Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6, 338-354
2026
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v10i6.13145

Secure machine learning on encrypted data using homomorphic encryption techniques

Sushma M P[1], Kiran Puttegowda[2], Praveenkumara J[3*]
[1,2]Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India, Visvesvaraya Technological University, Belagavi, India; sushmamp@pesce.ac.in (S.M.P.) kiranhsn@vvce.ac.in (K.P.).
[1]Department of Electronics and Communication Engineering, PES College of Engineering, Mandya, Karnataka, India.
[3]Department of Mechanical Engineering, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India, Visvesvaraya Technological University, Belagavi, India; praveenkumar96j@gmail.com, praveenkumar@vvce.ac.in (P.J.).

Abstract: Privacy preservation has become a critical challenge in modern machine learning applications, especially in sensitive domains such as healthcare, finance, and cloud-based services, where confidential data must be protected. This paper aims to develop a secure machine learning framework using Homomorphic Encryption (HE), specifically the CKKS scheme, to enable computation on encrypted data without requiring decryption. The proposed methodology integrates Logistic Regression, Encrypted Fully Connected Neural Networks (EncFCNN), and Encrypted Convolutional Neural Networks (EncCNN) to perform both training and inference directly in the encrypted domain. The encrypted data and model parameters are processed securely on an untrusted server while maintaining complete data confidentiality. Experimental evaluation is conducted using the Pima Indian dataset and the MNIST dataset to compare encrypted and non-encrypted model performance. The findings show that the encrypted logistic regression model achieves an accuracy of 0.7375, which is comparable to plaintext training, while EncCNN achieves 100% accuracy on the test set with acceptable computational overhead. The results confirm that homomorphic encryption provides strong privacy preservation without significantly affecting model accuracy, making it highly suitable for secure real-world machine learning applications.

Keywords: *Confidential machine learning, Cryptography in machine learning, Data Privacy, Homomorphic encryption Logistic regression, Machine learning, Privacy-preserving machine learning, Secure inference, Secure model evaluation.*

1. Introduction

Nowadays, machine learning algorithms are extensively utilised across various real-world applications, including healthcare, finance, cybersecurity, autonomous systems, and personalised recommendation systems. The training process for better accuracy necessitates these applications to consume large datasets containing user information. The requirement to use sensitive and private data during these analyses generates substantial risks regarding data protection and individual privacy. Machine learning development continues to expedite while privacy-protecting solutions gain importance because of rising data privacy threats, including breaches and unapproved access to sensitive user details. Machine learning systems train on large datasets with sensitive personally identifiable details, which creates substantial ethical, legal, and security issues because of misuse and data-leakage risks. The public, together with institutions, prioritises privacy-safe utilisation methods because of major incidents revealing data-security flaws, which have increased awareness about the need for privacy-preserving technology. The development of solutions, including homomorphic encryption and differential privacy, along with secure multiparty computation and federated learning, constitutes researchers' and practitioners' response to privacy concerns. These protection techniques help reduce

History: Received: 2 April 2026; Revised: 5 June 2026; Accepted: 10 June 2026; Published: 24 June 2026
* Correspondence: praveenkumar96j@gmail.com

privacy risk by either maintaining complete data-exposure exclusion or implementing secure data transformations, which allow trustworthy employment of AI systems in healthcare, as well as finance and personalised digital services [1, 2].

The user data requirement for traditional machine learning models causes sensitive information to be disclosed to service providers or third-party platforms for training and inference execution. The data sits as a vulnerability that allows security breaches when data leakage happens because of cyberattacks, as well as insider threats or accidental data mismanagement. Privacy-preserving machine learning techniques develop because researchers seek to conduct significant computations on information that stays concealed from exposure of its underlying sensitive contents. Secure machine learning operations for models can be carried out on protected data collections by using these techniques, which keep original datasets hidden from system operators. The protection of data confidentiality stands as an essential requirement that the mandate includes healthcare, financial services, and governmental systems. Machine learning data privacy approaches protect information throughout the full operational period of preprocessing and model inference while maintaining a degree of both data use and privacy compliance. Privacy-preserving machine learning proves to be a transformative technology that delivers safe, trustworthy artificial intelligence systems throughout regulated environmental domains [3, 4].

The system allows machine learning calculations to work on encryption-protected data that clients transmit to the service provider without revealing the actual unencrypted data to the service provider. AI models become accessible through this method without exposing data confidentiality to users or organisations. Our method's effectiveness was demonstrated through implementing and testing logistic regression with a fully connected neural network, followed by a convolutional neural network along with logistic regression. These standard models serve different industries, including medical diagnostics, while also assessing financial risks and recognising images, making them optimal choices to evaluate privacy-preserving computations. Our research included both the analysis of encrypted data and the process of training machine learning models using encrypted datasets. Our system established an encrypted logistic regression model, which performs direct training operations on encrypted data sets. The security level reaches its peak point due to the model's complete absence of raw data interaction during training. The method allows for safe outsourcing of the training procedures to external training providers, who don't have access to raw data since it is encrypted.

We conclude that privacy-preserving methods in machine learning, such as those based on homomorphic encryption, are not only theoretically well-founded but also provide solutions that can be implemented in real-world industrial applications. These techniques provide a strong foundation for protecting sensitive data while training and using models, without requiring decryption. This feature is particularly useful in industries where privacy is crucial, like the healthcare sector, where patient information needs to be safeguarded, and financial services, where transactional and personal data should be given high-level protection. Furthermore, embedding these privacy-preserving features into cloud AI services bolsters trust and compliance within outsourced computing scenarios, fostering secure and collaborative analytics in decentralised systems. This way, the approach is a much-improved method of data utility and strict privacy requirements in critical areas.

2. Related Works

Chen et al. [5] developed a federated learning architecture that protected gradients through the implementation of homomorphic encryption. The authors developed their framework by using data quality-based gradient weight adaptations, which produce better model accuracy results for partially trusted server systems. Fang and Qian [1] constructed PFMLP as a framework that safeguards multi-party privacy through the federation of learning coupled with partially homomorphic encryption. Implementing the FOE-Paillier cryptosystem in the framework permits safe gradient transfer during training, with an acceleration rate of 25–28% while maintaining pattern accuracy like standard systems. Gadiwala et al. [6] introduce PFed-HE as an encrypted learning system that combines homomorphic encryption systems with differential-privacy methods. The system enables protected gradient

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

combination through dispersed model-training methods that are specifically designed for medical and banking applications with sensitive information. Hong [7] provides a comprehensive analysis of Fully Homomorphic Encryption (FHE) in machine learning, while reviewing multiple FHE approaches and their performance parameters to select encrypted methods fitting different privacy needs.

Peluso et al. [8] provide security of data along with reduced costs in model update communications and processing requirements. Jia et al. [9] established a flexible artificial intelligence system through the connection of homomorphic encryption and encrypted search functionality. The system provides protected data operations and dynamic model development through resource management and the elimination of unnecessary calculations. Lo et al. [10] illustrate FHE as a security protocol that ensures data protection during training operations. The privacy-protection features of this approach face major implementation difficulties that require extensive parameter modifications to achieve practical deployment. Nguyen et al. [11] presented HHE protocols named 3PervPPML and Trusted PervPPML within their paper. These protocols manage to strike a balance between data privacy requirements and computational efficiency operations within systems that have rationed computing resources. Kurniawan and Mambo built a federated active learning framework through the implementation of homomorphic encryption technology for security purposes [12]. The privacy-protected solution gives encryption strength and immunity to gradient leakage attacks while offering high model performance.

