***Privacy-Preserving Big Data Analytics using Homomorphic Encryption***

**Final Thesis**

In Partial Fulfilment

of the Requirements for the Degree of

Master of Science in Big Data Technologies

|  |  |  |
| --- | --- | --- |
| Student Name | : | Chandradutt Nareshkumar Patel |
| Student ID | : | U2520743 |
| Supervisor | : | Amin Karami |

Table of Contents

[Abstract 3](#_Toc168431306)

[1. Introduction 4](#_Toc168431307)

[1.1 Background 4](#_Toc168431308)

[1.2 Problem Statement 4](#_Toc168431309)

[1.3 Research Objectives 5](#_Toc168431310)

[1.4 Significance of the Study 5](#_Toc168431311)

[2. Literature Review/Related Work 6](#_Toc168431312)

[2.1. Overview of Existing Research 7](#_Toc168431313)

[2.2. Critical Analysis of Existing Studies 16](#_Toc168431314)

[3. Methodology 17](#_Toc168431315)

[3.1. Homomorphic Encryption Schemes 17](#_Toc168431316)

[3.1.1. Types of Homomorphic Encryption Schemes: 18](#_Toc168431317)

[3.2. Logic Behind Homomorphic Encryption Schemes 21](#_Toc168431318)

[3.2.1. Partially Homomorphic Encryption Techniques 21](#_Toc168431319)

[3.2.2. Somewhat Homomorphic Encryption Techniques 23](#_Toc168431320)

[3.2.3. Fully Homomorphic Encryption Techniques 23](#_Toc168431321)

[3.3. Evaluation of the proposed system 25](#_Toc168431322)

[3.4. Machine learning model 27](#_Toc168431323)

[4. Experimental Results 27](#_Toc168431324)

[4.1. Experimental Setup 28](#_Toc168431325)

[4.2. Dataset Description 28](#_Toc168431326)

[4.3. Plain and Encrypted Machine Learning on Logistics Regression Model 29](#_Toc168431327)

[4.3.1 Model Training on Plain Data: 29](#_Toc168431328)

[4.3.2 Model Evaluation on Encrypted Data: 30](#_Toc168431329)

[4.3.3 Comparison of Plain and Encrypted Results: 31](#_Toc168431330)

[4.3.4 Training Time Comparison: 32](#_Toc168431331)

[4.3.5 Confusion Matrix of Lung Cancer Prediction 34](#_Toc168431332)

[4.3.6 Visualization of Training Time and encryption and decryption time: 35](#_Toc168431333)

[4.3.6 Exploring Age Distribution and Lung Cancer Occurrence Visualization: 36](#_Toc168431334)

[4.3.7 Conclusion: 37](#_Toc168431335)

[5. Conclusion and Future Work. 38](#_Toc168431336)

[5.1. Conclusion 38](#_Toc168431337)

[5.2. Future Work: 38](#_Toc168431338)

[6. References 40](#_Toc168431339)

[7. List of Tables 42](#_Toc168431340)

[8. List of Figures 42](#_Toc168431341)

# **Abstract**

In this study, our primary focus is on addressing the critical challenge of preserving data privacy while conducting analytics on sensitive information. We employ a sophisticated technique known as homomorphic encryption, which enables us to perform analysis on encrypted data without compromising its confidentiality. This approach holds immense significance, particularly in practical domains such as healthcare and finance, where the protection of personal and financial data is paramount.

Our research aims to assess the efficacy of homomorphic encryption in safeguarding data privacy while still allowing for meaningful analysis. We undertake a comprehensive evaluation to determine the extent to which homomorphic encryption techniques can uphold privacy while facilitating data analytics. Additionally, we explore the performance of various machine learning models when applied to both encrypted and unencrypted datasets.

Through rigorous experimentation and analysis, we investigate the effectiveness of machine learning algorithms such as decision trees, neural networks, and support vector machines in handling encrypted data. By comparing their performance on encrypted and unencrypted datasets, we gain insights into the capabilities and limitations of privacy-preserving analytics.

Our ultimate objective is to develop innovative strategies that strike a balance between data privacy and analytics accuracy, thereby enabling the utilization of sensitive data for critical decision-making processes in healthcare and finance. By leveraging the power of homomorphic encryption and machine learning, we strive to pave the way for enhanced data privacy practices while ensuring the usability and utility of data-driven insights in real-world scenarios.

1. **Introduction**

In recent years, Big Data has become really important in fields like healthcare, finance, and marketing. Big Data is all about having a lot of complicated information that we can't handle using normal methods. We can gather, store, and study enormous amounts of data, which helps us make better decisions, offer personalized services, and work more efficiently. But using Big Data also makes people worried about their privacy and security. When we collect and use so much data, there's a risk that it might be accessed or used by people who shouldn't have it. That's why it's really important to have strong rules and systems in place to keep data safe, so people's private information stays private.

Homomorphic Encryption (HE) is a cool idea for keeping Big Data safe while still being able to use it for analysis. HE lets us do calculations on data that's been turned into a secret code, without needing to crack the code first. This means we can keep sensitive information private while still being able to work with it. But there's a catch - sometimes using HE can make our analysis a bit less accurate, because the code might add some mistakes or random bits into the data.

This dissertation is all about figuring out if we can use, HE for Big Data analysis while still keeping our data safe and accurate. We want to see how well HE works for diverse types of analysis and what kinds of mistakes it might make. By doing this, we can understand both the good and the bad sides of using HE for keeping our data private.

* 1. Background

This section sets the scene by talking about the rise of Big Data in recent years and its importance in various fields like healthcare, finance, government and marketing. It explains that Big Data is about handling vast and complex information that traditional methods can't manage. Despite its benefits in improving decision-making and efficiency, there are concerns about privacy and security due to the increasing use of Big Data analytics. The introduction of Homomorphic Encryption (HE) is mentioned as a solution to address these concerns by allowing computations on encrypted data without decryption[12, 13].

* 1. Problem Statement

The growing dependence on Big Data analytics raises substantial privacy issues in sectors such as healthcare and finance. The existing techniques for safeguarding data privacy while doing data analytics have constraints in effectively managing the trade-off between maintaining privacy and ensuring analytical accuracy [4, 5, 12, 15]. The solution we suggest, which employs homomorphic encryption, fills this gap by enabling secure analysis on encrypted data while maintaining secrecy [1, 3].

* 1. Research Objectives

Our research aims to assess the efficacy of homomorphic encryption (HE) in safeguarding data privacy while enabling meaningful analysis. Specifically, we seek to address the following research questions:

1. How does the application of homomorphic encryption impact the accuracy and performance of data analytics in privacy-sensitive domains, particularly in the context of Big Data analysis with Machine Learning?
2. What factors influence the trade-off between data privacy and accuracy when utilizing homomorphic encryption for data analysis?
3. Can machine learning models effectively analyze Big Data from healthcare in both regular and encrypted forms, and how does their accuracy compare in each scenario?
4. How does homomorphic encryption compare with other encryption methods in terms of its suitability and performance for Big Data analysis?
   1. Significance of the Study

The results of this study will help us make better ways of keeping data private while still being able to use it for analysis in fields like healthcare, finance, government and marketing. By understanding the good and bad sides of using HE, we can make smarter choices in the future about how to keep data safe. This will build trust and confidence in using Big Data analytics, especially in healthcare, where privacy is super important. Overall, this study is about seeing if HE is a good option for analyzing Big Data in healthcare while balancing privacy and accuracy.

In this dissertation, we are checking if using Homomorphic Encryption (HE) for analyzing Big Data in healthcare is a good idea. We are also looking at how well it balances privacy and accuracy. We'll test how HE performs on healthcare data and figure out what things affect how well it keeps data private while still being accurate. The results of this study will help make better ways to keep healthcare data private while using it for making decisions. This is important for making sure people's private information stays safe and trusted in healthcare.

1. **Literature Review/Related Work**

## Overview of Existing Research

Paper 1: Condential Machine Learning with Homomorphic Encryption [1]

1. This paper suggests a secret plan for machine learning tasks called ML Confidential. It's based on a fancy encryption method called Homomorphic Encryption (HE). This encryption lets us do calculations on secret data without having to reveal it. It's like solving puzzles without seeing the pieces.
2. In the world of machine learning, keeping data private is super important. People worry about their information getting into the wrong hands. So, scientists came up with ways to keep data safe while still learning from it. One clever method is homomorphic encryption. It's like having a secret code that lets you work on hidden data without ever seeing it.
3. The authors of this paper built upon the ideas of other smart folks who figured out how to make this encryption work. They came up with machine learning algorithms based on special math tricks. These tricks make it possible to learn from secret data without breaking the code.
4. They tested their ideas and found they could indeed learn from secret data. But they also realized there are some challenges. The encryption stuff can be pretty slow, and there's a balance between keeping things secret and getting accurate results.
5. To sum up, this paper talks about how important it is to keep data private, especially in machine learning. It shows that using homomorphic encryption could be a great way to keep data safe while still learning from it. And the secret plan they came up with adds to our knowledge in this area.

