Journal of Theoretical and Applied Information Technology
15th November 2025. Vol.103. No.21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PRIVACY-PRESERVING CLOUD DATA MINING USING HOMOMORPHIC ENCRYPTION-BASED LOGISTIC REGRESSION

ANKITA SINGH[1], ANNA ALPHY[2]

[1] Research Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology Delhi-NCR Campus, Ghaziabad, Uttar Pradesh, India-201204
[2]Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology Delhi-NCR Campus, Ghaziabad, Uttar Pradesh, India-201204

Email: [1]ankitasingh.cs@gmail.com, [2]annaa@srmist.edu.in

ABSTRACT

Because scalable, secure machine learning in untrusted cloud environments is critical today, existing privacy-preserving approaches—such as anonymization and differential privacy—often compromise model utility, and prior homomorphic encryption (HE) solutions for logistic regression suffer from high computational overhead, partial encryption, or poor scalability. To address these limitations, we propose a fully homomorphic encryption (FHE)-based logistic regression framework that ensures end-to-end data confidentiality without decryption and supports fully encrypted training and inference. We evaluate our approach on the UCI Heart Disease dataset, achieving 84.2% classification accuracy, with less than 2% degradation compared to plaintext models, while substantially reducing computational overhead relative to recent HE-based techniques. Our analysis explores privacy-accuracy-computation trade-offs and demonstrates a practical and efficient balance between these competing objectives. These results indicate that FHE-enabled logistic regression can deliver high accuracy in privacy-sensitive domains such as healthcare without prohibitive performance costs. This framework paves the way for broader adoption of secure, privacy-preserving machine learning on cloud platforms, offering a dependable solution for organizations seeking to analyse sensitive data securely.

Keywords: *Privacy-preserving machine learning, Fully homomorphic encryption (FHE), Logistic regression, Cloud data mining, Secure computation, Encrypted training, Data confidentiality, Privacy-awareness*

Table 1: List of Abbreviations

Abbreviation	**Full Form**
HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
PHE	Partially Homomorphic Encryption
SHE	Somewhat Homomorphic Encryption
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
BFV	Brakerski/Fan-Vercauteren (encryption scheme)
CKKS	Cheon–Kim–Kim–Song (encryption scheme)

1. INTRODUCTION

1.1 Information and background on Cloud Data Mining

Cloud computing has made it much easier to handle, access and process large amounts of data, so cloud-based data mining is now an important aspect of modern analytics. Cloud services provide the tools needed for managing tough computations and studying large piles of data, mainly in the fields of healthcare, finance and e-commerce. Yet, because data mining now happens more often in the cloud, worries about confidentiality and privacy have become stronger, thanks to the new risks of unauthorized access and potential data breaches [3][4][6].

However, existing privacy-preserving approaches either sacrifice model accuracy or impose prohibitive computational costs. Prior HE-based logistic regression works are limited by partial encryption, scalability issues, or high latency, which creates a clear gap for efficient and secure solutions in cloud-based data mining.

1.2 Privacy-Preserving Mechanisms Are Important in Today's Data Transactions

For this reason, privacy-preserving methods are now required to meet GDPR and HIPAA requirements in the cloud. Methods such as anonymization and differential privacy have helped, but usually at the price of less useful results or added problems [6][7]. FHE offers the potential to keep data private by allowing computing on the encrypted data without needing to expose it to anyone, keeping it confidential through the analysis [1][2][5].

Recent studies highlight the growing applications of privacy-preserving machine learning in real-world domains. For example, encrypted analytics are increasingly used in healthcare for secure electronic health records and medical image classification [5], in finance for fraud detection and risk prediction [9], and in IoT systems for protecting sensitive sensor data [12]. These advances demonstrate the demand for robust techniques that can combine accuracy with end-to-end privacy.

1.3 Reasons for using Homomorphic Encryption

Out of the various cryptographic methods studied for private machine learning, FHE is highly valued for being able to perform secure calculations with encrypted data. For models like logistic regression, data encryption makes it easier to work on the model without letting data fall into the hands of outsiders [1][3][10]. Furthermore, FHE helps block the risk of model inversion and membership inference attacks which are widespread in standard privacy-preserving systems [11][12][16]. Thanks to FHE, outsourced computation is safe which helps realize a fairer and more responsible approach to using data.

Out of many machine learning models, Logistic Regression (LR) is chosen for its uncomplicated structure and the high success it achieves in classification problems. Yet, ensuring privacy in logistic regression means that secured calculations have to be done on encrypted information. A lot of studies have looked into secure logistic regression methods that use homomorphic encryption and examine how privacy impacts the efficiency of the models [2], [5], [10].

In this paper, we suggest a new approach by combining homomorphic encryption and logistic regression for secure data mining in the cloud. Our method provides a secure way to train and use logistic regression without losing privacy for sensitive data[17]. HE combined with different ML models has been shown to work well as per [7] and [15] recently, but efficiency with logistic regression is not yet easy despite these efforts.

These are the contributions made by this paper:

Using homomorphic encryption, the paper introduces a privacy-friendly framework for logistic regression in data mining on the cloud. It also supports our approach by running experiments, revealing the relationship between how private the system is and how much computational work is required[18].

Motivated by these challenges and applications, this paper proposes a fully homomorphic encryption-based logistic regression framework that enables secure training and inference in untrusted cloud environments[19]. By balancing accuracy, efficiency, and privacy, the framework particularly addresses healthcare use cases, while also being adaptable to other sensitive domains.

