows the decrypted result of predicted outcome produced by the machine learning model.

4.5 Accuracy evaluation of Random Forest classifier

Accuracy measures how accurately the system model operates. Generally speaking, it is the proportion of correctly expected observance to all observational data. Equation (1) is used to describe the accuracy as

$$Accuracy = \frac{t_{pos} + t_{neg}}{t_{pos} + t_{neg} + f_{pos} + f_{neg}} \qquad (1)$$

In the present study, NSL-KDD attributes were chosen at random to minimize the dataset size according to the model trained. The complexity of the system and training time can both be decreased by choosing features randomly.The features in a traffic record can be categorized into four groups: host-based, intrinsic, time-based, and content. These features give information on the interaction with the traffic input by the IDS. The various feature categories are described in table 4.

- Without examining the payload, intrinsic properties can be inferred from the header of a packet, which contains the essential details of the packet. Features 1-6 can be found in the category.
- Since packets are supplied in numerous pieces rather than all at once, content features contain details about the original packets. The structure can obtain the payload using this data. Features 10 to 22 are included in this category.
- Time-based features analyze traffic input in more than a two-second window as well as save details like the number of times it tried to join a single server. Rather than providing details about the substance of the traffic input, those features primarily provide counts as well as rates. Features 23–31 are part of the category.
- Related to Time-based attributes, Host-based features examine throughout a sequence of making connections as opposed to a two-second window (how several connections across x connections are established with the same host and how many queries are made to that host). These functions are made to handle assaultswhich lasts lengthier than a 2-second window of time. In this region, the Features 32 through 41 can be found.

The feature types in this data set can be broken down into 4 types:

- Four Categorical Features: (2, 3, 4, and 41)
- Six Binary Features: (7, 12, 14, 20, 21, and 22)
- Twenty-three Discrete Features: (8, 9, 15, 19, 23–41)
- Ten Continuous Features: (1, 5, 6, 10, 11, 13, 16, 17, 18, and 19)

The data analysis revealed that, while it might not be advantageous in all situations, this strategy performed effectively in the present system. Thus, the Random Forest classification model was trained using several NSL-KDD based on feature instances, as well as various parameters were used to evaluate the effectiveness of the system. The table 4 and figure 15 shows a graphical depiction of the accuracy rate for different feature subsets.

Due to their ability to handle irregular data sets, random forest classifiers were used to achieve a total accuracy of almost 99% on all four extracted features. A random forest may identify data imbalances, and it uses the bootstrap approach to raise minority class incidence while lowering data misclassification and improving accuracy. In a similar manner, Random Forest undertakes a thorough comprehensive investigation of all options along its branch, producing an amazing performance. The classifiers generally try to keep costs down, which helps them become better predictors for malware detection. They are therefore a useful tool for detecting network intrusions.

