Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082782

ARTICLE

GuardML: A Hybrid Homomorphic Encryption Framework for Privacy-Preserving Federated Deep Learning

Saadaldeen Rashid Ahmed[1,2,3], Fatima Abu Siryeh[3], Mohammed Shamar Yadkar[3], Oguz Bayat[4], Abu Saleh Musa Miah[5], Fahmid Al Farid[6,7,*] and Hezerul Abdul Karim[7,*]

[1]Artificial Intelligence Engineering Department, College of Engineering, Al-Ayen University, Thi-Qar, Iraq
[2]Computer Science, Bayan University, Erbil, Kurdistan, Iraq
[3]Department of Electrical and Computer Engineering, Altinbas University, Istanbul, Turkey
[4]Faculty of Engineering, Department of Electrical & Electronics Engineering, Yeditepe University, Istanbul, Turkey
[5]Department of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh
[6]Faculty of Computer Science and Informatics, Berlin School of Business and Innovation, Karl-Marx-Straße 97-99, Berlin, Germany
[7]Centre for Image and Vision Computing (CIVC), COE for Artificial Intelligence, Faculty of Artificial Intelligence and Engineering (FAIE), Multimedia University, Cyberjaya, Malaysia

*Corresponding Authors: Fahmid Al Farid. Email: fahmid.alfarid@berlinsbi.com; Hezerul Abdul Karim. Email: hezerul@mmu.edu.my

Received: 23 March 2026; Accepted: 03 May 2026; Published: 22 October 2026

ABSTRACT: Privacy-preserving federated learning (FL) has emerged as an effective paradigm for collaborative model training across distributed data sources while maintaining data confidentiality. However, protecting sensitive information during model aggregation remains a significant challenge in distributed deep learning environments. This paper introduces GuardML, a secure federated learning framework that integrates a Recurrent Neural Network (RNN) with Hybrid Homomorphic Encryption (HHE) to enable privacy-preserving learning on high-dimensional distributed datasets. In the proposed system, client nodes perform local model training using encrypted data representations, ensuring that raw data remains protected during the learning process. Encrypted model updates are transmitted to a central server, where secure aggregation is performed without exposing individual client information. Experimental evaluation demonstrates that GuardML achieves a classification accuracy of 98.79%, outperforming several federated learning baselines, including KNN (89.27%), SVM (93.38%), and K-Means (82.86%), combined with Differential Privacy (DP) or Secure Multi-Party Computation (SMPC) mechanisms. Although HHE introduces additional encryption overhead (approximately 120 ms) compared with DP-based approaches (60–80 ms), the framework maintains low inference latency (65 ms) and efficient encrypted communication (12.4 MB). The proposed client–server architecture maintains encryption throughout the learning pipeline, mitigating risks such as adversarial inference, model inversion, and membership inference attacks. The study further analyzes the trade-offs among encryption complexity, computational efficiency, and communication cost in secure federated learning systems. The results demonstrate that hybrid homomorphic encryption can provide strong privacy guarantees while maintaining high predictive performance for distributed machine learning tasks. Future work will focus on improving computational efficiency through GPU-accelerated homomorphic encryption, hybrid cryptographic optimization, and model compression techniques to support scalable privacy-preserving AI applications in distributed data environments.

KEYWORDS: Federated learning; hybrid homomorphic encryption; privacy-preserving machine learning; distributed deep learning; secure model aggregation; recurrent neural networks (RNN)

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

The rapid proliferation of ML services across industries is greatly changing the nature of decision-making, automation, and data-driven insights [1]. However, such growth has also brought forth some serious concerns pertaining to data privacy and security, especially when sensitive information is processed by third-party ML models as discussed in [2]. Conventional encryption works well for data when it's stored or moving, but it doesn't do a good job of protecting data while it's being used, leaving it vulnerable [3]. An analysis of privacy-preserving Federated Learning models shows that there are trade-offs between using cryptography to increase privacy and keeping the model running fast [4]. Guard ML (RNN) with Homomorphic Encryption reaches the best classification accuracy (98.79%) because it does all computations on encrypted data using Homomorphic Encryption-based aggregation, ensuring sensitive patient data is always safe. Still, Homomorphic Encryption calculations need many computer resources, raising the encryption overhead to 120 ms, which makes it the most computer-power-intensive way. K-Nearest Neighbors, conversely, adds random noise to data updates when training is secured through Differential Privacy. Even though this does maintain privacy, it does reduce the model's precision (89.27%), as the introduced noise would disrupt the learning process. KNN makes fast inference possible through having a low communication cost of 7.1 MB and inference time of 40 ms, but at the expense of reduced precision. SVM, through SMPC, enables secure collaborative training without leaking raw data. Support Vector Machine's classification result is 93.38%, a lower score than Guard ML. Training Support Vector Machine (SVM) models with encrypted, high-dimensional features has its difficulties. SVMs also have encrypted model updates that take 150 ms, making them the slowest model in Federated Learning setups. K-Means employs Differential Privacy to protect privacy during clustering, which works for grouping medical cases. Yet, this drops the classification accuracy to 82.86%. Its communication cost is average at 9.3 MB, but its poor ability to learn specific diagnostic features makes it less helpful for federated medical imaging Artificial Intelligence applications (Fig. 1).