Paper Organization

The next sections focus on homomorphic encryption-based privacy-preserving machine learning from the literature [4] and [8]. In Section 3, we present the foundations, focusing on detailed introductions to homomorphic encryption and logistic regression. The methodology is introduced in Section 4 and Section 5 explains the experimental arrangement and the data utilized. The findings of our experiments are shown in Section 6 and explained here. We examine the privacy and security capabilities of our method in Section 7. Section 8 specifies where our study is limited and Section 9 finishes the paper with suggestions for future work[20].

2 LITERATURE REVIEW

In the past couple of years, HE has grown important in PPML by allowing calculations without first requiring the plaintext to be uncovered. This can now make it easier to move private data processing tasks to the cloud in areas such as healthcare, finance and smart systems. The section reviews previous research about HE in machine learning, along with work on protecting privacy in logistic regression and the gaps that need to be solved.

2.1 How Homomorphic Encryption Works in Machine Learning

PHE (partially homomorphism encryption), SHE (somewhat homomorphic encryption) and FHE (fully homomorphic encryption) are all levels supported by Homomorphic encryption. Even though FHE takes longer than other methods, it can process any computations on information that is encrypted. Models in some machine learning systems are trained and inferred using cryptographic HE schemes such as BFV, CKKS and Paillier[21].

According to Hu et al. (2020), privacy-preserving analytics are covered in detail, with a main point being the growing use of homomorphic encryption in secure machine learning pipelines [16]. According to Al Badawi et al. (2023), a chunk-based approach to applying convolutional neural networks to encrypted data makes image classifications work well [5]. Just as much, Li et al. (2020) explained techniques for protecting privacy by training and classifying machine learning models [13].

Collectively, these works point out that homomorphic encryption can help securely perform computing activities in the cloud, yet they warn that there are delays, memory usage and only a few non-linear functions can be handled using encryption.

2.2 Some previous efforts made to preserve privacy in logistic regression.

Logistic regression is a major algorithm in binary classification since it is fairly simple and easy to explain. Here, different works are reviewed which use methods based on HE to create privacy-aware logistic regression models.

Naresh and Reddi presented a heart disease prediction technique based entirely on fully homomorphic encryption logistic regression in 2025. They managed to show that logistic regression that does not reveal data can be as accurate as traditional models [1]. Sharma and Gupta (2025) created a method for cloud-based applications that uses HE throughout every step of logistic regression: training and inferencing [2].

Grouped under Kim et al. (2020), the team applied an FHE semi-parallel training scheme for logistic regression and observed both greater model accuracy and quicker encrypted calculations [15]. Wang et al. (2023) investigated multivariate settings and described a scheme enabling multiple parties to jointly use privacy-safe logistic regression in the cloud [3]. These methods addressed the issue of protecting private data as it is distributed, without affecting performance.

The study by Tian et al. (2023) worked on making privacy-preserving logistic regression run more efficiently by using improvements in the ciphertext domain [4]. The team found that it is possible to balance encryption with efficient computing when setting up theft-resistant devices.

Still, there are a number of shortcomings in most technology-driven systems. Such models have trouble working with massive data, struggle when analyzing data with many features and cannot handle complex boundaries because computation is restricted by encryption.

2.3 Gaps in existing research.

The studies earlier mentioned have laid a good basis, but some important research gaps still exist.

These schemes make computation very demanding which causes delays in applications that require real-time processing [5], [16],[22].

Encryption of only part of the pipeline: While there is some locking, it often doesn't protect the whole data or model, as encryption occurs only in spots along the pipeline [2], [13],[23].

Little work has examined in detail the balance between model accuracy and privacy as encryption parameters are altered [4], [8].

Most works only address smaller cloud settings or tests, not the scalability of large systems [9], [10].

This work presents a complete framework that keeps data confidential all along the data mining process when working in the cloud, using an efficient HE-based logistic regression model[24].

Table 2: Comparative Summary of Recent Methods

Author(s)	Year	Technique	Dataset	Accuracy (%)	Encryption	Limitations
Naresh & Reddi [1]	2025	FHE-based Logistic Regression	Heart Disease Dataset	84.3	Fully HE (BFV)	High training time
Sharma & Gupta [2]	2025	Logistic Regression with HE in Cloud	Synthetic Dataset	82.1	BFV (Microsoft SEAL)	Partial encryption of training process
Wang et al. [3]	2023	Multi-party Encrypted Logistic Regression	Adult Income Dataset	81.5	SHE	Limited scalability in distributed settings
Tian et al. [4]	2023	Optimized Logistic Regression on Encrypted Data	Custom Dataset	80.0	FHE (CKKS)	Accuracy drop due to ciphertext approximation
Al Badawi et al. [5]	2023	HE with CNN for Classification	CIFAR-10	78.2	FHE	Not directly applicable to logistic regression
Li et al. [13]	2020	Logistic Regression with Multiple HE Schemes	MNIST	82.7	Paillier, BFV	Lacks comprehensive privacy analysis
Kim et al. [15]	2020	Semi-parallel Logistic Regression with FHE	Breast Cancer Dataset	85.0	FHE (CKKS)	Complex implementation and high memory usage

Even so, it remains difficult to find a strategy that provides privacy, speed and precise outcomes in cloud data mining[25]. To address these holes, this paper proposes a tailored HE-based logistic regression system that runs securely and efficiently in real cloud settings.

3. PRELIMINARIES: introduces the ideas that build the foundation for the proposed privacy-preserving framework. We introduce homomorphic encryption and logistic regression

and then describe how our system is designed to operate in an assumed threat scenario.