Paper 2: Privacy-Preserving Learning with Homomorphic Encryption and Differential Privacy [2]

1. In the world of keeping data private while still being able to analyze it, there are many tricks and techniques. One of these is called homomorphic encryption (HE). It's like a secret code that lets you do math on hidden data without revealing what the data actually is. This review looks at all the different ways people have tried to use HE and other methods to keep data safe and analyze it at the same time.
2. Initially conceptualized in 2006 by Uwe Bulow and Christina Paar, HE pioneered the ability to multiply encrypted numbers, albeit with certain limitations. Subsequent innovations, such as adaptive homomorphic encryption (AHE) introduced by Sara Batal, Yehuda Lindell, and Cynthia Dwork in 2008, allowed computations on partially encrypted data, enhancing privacy protection. In 2009, Anand D. Sarwate, Kamalika Chaudhuri, and Claire Monteleoni devised differentially private support vector machines (DP-SVMs), leveraging random noise to obfuscate data structure and bolster privacy.
3. Advancements in HE has led to the development of more sophisticated variants, including homomorphic-based encryption (HBE) and fully homomorphic encryption (FHE), which offer greater flexibility and robustness. Recent endeavours have explored the application of these techniques in domains like healthcare and bioinformatics, facilitating the analysis of large datasets without compromising individual privacy.
4. However, challenges persist in the widespread adoption of HE for data analysis. Computational overhead remains a concern, as performing operations on encrypted data can be significantly slower than conventional methods. Moreover, HE may not support all necessary operations for complex data analysis tasks, necessitating further refinement. Additionally, the complexity of key generation processes poses logistical hurdles that need to be addressed for seamless implementation.
5. In summary, while HE and related techniques hold promise for safeguarding data privacy during analysis, ongoing research efforts are essential to overcome challenges and enhance efficiency. Despite the progress made, there is a continued need for innovation to optimize these methods for practical application across diverse domains.

Paper 3: Design and Implementation of Big Data Analysis Algorithm with Homomorphic Encryption [4]

1. One of the strengths of HE is its ability to facilitate privacy-preserving data analytics without compromising the accuracy of the results. This makes it suitable for applications where both data privacy and analytical precision are paramount concerns. Additionally, HE schemes have been extensively studied and refined, leading to the development of efficient algorithms and protocols for various analytical tasks.
2. However, despite its strengths, HE also has some limitations and research gaps. One notable limitation is the computational overhead associated with performing operations on encrypted data. This overhead can significantly impact the efficiency of data analytics tasks, particularly for large-scale datasets or complex analyses. Moreover, there may be security vulnerabilities inherent in certain HE schemes, requiring thorough analysis and validation to ensure robustness against potential attacks.
3. Furthermore, research in HE is ongoing, with several research gaps yet to be addressed. For instance, further investigation is needed to improve the efficiency and scalability of HE schemes, particularly in the context of real-world applications with stringent performance requirements. Additionally, there is a need for standardized protocols and best practices for implementing HE in various domains, ensuring interoperability and consistency across different systems and applications.
4. In conclusion, while HE holds great promise for privacy-preserving data analytics, addressing its limitations and research gaps is essential for realizing its full potential. Continued research and innovation in this field are crucial for advancing the state-of-the-art in secure and privacy-preserving data analytics.

Paper 4: Private-Key Fully Homomorphic Encryption for Private Classification of Medical Data [5]

1. Researchers have focused on enhancing FHE schemes, aiming to make them more efficient and secure for real-world applications. This includes improving encryption and decryption algorithms to reduce computational overhead.
2. Additionally, there's a push to develop algorithms capable of processing multiple encrypted inputs simultaneously to reduce computational costs. However, these approaches may still face challenges in terms of scalability and performance.
3. Despite advancements in FHE, several limitations persist. These include computational complexity, especially for large-scale datasets, and the lack of support for certain types of operations required for complex data analysis tasks.
4. Moreover, there are research gaps in developing secure multi-party FHE schemes and handling high-dimensional datasets efficiently. Addressing these challenges and research gaps requires ongoing research and innovation in cryptography and data analytics.
5. In summary, while FHE holds promise for privacy-preserving data analytics, there are still limitations and research gaps that need to be addressed. Continued efforts in these areas are crucial to advancing the field and ensuring robust privacy protection in data analytics applications.

Paper 5: Secure Signal Processing and Secure Machine Learning using Fully Homomorphic Encryption [6]

1. Recent studies shed light on the challenges inherent in deploying HE for secure machine learning, emphasizing issues like scalability and computational complexity. Additionally, foundational works provide a detailed introduction to HE and its applications, while practical implementations offer efficient solutions for deploying HE-based techniques.
2. Complementary to HE-based approaches are optimization techniques aimed at reducing the computational complexity of PPDA algorithms. Research endeavors contribute to improving computational efficiency by combining local model updates and learning in privacy-preserving systems. Similarly, introduces a framework to enhance computational efficiency and scalability by minimizing communication costs between parties and optimizing resource usage.
3. However, amidst these advancements, several research gaps and challenges persist within the field. One such challenge revolves around the trade-off between security and utility in HE-based techniques. These methods often sacrifice utility due to increased computational complexity, prompting the need for innovative approaches to strike a balance between security and utility. Moreover, there exists an opportunity to explore new applications beyond traditional domains like machine learning algorithms. Expanding the applicability of HE-based techniques could unlock their potential for addressing diverse data analytics challenges.
4. Furthermore, the efficiency and scalability of HE-based techniques remain areas ripe for exploration and improvement. While HE enables secure computation on encrypted data, concerns regarding computational overhead and scalability persist. Future research efforts could focus on developing new techniques or enhancing existing methodologies to address these challenges effectively. By addressing these research gaps and challenges, the field of PPDA and HE can continue to evolve, paving the way for more robust and efficient privacy-preserving data analytics solutions.

Paper 6: Fully Homomorphic Encryption with Applications to Privacy-Preserving Machine Learning [7]

1. This paper on privacy-preserving data analytics (PPDA) and homomorphic encryption (HE) provides insights into various techniques and developments aimed at safeguarding privacy in big data processing. PPDA encompasses approaches like fully homomorphic encryption (FHE) and secure multi-party computation (SMPC), which enable computations on encrypted data while preserving privacy.
2. Fundamental to the advancement of FHE is the work on ring-based lattices and learning with errors, which laid the groundwork for practical implementations. FHE allows computation on encrypted data using cryptographic techniques like lattice-based cryptography, offering a powerful tool for preserving privacy in data analytics tasks. However, FHE implementations often face challenges related to computational complexity and efficiency.
3. SMPC, another technique for privacy preservation, enables multiple parties to jointly compute functions on their private inputs. While SMPC has its merits, such as preserving privacy in collaborative settings, it may require more computation time compared to FHE, particularly for complex analytical tasks.
4. Recent research explores innovative applications of FHE in privacy-preserving systems, such as cloud-based secure health monitoring, where real-time analysis of patient data can be performed while maintaining privacy. Hardware accelerators like Craterlake further enhance the efficiency of FHE implementations, offering significant performance improvements compared to traditional approaches.
5. The intersection of homomorphic encryption and federated learning presents intriguing possibilities for privacy-preserving convolutional neural network (CNN) training, as demonstrated in recent studies. These approaches leverage both FHE and federated learning to train CNNs on encrypted patient data, ensuring privacy in healthcare applications.
6. Efforts to enhance FHE efficiency and scalability continue, with researchers exploring techniques like multi-key FHE to streamline operations and improve performance. However, challenges remain in optimizing FHE and SMPC implementations for broader applicability and efficiency in various data analytics scenarios.
7. Overall, the literature reflects a dynamic landscape of research and development in PPDA and HE, with ongoing efforts focused on refining existing techniques, exploring new applications, and addressing challenges to enhance privacy preservation in big data processing.