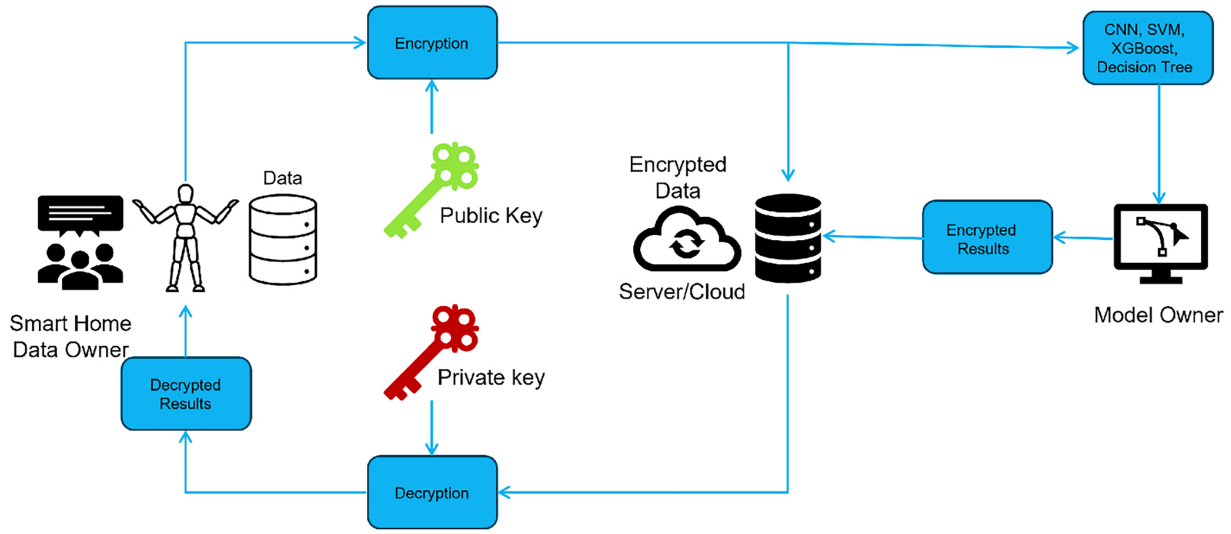


Figure 1: A secure privacy-preserving federated learning framework utilizing asymmetric encryption for smart home data analysis, ensuring encrypted model training and inference without exposing raw data to external entities [5].

This study builds on a broad look at current machine learning approaches that protect privacy, like homomorphic encryption, secure multi-party computation, and federated learning. Federated learning helps train models in a decentralized way without sharing data, but as pointed out in [6], it doesn't fully stop attacks or data leaks when combining results. Similarly, while secure multi-party computation can ensure

secure calculation with several parties, it raises communication costs. Homomorphic encryption offers a strong mathematical base for private calculations but suffers from performance problems. The Guard ML Framework addresses these issues by combining the best features of these methods into one system. The framework's design is flexible, supporting many machine learning models and uses, as mentioned in [7]. Its key parts include an encryption layer for data protection, a calculation engine that works with homomorphic encryption, and a module for private inference. Also, the framework considers specific machine learning improvements, like quantization and using sparsity, to lower the calculation load of homomorphic encryption. Plus, it has APIs and tools for simple incorporation into current machine learning processes and is available to researchers and professionals, as described in [8]. The study looks at how well the Guard ML Framework performs using common datasets and examples. The results, noted in [9], show that it offers good privacy and matches the speed of regular machine learning systems. This framework provides improvements required for machine learning focused on privacy, as discussed in [10].