Table 4: Accuracy of Random Forest Classifier

Feature sets	Accuracy (%)
1	99.882
2	99.890
3	99.885
4	99.891

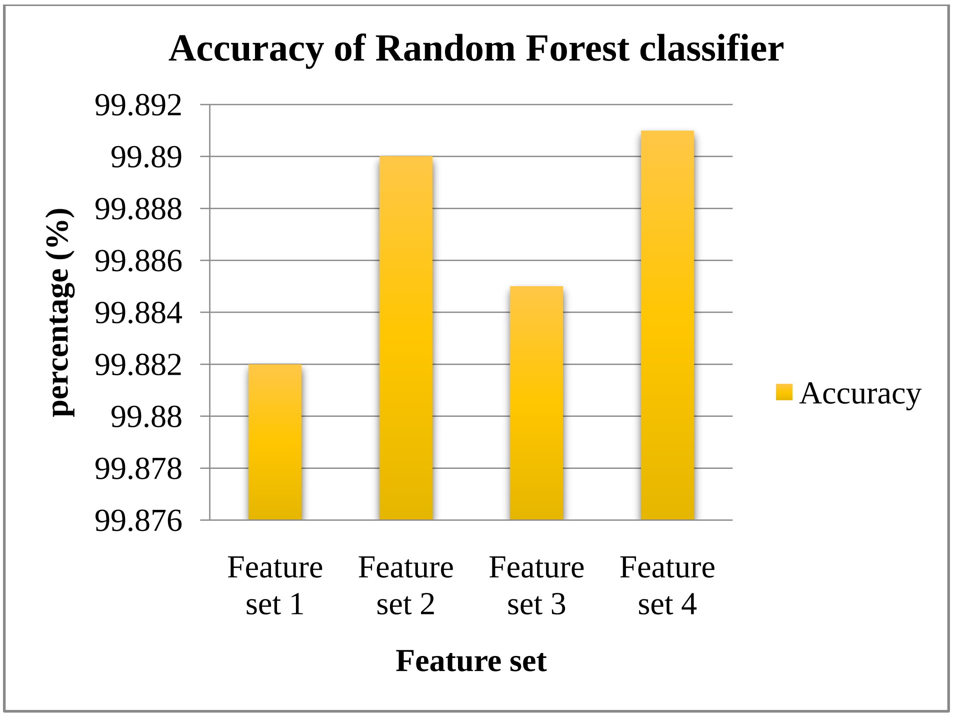


Figure 15: accuracy of Random Forest classifier

4.6 Evaluation outcomes of proposed method

The random forest classification system is a controlled machine learning mechanism that is very versatile, simple to use and gives excellent results with or without section depicts. It is straightforward, and the reality it is able to be applied to both regression and classification tasks—which that make up the majority of today's machine learning tasks—increases its acceptance. An ensemble technique called random forests produces numerous decision trees as well as combines them so that the system can predict the classification of the data more precisely and consistently. Figure 16 displays a diagram that explains how Random Forest functions.

When opposed to the MGA-SVMHGS-PSO-ANN methodology, K-Means with canopy (Euclidean), as well as BAT-MC, the predicted approach Random Forest achieves enhanced performance which is tabularized in a table 5. The Random Forest approach yields more accurate findings. The Random Forest classifier was used to get an accuracy level of 99.887% in this case**.**

Table 5: Evaluation outcomes of proposed method

Author & year	Method	**Accuracy (%)**
(Hosseini & Zade, 2020)	MGA-SVMHGS-PSO-ANN method	99.3
(Vinutha & Poornima, 2019)	K- Means with canopy (Euclidean)	90.63
(Su et al., 2020b)	BAT-MC	84.25
	Proposed RF	99.887