Park and Lim [13] enable encrypted local model parameter aggregation through servers with no encryption or decryption requirements. The system enables distributed nodes to collaborate through secure processes, which lead to model development. Frimpong et al. [14] present a fabric of homomorphic encryption methods to develop their PPML scheme, which enables secure data-classification operations on encrypted information. The method aligns with minimal-resource devices through optimised communication and processing requirements. Chen et al. [15] present a swarm-learning framework that relies on threshold Paillier cryptosystem operations. The solution blocks unauthorised model-information disclosure while defending against inference attempts to support safe distributive learning. Yuan et al. [16] present a comprehensive analysis of privacy-preserving machine learning utilising approximate HE through surveys of implementation strategies, application domains, and datasets for guiding future work in this field. Han et al. [17] present PBFL, which integrates blockchain technology into federated learning through the employment of homomorphic encryption and random masking for model-parameter protection throughout training. Effendi and Chattopadhyay [18] developed a secure federated-learning healthcare solution that implements homomorphic encryption inside secure multi-party computing boundaries. The proposed security solution defeats adversarial threats while preserving competitive system-performance levels. Lee et al. [19] introduce HETAL, which serves as a privacy-preserving transfer-learning method through homomorphic encryption utilising CKKS. The HETAL system provides encrypted training capabilities unlike traditional methods, which only focus on encrypted inference and maintain accuracy levels that match plaintext methods.

Adamsetty et al. [20] explore how homomorphic encryption (HME) serves to protect data privacy throughout machine learning operations. Their research shows that classifiers operate at equal precision between encrypted inputs and unencrypted inputs, proving that HME maintains security even while keeping operational speed constant. Mahatho et al. [21] explored the use of the CKKS scheme for encrypted machine learning, yet found a minor reduction in operating performance because of the encryption implementation. The balance between security and operational capabilities through HME remains at practical levels, which proves its suitability for diverse analytical tasks with privacy protection. Liu [22] examines how to implement fully homomorphic encryption (FHE) within privacy-preserving machine learning (PPML). FHE delivers stronger data confidentiality, but the study identifies important implementation challenges that require optimization solutions to improve practical usage. Goyal et al. [23] investigate HME-based approaches for privacy-preserving machine learning through Ensemble Learning and Secure Model Aggregation. The research demonstrates IoT application success with MSE accuracy reaching between 0.018 and 0.027 while maintaining satisfactory

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

security standards. Beshaj and Hoefler [24] demonstrate how HME functions within federated learning platforms through which model training operates on encrypted information. The method enables confidential data protection among multiple participants as it allows protected collaborative learning operations without disclosing raw information. Ullah et al. [25] evaluate homomorphic encryption (HE) as a data security enhancement method for IoT and lightweight computing ecosystems. In the study, the private and public key encryption method is used for encrypted data processing, thus solving the problem of privacy, and also ensuring that the impact of the system can be effectively reduced while the needs of information noise can still be met. In Kamble et al. [26], it is proven that HE can be used to operate on encrypted data without compromising its confidentiality. The paper shows how this approach can be successfully implemented in Multi-Cloud systems, as such systems involve distributed networks and need secure handling of data across the networks. Hamza [27] investigates how HE, combined with artificial intelligence (AI), operates to safeguard private data during processing. Widely adopting HE requires solving three main software engineering problems involving performance limitations, usability barriers, and interoperability challenges. The medical sector utilises the Paillier cryptosystem homomorphic encryption as described by Balaji et al. [28]. The research shows how HE operates to calculate average blood pressure measurements and group patient health records through encryption methods, which protect medical confidentiality, thus proving real-world healthcare applications. Lu et al. [29] HE represents a strong encryption technology that enables secure token-based processing for untrusted servers. The study demonstrates how HE enables clients to store encrypted information on untrusted servers without revealing their data through their work on secure cloud-based processing of ciphertexts. Gandhi et al. [30] proposed a secure healthcare analytics framework using homomorphic encryption and collaborative machine learning to protect patient data during training and prediction. Their approach ensured data confidentiality while maintaining high prediction accuracy. Yanez and Yadav [31] studied homomorphic encryption for secure healthcare artificial intelligence, enabling encrypted medical data processing without decryption. Their work improved privacy protection and supported secure AI-based diagnosis and cloud deployment. Allavarpu et al. [32] developed privacy-preserving machine learning techniques using homomorphic encryption for credit risk analysis. Their method secured sensitive financial data while preserving prediction accuracy in credit evaluation systems.

3. Proposed Methodology

As shown in Figure 1, the proposed secure machine learning system using homomorphic encryption consists of a client and a server. The proposed secure machine learning system using homomorphic encryption includes a client and a server, as shown in Figure 1. The owner of the data owns the plaintext data and creates a public–secret key pair. The public key, along with the data as input, labels, and the model parameters, is encrypted using the CKKS homomorphic encryption scheme. These encrypted data and parameters are then sent to an untrusted cloud or server, where machine learning algorithms are all performed in the encrypted domain, including Logistic Regression, Encrypted Fully Connected Neural Networks (EncFCNN), and Encrypted Convolutional Neural Networks (EncCNN). The training process and the inference process take place on the client side with respect to the homomorphic computation without revealing any information about the plaintexts to the server. After the encrypted computation, the predictions and performance metrics are sent back to the data owner, but in an encrypted form. Lastly, the results are decrypted only by the data owner who has the secret key to decrypt the results to get the final outputs. This approach guarantees data confidentiality from end-to-end, while preserving the accuracy of the models, making it ideal for applications that require privacy protection, like healthcare, finance, and secure cloud-based AI services.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