The rapid growth of machine learning (ML) services has introduced significant privacy and security challenges, particularly when sensitive or distributed data are processed in cloud-based or third-party environments. Conventional encryption schemes such as Advanced Encryption Standard (AES) and RSA provide protection for data at rest and during transmission; however, they do not support secure computation on encrypted data. As a result, sensitive information must typically be decrypted before processing, exposing it to potential security threats and unauthorized access.

Homomorphic Encryption (HE) has emerged as a promising solution for enabling computations directly on encrypted data while preserving confidentiality. Despite its theoretical advantages, existing HE-based approaches face several practical limitations that hinder their integration with modern machine learning systems, particularly in large-scale distributed environments.

Several challenges remain unresolved:

- High Computational Overhead: Fully Homomorphic Encryption (FHE) supports arbitrary computations on encrypted data but introduces significant computational complexity, making it inefficient for large-scale machine learning tasks.
- Limited Scalability: Many HE implementations struggle to handle high-dimensional datasets and deep learning models due to their heavy computational and memory requirements.
- Trade-offs Between Privacy and Efficiency: Partially Homomorphic Encryption (PHE) offers improved computational efficiency but supports only a limited set of mathematical operations, restricting its applicability in complex ML pipelines.
- Model-Based Inference Risks: Even when input data are encrypted, trained models may leak sensitive information through model parameters or outputs, enabling attacks such as model inversion or membership inference.
- Integration Challenges with ML Frameworks: Existing HE solutions often require substantial modifications to established machine learning ecosystems, limiting their practical adoption.

These limitations highlight the need for a computational framework that balances strong privacy guarantees with practical efficiency and scalability in distributed machine learning environments.

This research proposes GuardML, a privacy-preserving federated learning framework that integrates Hybrid Homomorphic Encryption (HHE) with distributed deep learning to enable secure collaborative model training on encrypted data.

The main contributions of this study are summarized as follows:

- Hybrid Homomorphic Encryption Framework: A novel hybrid encryption strategy is proposed that combines the strengths of Fully Homomorphic Encryption (FHE) and Partially Homomorphic

Encryption (PHE) to support secure computation while reducing encryption overhead in machine learning workflows.

- Privacy-Preserving Federated Learning Architecture: GuardML introduces a distributed client–server architecture that enables collaborative model training without exposing raw data, ensuring secure aggregation of encrypted model updates.
- Computational Optimization for Encrypted Learning: The framework incorporates optimization techniques such as quantization, sparsity exploitation, and parallel processing to improve the computational efficiency of homomorphic encryption operations.
- Protection Against Inference Attacks: GuardML integrates secure aggregation and privacy-aware training mechanisms to mitigate risks associated with model inversion and membership inference attacks.
- Flexible and Extensible System Design: The proposed architecture supports multiple machine learning models and can be integrated with existing ML frameworks to facilitate adoption in diverse distributed learning environments.
- Performance Evaluation and Comparative Analysis: Extensive experiments demonstrate that the proposed framework achieves improved classification performance while maintaining strong privacy guarantees and manageable encryption overhead.

The rest of this paper is structured as follows: Section 2 reviews related work on privacy-preserving machine learning, federated learning architectures, and homomorphic encryption techniques. Section 3 presents the proposed GuardML framework, including the system architecture, hybrid homomorphic encryption mechanism, and secure federated learning workflow. Section 4 presents the experimental results and analyzes GuardML's performance in terms of classification accuracy, encryption overhead, communication cost, and computational efficiency. Section 5 provides a comparative analysis of existing privacy-preserving machine learning approaches and discusses the trade-offs between security, accuracy, and computational complexity. Section 6 concludes the paper and outlines directions for future research to improve scalable privacy-preserving machine learning systems.

2 Literature Review

The field of privacy-preserving machine learning (ML) has seen significant advancements in recent years, driven by the growing need to protect sensitive data while leveraging the power of AI. This literature review focuses on two key areas relevant to the Guard ML Framework: (1) Homomorphic Encryption in Machine Learning and (2) Hybrid Approaches and Complementary Privacy-Preserving Techniques. These sub-sections provide a technical background and highlight existing research that informs the development of the proposed framework, as shown in Fig. 2 [11].