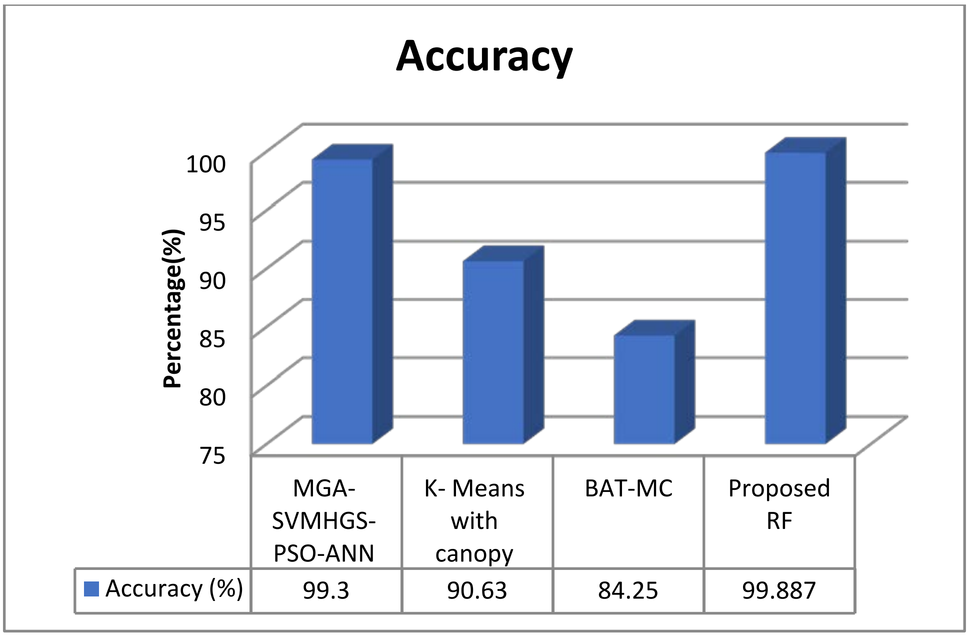


Figure 16: evaluation outcomes of proposed system

This research revealed an FHE-enhanced version of the standard Logistic Regression algorithm because it includes several Homomorphic Encryption-incompatible procedures and operations. To defend both model knowledge and data against numerous attackers, logistic regression on encrypted files will become increasingly crucial. Effective arithmetic evaluations of encrypted real figure data are made possible by the homomorphic encryption (HE) for such arithmetic of approximate numbers approach, which stimulates the development of privacy-preserving machine learning techniques and algorithms. Real-time predictions are made possible by the fully-homomorphic encryption used in the reasoning phase on the encrypted form. A logistic regression with completely homomorphic encryption is utilized to forecast encrypted data. Our findings, therefore, assist the creation of potential functions and for privacy-preserving logistic regression.

According to the experimental findings, our method performed better at classifying data from the NSL-KDD dataset as well as the majority of real datasets. Additionally, because our technique does not need any non-polynomial approximation operations during the training step, the estimated duration of iteration was efficient. Research permits the use of FHE across the entire logistic regression process, from training to inference.

5. Discussion

Although there are many applications where completely homomorphic encryption is presently beneficial, some of them cannot employ it because of the restrictions of this method. Much current research doesn't really cover the privacy of the secret keys, as well as homomorphic encryption uses, which are typically client-server instances in which both the data as well as the methodology should be kept under wraps. Here, a decision is based on the application's specific circumstances.

Natural protection against attacks on the CKKS system, which effectively retrieves the cipher text’s encryption interference, is to change the decryption method, thereby it produces no output but just an approximation that is independent of the secret key as well as encryption randomness. Simple countermeasures are all that is needed to lessen the impact of the assaults that are covered in the paper.

According to the testing findings, our algorithm performed better at classifying data from datasets. Additionally, because of our technique during the iteration process, the average amount of time of the iterative process was effective. And for the logistic regression, our training approach can be used in conjunction with reducing FHE inference algorithms. Our research thus makes it possible for a whole logistic regression method, including training and inference, to use FHE.

The method uses the cloud's ability to perform more sources of energy operations to streamline and lessen the need for real-time threat detection. The NSL-KDD dataset was employed for testing the proposed methodology. The chosen model functioned well, according to a variety of performance measures, including accuracy, as demonstrated by the findings.

Due to their ability to handle irregularly distributed data, random forest classifiers were used to achieve an average accuracy of 99.88% for all four extracted features. A random forest may identify data imbalances, and it employs the bootstrap procedure to raise minority class incidence while lowering data misinterpretation and improving accuracy. Additionally, the classifier seeks to reduce costs by improving predictions for malware detection. They are therefore a useful tool for detecting attacks.

These advances are well positioned to help the field, which is data-rich and privacy-conscious. Despite needing to share the data, confidentiality categorization techniques might enable personal access to a predictive technique. FHE is used for a range of statistics as well as machine learning problems, including the secret evaluations of deep neural networks. There will continually be more uses for FHE as its efficiency and usefulness increase.

5.1 Future work

There are two additional elements to take into account in addition to the known computing expenses when using the homomorphic encryption algorithm. One illustration is the fact that the majority of popular homomorphic encryption techniques lacked multi-user capabilities. In theoretical research, we assumed that all data providers have the same encryption key, but in practice, this is hardly true while there are multiple data users. Multiple data owners might or might not be trusted in this situation. These data owners don't want to use cooperatively encrypted data to train a system.

A homomorphic encryption algorithm involving multiple keys could be used to resolve this issue practically because it ensures that the data will be encrypted with many independent keys as well as allows the training data to be retrieved from the various data holders, each of whom has a unique key. One disadvantage is that in order for homomorphic encryption to function properly, significant structural changes, as well as specific client-server programs, as well as specific client-server programs are needed. Without first getting user authorization, industrial companies are not permitted to use this method for scientific analysis. This could increase overall costs and keep the business focused on alternative solutions that are better at encrypting analytical activities and don't give a damn about privacy issues..