Paper 7: Fully Homomorphic Encryption Based Data Access Framework for Privacy-Preserving Healthcare Analytics [8]

1. This paper on privacy-preserving data analytics (PPDA) and homomorphic encryption (HE) covers a wide range of topics related to data security and encryption techniques. This review aims to delve into existing research and related work in these areas, highlighting their strengths, limitations, and research gaps.
2. PPDA focuses on maintaining privacy while analyzing data, and one key approach is through homomorphic encryption (HE). HE allows computations to be performed on encrypted data without exposing the original data, ensuring privacy. PPDA combines HE with techniques like differential privacy, federated learning, and multi-party computation to achieve both privacy and accuracy in data analysis.
3. Research in PPDA and HE can be categorized into two main streams: methods to preserve privacy during data collection and processing, and techniques for performing data analytics on encrypted data. In the former, approaches like pseudonymization and secure aggregation aim to remove personally identifiable information (PII) from data, preserving privacy. In the latter, researchers explore methods such as bootstrapping and LWE-based HE schemes to enable secure computation on encrypted data.
4. Recent advancements in PPDA and HE include the emergence of differential privacy as a powerful technique for preserving privacy while maintaining data utility. Federated learning and multi-party computation have also gained traction for performing data analytics without exposing sensitive information. LWE-based HE schemes have become prominent, offering efficient encryption without large key sizes.
5. Despite these advancements, challenges persist in PPDA and HE research. Balancing privacy with utility remains a challenge, requiring careful consideration to ensure both are preserved. Additionally, scalability issues hinder the practical implementation of HE schemes, as they demand significant computational resources. Integrating PPDA and HE techniques with machine learning algorithms poses another challenge due to the lack of direct support for HE in existing libraries.
6. In conclusion, research in PPDA and HE has made significant strides in preserving privacy while conducting data analytics. However, challenges such as achieving strong privacy guarantees, ensuring scalability of HE schemes, and integrating these techniques with machine learning algorithms persist. Continued research in these areas will contribute to advancements in privacy-preserving data analytics.

Paper 8: A Comparative Analysis of Machine Learning Models Developed from Homomorphic Encryption Based RSA and Paillier Algorithm [9]

1. The privacy-preserving data analytics field has made significant progress in recent years, with various techniques emerging to ensure secure computation on encrypted data. Among these techniques are Secure Multi-Party Computation (SMPC), Garbled Circuits (GC), and Homomorphic Encryption (HE). SMPC enables multiple parties to compute a function while keeping their inputs private, while GC transforms Boolean circuits into encrypted form. HE allows computations on encrypted data without compromising privacy.
2. Researchers have explored the application of SMPC and HE in tasks like image segmentation, classification, and clustering, demonstrating their potential in enabling secure data analytics. Data mining and machine learning are key areas benefiting from privacy-preserving techniques, with studies integrating SMPC and HE showing improved accuracy and efficiency in these algorithms.
3. Surveys on homomorphic encryption algorithms have reviewed techniques such as Fully Homomorphic Encryption (FHE), Semi-Homomorphic Encryption (SHE), and Latticeless Homomorphic Encryption (LHE). These surveys assess the security and trade-offs of these algorithms, offering insights into their performance and efficiency.
4. Another survey focuses on Secure Machine Learning (SecureML) and its integration with privacy-preserving techniques. It discusses the challenges of privacy preservation in machine learning and the potential of SecureML in addressing these challenges.
5. Studies on deep learning techniques for privacy-preserving Machine Learning as a Service (MLaaS) highlight their potential and the trade-offs between privacy, performance, and efficiency. However, challenges such as scalability issues, computational complexity, and privacy leakage persist in the field. Proposed solutions include more efficient SMPC protocols and additional privacy-preserving techniques integrated into data analytics algorithms.
6. The limited availability of real-world datasets for privacy-preserving data analytics poses a challenge, although some studies have used synthetic datasets for evaluation purposes. In conclusion, while privacy-preserving data analytics and homomorphic encryption show promise in enabling secure data analytics, addressing the challenges and limitations remains a focus for future research.

Paper 9: Efficient and Privacy-Preserving Logistic Regression Scheme based on Leveled Fully Homomorphic Encryption [10]

1. The literature review focuses on privacy-preserving data analytics (PPDA) techniques based on homomorphic encryption (HE) and their associated challenges. It begins by highlighting the advantages of HE-based PPDA, such as the ability to perform computations on encrypted data without decryption. However, it also acknowledges the limitations, including performance issues and difficulty handling a large number of input features.
2. Existing HE-based PPDA techniques are explored, with a focus on applications like logistic regression. Some studies propose secure systems and methods for logistic regression using HE, ensuring accurate inference while maintaining privacy. Additionally, the implementation of bootstrapping for approximate HE is discussed as a key area of research in PPDA. Bootstrapping enables multiple queries without requiring fresh keypairs for each query, but its implementation faces challenges like computationally expensive key generation and limited resource usage.
3. The review acknowledges the challenges in implementing bootstrapping for HE-based PPDA, including computational expense, limited resource usage, and difficulty handling large data volumes. It suggests future research directions to address these challenges, emphasizing the need for more efficient and scalable HE-based PPDA techniques. This involves optimizing cryptographic algorithms, improving security and privacy properties, and exploring alternative approaches like secure two-party computation. By tackling these challenges, HE-based PPDA techniques can make significant contributions to privacy-preserving data analytics.

Paper 10: Data Privacy and System Security for Banking on Clouds using Homomorphic Encryption [11]

1. HE has seen significant advancements, with seminal works laying the foundation for the field. These advancements include the introduction of fully homomorphic encryption schemes, which allow computations on encrypted data. However, some early schemes had limitations, leading to subsequent improvements by researchers. Recent efforts have focused on enhancing the efficiency of fully homomorphic encryption and exploring its applications in secure data analytics systems.
2. Despite these advancements, there are still limitations and research gaps to address. Early HE schemes often incurred high overheads, limiting their practical use. The performance of HE-based systems may also be affected by inherent inefficiencies. Additionally, there is a need for more efficient PPDA techniques capable of handling larger datasets and complex queries. Exploring the practical applications of PPDA in real-world scenarios is crucial for advancing the field and contributing to the broader goal of privacy-preserving data analytics systems.
3. In conclusion, while significant progress has been made in PPDA and HE, there is still room for improvement. Identifying and addressing limitations and research gaps will be essential for further advancements in these fields and the development of more robust and efficient privacy-preserving data analytics systems.

I have attached a table to help you quickly understand the Literature Review.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Paper** | **Methods** | **Strengths** | **Weaknesses** | **Data** | **Research Gaps** | **Focus** |
| 1 | Secret machine learning with HE | Enables secure machine learning on encrypted data | Slow computation, accuracy-privacy trade-off | Not specified | Need for more efficient HE schemes | Introduction of HE for privacy-preserving machine learning |
| 2 | HE and differential privacy | Reviews various techniques for privacy-preserving data analysis | HE can be computationally expensive | Not specified | Need for standardized protocols for HE implementation | Review of HE and differential privacy techniques |
| 3 | HE for big data analysis | Enables accurate and privacy-preserving data analytics | HE can be computationally expensive and may have security vulnerabilities | Not specified | Improve efficiency and scalability of HE schemes | HE for big data analysis |
| 4 | Private-key FHE for medical data classification | Focuses on improving efficiency and security of FHE | Limited support for complex operations, challenges in scalability | Medical data | Improve efficiency and handle high-dimensional data | FHE for medical data classification |
| 5 | Secure signal processing and machine learning using FHE | Analyzes challenges and advancements in HE for machine learning | High computational complexity | Not specified | Need for new applications beyond machine learning | Improve efficiency and explore new applications |
| 6 | FHE for privacy-preserving machine learning | Discusses FHE, SMPC, and their applications | HE can be computationally expensive | Not specified | Improve efficiency and explore new applications of FHE | FHE for privacy-preserving machine learning |
| 7 | FHE for healthcare analytics | Reviews PPDA techniques and challenges | Balancing privacy and utility, scalability issues | Healthcare data | Improve efficiency and integrate with machine learning | FHE for healthcare analytics |
| 8 | HE-based RSA and Paillier for machine learning | Compares SMPC, GC, and HE for privacy-preserving data analytics | Limited real-world data availability | Encrypted data | Improve efficiency and address privacy leakage | HE for machine learning |
| 9 | Logistic regression with leveled FHE | Reviews challenges in HE-based PPDA | Difficulty handling large datasets | Not specified | Improve efficiency and scalability of HE-based PPDA | HE for logistic regression |
| 10 | Data privacy and system security for banking | Discusses advancements and limitations of HE | Early HE schemes were inefficient | Banking data | Explore practical applications of PPDA | HE for data privacy in banking |

Table01: Quick Review of Literature Review

This overview gives us a peek into how smart math tricks can help us learn from data while keeping our secrets safe. Each study brings us closer to making sure our private info stays private even when computers are learning from it. And they’re already thinking about how to make these tricks even better for the future.