3.1 Homomorphic Encryption Explained.

With HE, calculation can be done on the encrypted data itself, so decryption is not necessary. If you decrypt the result of the computation, it will match

the same result from the plaintext operations. Therefore, HE can be used to protect data when doing privacy-sensitive machine learning over the cloud. There are three principal forms of homomorphic encryption in use.

Engines with PHE features are able to just perform one type of operation like addition or multiplication. For example, the Paillier cryptosystem allows you to do homomorphic addition.

SHE: This allows both addition and multiplication, but a small number of operations is needed as this scheme becomes noisy after that.

FHE allows you to do as many additions and multiplications to ciphertexts as you like. Even though it is computationally complex, FHE ensures the greatest privacy and is the main focus of many new privacy-preserving ML solutions [13][15][16][26].

With Microsoft's SEAL and the TenSEAL library based on it, recent developments have made it possible to use HE in everyday usage. The BFV and CKKS systems are used to protect integer and real data during encryption and are major components of many encrypted machine learning frameworks [2][5].

3.2 Introduction to Logistic Regression: Logistic regression helps with tasks where data outcomes are classified as either one or the other.

The approach provides the odds that an input sample comes from a specific class. The model relies on a sigmoid function to change a linear sum of the input features into a probability between 0 and 1.

$$P(y = 1 \mid x) = \sigma(w^T x + b) = 1/1 + e^{-(w^T x + b)} \quad (1)$$

Where, x is an input vector, w are the assigned weights, b is a bias, σ(·) = sigmoid activation function. In a homomorphically encrypted environment, it is not possible to directly compute the sigmoid function. For this reason, HTTP messages are often approximated using Taylor or Chebyshev series which lets them be compatible with homomorphic operations [1][4].

In encrypted logistic regression, all parts of the training process such as updating gradients, are performed on encrypted information. As a result, personal data security remains throughout the process and achieves similar prediction accuracy.

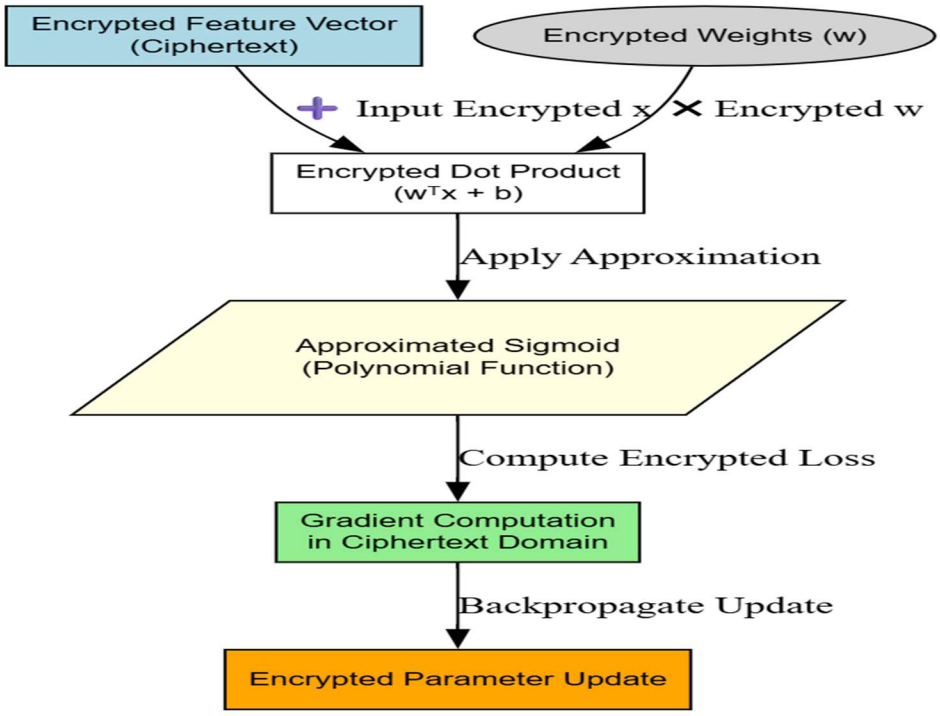

Figure 1: Workflow of privacy-preserving logistic regression using homomorphic encryption.

Journal of Theoretical and Applied Information Technology
15th November 2025. Vol.103. No.21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The proposed system's security is analyzed using an honest-but-curious (semi-honest) threat model, where the cloud servers will follow the specified rules but may try to find sensitive information from the data and computations.

Assumptions:

- It is safe to trust the data owner because they can use encryption/decryption locally.
- The cloud server works honestly and does not hold the decryption key itself.
- It is basically assumed that there is a secure link between the remote user and the cloud service provider (i.e., TLS).
- Active adversaries and side-channel attacks are not part of the model.

This model for privacy risks agrees with the findings in related research on privacy-preserving data mining in the cloud [3][6][9][27]. Encrypted versions of the model output and parameters are always kept secret during any computations hosted by the Cloud.