Fully Homomorphic Encryption does not ensure data integrity, in contrast to more conventional encryption techniques. Homomorphic encryption schemes do not ensure data security, in contrast to conventional encryption techniques. Whatever Big Data analytics architecture for privacy preservation may have limited effectiveness and accuracy due to the issue of data corruption or integrity losses in homomorphic encryption technologies. And although the federated framework model has been suggested as a means of enabling cooperative training, the privacy issue has grown due to the possibility of collaborative data being leaked by combining multiple databases, particularly in light of a wide range of privacy assaults, like the privacy risk from interconnections among databases. Researchers may be required to investigate certain security techniques for the ensemble learning aggregator to recognize malignant players depending upon their method updations due to the significant privacy issue in federated learning methods. Eventually, malicious attackers might access the computing infrastructure.

6. Conclusion

There are numerous uses for Machine Learning (ML) categorization. A user may choose to delegate categorization activities in the big data regime to lessen the user's own computational workload. An organisation could want to provide these clients a categorization model and categorization services in the interim. However, operations like network traffic demand sensitive information that either party might not wish to disclose. Particularly, the use of new technology by hackers and the rise in demand for the Cybercrime services have led to even more advanced dangers. Organizations and businesses are becoming more concerned about security issues as a result of the rise in these and other criminal activities. A major problem is also being presented by the increasing costs of an effective IT safety environment and workforce.Secure computations over encoded data is possible with fully homomorphic encryption (FHE), which does not require decoding. FHE allows categorization to be outsourced to the cloud without disclosing any information. Thus, this paper provided an effective solution to the privacy protection problem by ML methods.

This research uses NSL KDD dataset, which contains the attacks like r2l, DoS, u2r, and probe. One of the crucial processes in the data mining procedure, data pre-processing involves preparing and transforming the initial dataset. Data pre-processing includes a number of procedures, such as feature reduction, data cleansing, and feature construction. Feature selection and extraction are components of feature minimization. In data pre-processing, extraction of features, construction, and selection are all separate techniques. Feature extraction and feature selection can also be coupled, as can feature creation and feature selections, relying on the analysis of the situation. In this paper, feature extraction is followed; data that is high dimensional is converted to low dimensional information by feature extraction. The method of feature extraction decides what information may be gleaned from audit information is most helpful for analysis.

This research proposes a FHE mechanismfor encrypting the user information. Moreover, CKKS scheme is used in FHE; a public key and a secret key are produced in the CKKS public key encoding system. The secret key is required for decoding and should be kept hidden, whereas the public key has been needed for encoding and can be distributed. Moreover, ML mechanisms like Random Forest (RF) and Logistic Regression (LR) are employed in this research. The RF method is used to identify the malicious attacks and blocks the attacks from encrypted data in the server. In addition, LR method is utilized for predicting the encrypted information on server and the predicted informationis send to the user. It is examined that secure predictions over encoded data at a cloud server occurs during maintaining the user's data privacy by applying FHE to ML prediction. In this way, the user can safely offshore their data prediction work and minimize their own computational load. Moreover, the effectiveness of the presented framework is validated and contrasted with other existing methods for determining the effectiveness of the developed ML technique. The results indicated that the presented ML model has higher enhancement in accuracy.

References

Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2019). A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Computing Surveys*, *51*(4), 1–35. https://doi.org/10.1145/3214303

Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., & others. (2021). Homomorphic encryption standard. In *Protecting Privacy through Homomorphic Encryption* (pp. 31–62). Springer.

Alloghani, M., M. Alani, M., Al-Jumeily, D., Baker, T., Mustafina, J., Hussain, A., & J. Aljaaf, A. (2019). A systematic review on the status and progress of homomorphic encryption technologies. *Journal of Information Security and Applications*, *48*, 102362. https://doi.org/10.1016/j.jisa.2019.102362

Aloqaily, M., Otoum, S., Al Ridhawi, I., & Jararweh, Y. (2019). An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, *90*, 101842.

Alshammari, A., & Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data*, *8*(1), 1–24.

Aslett, L. J. M., Esperança, P. M., & Holmes, C. C. (2015). *Encrypted statistical machine learning: New privacy preserving methods*. https://doi.org/10.48550/ARXIV.1508.06845

Bai, B. (2022). Computer Technology of Environmental Design Specialty in Cloud Computing Environment. *International Conference on Multi-Modal Information Analytics*, 973–979.