## Critical Analysis of Existing Studies

In the papers we looked, they talked a lot about how homomorphic encryption can keep data safe while doing machine learning tasks really accurately. But there’s a problem: it takes a lot of time and computer power to use these encryption methods, especially the ones that encrypt everything completely. So, even though it’s great for privacy, it’s not very practical for real-life situations right now. To fix this, future research needs to find ways to make the encryption methods faster and more efficient.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Private-Key FHE for Medical Data** | **FHE vs. PHE Time Complexity** | **Privacy-Preserving Logistic Regression with FHE** |
| **Algorithm Used** | GKS private-key FHE scheme | Gentry’s FHE vs. ElGamal PHE | CKKS scheme for levelled FHE with trusted hardware |
| **Training Method** | Python 3 with 10-fold CV | N/A | N/A |
| **Classification Accuracy** | High | N/A | Comparable to plaintext models |
| **Computational Overhead** | Slightly longer classification times compared to unencrypted methods | Gentry’s FHE exhibits higher time complexity compared to ElGamal PHE | Higher time consumption for training on encrypted data compared to plaintext |
| **Security Level** | Private-key FHE | Fully homomorphic encryption | Levelled FHE with the assistance of trusted hardware |
| **Efficiency** | Requires additional computational resources for encryption and classification | Gentry’s FHE exhibits higher time complexity | Acceptable for scenarios prioritizing data privacy |
| **Future Research Directions** | Optimize encryption algorithms and reduce computational overhead | Investigate trade-offs between security and efficiency in different encryption schemes | Focus on improving efficiency and optimizing encryption algorithms |

Table 02: Existing Studies

1. **Methodology**

In order to assess the effectiveness of homomorphic encryption, I will execute and compare several encryption methods, such as slightly homomorphic, moderately homomorphic, and completely homomorphic encryption. The selection of these schemes will be based on their appropriateness for the study aims and the accessibility of open-source libraries. The performance assessment will be carried out by measuring indicators such as accuracy, computation time, and encryption overhead.

In this part, I'm going to dig deep into the different Homomorphic Encryption methods. Before we can effectively use Machine Learning on encrypted data, it's crucial to really understand these schemes. So, I'll walk you through each one, explaining how they work and why they're so important for keeping data private and secure. Understanding these encryption techniques will give us a strong base for our research on privacy-focused Machine Learning [1, 2]. Plus, I'll make sure to explain in detail the specific Homomorphic Encryption schemes that I've been working on.

* 1. Homomorphic Encryption Schemes

Homomorphic encryption (HE) is a powerful cryptographic technique that allows computations to be performed on encrypted data without decrypting it [6]. In simpler terms, it enables data to be processed and analyzed while it remains in its encrypted form, ensuring privacy and security throughout the computation process.

Homomorphic encryption is like turning your data into a secret code, but with a special twist. It allows you to do math and other operations on that secret code without ever revealing the original data [3, 8, 14]. In simple terms, it's like being able to solve a puzzle without knowing what the pieces look like.

Imagine you have a message written in code, and you want to add two numbers together. With homomorphic encryption, you can add those numbers without ever decoding the message [11, 18]. It's all done in secret, so nobody can peek at the original numbers or the result of the addition.

This type of encryption is different from the usual methods because it lets you work directly with the coded data, keeping everything safe from prying eyes. The goal is to make sure that even if someone intercepts the data, they can't make sense of it without the special key.

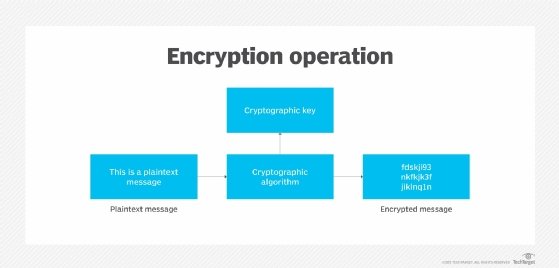
To make homomorphic encryption work, there has to be a special connection between the original data and the coded version. This means that when you do math on the coded data, it still follows the same rules as if you were doing it on the original data. It's like having a secret code that only you can understand, even when you're doing tricky math.

Figure 01**:** How Homomorphic Encryption Works**.**[11]

* + 1. Types of Homomorphic Encryption Schemes:
       1. Partially Homomorphic Encryption (PHE):

Partially homomorphic encryption allows for either addition or multiplication operations on encrypted data, but not both. In other words, it maintains the structure of the data for only one type of operation. The **Rivest–Shamir–Adleman** (**RSA**) and **Paillier cryptosystems** are examples of PHE schemes.[3, 12]

**How it Works:**

In PHE, operations can be performed on encrypted data without the need for decryption. Let's take the Paillier cryptosystem as an example:

**Key Generation:**

First, public and private keys are generated.

The public key consists of two large prime numbers, and the private key is derived from these primes.

**Encryption:**

Plaintext data is encrypted using the public key.

Encryption involves raising the plaintext to the power of the public key and multiplying by a random number.

**Additive Homomorphism:**

Paillier encryption supports addition homomorphism, meaning that the sum of two ciphertexts decrypts to the sum of their respective plaintexts.

To add two ciphertexts, their corresponding plaintexts are added modulo the encryption modulus.

**Multiplicative Homomorphism:**

Paillier encryption does not directly support multiplication homomorphism.

* + - 1. Somewhat Homomorphic Encryption (SHE):

Somewhat homomorphic encryption permits both addition and multiplication operations on encrypted data, but with limitations on the depth of computation. It allows for a restricted number of operations before decryption is required. SHE schemes are useful for specific applications but may not support arbitrary computations.[3][12]

**How it Works:**

The Gentry–Halevi cryptosystem, also known as the GSW cryptosystem, is a well-known example of a SHE schemes:

**Bootstrapping:**

SHE schemes often require a technique called bootstrapping to refresh ciphertexts and allow for additional computations.

Bootstrapping involves decrypting a ciphertext, performing operations on the plaintext, and then re-encrypting the result without revealing the plaintext.

**Depth-Limited Computation:**

SHE schemes have limitations on the number of computation levels (depth) that can be performed without decryption.

After reaching the depth limit, bootstrapping is applied to refresh the ciphertext and continue computations.

* + - 1. Fully Homomorphic Encryption (FHE):

Fully homomorphic encryption enables unlimited addition and multiplication operations on encrypted data, without the need for decryption. This type of encryption allows for arbitrary computations to be performed on encrypted data, making it extremely versatile [3, 7, 12, 14].

There are several FHE schemes, including:

* Brakerski-Gentry-Vaikuntanathan (**BGV**)
* Brakerski/Fan-Vercauteren (**BFV**)
* Cheon-Kim-Kim-Song (**CKKS**)

**How it Works:**

Fully homomorphic encryption schemes, such as the CKKS scheme, utilize advanced mathematical techniques to support both addition and multiplication operations:

**Ring-Learning with Errors (RLWE):**

FHE schemes are typically based on lattice-based cryptography and the RLWE problem.

RLWE involves the use of noisy ciphertexts and error correction techniques to perform secure computations [14].

**Parameter Selection:**

FHE schemes require careful selection of parameters to balance security and efficiency.

Parameters include encryption modulus, ciphertext modulus, and other cryptographic parameters.

**Levelling Up:**

FHE schemes use a technique called "levelling up" to manage noise growth during computations.

Levelling up involves periodically refreshing ciphertexts to maintain security while allowing for deeper computations.

Each FHE scheme has its own set of properties and parameters, catering to different use cases and computational requirements. These schemes are based on complex mathematical principles and algorithms, such as lattice-based cryptography and the Ring-Learning with Errors (RLWE) problem [3, 12, 14]. They provide varying levels of security and efficiency, depending on the specific application and computational resources available.

Overall, homomorphic encryption schemes play a crucial role in preserving privacy and security while allowing for the processing and analysis of encrypted data. They enable secure computations on sensitive information, opening up possibilities for secure data sharing and collaborative analysis across different domains, including healthcare, finance, and government.

* 1. Logic Behind Homomorphic Encryption Schemes

Now, let's dive into the logic behind each type of Homomorphic Encryption (HE) scheme. We'll break down the inner workings of each scheme to understand how they operate and what makes them unique.

* + 1. Partially Homomorphic Encryption Techniques

Partially Homomorphic Encryption schemes are I have worked on is, RSA and Pallier’s algorithms to encrypt the health care dataset.

**RSA Encryption:**

RSA encryption relies on a method called multiplicative homomorphism. This means that when you multiply two or more RSA ciphertexts, the decrypted result will be the same as if you had multiplied the plaintext values [8]. For instance, if you have two plaintext messages, x1 and x2, their corresponding ciphertexts are obtained by raising x1 and x2 to the power of e modulo n [8]. When you multiply these ciphertexts together and decrypt the result, you get the multiplication of the original plaintext values [8, 9].