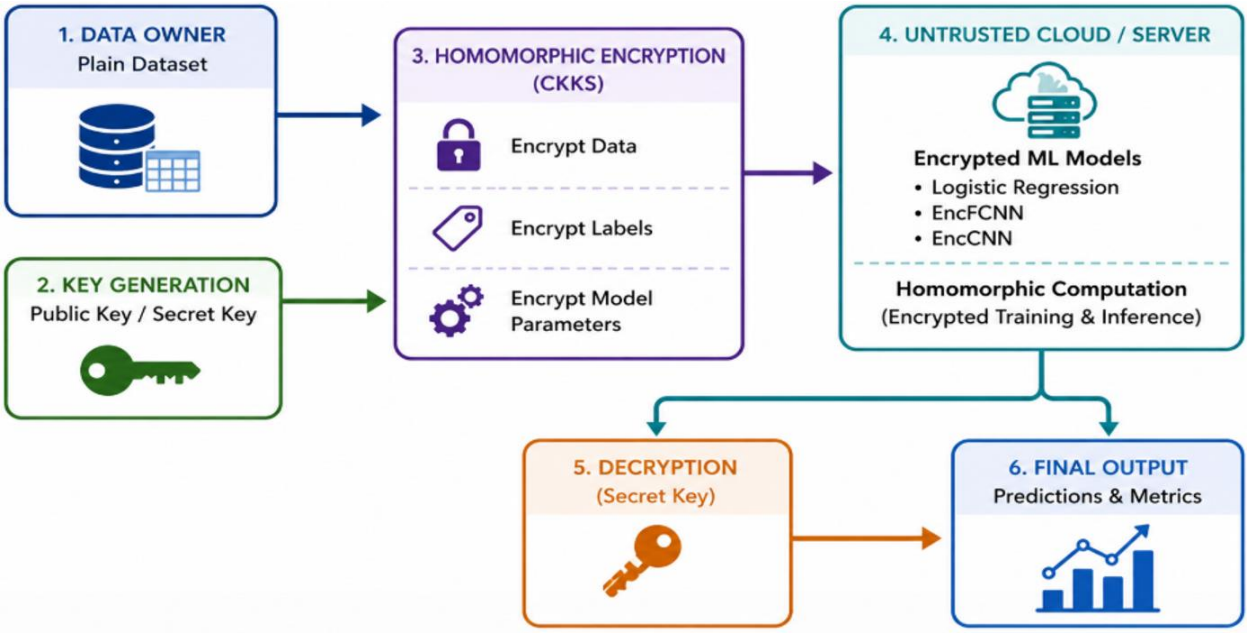


Figure 1.
Block diagram of the proposed privacy-preserving machine learning framework using homomorphic encryption.

We have developed our baseline Encrypted Fully Connected Neural Network (EncFCNN) model, and the detailed encryption workflow used in our study is shown in Figure 2. The figure shows a visual representation of the processing of encrypted data through the model while preserving their privacy. The EncFCNN model is used as a basis for computing privacy-preserving operations with homomorphic encryption. In contrast to the typical FCNN models, where sensitive data is fed into the network for training and computation, our EncFCNN is built to work with encrypted inputs, keeping sensitive data secure during the calculation process. The following are the crucial steps of the encryption workflow.

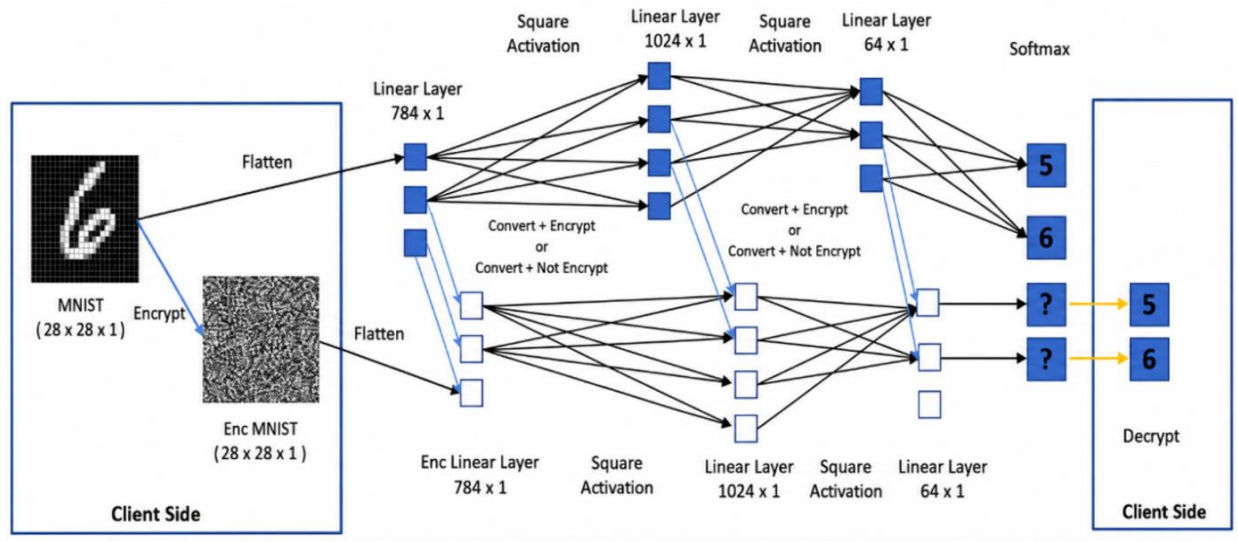


Figure 2.
Proposed Enc FCNN model and THE Encryption Workflow.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

3.1. Data Evaluation on Encrypted Model

Homomorphic encryption demonstrates ways to do operations on encrypted data without letting the operator know specific information. In the context of machine learning, a linear unit is the essential part of logistic regression, convolutional neural networks, and fully connected neural networks, and it outputs the result as the matrix multiplication between the input and the weight, plus the bias. Therefore, the simplicity and repetitive nature of the linear unit enable such a structure to work with homomorphic encryption by encrypting machine learning weights and biases and directly applying them to the encrypted data. The service provider thus calculates the encrypted results and returns them to users with full privacy. Three types of machine learning models are compared within this section. We implemented a logistic regression, a simplified convolutional neural network model, and a fully connected neural network model within PyTorch to compare the performance between the encrypted model and the non-encrypted models.

3.2. CNN Model

Figure 3 represents the architecture of the proposed baseline Convolutional Neural Network (CNN) model for processing MNIST handwritten digit images. We implemented three machine learning models as baselines for the model encryption, encrypted data evaluation, and performance comparison. To represent the single unit of our model, let x = (x1, x2, ..., xn) be our input data with dimension n, the weights are represented as w = (w1, w2, ..., wn), and the bias is represented as a scalar b. The linear regression is represented as f(x) = xw + b. Meanwhile, the sigmoid function can be represented as Sigmoid(x) = 1 / (1 − e−x). The logistic regression uses the sigmoid function to distribute the outputs with a probability between [0, 1]. The logistic regression is basically f(x) = Sigmoid(wx + b), which gives a value within [0, 1].

For the fully connected neural networks, we combined a list of units that are fully connected to the input as a layer, and we can further implement hidden layers that are connected to it. For each unit of the layer, it truly is the single linear unit that is mentioned in logistic regression. It is more complicated to implement the Convolutional Neural Networks. Besides the linear layers in the model, the core part of the CNN is the convolutional layer, which enables the network to extract spatial information from the image and perform the recognition. The convolution layer works by moving and multiplying a kernel filter that extracts the image information. As for the activation function, a squared activation is used in order to parallel the EncCNN.