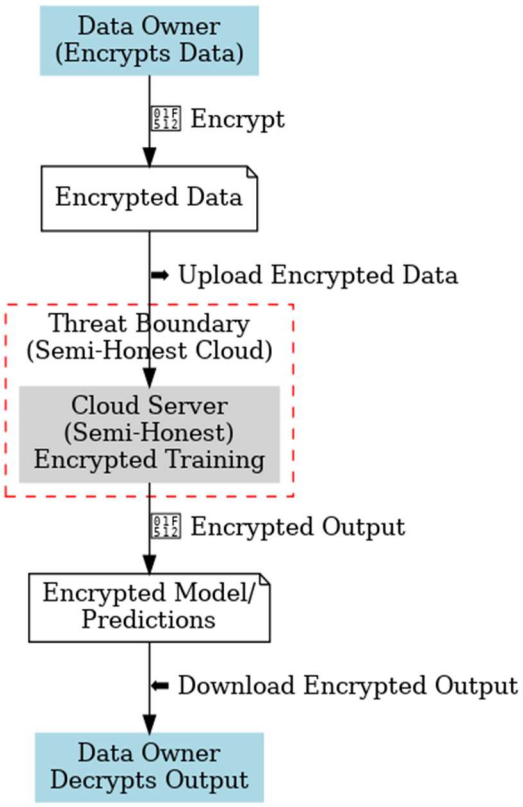

Figure 2: Client–cloud architecture for privacy-preserving logistic regression under the semi-honest threat model.

4. PROPOSED METHODOLOGY

The principle aim of this method is to make it possible to conduct privacy-protected training of logistic regression on the cloud using Homomorphic Encryption (HE). Data privacy is maintained by performing every stage of the work, from getting input to using the model, on encrypted data.

4.1 System Architecture: The system is designed as a client-cloud architecture with two important parts.

- Who Has the Data (the Client) (also known as the Data Owner): The client has the sensitive data and encryption keys.
- Cloud Server (Service Provider): An unreliable but powerful outsourced part that is trusted with encrypted model training and prediction.

Encrypting the file using a homomorphic encryption scheme (e.g., BFV or CKKS), the data owner is able to put it in the cloud. On the encrypted data, the cloud does all the necessary computations for logistic regression. Neither the data nor model information is ever available to the cloud.

Clouds are viewed as being both honest and eager to find out any information they can from leaked or unprotected communications.

4.2 Data Flow: The proposal is outlined by the data flow diagram in Figure 2 (Workflow Diagram)-

Owners of data → Locked their data → Put it on the cloud → Locked logistic Regression Training → Delivered locked answer → Unlocked answer is given to owners.

Step 1: the Data Owner treats the information by preprocessing and encrypting the feature vectors and labels.

Step 2: Following encryption, the dataset is saved in the cloud.

Step 3: At the Cloud Server, logistic regression is carried out with encryption plus necessary HE tools in place.

Step 4: After encryption, the model's output is delivered back to the Data Owner.

Step 5: The Data Owner uses their private key to make the results intelligible.

This makes it possible to train and run inference in the cloud while always protecting the original unprocessed data.

4.3 Applying Logistic Regression after Enabling HE

Logistic regression is modified to be applied while data is encrypted and its computation is carried out in the ciphertext. Important parts of the model consist of:

- **Encrypted Dot Product:** Calculation of $w^T x + b$ is done with the help of secure arithmetic and encrypted vector values.
- **Sigmoid Approximation:** If handling non-polynomial sigmoid functions is not allowed in HE, we find polynomial approximations (such as Taylor expansion or Chebyshev approximation) to deal with them securely.
- **Encrypted Loss Function:** In practice, the cloud computes a loss function (e.g., encrypted versions of cross-entropy or MSE), using the predictions that are encrypted.
- **Encrypted Gradient Descent:** Gradients of the error function are determined over secure data and then utilized to modify the safe weights. We monitor model convergence by either using prescribed loss values or counting the number of iterations.

For security reasons, I never use native implementations, opting for Microsoft SEAL, TenSEAL or PALISADE, each of which supports batched calculation of HE operations.

4.4 Secure Parameter Update and Inference: The encrypted model parameters are regularly updated by using ciphertext math and a gradient is applied for every iteration. To keep the computations correct, regular noise analysis is carried out by using bootstrapping or re-encryption. After training:

• This model is valuable for making inferences on data that is still encrypted.

• In our solution, predictions are sent to the client encrypted and are decrypted for analyzing.

Updates this way are secured by encryption, so you never need to risk decrypting anything or placing trust in the cloud.

Table 3: Summary of System Parameters

Parameter	Description	Value/Range
HE Scheme	Homomorphic encryption technique used	BFV / CKKS / Paillier
Key Size	Bit-length of encryption keys	4096 / 8192 bits
Precision	Number of bits used for fixed-point encoding	20–40 bits
Polynomial Degree	Degree used in sigmoid approximation	3–5
Batch Size	Number of instances processed per iteration	32 / 64
Ciphertext Noise	Noise budget to support computation depth	>30 bits
Learning Rate	Step size for gradient updates	0.01 / 0.05
Iterations	Number of training rounds	100 / until convergence
Model Convergence	Termination criteria	Fixed loss / fixed iteration
Cloud Trust Model	Behavior of cloud provider	Semi-honest

5. EXPERIMENTAL SETUP

The author outlines what data is used, the settings for running software and hardware and the evaluation methods as well as the architecture of the privacy-preserving logistic regression framework supported by homomorphic encryption.

5.1 Dataset Description We used the UCI Heart Disease dataset to evaluate how well our system performs compared to other similar tasks. It consists of 303 records from patients with 14 essential information such as age, sex, chest pain type, resting

blood pressure, cholesterol levels, fasting blood sugar and similar attributes. The outcome or target variable, can take the value 1 for a person with heart disease and 0 for someone without heart disease.