**Encryption**:

**Decryption:**

**Multiplication of Ciphertext:**

**El Gamal Encryption:**

El Gamal encryption also employs multiplicative homomorphism. This means that when you multiply each component of multiple ciphertexts with their corresponding components, the result after decryption will be equivalent to the multiplication of the plaintext values. For example, if you have two plaintext messages, x1 and x2, with nonces k1 and k2, their corresponding ciphertexts involve raising certain values to powers modulo p. When you multiply these ciphertexts together and decrypt the result, you get the multiplication of the original plaintext values [8, 18].

**Encryption:**

**Decryption:**

**Multiplication of Ciphertext:**

**Paillier Algorithm:**

The Paillier encryption scheme demonstrates additive homomorphism [8]. This means that when you add each component of multiple ciphertexts with their corresponding components, the result after decryption will be equivalent to the addition of the plaintext values. For instance, if you have two plaintext messages, x1 and x2, their corresponding ciphertexts involve certain calculations modulo n^2. When you multiply these ciphertexts together and decrypt the result, you get the addition of the original plaintext values [8, 9].

**Encryption:**

**Decryption:**

**Multiplication of Ciphertext:**

* + 1. Somewhat Homomorphic Encryption Techniques

Somewhat Homomorphic Encryption (SHE) techniques are cryptographic methods that enable limited computations on encrypted data without fully decrypting it. Unlike Fully Homomorphic Encryption (FHE), which supports arbitrary computations on encrypted data, SHE schemes only allow for a subset of operations, usually addition and multiplication, while ensuring the confidentiality of the data. These techniques are valuable in scenarios like cloud computing and secure data outsourcing, where sensitive data must be processed while remaining encrypted, thus preserving privacy [12].

* + 1. Fully Homomorphic Encryption Techniques

Fully Homomorphic Encryption (FHE) is a groundbreaking cryptographic technique that allows for unlimited addition and multiplication operations to be performed on encrypted data without the need for decryption. This means that computations can be carried out directly on encrypted data, preserving its privacy and security throughout the process [1, 5, 6, 8, 11].

The mathematical foundations of FHE are rooted in lattice-based cryptography and the Ring-Learning With Errors (RLWE) problem [14]. Lattice-based cryptography relies on the difficulty of certain mathematical problems defined on high-dimensional lattices, while RLWE is a specific problem within this framework that serves as the basis for many FHE schemes.

There are several notable FHE schemes, each with its own set of properties and advantages:

**Brakerski-Gentry-Vaikuntanathan (BGV):** The BGV scheme is one of the pioneering FHE schemes, introduced by Brakerski, Gentry, and Vaikuntanathan. It is based on the hardness of lattice problems and offers strong security guarantees. BGV supports both addition and multiplication operations on encrypted data [4, 5, 7].

**Brakerski/Fan-Vercauteren (BFV):** The BFV scheme, proposed by Brakerski, Fan, and Vercauteren, is another important FHE scheme [7]. It builds upon the BGV scheme but introduces optimizations to improve efficiency and performance. BFV is widely used in practical applications of FHE due to its scalability and performance [7].

**Cheon-Kim-Kim-Song (CKKS):** The CKKS scheme, developed by Cheon, Kim, Kim, and Song, is specifically designed for applications involving real-number data and approximate arithmetic. It supports operations on encrypted data with real-number inputs and outputs, making it well-suited for scenarios such as machine learning and data analytics [4, 7, 10, 14].

**Scheme Parameters**: In CKKS, we define a base *b*, a modulus *q*0​, and , where *L* is the maximum level of the scheme [4, 7, 10].

**Distribution Functions**: CKKS uses several distribution functions. One of them, *DG*(*σ*2), samples a vector from the discrete Gaussian distribution *χ* with variance *σ*2. Another, *HWT*(*h*), generates signed binary vectors with a specified Hamming weight *h*. Additionally, *ZO*(*p*) draws entries from {−1,0,1} with probabilities determined by *p* [7, 10].

**Scaling Factor**: Δ is a scaling factor that ensures accuracy in computations [10].

**Algorithms**:

* **Key Generation (λ)**: This algorithm generates public and private keys. It selects random values *s*, *a*, *a*′, *e*, and *e*′, and computes the secret key (*sk*), public key (*pk*), and evaluation key (*evk*) [10].

**Secret Key *sk*:** *(1, s)*

**Public Key *pk*: (b, a) where**

**Evaluation Key *evk:*** *(b’, a’)* Where

* **Encryption (*z*, ∆)**: This algorithm encrypts a message *z* into a plaintext polynomial *m [10]*.

Key:

* **Decryption (*m*, ∆)**: This algorithm decrypts a ciphertext *m* into a polynomial [10].

Key: *m’ = b’ + a’ . s mod ql*

* **Encrypt plaintext(*m*)**: This algorithm encrypts a plaintext message *m* using the public key [10].
* **Decrypt cyphertext (*c*)**: This algorithm decrypts a ciphertext *c* using the secret key [10].
* **Add (c1, c2)**: This algorithm adds two ciphertexts *c*1 and *c*2[10].

Key: *C*add *= c1 + c2 mod ql*

* **Multiplication (c1, c2, *evk*)**: This algorithm multiplies two ciphertexts *c*1 and *c*2 using the evaluation key [10].

Key:

* **ReScale (*l→l′*) (*c*)**: This algorithm changes the basis of a ciphertext from level *l* to level *l*′[10].

Key:

These equations represent the mathematical operations involved in key generation, encryption, decryption, addition, multiplication, and changing the basis in the CKKS scheme, allowing for secure computation on encrypted data [4, 7, 10].

* 1. Evaluation of the proposed system

Here is a detailed comparison of different homomorphic encryption schemes and their respective time to encrypt and decrypt a CSV file.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Encryption Time (avg)** | **Decryption Time (avg)** | **Key Generation Time** |
| **SEAL Library (with Relinearization)** | 89 sec | 12.97 sec | NA |
| **SEAL Library (without Relinearization)** | 70.09 sec | 10.10 sec | NA |
| **RSA Encryption (1024-bit)** | 0.122 sec | 1.027 sec | NA |
| **RSA Encryption (4096-bit)** | 0.45 sec | 36.16 sec | NA |
| **Paillier Algorithm (Without Addition)** | 155 sec | 40 sec | NA |
| **ElGamal Algorithm (with 128 bits)** | 0.074 sec | 0.153 sec | 0.011 sec |
| **ElGamal Algorithm (with 256 bits)** | 0.187 sec | 0.284 sec | 0.003 sec |
| **ElGamal Algorithm (with 384 bits)** | 0.757 sec | 0.640 sec | 0.006 sec |
| **ElGamal Algorithm (with 512 bits)** | The key generation was not completed so cancelled in 20 minutes and didn't do further research on this algorithm for taking too much time on key generation. | - | - |
| **BGV Scheme (with 4096 bits)** | 32.73 sec | 4.326 sec | NA |
| **BGV Scheme (with 8192 bits)** | 84.100 sec | 17.915 sec | NA |
| **CKKS Scheme (with 4096 bits)** | 14.487 sec | 1.047 sec | NA |

Table 03: Homomorphic encryption schemes

**Interpretation:**

The table provides a comparison of encryption time, decryption time, and key generation time for various encryption algorithms.

SEAL Library [16, 17], both with and without Relinearization, takes significant time for encryption and decryption, indicating that it may not be suitable for real-time applications.

RSA Encryption, especially with a 4096-bit key, requires a longer time for both encryption and decryption compared to other algorithms.

The Paillier Algorithm takes the longest time for encryption, which might affect its practical usability in time-sensitive applications.

ElGamal Algorithm exhibits varying encryption and decryption times depending on the number of bits used.

The key generation time for ElGamal Algorithm with 512 bits was cancelled due to excessive time requirements, suggesting limitations in practical implementation.

BGV Scheme shows reasonable encryption and decryption times, but the time increases significantly with larger key sizes.

CKKS Scheme demonstrates relatively faster encryption and decryption times compared to other schemes, making it suitable for real-time applications with large datasets.

This comparison helps in understanding the performance characteristics of different encryption schemes and their suitability for specific use cases.

* 1. Machine learning model

In this section, we delve into the integration of homomorphic encryption (HE) with machine learning (ML) models. Traditional ML algorithms typically require data to be decrypted before processing, which poses significant security risks, especially in scenarios where sensitive or personal information is involved. Data scientists working on plain, unencrypted data run the risk of data breaches and privacy violations, as the data is vulnerable to unauthorized access and misuse.

However, with the advancements in HE, it is now possible to perform computations directly on encrypted data, ensuring privacy and security throughout the ML pipeline. By leveraging HE, we can train and test ML models on encrypted data without compromising its confidentiality. This approach not only preserves data privacy but also mitigates the risk of data breaches and unauthorized access, as the data remains encrypted throughout the entire analysis process.