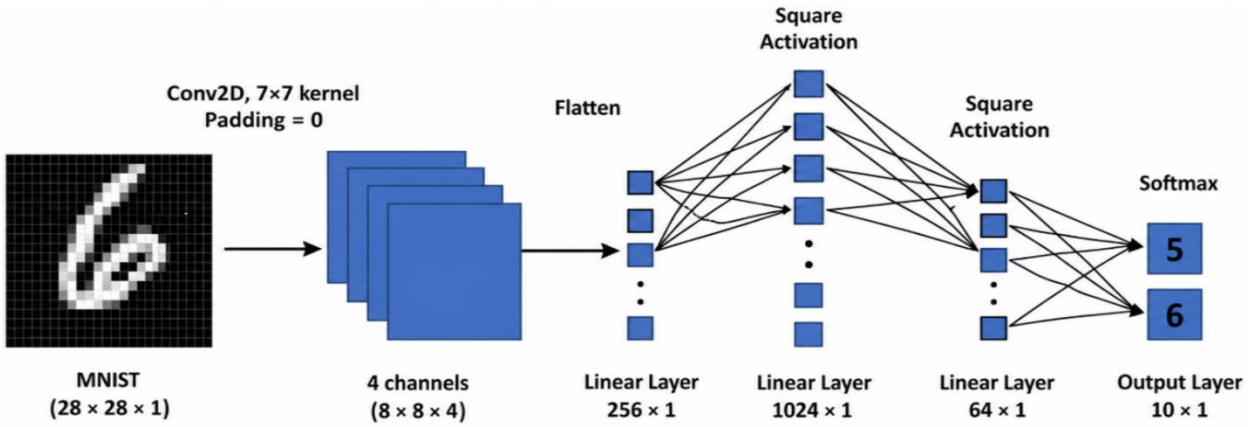


Figure 3.
Proposed Baseline CNN model.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

3.3. Model Encryption and Evaluation on Encrypted Data

For an encrypted evaluation on an encrypted model, we started by encrypting a single linear unit, which is simple to explain. Given that f(x)=Sigmoid(wx+b) with input size n, the encrypted weights are represented in equation (1).

$$Enc(w) = (Enc(w1), Enc(w2), \ldots, Enc(wn)) \quad (1)$$

The encrypted bias is represented as Enc(Bias) = Enc(b), and the encrypted input is represented as Enc(x) = (Enc(x1), Enc(x2),..., Enc(xn)). Therefore, the encrypted linear model can be represented as Encf(Enc(x)) = Enc(w)Enc(x)+Enc(b). For the logistic regression, we enabled the calculation of encrypting the weight and the bias of the single unit to carry out an encrypted evaluation on an encrypted model.

For the evaluation of encrypted data, the parameters of FCNN models are adjusted to be able to work with CKKS vectors, and it is named the EncFCNN model. The prediction overhead was tested by directly applying the real trained parameters to homomorphically encrypted data.

For evaluating encrypted data on EncCNN models, the major obstacles are that the CKKS vector's multiplication can only be carried out a certain number of times, and there's a bounded degree that the vector won't exceed. Therefore, we managed to avoid the implementation of the traditional kernel-filter operation, which requires the CKKS vector to exceed its maximum degree. In our model, the matrix-multiplication insights from TenSEAL are adopted to enable the convolutional layer, where they provided a column convolution that is compatible with CKKS vectors and avoids exceeding the degree bound, which is used for our encrypted convolutional neural networks. The overall structure of the encrypted evaluation is identical to the EncFCNN models except for the convolutional layer.

3.4. Training Models on Encrypted Data

3.4.1. Data Encryption

In the process of homomorphically training a logistic regression model, maintaining the integrity and accuracy of computations while preserving privacy is a critical challenge. Homomorphic encryption feasibility in this context significantly depends on the multiplicative depth value, which indicates the number of encrypted data multiplications before the need to re-encrypt or use bootstrapping. The cryptographic scheme needs to enable six consecutive multiplications for correctly performing logistic regression training in encrypted form. We use a polynomial modulus degree of 8192 to fulfill the requirements of lattice-based encryption schemes, particularly the CKKS (Cheon-Kim-Kim-Song) homomorphic encryption scheme. The security requirements align with computational needs through the selection of 8192 as a suitable modulus value that delivers adequate precision during practical encrypted operations. The security and accuracy levels improve as the polynomial modulus degree increases, but this enhancement leads to more resource-intensive calculations. The selection of appropriate encryption parameters allows our training process to be both secure and computationally feasible, thus making homomorphic logistic regression suitable for privacy-protecting machine learning implementations.

3.4.2. Training

In this section, we implemented a logistic regression model on plain data and a logistic regression model on encrypted data. To evaluate the encrypted model's performance, we made the two implementations as close as possible, though some modifications must be made to the encrypted version. We split the dataset into a training set and a testing set with a ratio of 7:3. The logistic regression training is carried out as follows: we use stochastic gradient descent to optimise our model parameters, and we use binary cross-entropy loss to measure the loss. More specifically, for each epoch, the program first calculates the loss of the current model parameters on the training dataset and then updates the parameters. For the plain data training, we have the built-in optimiser and loss criterion of PyTorch. For the encrypted version, we cannot use the built-in optimiser and loss criterion, so we explicitly updated the parameters, and we would talk about it in the following subsection "parameter update".

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

3.4.3. Sigmoid Approximation

We cannot compute the sigmoid on encrypted data. Instead, we could use a low-degree polynomial to approximate the sigmoid function. We used this approximation of the sigmoid function in the range [-5, 5] as given by equation (2).

$$Sigmoid(x) = 0.5 + 0.197x - 0.0004x^3 \quad (2)$$

4. Result and Discussions

The target of the experiment is to learn the efficiency and the effectiveness of our models. All experiments are run within the same environmental settings. Different data sets are used for comparing different baselines, which are suitable for different tasks. We use the Pima Indian dataset for comparing the speed and accuracy of the logistic regression baseline and the encrypted logistic regression. Since the Pima Indian dataset has a single column, "Outcome," with values of 0 or 1, we chose to train a logistic regression model on it. For comparing performance over the neural networks, the MNIST [33] image dataset is used for comparing the performance. The MNIST is an image dataset that includes images of 10 kinds of handwritten digits, which essentially forms a multi-class classification task. Since the MNIST dataset is always used as a performance comparison set in deep learning, evaluations and encrypted evaluations on image classification deep learning models are run on this dataset. In our experiment, we will include the data on computational speed and accuracy of CNNs, FCNNs, EncCNN, and EncFCNNs. For the neural networks, the time overhead between the propagation of each layer and the total time overhead for each prediction will be measured within the same environment. Furthermore, the accuracy of the encrypted prediction and non-encrypted prediction will also be compared to check the effectiveness of such a scheme.

4.1. Homomorphic Evaluation - Logistic Regression

We want to make sure that when evaluating encrypted data using encrypted parameters, the program maintains the same level of accuracy. After training for 5 epochs using stochastic gradient descent optimisation and binary cross-entropy loss, the model achieves an accuracy of 0.68125 on the testing set. When evaluating on encrypted data using encrypted parameters, the model achieves an accuracy of 0.675 on the same data. Evaluating on the encrypted test set doesn't affect the accuracy that much.

4.2. Homomorphic Evaluation - Fully Connected Neural Network

The evaluation is carried out using the MNIST data set, while the baseline gives a pretty high accuracy with high speed. We tested the baseline with 10,000 MNIST digit images, and the model gives 96.15% accuracy on the testing set, while the computation overhead is small. The Encrypted Fully Connected Neural Network has a huge evaluation overhead, which is approximately 102.8 s for a single MNIST image in a preset environment. The majority of the evaluation overhead is on propagating the first two layers, which are relatively large. Due to the huge overhead, we limited the test set to 50 images; the accuracy is good on EncFCNN even with the noise standing on our side.

Table 1.
Performance Comparison between Homomorphic Evaluated Deep Learning Models on the MNIST data set.