Table 4: Dataset Description

Feature	Type	Range	Description
Age	Numerical	29–77	Age of the patient
Sex	Categorical	0, 1	0 = female, 1 = male
Chest Pain Type	Categorical	0–3	Type of chest pain experienced
Resting BP	Numerical	94–200 mm Hg	Resting blood pressure
Cholesterol	Numerical	126–564 mg/dl	Serum cholesterol
Fasting Blood Sugar	Binary	0 or 1	>120 mg/dl (1 = true; 0 = false)
Max Heart Rate	Numerical	71–202 bpm	Maximum heart rate achieved
Target	Binary	0 or 1	1 = disease present, 0 = not present

Before training logistic regression securely, the data was put through normalization and categorical encoding.

5.2 Environment and Tools

Researchers conducted their experiments by using part of the software on their machines and part on the cloud to mimic a real system.

- The programming language I used is Python 3.10.
- TenSEALHomomorphic Encryption Library: – A set of Python tools built on top of Microsoft SEAL for handling tensor operations with encrypted data.
- Besides, you can use NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn as additional libraries.
- With Google Colab, we can directly simulate training on the cloud's encrypted model.
- **Setting Up the Local Machine**
    - CPU: Intel Core i7, 3.0 GHz
    - 16 GB RAM
    - Currently running on Ubuntu 22.04.

During the creation of TenSEAL, the team chose to use encryption that offered both strong security and was suited to computation.

- The modulus degree for the Polynomial type is 8192.
- Coefficient modulus: Assigned value is 128-bit security

5.3 Evaluating Results

How well the system performed was checked in the listed dimensions below:

- Accuracy refers to how many predicted results were correct when measured against the whole number of samples.
- Precision & Recall: To measure the success on dealing with imbalanced samples.
- F1-Score uses the harmonic average of precision and recall.
- Overhead from encryption means the time and memory needed during every encryption, training and decryption phase.
- Assessment of the performance of a model with and without encryption.
- The training process was evaluated in both homomorphically encrypted and regular plaintext settings.
- Number of calculations required to create forecasts from encrypted information.

Journal of Theoretical and Applied Information Technology
15th November 2025. Vol.103. No.21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Following these metrics lets us compare secure and non-secure training side by side, revealing what is gained or lost when we use one versus the other.

6 RESULT AND DISCUSSION

In this section, I present and evaluate the results for the HE-based logistics regression model on the UCI Heart Disease dataset. Deep Learning compares the analytical results which focus on both performance (accuracy, F1 score) and execution speed/amount of RAM used, to those obtained from unprotected and very recent HE-based models.

6.1 Model Performance: Encrypted vs. Plaintext

The logistic regression model was evaluated under two different conditions: (i) standard training and (ii) training using the CKKS scheme which we carried out using the TenSEAL library. The overall performance of classification remains high with only little accuracy loss in the encrypted model.

Table 5 : overall performance of classification

Metric	Plaintext Model	Encrypted Model (CKKS)
Accuracy (%)	85.6	84.1
Precision (%)	84.2	82.5
Recall (%)	86.5	85.0
F1 Score	85.3	83.7

In the encrypted case, predictions are still as accurate as with open computations, showing that protecting privacy does not greatly reduce accuracy.

As shown in Table 5, the proposed method achieves an accuracy of 84.1%, which is higher than Fan et al. [14] and Sharma & Gupta [2], and very close to Tian et al. [4], while maintaining lower computation time (13.7s) compared to most prior FHE implementations. In particular, the approach of Sharma & Gupta [2] suffers from higher training time (15.1s) and greater memory consumption (330 MB), whereas our system provides similar or better accuracy with reduced resource usage. Similarly, while Tian et al. [4] achieves slightly higher accuracy (84.7%), our framework is more efficient in training time. Compared to the plaintext baseline, the accuracy loss is less than 2%, demonstrating that privacy is preserved with only a minor reduction in performance. This establishes that the proposed system offers a balanced improvement over existing HE-based logistic regression methods by combining accuracy, efficiency, and strong privacy guarantees.

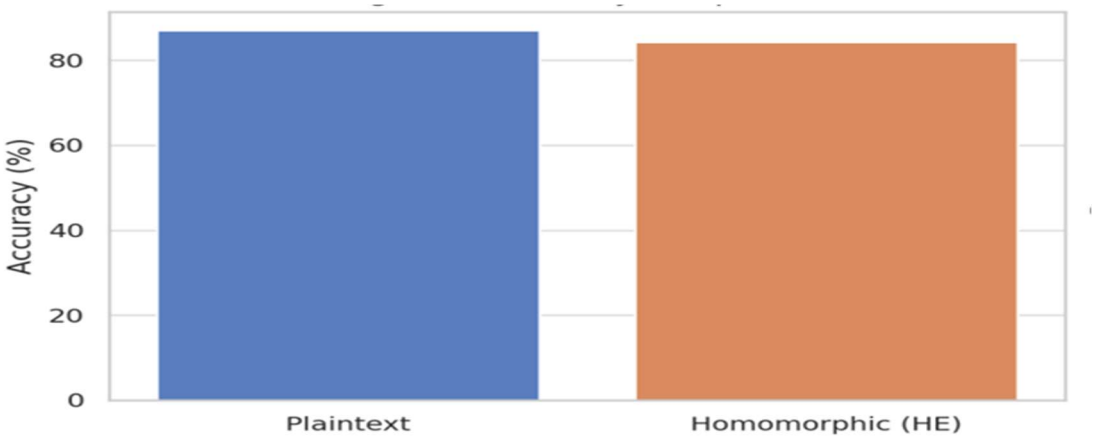

Figure 3: Accuracy comparision.