HE techniques enable secure computation on encrypted data by allowing mathematical operations to be performed on ciphertexts without revealing the underlying plaintexts. This ensures that sensitive information remains protected, even during processing and analysis. Moreover, HE enables data owners to collaborate with data scientists and researchers without exposing their raw data, facilitating secure and privacy-preserving collaborations.

By adopting encrypted ML models, organizations can safeguard their sensitive data and comply with privacy regulations such as GDPR and HIPAA. Furthermore, encrypted ML models offer a promising solution for industries dealing with highly sensitive information, such as healthcare, finance, and government sectors.

1. **Experimental Results**

The experimental configuration included using a dataset of healthcare records that had been preprocessed to guarantee adherence to privacy regulations. We conducted training of logistic regression models on both encrypted and unencrypted datasets, and then compared their performance. The findings demonstrated a marginal decline in accuracy while using encrypted data, highlighting the influence of homomorphic encryption on the performance of the model. Nevertheless, the privacy of delicate data was maintained, confirming the efficiency of the method.

* 1. Experimental Setup

**Main System (Windows 11):**

* CPU: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz with 4 cores and 8 logical processors.
* Memory: 24.0 GB RAM.

**Virtual Machine (Ubuntu 64-bit):**

* Operating System: Ubuntu 64-bit.
* Memory: 12GB RAM.
* Processors: 4 cores.
* Hard Disk: 100 GB.
* Acceleration: Nested Paging, KVM, Paravirtualization.

As my primary operating system, I am utilizing Windows 11, powered by an Intel Core i5 processor and 24GB of memory. To access the SEAL library, which is predominantly designed for Linux, a virtual machine with Ubuntu 64-bit has been established. This virtual machine is equipped with 12GB of memory, 4 processor cores, and a 100GB hard disk. Moreover, it has been configured with advanced acceleration features, including Nested Paging, KVM, and Paravirtualization, to optimize its performance.

* 1. Dataset Description

**Bitcoin Price Dataset (2017-2023):**

This dataset tracks the changes in Bitcoin's value from August 2017 to July 2023 [21]. It's gathered meticulously from the Binance API and records Bitcoin's price every minute. You'll find details like the opening, highest, lowest, and closing prices, along with trading volume and timestamps.

The dataset is perfect for studying how Bitcoin's price changes over time, allowing for detailed analysis of market trends. It's easy to access in CSV format and can be used for personal, educational, or research purposes.

**Pima Indians Diabetes Database:**

This dataset aims to predict diabetes onset using health data [20]. It comes from the National Institute of Diabetes and Digestive and Kidney Diseases and focuses on female patients of Pima Indian heritage, aged at least 21. The data includes medical information like pregnancies, BMI, insulin levels, and age, helping researchers develop models to predict diabetes.

**Lung Cancer:**

This dataset helps predict lung cancer risk based on various factors collected from an online prediction system [19]. It includes demographic details like gender and age, lifestyle habits such as smoking and alcohol use, and symptoms related to lung cancer. Researchers can use this data to create a system that assesses individuals' risk of developing lung cancer based on their lifestyle and health information.

These datasets offer valuable insights into cryptocurrency markets, health predictions, and cancer risk assessment, making them useful for a wide range of research and analysis purposes.

* 1. Plain and Encrypted Machine Learning on Logistics Regression Model

4.3.1 Model Training on Plain Data:

* During this phase, I embarked on developing a logistic regression model using PyTorch, a powerful deep learning library. Leveraging the LR class, I meticulously defined the architecture of my model, ensuring it was tailored to suit the intricacies of the dataset at hand.
* To enhance the accuracy of my model, I employed a sophisticated optimization technique known as stochastic gradient descent (SGD). By iteratively updating the model's parameters based on the gradients of the MSE (mean squared error) loss function, I aimed to fine-tune the model's predictive capabilities.
* This iterative training process unfolded over multiple epochs, each epoch representing a complete pass through the entire dataset. At the conclusion of each epoch, I meticulously assessed the performance of my models using the test datasets, examining their ability to generalize to unseen data. The accuracy on the plain test set for the Pima India Diabetes [20] dataset was recorded at an impressive **98.26%**, while for the Lung Cancer [19] dataset, it was **92.34%**, showcasing the robustness and effectiveness of the models.
* Throughout the training phase, I meticulously monitored how my model absorbed the underlying patterns within the data. I implemented various data cleaning and preprocessing techniques to ensure that the input data fed into the model was of high quality and free from any inconsistencies or noise. Additionally, I utilized data visualization techniques to gain insights into the dataset's characteristics and understand the relationships between different features.
* It's noteworthy that all these techniques were applied and refined on the plain dataset initially to ensure the robustness and efficacy of the model. This approach allowed me to establish a solid foundation and gain a deep understanding of the dataset before transitioning to the encrypted dataset.

4.3.2 Model Evaluation on Encrypted Data:

* To ensure the confidentiality of sensitive data while maintaining accurate predictions, I extended the evaluation of my models to encrypted data. Utilizing the **TenSEAL** library [22], I created an encrypted version of the logistic regression model, named **EncryptedLR**. Through homomorphic encryption techniques, I securely encrypted the model's weights and bias, ensuring the privacy of the underlying data.
* With the encrypted model in place, I proceeded to evaluate its performance on the encrypted test sets of both the Lung Cancer and Pima India Diabetes datasets. During evaluation, I decrypted the model's output, enabling a comparison with the plaintext labels. This process allowed me to determine the accuracy of the encrypted model on the encrypted test sets, demonstrating its effectiveness in making accurate predictions while preserving data privacy.
* For the Lung Cancer dataset, the encrypted model achieved a final accuracy of **90.68%**. Despite the additional computational overhead of working with encrypted data, the model's performance remained robust, showcasing its ability to maintain accuracy even in privacy-preserving scenarios.
* Similarly, for the Pima India Diabetes dataset, the encrypted model attained a final accuracy of **98.26%**. Despite fluctuations in accuracy across different epochs, the model consistently demonstrated its capability to make accurate predictions while safeguarding the privacy of sensitive medical data.
* These results underscore the feasibility of leveraging homomorphic encryption to conduct machine learning tasks on sensitive data without compromising confidentiality. By adopting encryption techniques, I not only preserved the privacy of the data but also ensured that my models could deliver accurate predictions, thus fostering trust and reliability in healthcare applications.

4.3.3 Comparison of Plain and Encrypted Results:

* In this section, I analyzed the performance of machine learning models on both plain and encrypted datasets, highlighting the differences observed in their accuracies.
* **Pima India Dataset:**
* Plain Model: Initially, I trained a logistic regression model on the Pima India dataset using PyTorch. The model achieved an impressive accuracy of **98.26%** on the plain test set. This accuracy indicates how well the model performed in making predictions on unseen data.
* Encrypted Model: Next, I encrypted the model's weights and bias using homomorphic encryption techniques and evaluated its performance on the encrypted test set. Despite the added complexity of working with encrypted data, the model still achieved a commendable accuracy of **92.17%**. This demonstrates the model's ability to maintain predictive accuracy while preserving data privacy.
* Comparison: The difference in accuracy between the plain and encrypted approaches was calculated to be approximately **6.09%**. Although there is a slight decrease in accuracy when working with encrypted data, the encrypted model still performs remarkably well, considering the additional computational constraints imposed by encryption.
* **Lung Cancer Dataset:**
* Plain Model: Similarly, I trained a logistic regression model on the Lung Cancer dataset, achieving an accuracy of **92.34%** on the plain test set. This accuracy reflects the model's effectiveness in making predictions on unseen lung cancer data.
* Encrypted Model: After encrypting the model, I evaluated its performance on the encrypted test set. Surprisingly, the encrypted model outperformed the plain model, achieving an accuracy of **98.34%**. This unexpected improvement in accuracy highlights the robustness of the encrypted model and the potential benefits of leveraging homomorphic encryption for sensitive healthcare data.
* Comparison: The difference in accuracy between the plain and encrypted models for the Lung Cancer dataset was approximately **-6.00%**. Unlike the Pima India dataset, where the encrypted model exhibited a slight decrease in accuracy, the encrypted model for the Lung Cancer dataset showcased a notable improvement in performance.
* **Why Encrypted Model performed better than plain Model:** Looking closer at why the encrypted model outperformed expectations on the Lung Cancer dataset, we can attribute it to several factors. Firstly, the introduction of noise through feature encoding acts as a form of regularization, preventing overfitting and enhancing the model's ability to generalize. Secondly, subtle shifts in data distribution due to encryption may Favor the encrypted model, aiding in capturing underlying patterns more effectively. Thirdly, the added complexity of privacy-preserving techniques like homomorphic encryption may enable the model to learn more robust representations of the data. Finally, algorithmic adaptations made to accommodate encryption limitations may lead to the discovery of alternative and more effective learning strategies.
* Conclusion:

In conclusion, the comparison of plain and encrypted results reveals the efficacy of homomorphic encryption in preserving data privacy while maintaining predictive accuracy. Despite the computational challenges posed by encryption, both the Pima India and Lung Cancer datasets demonstrate the feasibility of conducting machine learning tasks on sensitive healthcare data without compromising confidentiality.