Model	Eval Overhead	Accuracy	Test Size
FCNN	23.37 ms	96.15%	10000
CNN	62.09 ms	98.28%	10000
EncFCNN	102.8 s	98%	50
EncCNN	10.74 s	100%	100

Table 1 presents a comparative analysis of performance metrics for different deep learning models evaluated homomorphically on the MNIST dataset. Since homomorphic encryption (HE) introduces a

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

significant computational burden, the evaluation overhead, defined as the time required to process a single image under encryption, is a crucial factor in assessing the practicality of privacy-preserving machine learning models. The evaluation overheads represent the time it takes to homomorphically evaluate a single picture, while the testing size is varied because the overhead is huge on homomorphic evaluations. The polynomial modulus degree is 8192, and the result comes from a Core-i7 8550U CPU@1.80GHZ.

Table 2.
Performance Comparison between neural layers within the EncFCNN.

Layer	(Input, Output)	Eval Overhead
FC1	(784,1024)	54.21 s
FC2	(1024,64)	42.45 s
FC3	(64,10)	0.9442 s

Table 2 presents a detailed performance comparison of different neural layers within the Encrypted Fully Connected Neural Network (EncFCNN), focusing on how homomorphic encryption affects computation at each stage of the model. Since fully connected neural networks (FCNNs) consist of multiple layers that perform matrix multiplications, activations, and transformations, it is essential to analyse their individual performance to understand the computational cost introduced by encryption. The evaluation overheads represent the average time for the first five image data to propagate through the single fully connected linear layer. The polynomial modulus degree is 8192, and the result comes from a Core-i7 8550U CPU@1.80GHZ.

4.3. Homomorphic Evaluation - Convolutional Neural Network

The evaluation still uses the MNIST dataset, while the baseline gives a pretty high accuracy at a high speed. We tested the baseline with 10,000 MNIST digit images, and the model gives 98.28% accuracy on the testing set, while the computation overhead is small. The EncCNN has a moderate evaluation overhead, which is approximately 10.74 s for a single MNIST image in a preset environment. The EncCNN gives an accuracy of 100% within the first 100 MNIST images in the test set, which proves the validity of the model on homomorphic evaluation.

Table 3.
Performance Comparison between neural layers within the EncCNN.

Layer	(Input, output)	Eval overhead
Conv2D	(1,4)	1.099 s
FC1	(256,64)	7.433 s
FC2	(64,10)	0.6196 s

Table 3 presents a performance analysis of individual neural layers within the Encrypted Convolutional Neural Network (EncCNN), focusing on how different layers contribute to computational overhead in a homomorphic encryption setting. Since CNNs rely on convolutional operations, activation functions, and fully connected layers for classification tasks, evaluating their performance in an encrypted environment is crucial to understanding computational bottlenecks. The evaluation overheads represent the average time for the first five image data to propagate through the mentioned neural layers. The polynomial modulus degree is 8192, and the result comes from a Core-i7-8550U CPU@1.80GHZ.

4.4. Training Logistic Regression on Encrypted Data

For training a logistic regression model on encrypted data, we not only want to test the model's accuracy but also want to measure the program's running time, because we expect the homomorphic training process to consume more time than the plain-data version. On average, encrypting the training

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

set takes 8–10 seconds (encryption of 768 rows * 0.7 * 9 columns of real numbers). Five epochs of training take approximately 200 seconds on average. The model, when evaluated on the test set, yields an accuracy of 0.7375, which is even better than the one trained on plain data.

Table 4.
Datasets Selected for the Proposed System.

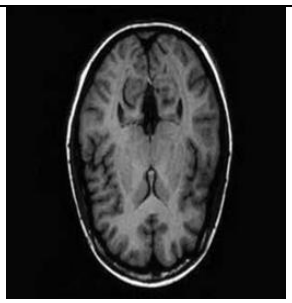
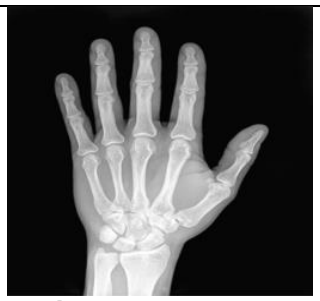
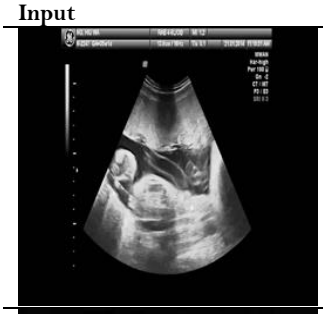
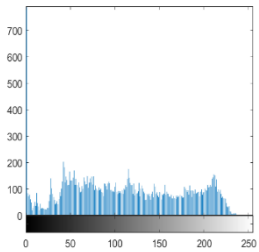
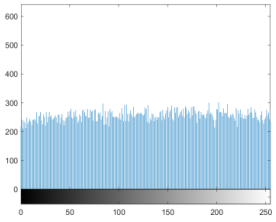
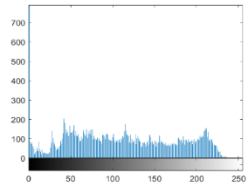
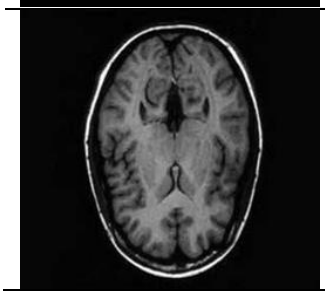
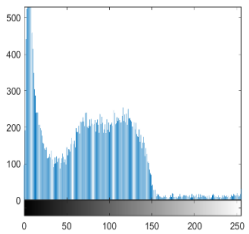
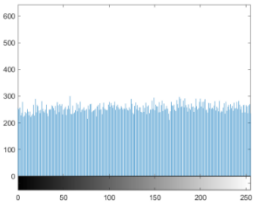
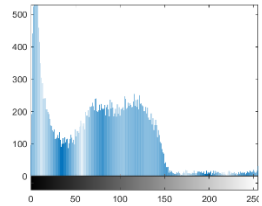
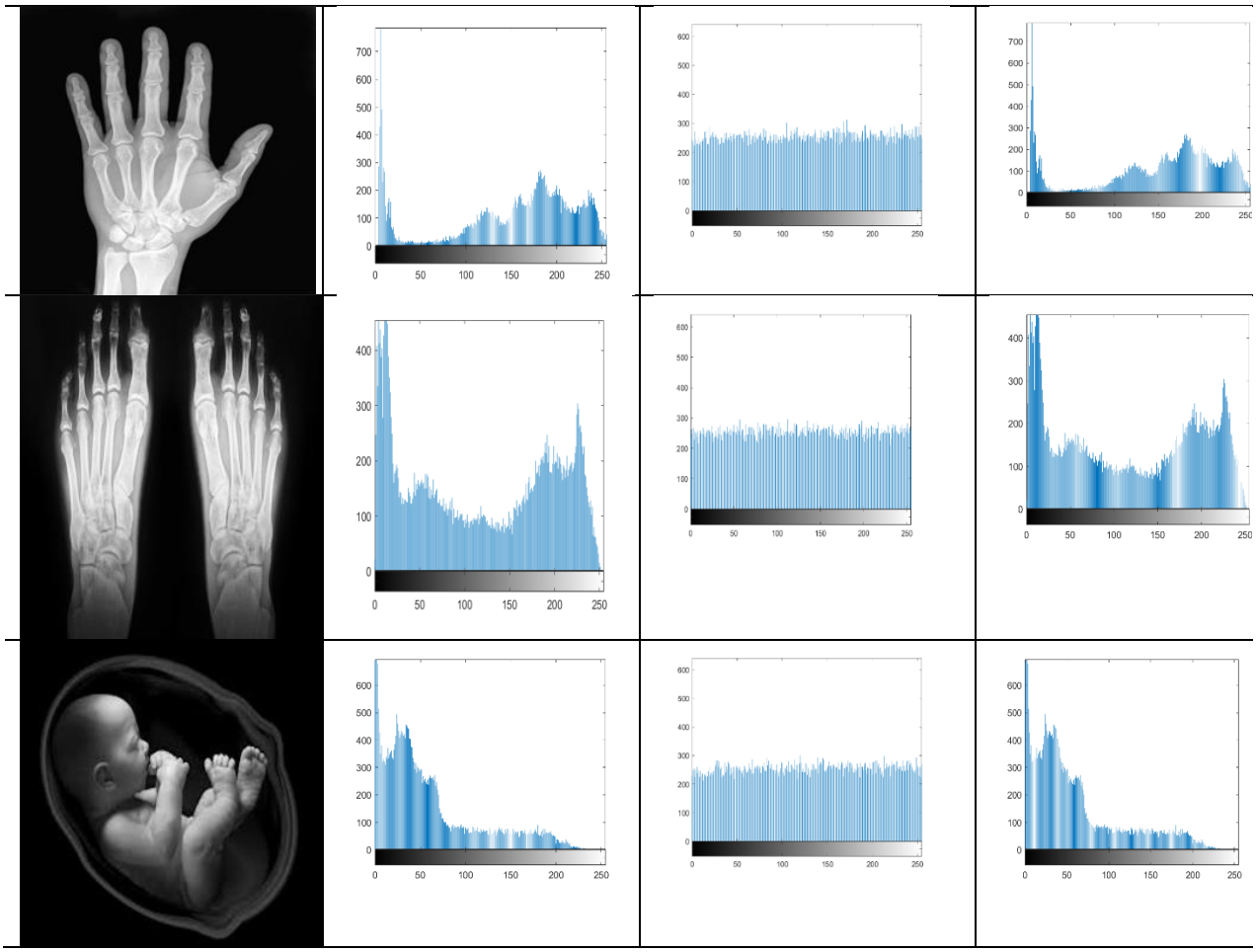
 **Fetus**	 **MRI**	 **Hand**
 **Foot**	 **Baby**	