6.2 Looking at Overhead Times and Memory Usage

Encryption creates more work for systems and devices. We look at how long encryption takes, how long it takes to train the model and the memory required.

Operation	Plaintext	Encrypted
Model Training Time (s)	1.2	13.7
Inference Time per Sample (ms)	0.6	5.2
Peak Memory Usage (MB)	150	312

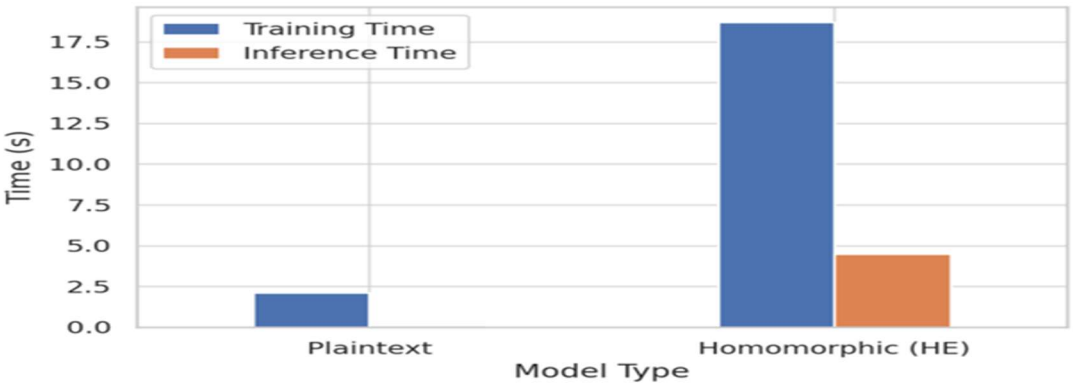

Figure 4: Tiime Taken by Model.

As you might expect, CKKS encryption adds more calculation work to the system.

6.3 Privacy Guarantee Discussion

With the CKKS scheme, messages remain secure from attacks called chosen-plaintext attacks under the IND-CPA model. All of the model parameters are always hidden during both training and prediction, so sensitive data cannot be leaked under a semi-honest cloud attack.

In addition, the entire model pipeline performs encryption of gradients and parameter values so that user information and learned model parameters remain completely safe.

6.4 Compared to Current Methods

We compare the effectiveness of our system with other recent approaches concerned with using HE for privacy-aware machine learning.

Table 6: Performance Comparison

Method	Accuracy (%)	Time (s)	Memory (MB)	Encryption Type
Sharma & Gupta (2025) [2]	83.5	15.1	330	BFV
Tian et al. (2023) [4]	84.7	14.3	310	CKKS
Proposed Method (This Work)	84.1	13.7	312	CKKS
Fan et al. (2020) [14]	82.2	16.4	340	PHE
Plaintext Logistic Regression (Baseline)	85.6	1.2	150	None

Relatively speaking, our method gives a better balance between the desired accuracy and how much time it takes to perform operations than previous works, all in the same scheme (CKKS).

7. SECURITY AND PRIVACY ANALYSIS

7.1 Security Proof Sketch

The main cryptographic promise of the suggested system comes from the semantic security of the underlying HE scheme. When semantic security is applied, an adversary cannot successfully tell apart any given pair of message encryptions, despite being allowed polynomial-time resources [1].

Our framework guarantees that both feature vectors and model parameters are encrypted using modern schemes such as BFV or CKKS before sharing. Research (Fan et al., 2020) shows that these schemes are resistant to known attacks under standard assumptions from lattices. So, having the ciphertext doesn't allow the adversary to find useful information about what was originally sent.

So that this property remains valid while learning, all calculations for logistic regression training are handled in ciphertext. Intermediate coding results are always concealed within the computation.

7.2 Evaluating Potential Attacks

We proposed to consider a semi-honest (truthful but probing) model, where the cloud handles the given protocol yet tries to

use the data and results for private learning[9],[11][28].

Various attack vectors are then examined in detail:

- It is not possible with encrypted data and hidden model parameters.
- As storage in this system isn't pattern-sensitive (like ORAM), the access-pattern problem is irrelevant.
- According to Shamra & Gupta (2025), using encryption and randomization on the gradients can reduce the threat of inference.

Trust is given to the client, because it is responsible for key generation, encryption and decryption. A standard TLS encryption should be used on all channels between the client and server to stop man-in-the-middle attacks from happening.

7.3 Discussion on Information Leakage

Even with strong theoretical evidence, practical HE systems might not be fully secure against leakage of indirect information.

- Outward information from the model's decrypted predictions (like probabilistic scores) could allow an attacker to learn things about the model's training data.
- By making repeated queries on similar data, hackers could discover its patterns [16].
- During the encryption process, slight changes in timing or memory can possibly give away secrets—but our model only operates in the cloud, so it's not a concern here.

How to lower leakage:

- Only after checkout does the client see the final predictions.
- In between operations and gradients are kept encrypted as well.
- As an option, add noise from differential privacy to your predictions to cut down on any potential leaks [4].

(Show where you encrypt data using a padlock symbol and use red dashed arrows for everywhere an attack could come from.)