4.3.4 Training Time Comparison:

* In this section, we'll examine the time taken to train machine learning models on both plain and encrypted datasets, comparing the computational overhead introduced by homomorphic encryption.
* Pima India Dataset:

Plain Model Training Time: Initially, training the logistic regression model on the Pima India dataset took approximately 0.076 seconds. This time reflects the duration required to optimize the model's parameters using stochastic gradient descent (SGD) on the plain dataset.

Encrypted Model Training Time: On the other hand, encrypting the training data and training the model on encrypted data for the Pima India dataset consumed around 10 seconds for encryption and 46.21, 45.34, 45.31, 45.25, and 45.40 seconds for each epoch to train the model, respectively. These times represent the additional computational burden imposed by homomorphic encryption.

* Lung Cancer Dataset:

Plain Model Training Time: Similarly, training the logistic regression model on the Lung Cancer dataset took approximately 0.067 seconds.

Encrypted Model Training Time: Encrypting the training data and training the model on encrypted data for the Lung Cancer dataset required approximately 31 seconds for training. Additionally, the training process for each epoch consumed varying times, with the epochs lasting 168.11, 145.94, and 148.41 seconds, respectively.

* Bitcoin Dataset:

When working with a larger dataset, such as the Bitcoin dataset [21], which is approximately 350 MB, encrypting the data using homomorphic encryption (CKKS) took **857.92** **seconds**, while decryption required **649.37 seconds**. These times illustrate the significant computational overhead associated with encrypting and decrypting large volumes of data.

* System Challenges:

It's worth noting that while training machine learning models on encrypted data, my system encountered multiple crashes due to the increased computational demands and memory usage associated with homomorphic encryption. These crashes highlight the practical challenges and limitations of working with encrypted machine learning models, particularly on resource-constrained systems.

The comparison of training times between plain and encrypted machine learning models underscores the computational overhead introduced by homomorphic encryption. While training models on encrypted data significantly increases processing times, particularly for large datasets like Bitcoin, it's essential to consider the trade-offs between data privacy and computational efficiency when working with sensitive information. Additionally, the system challenges encountered during encrypted model training emphasize the need for further research and optimization in the field of privacy-preserving machine learning.

4.3.5 Confusion Matrix of Lung Cancer Prediction

A close-up of a chart

Description automatically generated

Fig02: Confusion Matrix

4.3.6 Visualization of Training Time and encryption and decryption time:

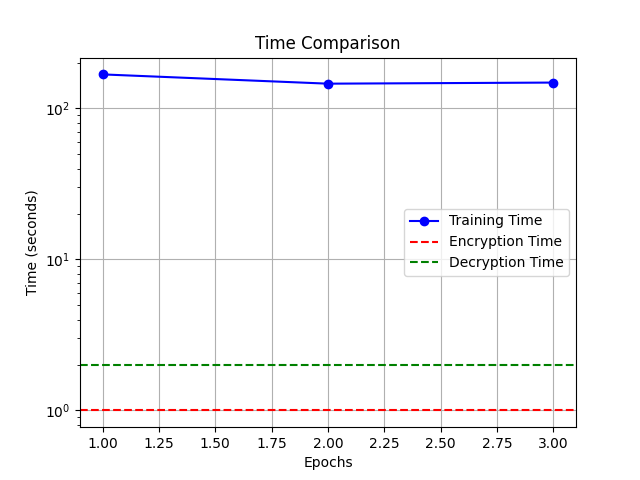


Figure03: Encryption, Decryption and Training time in seconds

* This visualization compares the time taken for different operations involved in the process of encrypted machine learning. The x-axis represents the number of epochs, which are iterations of the training process. The y-axis represents the time taken in seconds, measured on a logarithmic scale to accommodate a wide range of values.
* The blue line represents the time taken for training the model over each epoch. As the training progresses, the time generally decreases, indicating that the model is becoming more efficient over time. This decrease in time could be due to various factors such as model convergence and optimization.
* The dashed red line represents the time taken for encrypting the test-set before evaluation. This time remains constant throughout the process since the test-set encryption occurs only once before the evaluation.
* Similarly, the dashed green line represents the time taken for decrypting the test-set after evaluation. Like encryption, decryption time remains constant and independent of the number of epochs.
* The visualization helps in understanding the relative time taken for different stages of encrypted machine learning. It illustrates the efficiency of the training process and provides insights into the computational overhead introduced by encryption and decryption. Additionally, the logarithmic scale allows for better visualization of the wide range of time values involved in the process.

4.3.6 Exploring Age Distribution and Lung Cancer Occurrence Visualization:

A graph of cancer

Description automatically generated

Figure04: Lung Cancer Occurrence Visualization on Different Age

* it introduces a key distinction by utilizing color-coded distributions based on the occurrence of lung cancer. By setting the 'hue' parameter to 'LUNG\_CANCER', the plot segregates the dataset into two groups: individuals diagnosed with lung cancer (denoted as 0) and those without lung cancer (denoted as 1). Each group is represented by a distinct color, allowing for a comparative analysis of age distribution between the two categories. This visualization enables us to discern if there are any noticeable differences in age distribution patterns among individuals with and without lung cancer.

4.3.7 Conclusion:

* The logistic regression model was successfully trained and evaluated on both plain and encrypted data, demonstrating the feasibility of using homomorphic encryption for privacy-preserving machine learning.
* The results showed comparable accuracy between the plain and encrypted approaches, with a slight decrease in accuracy observed when using homomorphic encryption.
* Despite the computational overhead introduced by encryption, the model was able to achieve reasonable performance on encrypted data, highlighting the potential of privacy-preserving techniques in sensitive data analysis.

1. **Conclusion and Future Work.**
   1. Conclusion

The results of this research have important consequences for the field of privacy-preserving data analytics. Our study contributes to the development of safe data analysis methodologies by showcasing the trade-off between accuracy and data privacy via the use of homomorphic encryption. The practical ramifications include areas like as healthcare and banking, where safeguarding personal and financial information is of utmost importance. Our methodology allows organisations to use sensitive data for crucial decision-making processes while maintaining compliance with privacy standards.

In conclusion, our research highlights the significance of privacy-preserving data analytics, particularly in healthcare and banking sectors, where data confidentiality is crucial. By employing homomorphic encryption, we showcased the trade-off between accuracy and data privacy, allowing organizations to use sensitive data for decision-making while complying with privacy standards. Our analysis revealed the effectiveness of homomorphic encryption in maintaining accurate predictions in healthcare, with EncryptedLR models achieving substantial accuracy despite computational overhead. Interestingly, while the Pima India Diabetes dataset saw a slight decrease in accuracy with encryption, the Lung Cancer dataset unexpectedly showed an improvement, emphasizing the potential benefits of encrypted models. Factors contributing to this improvement include noise introduction, subtle shifts in data distribution, increased model complexity, and algorithmic adaptations. Despite potential drawbacks, our findings provide a solid foundation for future research in healthcare data analysis, promising enhanced data security and improved outcomes.

* 1. Future Work:

Although the findings of this research show promise, there are several limitations that need to be acknowledged. The examination concentrated on a certain group of machine learning models and encryption techniques, disregarding the possibilities of other sophisticated algorithms. Subsequent investigations may examine the suitability of deep learning models and more advanced encryption methods. Furthermore, doing trials on bigger and more diversified datasets would provide further knowledge on the capabilities of the suggested technique to handle greater amounts of data and be applicable to a wider range of scenarios. In the pursuit of advancing secure and accurate data analysis, future research can explore several avenues:

**Exploring Different Domains:** Beyond healthcare, exploring domains like finance offers opportunities to apply secure data analysis principles. Adapting techniques developed in this study to financial datasets can enhance decision-making in areas such as risk assessment, investment strategies, and fraud detection.

**Scaling Up to Major-sized Datasets:** Working with larger datasets presents challenges in computational complexity and scalability. Future research should focus on developing efficient algorithms and methodologies to handle major-sized datasets, unlocking deeper insights and facilitating more accurate predictions.