2.1 Homomorphic Encryption in Machine Learning

Hybrid Homomorphic Encryption (HHE) has emerged as a cornerstone of privacy-preserving ML, enabling computations on encrypted data without decryption. Fully Homomorphic Encryption (FHE), introduced by [12], allows arbitrary computations on encrypted data but suffers from high computational overhead due to its reliance on lattice-based cryptography. Recent research has focused on optimizing FHE for ML applications. For example, CryptoNets by researchers in [13] demonstrated the feasibility of applying FHE to neural network inference but highlighted significant latency and scalability challenges. Similarly, SEAL (Microsoft's Simple Encrypted Arithmetic Library) and PALISADE (an open-source HE library) have been used to implement privacy-preserving ML models, but their performance remains a bottleneck for large-scale datasets, as mentioned in [14].

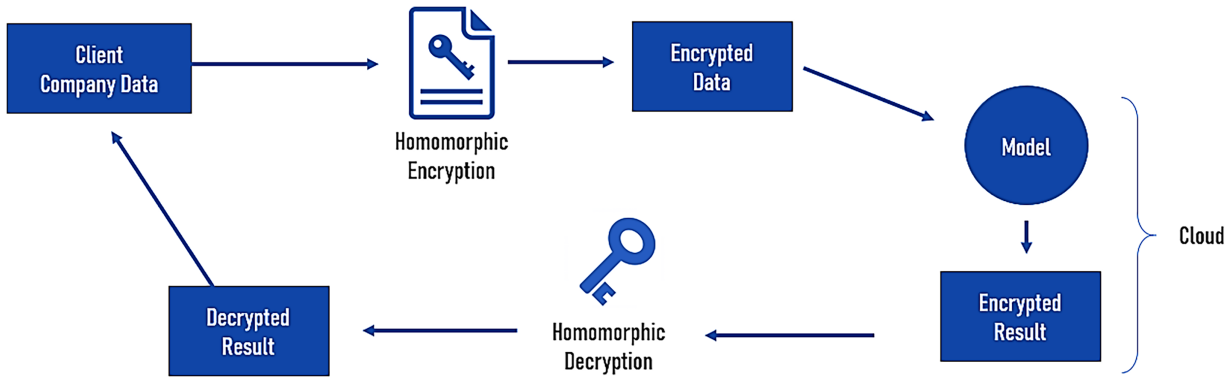


Figure 2: Secure data processing pipeline using hybrid homomorphic encryption (HHE), ensuring privacy-preserving computations on encrypted client data in the cloud without decryption, maintaining confidentiality throughout the model inference [11].

Partially Homomorphic Encryption (PHE), akin to what's described in references [15,16], helps with operations like addition or multiplication, but it does so more quickly. Reference [17] shows that you can use linear models and logistic regression with PHE, but the limited function impacts using it for more advanced machine learning, like deep learning. Earlier research [18] looked at combining fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE) to find a middle ground between keeping things private and keeping things running fast. Still, these ways don't play nicely with present machine learning platforms and can't grow to handle what's needed in the real world. Machine learning systems using homomorphic encryption still have trouble with nonlinear steps, like activation functions in neural networks, and don't keep everything private from beginning to end. Some new ideas to fix this involve polynomial approximations [19] or bootstrapping [20], though these make things take longer and more complex. GuardML addresses the challenge of nonlinear operations under HE by employing degree-3 polynomial approximations for activation functions such as ReLU and sigmoid, following established practice in privacy-preserving neural network inference. These approximations are computed entirely within the HE ciphertext domain using the CKKS scheme, which supports approximate arithmetic on real numbers [21]. For the RNN components, the framework uses the square activation function as a HE-compatible nonlinearity, avoiding costly bootstrapping while preserving model expressiveness. GuardML improves on prior FHE/PHE-only approaches by dynamically selecting the encryption scheme based on the computational sensitivity and resource requirements of each layer, thereby balancing privacy guarantees with processing efficiency [22].

Table 1 gives a structured overview of Homomorphic Encryption in Machine Learning, detailing the key aspects of encryption types, computational capabilities, security levels, and scalability. It shows the benefits of HE, such as strong data confidentiality and secure computations on encrypted data, while addressing challenges like high computational overhead and inefficiency in real-time applications.