Barkataki, S., & Zeineddine, H. (2015). On achieving secure collaboration in supply chains. *Information Systems Frontiers*, *17*(3), 691–705. https://doi.org/10.1007/s10796-013-9448-3

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

Bost, R., Popa, R. A., Tu, S., & Goldwasser, S. (2015). Machine Learning Classification over Encrypted Data. *Proceedings 2015 Network and Distributed System Security Symposium*. Network and Distributed System Security Symposium, San Diego, CA. https://doi.org/10.14722/ndss.2015.23241

Cai, F., Zhu, N., He, J., Mu, P., Li, W., & Yu, Y. (2019). Survey of access control models and technologies for cloud computing. *Cluster Computing*, *22*(3), 6111–6122.

Chabanne, H., De Wargny, A., Milgram, J., Morel, C., & Prouff, E. (2017). Privacy-preserving classification on deep neural network. *Cryptology EPrint Archive*.

Chauhan, K. K., Sanger, A. K. S., & Verma, A. (2015). Homomorphic Encryption for Data Security in Cloud Computing. *2015 International Conference on Information Technology (ICIT)*, 206–209. https://doi.org/10.1109/ICIT.2015.39

Chen, H., Gilad-Bachrach, R., Han, K., Huang, Z., Jalali, A., Laine, K., & Lauter, K. (2018). Logistic regression over encrypted data from fully homomorphic encryption. *BMC Medical Genomics*, *11*(S4), 81. https://doi.org/10.1186/s12920-018-0397-z

Chen, H., Huang, Z., Laine, K., & Rindal, P. (2018). Labeled PSI from fully homomorphic encryption with malicious security. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 1223–1237.

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. In T. Takagi & T. Peyrin (Eds.), *Advances in Cryptology – ASIACRYPT 2017* (Vol. 10624, pp. 409–437). Springer International Publishing. https://doi.org/10.1007/978-3-319-70694-8_15

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

D’Alconzo, A., Drago, I., Morichetta, A., Mellia, M., & Casas, P. (2019). A survey on big data for network traffic monitoring and analysis. *IEEE Transactions on Network and Service Management*, *16*(3), 800–813.

Das, D. (2018). Secure cloud computing algorithm using homomorphic encryption and multi-party computation. *2018 International Conference on Information Networking (ICOIN)*, 391–396.

DesLauriers, J., Kiss, T., Ariyattu, R. C., Dang, H.-V., Ullah, A., Bowden, J., Krefting, D., Pierantoni, G., & Terstyanszky, G. (2021). Cloud apps to-go: Cloud portability with TOSCA and MiCADO. *Concurrency and Computation: Practice and Experience*, *33*(19), e6093.

Dugan, T., & Zou, X. (2017). Privacy-preserving evaluation techniques and their application in genetic tests. *Smart Health*, *1–2*, 2–17. https://doi.org/10.1016/j.smhl.2017.03.003

Gadekar, D. P., Sable, N. P., & Raut, A. H. (2019). Exploring Data Security Scheme into Cloud Using Encryption Algorithms. *International Journal of Recent Technology and Engineering (IJRTE), Published By: Blue Eyes Intelligence Engineering & Sciences Publication, ISSN*, 2277–3878.

Gahi, Y., Guennoun, M., Guennoun, Z., & El-Khatib, K. (2012). On the use of Homomorphic Encryption to Secure Applications, Services, and Routing Protocols. *European Journal of Scientific Research*, *88*(3), 416–438.

Gai, K., Qiu, M., Li, Y., & Liu, X.-Y. (2017). Advanced Fully Homomorphic Encryption Scheme Over Real Numbers. *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, 64–69. https://doi.org/10.1109/CSCloud.2017.61

Gao, M., Ma, L., Liu, H., Zhang, Z., Ning, Z., & Xu, J. (2020). Malicious network traffic detection based on deep neural networks and association analysis. *Sensors*, *20*(5), 1452.

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/ACCESS.2019.2923640

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, 169–178.

Gharaibeh, A., Salahuddin, M. A., Hussini, S. J., Khreishah, A., Khalil, I., Guizani, M., & Al-Fuqaha, A. (2017). Smart cities: A survey on data management, security, and enabling technologies. *IEEE Communications Surveys & Tutorials*, *19*(4), 2456–2501.

Gorantala, S., Springer, R., Purser-Haskell, S., Lam, W., Wilson, R., Ali, A., Astor, E. P., Zukerman, I., Ruth, S., Dibak, C., & others. (2021). A general purpose transpiler for fully homomorphic encryption. *ArXiv Preprint ArXiv:2106.07893*.