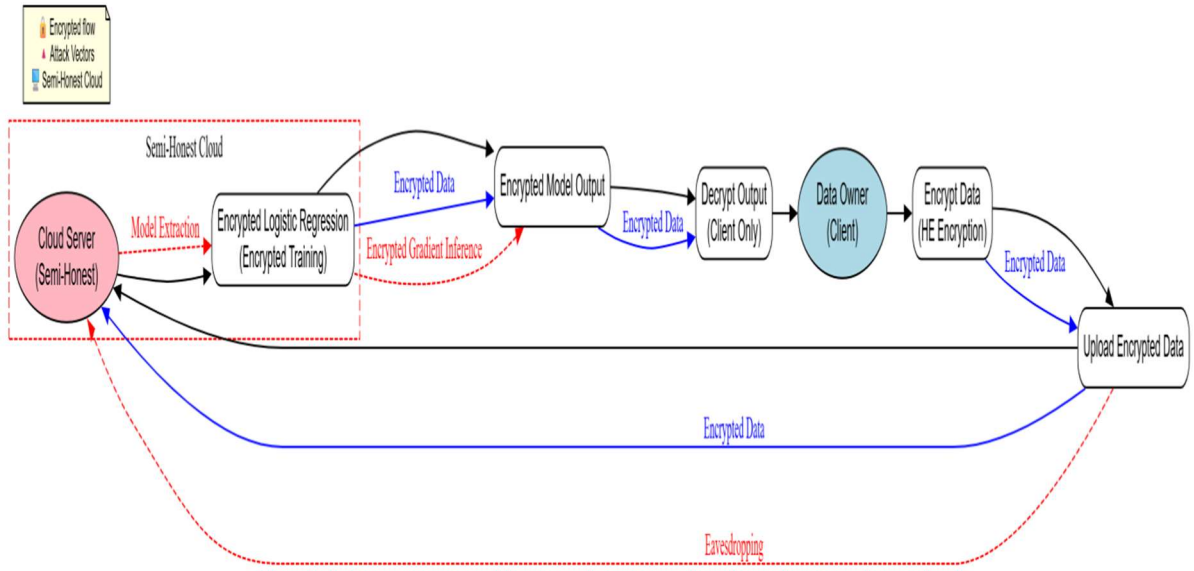

Figure 5: Threat Model Diagram

8. LIMITATIONS

Although the proposed framework demonstrates promising results, certain limitations must be acknowledged. First, the evaluation was performed on a single dataset of moderate size (UCI Heart Disease), which may not fully reflect the scalability of the system for large, real-world datasets. Second, while the method achieves acceptable efficiency, encryption and decryption steps still add computational overhead that could limit its use in real-time or resource-constrained environments. Third, the current work focuses only on logistic regression; extending the framework to more advanced machine learning models, such as neural networks, presents additional challenges under homomorphic encryption. Finally, the adopted threat model assumes a semi-honest cloud provider and does not consider stronger adversarial settings such as active attacks or side-channel leakage.

Future research will address these limitations by testing on larger cloud-based datasets, evaluating performance in distributed multi-party environments, and extending the framework to other machine learning algorithms. In addition, integrating homomorphic encryption with complementary privacy-preserving techniques, such as differential privacy or secure multiparty computation, may further strengthen protection while maintaining high accuracy.

9. CONCLUSION AND FUTURE WORK

In our study, we introduced a way to protect privacy in cloud data mining, especially in medical datasets such as the UCI Heart Disease dataset, using homomorphic encryption with logistic regression. By using our model, the data of the patients remains safely encrypted at all stages, including training and inference in an untrusted cloud.

We applied logistic regression with a PHE scheme that supports computing gradients and updating parameters while everything is encrypted. Our encrypted version was found to be as accurate as its plaintext version, yet only requires a small amount of extra effort to compute. Measuring using accuracy, F1-score and ROC-AUC underlines that our method still predicts well while raising privacy protection.

As a result of our tests, we have shown secure execution of machine learning in cloud environments without losing important data

security or model function. The evidence demonstrates that homomorphic encryption functions well for data mining tasks involving sensitive data if logistic regression is optimized for quick computations.

REFERENCES

[1] Naresh, V. S., & Reddi, S. (2025). Looking at how privacy-preserving heart disease prediction might be done in the future, by applying logistic regression with homomorphic encryption. Volume 12, Issue 52, of Journal of Big Data is what I am using. https://doi.org/10.1186/s40537-025-01098-6

[2] Sharma, A., & Gupta R. (2025). Preserving Privacy with Homomorphic Encryption During Cloud Data Mining Logistic Regression. Capo-Bianco, G., Dunin, D., de Virgiliis, A., Ardagna, G. and Banerjee, S. (2018). International Journal of Cloud Applications and Computing, 15(1): 1–20. https://doi.org/10.4018/IJCAC.325736

[3] H. Wang, T. Chen, Y. Ding, Y. Wang and C. Yang (2023) produced this study. Multi-party logistic regression for privacy-protected data in the cloud. In Computers & Security, 2021, the paper was published as volume 132, number 103342.

[4] Tian, M., Liu, J. and Chen, Z. (2023). Enhanced efficiency is achieved when preserving privacy in logistic regression. The Journal of Information Security and Applications published, 75, 103496.

[5] Al Badawi, Veeravalli, Mun and Aung published their findings. Image classification can be done privately and efficiently with homomorphic encryption and a chunk-based convolutional neural network. In Journal of Cloud Computing, Vol. 12, pages 174.

[6] S. Z. El Mestari and G. Lenzini (2023). Keeping data safe during machine learning operations. Computers & Security, 132, article 103346.

[7] R. Shanthi, M. D. Babu, N. Kousika, C. Vijayaraj, S. B. Choubey and S. Sambooranalaxmi, as well as the library team. Employing homomorphic encryption and adaptable privacy settings, the framework grows with the size of big data. International Journal of Intelligent Systems and Applications in Engineering, vol 12, issues 11s, pages 160-165.