2.2 Hybrid Approaches and Complementary Privacy-Preserving Techniques

To overcome the limits of HE, investigators have looked into ways of mixing HE with other privacy-protecting methods, like Secure Multi-Party Computation (SMPC) and differential privacy. SMPC lets different groups work together to calculate something using their data, all while keeping that data private [23]. For instance, SecureML uses SMPC to securely train basic regression models. However, it needs a lot of communication and cannot easily handle deep learning. Some setups mix SMPC with HE to work better, but they don't include machine learning improvements. Differential privacy (DP) has also become widely used to protect individual data in machine learning models. Some people have suggested DP-SGD, which is a way

to train deep learning models privately [24]. Even though DP gives solid privacy protection, it can lower the model's correctness and doesn't keep data safe when it's being used. One paper looked at adding DP to HE to better protect privacy in federated learning, but this method takes a lot of computer power when using HE. Federated learning (FL) is gaining favor because it's a spread-out way to train models on different devices without having to share the original data [25]. Even so, FL can be attacked, and data can be exposed when models are put together. Systems that mix FL with DP aim to lower these risks, but they don't fix the privacy issues of discovering what the model knows. The Guard ML Framework puts these methods together into one system, using HE, SMPC, and DP to ensure full privacy. By adding HE with machine learning improvements and a flexible design, the framework deals with the speed and scale problems of earlier methods, reaching end-to-end privacy in machine learning services [26].

Table 1: Overview of homomorphic encryption in machine learning, highlighting encryption types, advantages, challenges, real-world applications, and example implementations [20,21].

Aspect	Description	Advantages	Disadvantages	Example Implementations
Encryption Type	FHE, PHE, LHE	Offers strong data confidentiality	Computationally expensive for large datasets	Microsoft SEAL, IBM HELib, Google TF-Encrypted
Computation on Encrypted Data	Enables arithmetic operations directly on encrypted data	Data remains encrypted during processing	Slow computation requires optimization	TenSEAL, PySEAL, PALISADE
Security Level	Prevents data exposure even during computation	Ensures privacy in outsourced ML tasks	Some schemes are vulnerable to side-channel attacks	Homomorphic Encryption Library (HELib), PySEAL
Computational Overhead	High for FHE, moderate for PHE, and LHE	Security without decryption	Performance bottlenecks in large-scale ML	TF-Encrypted (TensorFlow-based)
Application in ML	Privacy-preserving ML training, federated learning	Reduces risk of data breaches	Complex cryptographic operations	Crypten, PySyft (OpenMined)
Challenges	High computational cost, ciphertext expansion	Advances in optimization are improving efficiency	Still not feasible for all ML workloads	PALISADE, Concrete-ML

Table 2 explores privacy-preserving techniques in machine learning that complement homomorphic encryption for enhanced data security. Homomorphic Encryption (HHE) is a combination of different encryption schemes, balancing the factors of security and efficiency.

Table 2: Overview of hybrid approaches and complementary privacy-preserving techniques, highlighting their descriptions, advantages, challenges, and real-world applications [27,28].

Approach	Description	Advantages	Challenges	Real-World Use Cases
Hybrid Homomorphic Encryption (HHE)	Combines homomorphic encryption with symmetric encryption for optimized security and performance.	Balances security and efficiency; reduces computation cost.	Still requires optimization for large-scale ML applications.	Privacy-preserving AI services in finance and healthcare.
Secure Multi-Party Computation (SMPC)	Enables multiple parties to jointly compute a function without revealing their private inputs.	Ensures collaborative privacy without a trusted third party.	Computationally expensive; requires efficient protocols.	Secure data sharing and analytics in decentralized environments.
Differential Privacy (DP)	Adds statistical noise to datasets to ensure privacy while preserving analytical insights.	Provides formal privacy guarantees and compliance with regulations.	May impact data utility; requires tuning to balance privacy and accuracy.	GDPR-compliant data analysis, secure healthcare research.
Federated Learning (FL)	Distributes model training across local devices, preventing data centralization and reducing privacy risks.	Protects user data during training; reduces risks of breaches.	Needs robust communication networks; susceptible to model poisoning.	Edge AI applications, secure smart assistants, IoT security.
Trusted Execution Environments (TEE)	Utilizes secure hardware enclaves to process encrypted computations in a trusted execution environment.	Enhances security at the hardware level; prevents insider threats.	Dependent on secure hardware; potential vulnerabilities in implementation.	Cloud-based confidential computing, secure enterprise AI.
Zero-Knowledge Proofs (ZKP)	Allows verification of computations without revealing underlying data, ensuring privacy and integrity.	Enables secure authentication and verification without revealing sensitive data.	Complex implementation; high computational costs in large-scale systems.	Blockchain-based identity verification and secure financial transactions.