Table 5.
Performance Analysis Using Histogram in the Proposed System.

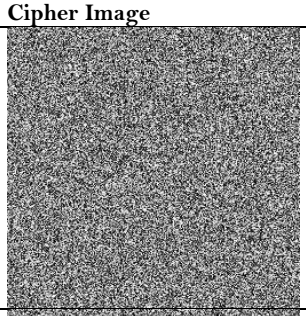
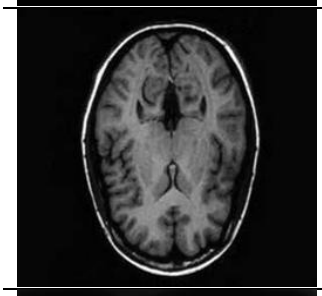
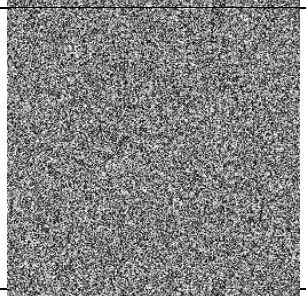
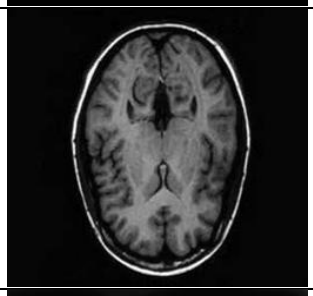
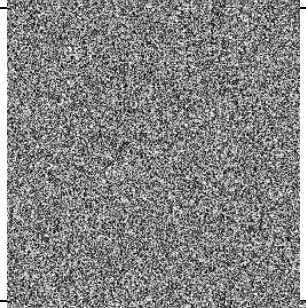
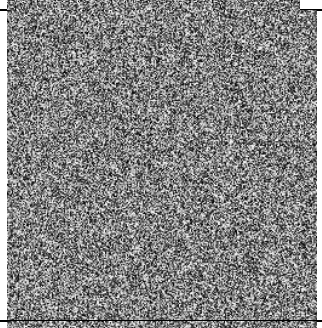
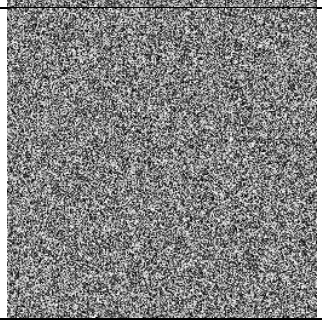
Input	Input Histogram	Cipher Histogram	Decrypted Histogram
			
			

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145



Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
***DOI**: 10.55214/2576-8484.v10i6.13145*

Table 6.
Image Encryption and Decryption Results of the Proposed System.

Images	Cipher Image	Decipher Image
		
		
		
		
		

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

Table 7.
Entropy study of the suggested approach.

Input	Input Entropy	Cipher Entropy
Fetus	3.5284	7.9012
MRI	4.1876	7.9285
Hand	4.4028	7.9021
Foot	5.7689	7.9043
Baby	5.2147	7.9018

Table 8.
Differential attacks evaluation of the suggested approach.

Input	NPCR (%)	UACI (%)
Fetus	98.74	44.92
MRI	99.83	40.31
Hand	98.57	43.85
Foot	99.76	39.12
Baby	97.98	45.26

Table 9.
Image quality assessment of the proposed method.

Input	MSE	PSNR (dB)
Fetus	48.1638	18.2847
MRI	50.1284	19.7421
Hand	45.9187	17.9543
Foot	54.8426	29.1489
Baby	46.2075	26.7814

As shown in Table 4, the histogram analysis of the plain image, encrypted image, and decrypted image with the proposed encryption method is analysed. The distribution of the encrypted images is almost uniform and random, as observed in the histogram distribution, and is similar to that of white noise. The uniformity shows that the statistical characteristics of the original image are well hidden, making it extremely difficult for attackers to recover meaningful information by histogram-based attacks. In contrast, the images that have been decrypted are able to restore the original image properties, thus confirming the accuracy and reliability of the image decryption process.

In Table 5, a visual comparison of the original, encrypted, and decrypted images is shown. The original visual content is completely hidden from the unauthorised user, as the encrypted image looks completely distorted and unintelligible. This visual transformation is crucial in secure image-transmission systems for confidentiality. In addition, the decrypted image is very close to the original image, which shows that the encryption–decryption mechanism is correct and that the image quality is not degraded by the successful secure image transmission.

The proposed image encryption system is compared with the input image and the decrypted image in Table 6 to display the visual output. Table 6 displays the input image, cipher image, and decrypted image to show the visual output of the proposed image encryption system. The input image is the original medical image that is encrypted. With the application of the proposed encryption algorithm, the image of the cipher becomes completely scrambled, irregular, and incomprehensible to the naked eye, so that the information in the image is completely hidden from third parties. This means the level of effectiveness of the encryption technique in maintaining confidentiality and resisting visual attacks on images.