Han, J.-L., Yang, M., Wang, C.-L., & Xu, S.-S. (2012). The Implemention and Application of Fully Homomorphic Encryption Scheme. *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 714–717. https://doi.org/10.1109/IMCCC.2012.173

Hosseini, S., & Zade, B. M. H. (2020). New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. *Computer Networks*, *173*, 107168. https://doi.org/10.1016/j.comnet.2020.107168

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

Jiang, Y., Hamer, J., Wang, C., Jiang, X., Kim, M., Song, Y., Xia, Y., Mohammed, N., Sadat, M. N., & Wang, S. (2018). SecureLR: Secure logistic regression model via a hybrid cryptographic protocol. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *16*(1), 113–123.

Jung Hee Cheon & Jinsu Kim. (2015). A Hybrid Scheme of Public-Key Encryption and Somewhat Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, *10*(5), 1052–1063. https://doi.org/10.1109/TIFS.2015.2398359

Khedr, A., Gulak, G., & Vaikuntanathan, V. (2016). SHIELD: Scalable Homomorphic Implementation of Encrypted Data-Classifiers. *IEEE Transactions on Computers*, *65*(9), 2848–2858. https://doi.org/10.1109/TC.2015.2500576

Kim, M., Harmanci, A. O., Bossuat, J.-P., Carpov, S., Cheon, J. H., Chillotti, I., Cho, W., Froelicher, D., Gama, N., Georgieva, M., & others. (2021). Ultrafast homomorphic encryption models enable secure outsourcing of genotype imputation. *Cell Systems*, *12*(11), 1108–1120.

Kim, S., Kim, J., Kim, M. J., Jung, W., Kim, J., Rhu, M., & Ahn, J. H. (2022). BTS: An accelerator for bootstrappable fully homomorphic encryption. *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 711–725.

Kim, S., Omori, M., Hayashi, T., Omori, T., Wang, L., & Ozawa, S. (2018). Privacy-Preserving Naive Bayes Classification Using Fully Homomorphic Encryption. In L. Cheng, A. C. S. Leung, & S. Ozawa (Eds.), *Neural Information Processing* (Vol. 11304, pp. 349–358). Springer International Publishing. https://doi.org/10.1007/978-3-030-04212-7_30

Kulkarni, V. Y., & Sinha, P. K. (2012). Pruning of random forest classifiers: A survey and future directions. *2012 International Conference on Data Science & Engineering (ICDSE)*, 64–68.

Kumar, A., & Lim, T. J. (2019). EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 289–294. https://doi.org/10.1109/WF-IoT.2019.8767194

Kwak, M., & Cho, Y. (2021). A Novel Video Steganography-Based Botnet Communication Model in Telegram SNS Messenger. *Symmetry*, *13*(1), 84. https://doi.org/10.3390/sym13010084

Li, J., Song, D., Chen, S., & Lu, X. (2012). A simple fully homomorphic encryption scheme available in cloud computing. *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, 214–217. https://doi.org/10.1109/CCIS.2012.6664399

Li, P., Li, J., Huang, Z., Gao, C.-Z., Chen, W.-B., & Chen, K. (2018). Privacy-preserving outsourced classification in cloud computing. *Cluster Computing*, *21*(1), 277–286. https://doi.org/10.1007/s10586-017-0849-9

Liu, L., Cao, Z., & Mao, C. (2018). A note on one outsourcing scheme for big data access control in cloud. *International Journal of Electronics and Information Engineering*, *9*(1), 29–35.

Mangayarkarasi, R., Vanitha, M., Subhashini, R., Deepa, M., & Angulakshmi, M. (2020). Intrusion Detection System using Random Forest. *Solid State Technology*, 9495–9511.

Mittal, D., Kaur, D., & Aggarwal, A. (2014). Secure data mining in cloud using homomorphic encryption. *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* 1–7.