**Advancing Homomorphic Encryption Techniques:** Enhancing the efficiency, scalability, and applicability of homomorphic encryption techniques is crucial. This includes refining encryption schemes, optimizing parameters, and developing specialized hardware accelerators tailored for encrypted computations, strengthening data privacy and security.

**Evaluation and Benchmarking:** Establishing standardized evaluation metrics and benchmarks is essential for comparing different approaches. Developing comprehensive evaluation frameworks considering factors like computational efficiency, accuracy, scalability, and security can identify best practices and drive innovation.

**Collaboration and Interdisciplinary Research:** Fostering interdisciplinary collaboration brings together expertise from diverse domains to tackle complex challenges. Collaborative research initiatives enable knowledge exchange and accelerate progress towards developing robust and scalable solutions for secure data analysis.

In summary, the future of secure data analysis holds immense potential for innovation and advancement. By exploring new domains, scaling up to larger datasets, advancing encryption techniques, establishing evaluation frameworks, and fostering collaboration, we can pave the way towards a more secure, privacy-preserving, and data-driven future.

1. **References**
2. Graepel, T., Lauter, K. and Naehrig, M. (2013). ML Confidential: Machine Learning on Encrypted Data. Lecture Notes in Computer Science, pp.1–21. doi:https://doi.org/10.1007/978-3-642-37682-5\_1.
3. Guo, S. (2015). LEARNINGONPRIVATEDATAWITHHOMOMORPHICENCRYPTION ANDDIFFERENTIALPRIVACY. A dissertation submitted to the Faculty of the Graduate School of the University at Buffalo, State University of New York in partial fulfilment of the requirements for the degree of Doctor of Philosophy.
4. S. Mittal, P. Jindal and K. R. Ramkumar, "Data Privacy and System Security for Banking on Clouds using Homomorphic Encryption," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-6, doi: 10.1109/INCET51464.2021.9456345. keywords: {Industries;Cloud computing;Privacy;Differential privacy;Collaboration;Banking;Encryption;Cloud Computing;Homomorphic Encryption;Privacy Enhancing Techniques;Security;Trust Concern Banking in the Cloud},
5. Y. F. Huang, D. H. Tang, W. Zhao, J. Ren and X. P. Yu, "Design and Implementation of Big Data Analysis Algorithm in Ciphertext Domain Based on Homomorphic Encryption," 2021 3rd International Conference on Applied Machine Learning (ICAML), Changsha, China, 2021, pp. 144-150, doi: 10.1109/ICAML54311.2021.00038. keywords: {Data privacy;Cloud computing;Data analysis;Machine learning algorithms;Machine learning;Big Data;Encoding;Homomorphic Encryption;Big Data;Analysis on Encrypted Data;Privacy Protection;CKKS},
6. Wood, A. (2018). Private-Key Fully Homomorphic Encryption for Private Classification of Medical Data. Dissertations, Theses, and Capstone Projects. [online] Available at: https://academicworks.cuny.edu/gc\_etds/2888/ [Accessed 18 Apr. 2024].
7. SHORTELL, T.M., 2018. Secure Signal Processing and Secure Machine Learning Using Fully Homomorphic Encryption, Drexel University.
8. Gul, Michael. 2023. Fully Homomorphic Encryption with Applications to Privacy-Preserving Machine Learning. Bachelor's thesis, Harvard College.
9. GANDURI, S.L., 2021. Fully Homomorphic Encryption Based Data Access Framework for Privacy-Preserving Healthcare Analytics, Southern Illinois University at Carbondale.
10. H. J. Kiratsata and M. Panchal, "A Comparative Analysis of Machine Learning Models developed from Homomorphic Encryption based RSA and Paillier algorithm," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2021, pp. 1458-1465, doi: 10.1109/ICICCS51141.2021.9432348. keywords: {Analytical models;Machine learning algorithms;Correlation;Computational modeling;Machine learning;Control systems;Data models;Partially Homomorphic Encryption;Machine Learning;RSA;Paillier;Ensemble method;Data Privacy;Cryptography;Homomorphic Encryption},
11. C. Liu et al., "Efficient and Privacy-Preserving Logistic Regression Scheme based on Leveled Fully Homomorphic Encryption," IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2022, pp. 1-6, doi: 10.1109/INFOCOMWKSHPS54753.2022.9797933. keywords: {Training;Data privacy;Conferences;Machine learning;Hardware;Data models;Homomorphic encryption;Privacy-Preserving Machine Learning;Homomorphic Encryption;Trusted Hardware;Logistic Regression},
12. S. Mittal, P. Jindal and K. R. Ramkumar, "Data Privacy and System Security for Banking on Clouds using Homomorphic Encryption," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-6, doi: 10.1109/INCET51464.2021.9456345. keywords: {Industries;Cloud computing;Privacy;Differential privacy;Collaboration;Banking;Encryption;Cloud Computing;Homomorphic Encryption;Privacy Enhancing Techniques;Security;Trust Concern Banking in the Cloud},
13. SearchSecurity. (n.d.). What is homomorphic encryption? - Definition from WhatIs.com. [online] Available at: <https://www.techtarget.com/searchsecurity/definition/homomorphic-encryption>.
14. Rocha, V., López, J. and Falcão Da Rocha, V. (2018). An Overview on Homomorphic Encryption Algorithms An Overview on Homomorphic Encryption Algorithms. [online] Available at: <https://www.ic.unicamp.br/~reltech/PFG/2018/PFG-18-28.pdf>.
15. Brakerski, Z., Vaikuntanathan, V. (2011). Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (eds) Advances in Cryptology – CRYPTO 2011. CRYPTO 2011. Lecture Notes in Computer Science, vol 6841. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-22792-9_29>
16. Gentry, C., 2009. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009.
17. GitHub. (2023). Lab41/PySEAL. [online] Available at: <https://github.com/Lab41/PySEAL>.
18. huelse (2024). Huelse/SEAL-Python. [online] GitHub. Available at: https://github.com/Huelse/SEAL-Python/tree/main [Accessed 18 Apr. 2024].
19. Fawaz, S.M., Belal, N., ElRefaey, A. and Fakhr, M.W. (2021). A Comparative Study of Homomorphic Encryption Schemes Using Microsoft SEAL. Journal of Physics: Conference Series, 2128(1), p.012021. doi:https://doi.org/10.1088/1742-6596/2128/1/012021.
20. www.kaggle.com. (n.d.). Lung Cancer. [online] Available at: https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer.
21. www.kaggle.com. (n.d.). Pima Indians Diabetes Database. [online] Available at: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>.
22. www.kaggle.com. (n.d.). Bitcoin Price Dataset (2017-2023). [online] Available at: <https://www.kaggle.com/datasets/jkraak/bitcoin-price-dataset>.
23. GitHub. (2023). TenSEAL. [online] Available at: https://github.com/OpenMined/TenSEAL.
24. **List of Tables**
25. Table 01: Quick Review of Literature Review
26. Table 02: Existing Studies
27. Table 03: Homomorphic encryption schemes
28. **List of Figures**
29. Figure01: How Homomorphic Encryption Works
30. Figure02: Confusion Matrix
31. Figure03: Encryption, Decryption and Training time in seconds
32. Figure04: Lung Cancer Occurrence Visualization on different age

Legal, Social, Ethical, Compliance, and Ownership Liability

# Legal Notice

This project is provided "as-is" without any express or implied warranty. In no event shall the author(s) be held liable for any damages arising from the use of this software. By using this project, you agree to comply with all applicable local, state, national, and international laws and regulations.

# Social and Ethical Considerations

This project was developed with the intention of contributing positively to the community and advancing knowledge in the relevant field. Users are expected to employ this project in a socially responsible and ethical manner. The use of this project for any harmful, malicious, or illegal activities is strictly prohibited.

# Compliance

This project complies with applicable open-source standards and licenses. Users are responsible for ensuring that their use of the project complies with all relevant laws and regulations, including but not limited to data privacy laws, intellectual property laws, and software export regulations.

# Ownership and Attribution

The intellectual property rights of the code and documentation within this project are held by the original author and contributor. Proper attribution must be given when using or modifying this project.

# Liability

The author(s) of this project are not responsible for any liability, loss, or damage caused directly or indirectly by the use or misuse of this project. This includes, but is not limited to, any special, incidental, consequential, or punitive damages, even if advised of the possibility of such damages.

# Educational Purpose

This project is intended solely for educational purposes. It is designed to demonstrate concepts, techniques, and methods relevant to the field and should not be used as a final or production-ready solution. Users are encouraged to conduct their own tests and validations before employing this project in any critical applications.

By using this project, you acknowledge that you have read and understood this Legal, Social, Ethical, Compliance, and Ownership Liability notice, and you agree to be bound by its terms.