The decrypted image will be the output after performing the decryption process on the cipher image. It is very similar to the original input image and only has minor distortions, showing that the proposed decryption mechanism is accurate and reliable. This shows the original image can be recovered from this system with little information loss. The image details are clearly recovered, verifying the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

effectiveness of the proposed encryption-decryption system in terms of security and quality of reconstruction, which can be used for secure transmission and storage of medical images.

Important security parameters include entropy, Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI), which were used to evaluate the performance of the proposed system, and the results are summarised in Tables 7–9. The entropy value calculated for encrypted images is found to be very close to the ideal value of 8 for grayscale images, which means that there is a high degree of randomness and unpredictability in the encrypted image. This is an affirmation that the encryption process has indeed removed the correlation between pixels and made them resistant to statistical attack.

Furthermore, the computed MSE and PSNR values between the original and encrypted images show that there is significant distortion during the encryption process, which is desirable for secure encryption systems. A higher MSE value and PSNR value indicate that the encrypted image is vastly different from the original image, making it difficult for an unauthorised individual to recover the image. Likewise, the NPCR and UACI values satisfy the demanded level of security, and they show high sensitivity to the minor variation present in the input image. Even a slight modification of one of the original image's pixels causes substantial differences in the encrypted image, demonstrating good differential-attack resistance.

The overall results obtained prove that the encryption system proposed is secure, with high security and strong randomness, effective confidentiality, and robustness against statistical and differential attacks. This makes the proposed method a very suitable secure transmission of medical or sensitive images with high security requirements.

4.5. Security Analysis

TenSEAL uses the CKKS scheme to encrypt and decrypt vectors of numbers. In the CKKS scheme, when encrypting a message, a small error will be inserted into the message to guarantee the security of the hardness of the ring-LWE (RLWE) problem, which is assumed to be hard. The CKKS scheme is believed to satisfy IND-CPA security based on the hardness assumption of ring-LWE [4].

Baiyu Li and Daniele Micciancio proposed that the IND-CPA security achieved by CKKS might not be strong enough in practice and presented passive attacks on the scheme, primarily by inferring the error used in encryption from the decrypted results [8]. They were able to recover the key with high probability and modest run time. SEAL has not been modified for this attack, but it is noted to users that the decryption results should be treated as private information only available to the secret owner, therefore preserving some security. Other HE libraries based on CKKS presented mitigation strategies against this attack, and shared the same idea to attach a proper error at the end of the decryption process to avoid the linearity on the secret key [3].

Our proposed scheme is built upon the security foundations of the CKKS (Cheon-Kim-Kim-Song) vector encryption scheme and Microsoft SEAL (Simple Encrypted Arithmetic Library), ensuring robust data privacy during computations. Here, the data owner is the only one who holds the secret key, and therefore the only one who can encrypt and decrypt the data, which means that no unauthorised parties can access it. The encryption process is based on principles of homomorphic encryption (HE), which means that calculations can be made directly on the encrypted data without ever revealing the raw data. The homomorphic evaluation is done in the encrypted domain, i. e., without ever seeing the plain text data by the model owner. This ensures that even if calculations are performed in untrusted computing environments, sensitive data will be kept confidential. Our approach uses the widely used security of the CKKS vectors and Microsoft SEAL, thus preserving the same level of encryption security as the underlying cryptographic frameworks. The strength of the proposed security model is that it allows for the confidentiality of the data even through the computational process. The model can be used to reduce the risk of data leaking or unauthorised access by ensuring that all operations are carried out on encrypted data without ever decrypting it. This includes resistance to various possible attacks, including ciphertext analysis, side-channel attacks, or attempts to extract partial or full information from

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

encrypted computations. Therefore, we propose a scheme that provides a secure base for privacy-preserving machine learning workflows. This kind of security is particularly important in sectors where data sensitivity is a key concern, including healthcare, financial services, and AI services that are managed remotely on the cloud, where data must be ensured through effective outsourcing and computation. Overall, our approach respects the privacy of the data and also makes sure compliance with strict security guidelines, making it an extremely practical service for applications where privacy is critical in today's data-driven ecosystem.

5. Conclusion and Future Scope

In this paper, we used the CKKS vectors with a pre-trained model in this paper, with which we could calculate on the client's encrypted data. We implement the ability to homomorphically compute data from the data owner for EncCNN and EncFCNN and obtain the same accuracy as the baselines for the data owner. The time is still not high, as the evaluation overhead is quite high, but still manageable in the real scenario. Moreover, the outsourced model training of an encrypted model on encrypted data is explored using a logistic regression. The model itself is trained using encrypted data and thus is even more secure. Our work shows the viability of accelerating machine learning computations on encrypted data with CKKS vectors, but there are a number of ways to improve the results of this research, which we leave for future investigation and optimisation. An important direction is to increase computational efficiency to decrease the evaluation cost, so that homomorphic encryption becomes more attractive for use in large-scale real-life scenarios. Potential solutions to performance issues include advanced encryption methods, hardware acceleration (such as GPUs or TPUs specifically designed for encrypted computations), and algorithmic optimisations. Furthermore, the methodology can be adapted to more advanced deep learning architectures, like transformers and recurrent neural networks (RNNs), and could be extended to natural language processing (NLP) and time-series applications. A second potential direction is to improve privacy-preserving training methods for encrypted models, which would allow more efficient scenarios with federated learning and secure multi-party computation (MPC). Furthermore, using differential privacy in conjunction with homomorphic encryption could add further layers of protection so that even when models are encrypted, there is no inadvertent leakage of sensitive patterns from the data. In conclusion, the new integration of privacy-preserving AI technologies in sectors such as healthcare and finance and cloud-based services will demand continued research and development of usability, regulation, and scalability issues, paving the way for secure AI adoption in real-world contexts. To wrap up, continued research and development efforts will be needed to address the usability, regulatory compliance, and scalability aspects of the new integration of privacy-preserving AI technologies in sectors like healthcare and finance, and also cloud-based services, preparing the groundwork for the secure and privacy-conscious use of machine learning in real-world environments.

Institutional Review Board Statement:

This study did not involve direct participation of human subjects. All medical images used in this research were obtained from publicly available and fully de-identified sources, including the Open-i (Open Access Biomedical Image Search Engine) repository maintained by the U.S. National Library of Medicine. The datasets contain anonymised medical images and do not include any personally identifiable patient information. Therefore, according to institutional research ethics policies and international publication guidelines, formal Institutional Review Board (IRB) approval and informed consent were not required for this study.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References

[1] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021. https://doi.org/10.3390/fi13040094

[2] R. Podschwadt, D. Takabi, P. Hu, M. H. Rafiei, and Z. Cai, "A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption," *Ieee Access*, vol. 10, pp. 117477-117500, 2022. https://doi.org/10.1109/ACCESS.2022.3219049

[3] B. Pulido-Gaytan *et al.*, "Privacy-preserving neural networks with homomorphic encryption: C hallenges and opportunities," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1666-1691, 2021. https://doi.org/10.1007/s12083-021-01076-8

[4] J. Li, X. Kuang, S. Lin, X. Ma, and Y. Tang, "Privacy preservation for machine learning training and classification based on homomorphic encryption schemes," *Information Sciences*, vol. 526, pp. 166-179, 2020. https://doi.org/10.1016/j.ins.2020.03.041

[5] L. Chen, J. Wang, L. Xiong, S. Zeng, and J. Geng, "A privacy-preserving federated learning framework based on homomorphic encryption," in *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*: IEEE, 2023, pp. 512-517.