Mohammed, S. J., & Taha, D. B. (2021). From cloud computing security towards homomorphic encryption: A comprehensive review. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 19*(4), 1152. https://doi.org/10.12928/telkomnika.v19i4.16875

Mukundan, R., Madria, S., & Linderman, M. (2014). Efficient integrity verification of replicated data in cloud using homomorphic encryption. *Distributed and Parallel Databases*, *32*(4), 507–534. https://doi.org/10.1007/s10619-014-7151-0

Park, H., Kim, P., Kim, H., Park, K.-W., & Lee, Y. (2018). Efficient machine learning over encrypted data with non-interactive communication. *Computer Standards & Interfaces*, *58,* 87–108. https://doi.org/10.1016/j.csi.2017.12.004

Podgorelec, B., Turkanović, M., & Karakatič, S. (2019). A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. *Sensors, 20*(1), 147.

Raj, K. J., Indu, I., & Anand, P. R. (2015). Homomorphic encryption with privacy protection for improved security in cloud networks. *International Journal of Applied Engineering Research, 10*(77), 455–459.

Sajay, K., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1–10.

Sgaglione, L., Coppolino, L., D'Antonio, S., Mazzeo, G., Romano, L., Cotroneo, D., & Scognamiglio, A. (2019). Privacy Preserving Intrusion Detection Via Homomorphic Encryption. *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE),* 321–326. https://doi.org/10.1109/WETICE.2019.00073

Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020a). BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset. *IEEE Access, 8,* 29575–29585. https://doi.org/10.1109/ACCESS.2020.2972627

Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020b). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access*, *8,* 29575–29585.

Sun, P. J. (2019). Privacy protection and data security in cloud computing: A survey, challenges, and solutions. *IEEE Access*, *7*, 147420–147452.

Teng, H., Liu, Y., Liu, A., Xiong, N. N., Cai, Z., Wang, T., & Liu, X. (2019). A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities. *Future Generation Computer Systems*, *94,* 351–367.

Teng, L., Li, H., Yin, S., & Sun, Y. (2020). A Modified Advanced Encryption Standard for Data Security. *Int. J. Netw. Secur.*, *22*(1), 112–117.

Viand, A., Jattke, P., & Hithnawi, A. (2021). Sok: Fully homomorphic encryption compilers. *2021 IEEE Symposium on Security and Privacy (SP)*, 1092–1108.

Vinutha, H. P., & Poornima, B. (2019). Analysis of NSL-KDD Dataset Using K-Means and Canopy Clustering Algorithms Based on Distance Metrics. In A. N. Krishna, K. C. Srikantaiah, & C. Naveena (Eds.), *Integrated Intelligent Computing, Communication and Security* (Vol. 771, pp. 193–200). Springer Singapore. https://doi.org/10.1007/978-981-10-8797-4_21

Wood, A., Shpilrain, V., Najarian, K., Mostashari, A., & Kahrobaei, D. (2018). Private-Key Fully Homomorphic Encryption for Private Classification. In J. H. Davenport, M. Kauers, G. Labahn, & J. Urban (Eds.), *Mathematical Software – ICMS 2018* (Vol. 10931, pp. 475–481). Springer International Publishing. https://doi.org/10.1007/978-3-319-96418-8_56

Xiang, G., Yu, B., & Zhu, P. (2012). A algorithm of fully homomorphic encryption. *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, 2030–2033. https://doi.org/10.1109/FSKD.2012.6234023

Yang, P., Xiong, N., & Ren, J. (2020). Data security and privacy protection for cloud storage: A survey. *IEEE Access*, *8*, 131723–131740.

Yasuda, M., Shimoyama, T., & Kogure, J. (2014). Secret computation of purchase history data using somewhat homomorphic encryption. *Pacific Journal of Mathematics for Industry*, *6*(1), 5. https://doi.org/10.1186/s40736-014-0005-x

Yu, X., Zhao, W., Huang, Y., Ren, J., & Tang, D. (2022). Privacy-Preserving Outsourced Logistic Regression on Encrypted Data from Homomorphic Encryption. *Security and Communication Networks*, *2022*, e1321198. https://doi.org/10.1155/2022/1321198

Zhang, L., Cai, Z., & Wang, X. (2016). Fakemask: A novel privacy preserving approach for smartphones. *IEEE Transactions on Network and Service Management*, *13*(2), 335–348.

Zhao, F., Li, C., & Liu, C. F. (2014). A cloud computing security solution based on fully homomorphic encryption. *16th International Conference on Advanced Communication Technology*, 485–488. https://doi.org/10.1109/ICACT.2014.6779008