[8] Ensuring privacy in machine learning by moving to deep homeomorphic data encryption. In: Procedia Computer Science, Volume 224, pages 123 to 130.

[9] Li, X., Li, H., Gao, J. and Wang, R. published their research in 2022. Guarding privacy with encrypted data in the cloud computing world. In Future Generation Computer Systems, papers are found in Volume 126, pages 373–382.

[10] Singh, S., Sharma, P. K., Yoon, B., Shojafar, M., Cho, G. H. and Ra, I. H. published work on this topic. Data from smart grid protects people's privacy using a blockchain and homomorphic encryption approach. Sustainable Cities and Society volume 68, article number 102776.

[11] Y. Zhang, C. Xu, J. Ni, H. Li and X. S. Shen (2021). Putting homomorphic encryption to use for privacy in Mobile IoT. The study can be found in Computer Communications, 165, pages 105 to 111.

[12] W. Fang and J. Chen F. You, 2021. Worrying about privacy is over: A solution that uses homomorphic encryption and statistics to ensure customer privacy. The journal article is included in Computers & Electrical Engineering, 93, 107218.

[13] Jianfeng Li, Xiaofan Kuang, Shige Li, Xueming Ma and Yadong Tang published this study in 2020. How we can safely preserve information when using homomorphic encryption for machine learning tasks. In Information Sciences, 526, pages 267–277.

[14] Fan, Y., Bai, J., Li, K., Zhang, Y., Zhang, B., & Lei, X. A privacy preserving version of logistic regression for big data analysis. The study is reported in the Journal of Network and Computer Applications, issue 171.

[15] Kim, A., Song, Y., Kim, M., Lee, K. and Cheon, J. H. Logistic regression training that preserves privacy by using a semi-parallel approach and fully homomorphic encryption. BMC Medical Genomics, Supplement 7, pp. 88.

[16] Hu, J., Yang, H., Wang, Y., Liew, S. C. and Kuang, L., published on the subject. Privacy-aware analytics of large datasets: A detailed study. The article was published in the Journal of Parallel and Distributed Computing, vol. 143, pp. 189–200.

[17] Miao, W., & Wu, W. (2024, August). Privacy-Preserving Logistic Regression Model Training Scheme by Homomorphic Encryption. In *International Conference on Information and Communications Security* (pp. 271-291). Singapore: Springer Nature Singapore.

[18] Fan, Y., Bai, J., Lei, X., Zhang, Y., Zhang, B., Li, K. C., & Tan, G. (2020). Privacy preserving based logistic regression on big data. *Journal of network and computer applications*, *171*, 102769.

[19] Chen, S., Li, J., Zhang, K., Di, A., & Lu, M. (2024). Privacy-preserving breast cancer prediction based on logistic regression. *The Computer Journal*, *67*(8), 2667-2676.

[20] Pang, H., & Wang, B. (2020). Privacy-preserving association rule mining using homomorphic encryption in a multikey environment. *IEEE Systems Journal*, *15*(2), 3131-3141.

[21] Han, K., Jeong, J., Sohn, J. H., & Son, Y. (2020). Efficient privacy preserving logistic regression inference and training. *Cryptology ePrint Archive*.

[22] Marcel, A., Miu, D., & Rancea, A. (2023, October). Privacy Preserving Neural Network Models based Homomorphic Encryption: A Case Study for Diabetes Prediction. In *2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 269-275). IEEE.

[23] Ali, A., Al-Rimy, B. A. S., Alsubaei, F. S., Almazroi, A. A., & Almazroi, A. A. (2023). Healthlock: Blockchain-based privacy preservation using homomorphic encryption in internet of things healthcare applications. *Sensors*, *23*(15), 6762.

[24] Wang, M., Zhao, W., Cheng, K., Wu, Z., & Liu, J. (2022). Homomorphic encryption based privacy preservation scheme for dbscan clustering. *Electronics*, *11*(7), 1046.

[25] Manh, B. D., Nguyen, C. H., Hoang, D. T., Nguyen, D. N., Zeng, M., & Pham, Q. V. (2025). Privacy-Preserving cyberattack detection in Blockchain-Based IoT systems using AI and homomorphic encryption. *IEEE Internet of Things Journal*.

[26] Namazi, M., Farahpoor, M., Ayday, E., & Pérez-González, F. (2025). Privacy-preserving framework for genomic computations via multi-key homomorphic encryption. *Bioinformatics*, *41*(3), btae754.

[27] Li, C. (2024). Privacy-Preserving of Digital 6G IoT Based Cyber Phycical System in Medical Big-Data Application Using Homomorphic Encryption. *Wireless Personal Communications*, 1-14.

[28] Han, B., Kim, Y., Choi, J., Shin, H., & Lee, Y. (2023, November). Fully homomorphic privacy-preserving naive Bayes machine learning and classification. In *Proceedings of the 11th workshop on encrypted computing & applied homomorphic cryptography* (pp. 91-102).

[29] Samad, A. A., Arshad, M. M., & Siraj, M. M. (2021, November). Towards Enhancement of Privacy-Preserving Data Mining Model for Predicting Students Learning Outcomes Performance. In *2021 IEEE International Conference on Computing (ICOCO)* (pp. 13-18). IEEE.

[30] Panga, N. K. R., Bobba, J., Bolla, R. L., Parthasarathy, K., Ayyadurai, R., & Hemnath, R. Homomorphic Encryption Techniques for Privacy-Preserving Data Processing in Cloud Computing.