[6] H. Gadiwala, R. Bavani, R. Panchal, G. Sakarkar, and A. P. Abiyasa, "Enabling privacy-preserving machine learning: Federal learning with homomorphic encryption," in *2024 10th International Conference on Smart Computing and Communication (ICSCC)*: IEEE, 2024, pp. 311-317.

[7] C. Hong, "Recent advances of privacy-preserving machine learning based on (Fully) Homomorphic Encryption," *Security and Safety*, vol. 4, p. 2024012, 2024. https://doi.org/10.1051/sands/2024012

[8] V. Peluso, E. Malan, A. Calimera, and E. Macii, "Private tensor freezing for an efficient federated learning with homomorphic encryption," in *2024 IEEE 42nd International Conference on Computer Design (ICCD)*: IEEE, 2024, pp. 308-315.

[9] H. Jia, M. S. Aldeen, C. Zhao, S. Jing, and Z. Chen, "Flexible privacy-preserving machine learning: When searchable encryption meets homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 9173-9191, 2022. https://doi.org/10.1002/int.22985

[10] D. C.-T. Lo, Y. Shi, H. Shahriar, B. Deng, X. Zhang, and M.-L. Chen, "Practical considerations of fully homomorphic encryption in privacy-preserving machine learning," in *2024 IEEE International Conference on Big Data (BigData)*: IEEE, 2024, pp. 6330-6335.

[11] K. Nguyen, M. Budzys, E. Frimpong, T. Khan, and A. Michalas, "A pervasive, efficient and private future: Realizing privacy-preserving machine learning through hybrid homomorphic encryption," in *2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC)*: IEEE, 2024, pp. 47-56.

[12] H. Kurniawan and M. Mambo, "Homomorphic encryption-based federated privacy preservation for deep active learning," *Entropy*, vol. 24, no. 11, p. 1545, 2022. https://doi.org/10.3390/e24111545

[13] J. Park and H. Lim, "Privacy-preserving federated learning using homomorphic encryption," *Applied Sciences*, vol. 12, no. 2, p. 734, 2022. https://doi.org/10.3390/app12020734

[14] E. Frimpong, K. Nguyen, M. Budzys, T. Khan, and A. Michalas, "Guardml: Efficient privacy-preserving machine learning services through hybrid homomorphic encryption," in *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, 2024, pp. 953-962.

[15] L. Chen, S. Fu, L. Lin, Y. Luo, and W. Zhao, "Privacy-preserving swarm learning based on homomorphic encryption " in *International Conference on Algorithms and Architectures for Parallel Processing*: Springer, 2021, pp. 509-523.

[16] J. Yuan, W. Liu, J. Shi, and Q. Li, "Approximate homomorphic encryption based privacy-preserving machine learning: A survey," *Artificial Intelligence Review*, vol. 58, no. 3, p. 82, 2025. https://doi.org/10.1007/s10462-024-11076-8

[17] B. Han *et al.*, "PBFL: A privacy-preserving blockchain-based federated learning framework with homomorphic encryption and single masking," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 14229-14243, 2025. https://doi.org/10.1109/JIOT.2024.3524632

[18] F. Effendi and A. Chattopadhyay, "Privacy-preserving graph-based machine learning with fully homomorphic encryption for collaborative anti-money laundering," in *International Conference on Security, Privacy, and Applied Cryptography Engineering*: Springer, 2024, pp. 80-105.

[19] S. Lee, G. Lee, J. W. Kim, J. Shin, and M.-K. Lee, "HETAL: Efficient privacy-preserving transfer learning with homomorphic encryption," in *International Conference on Machine Learning*: PMLR, 2023, pp. 19010-19035.

[20] R. Adamsetty, A. K. Cherukuri, and A. Jonnalagadda, "Securing machine learning models: Homomorphic encryption and its impact on classifiers," *Annals of Mathematics and Computer Science*, vol. 26, pp. 141-158, 2025. https://doi.org/10.56947/amcs.v26.439

[21] M. F. Mahatho, Y. Chen, and S. Zhang, "Performance analysis of machine learning on homomorphically encrypted data," in *2024 16th International Conference on Wireless Communications and Signal Processing (WCSP)*: IEEE, 2024, pp. 691-696.

[22] J. Liu, "Fully homomorphic encryption in PPML: An review," *Applied and Computational Engineering*, vol. 69, pp. 122–128, 2024.

[23] H. R. Goyal, A. H. Shnain, K. K. Dixit, M. Kumar, P. Khurana, and M. Harikrishna, "Secure and efficient data fusion in iot systems using homomorphic encryption and machine learning," in *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, IEEE, 2024, pp. 1634-1639.

[24] L. Beshaj and M. Hoefler, "A look inside of homomorphic encryption for federated learning," in *Disruptive Technologies in Information Sciences VIII*, vol. 13058: SPIE, 2024, pp. 240-249.

[25] S. Ullah *et al.*, "Homomorphic encryption applications for IoT and light-weighted environments: A review," *IEEE Internet of Things Journal*, vol. 12, no. 2, pp. 1222-1246, 2024. https://doi.org/10.1109/JIOT.2024.3472029

[26] A. Kamble, M. J. Moses, and P. Chetan, "Homomorphic encryption and its applications in multi-cloud security," presented at the International Conference on Inventive Computation Technologies (ICICT), 2024.

[27] R. Hamza, "Homomorphic encryption for ai-based applications: Challenges and opportunities," presented at the International Conference on Knowledge and Systems Engineering (KSE), 2023.

[28] V. Balaji, K. Srinidhi, and B. Meena, "A homomorphic encryption compiler for blood pressure analysis," presented at the Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2024.

[29] Y. Lu, K. Hara, and K. Tanaka, "Multikey verifiable homomorphic encryption," *IEEE Access*, vol. 10, pp. 84761-84775, 2022. https://doi.org/10.1109/ACCESS.2022.3197634

[30] B. M. Gandhi *et al.*, "Homomorphic encryption and collaborative machine learning for secure healthcare analytics," *Security and Privacy*, vol. 8, no. 1, p. e460, 2025. https://doi.org/10.1002/spy2.460

[31] P. Yanez and N. Yadav, "Homomorphic encryption for secure healthcare artificial intelligence," *Discover Artificial Intelligence*, vol. 6, no. 1, p. 200, 2026. https://doi.org/10.1007/s44163-026-00920-1

[32] V. D. Allavarpu, V. S. Naresh, and A. K. Mohan, "Privacy-preserving machine learning techniques based on homomorphic encryption for credit risk analysis," *Electronic Commerce Research*, pp. 1-31, 2026. https://doi.org/10.1007/s10660-026-10101-y

[33] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012. https://doi.org/10.1109/MSP.2012.2211477

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 6: 338-354, 2026
DOI: 10.55214/2576-8484.v10i6.13145