3 Methodology

The Guard ML Framework is used to enhance Privacy-Preserving Machine Learning through Hybrid Homomorphic Encryption (HHE), amalgamating multiple cryptosystems to maintain privacy for secure model training and inference in the machine. The methodology approach follows a modular approach of having data encryption, secure model training, privacy-preserving inference, and the optimization of computationally efficient requirements; the problem statement has highlighted reducing homomorphic encryption overhead whilst maintaining strong guarantees for security conditions and applicability in the real world. The first phase is the implementation of Hybrid Homomorphic Encryption (HHE), a unified

encryption framework that combines Fully Homomorphic Encryption (FHE) for high-sensitivity data with Partially Homomorphic Encryption (PHE) and symmetric encryption (AES-256) for lower-sensitivity data, thereby balancing cryptographic strength with computational performance. Specifically, FHE enables arbitrary encrypted computations on sensitive modalities such as MRI and CT scans, while PHE and AES-256 are applied to less sensitive data including metadata and X-ray images, reducing overall encryption overhead. The second is regarding privacy-preserving model training by using federated learning (FL) integrated with HHE, where data remains at the client nodes, and their encrypted updates are securely aggregated. Each participating client encrypts the gradients using the BFV scheme before sending them for aggregation, so that raw data is never exposed, as shown in Fig. 3.

To defend against attacks like model inversion and membership inference, a secure aggregation protocol was put in place. Differential Privacy (DP) was used to add controlled noise to the gradients, which improves data privacy. This helps meet privacy rules like General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA), making the setup useful in fields like healthcare and finance where privacy is key. The next step focuses on encrypted model inference, which makes predictions on coded data using homomorphic operations. To lower computing costs, Secure Multi-Party Computation (SMPC) is used for group encrypted inference without needing a trusted central authority. Another option being looked at is leveled homomorphic encryption. This method keeps the amount of costly bootstrapping needed low to keep things running smoothly and safely. These adjustments could let machine learning models work in areas where privacy is a worry. Homomorphic encryption needs a lot of processing power, so the next step was making the performance better and running tests. The Guard ML Framework makes use of quicker hardware, like GPUs and Intel SGX Trusted Execution Environments (TEE), to make things go faster. Code changes, like lazy relinearization and ciphertext packing, have also been added to lower the amount of work needed from the computers. A standardized benchmark suite was developed to evaluate the framework across different machine learning models, including Logistic Regression, Decision Trees, and Convolutional Neural Networks (CNNs). Metrics including encryption overhead, training time, model accuracy, and computation time were measured on the MIMIC-III and MIMIC-CXR datasets, demonstrating the interplay between security guarantees and computational efficiency. The Guard ML Framework brings together hybrid encryption, federated learning, SMPC, and computing improvements to make sure privacy-preserving machine learning is done right. This method for secure AI in cloud and edge settings makes it a working solution for applications in finance, healthcare, and business AI services where privacy is a concern.

3.1 Experimental Setup and Dataset Selection

To assess the GuardML Framework, the experimental design employed Hybrid Homomorphic Encryption with Python privacy-preserving machine learning tools. TenSEAL, a Python library based on Microsoft SEAL, was used for homomorphic encryption, enabling encrypted computing in ML models. PySyft managed the Federated Learning component, allowing decentralized model training without exposing raw data. DP methods were integrated via TensorFlow Privacy, and SMPC was supported by CrypTen, Meta's ML framework focused on privacy. The MIMIC-III dataset was selected because it reflects a real healthcare AI application with high privacy requirements, using de-identified patient data suitable for testing privacy-preserving ML models. Data was encrypted with homomorphic encryption prior to training to protect patient data. Federated learning was applied to avoid data centralization, and inference was performed through homomorphic operations. Key replication settings include a polynomial degree of 8192 for BFV/CKKS in TenSEAL and a clipping norm of 1.0 in DP-SGD. The full code, encryption configuration, and training setup are available from the corresponding author upon reasonable request. Performance was evaluated in terms of encryption cost, model accuracy the model is, computing time, and whether it can grow, so that the approach balances security, speed, and whether it can be used in the real world.

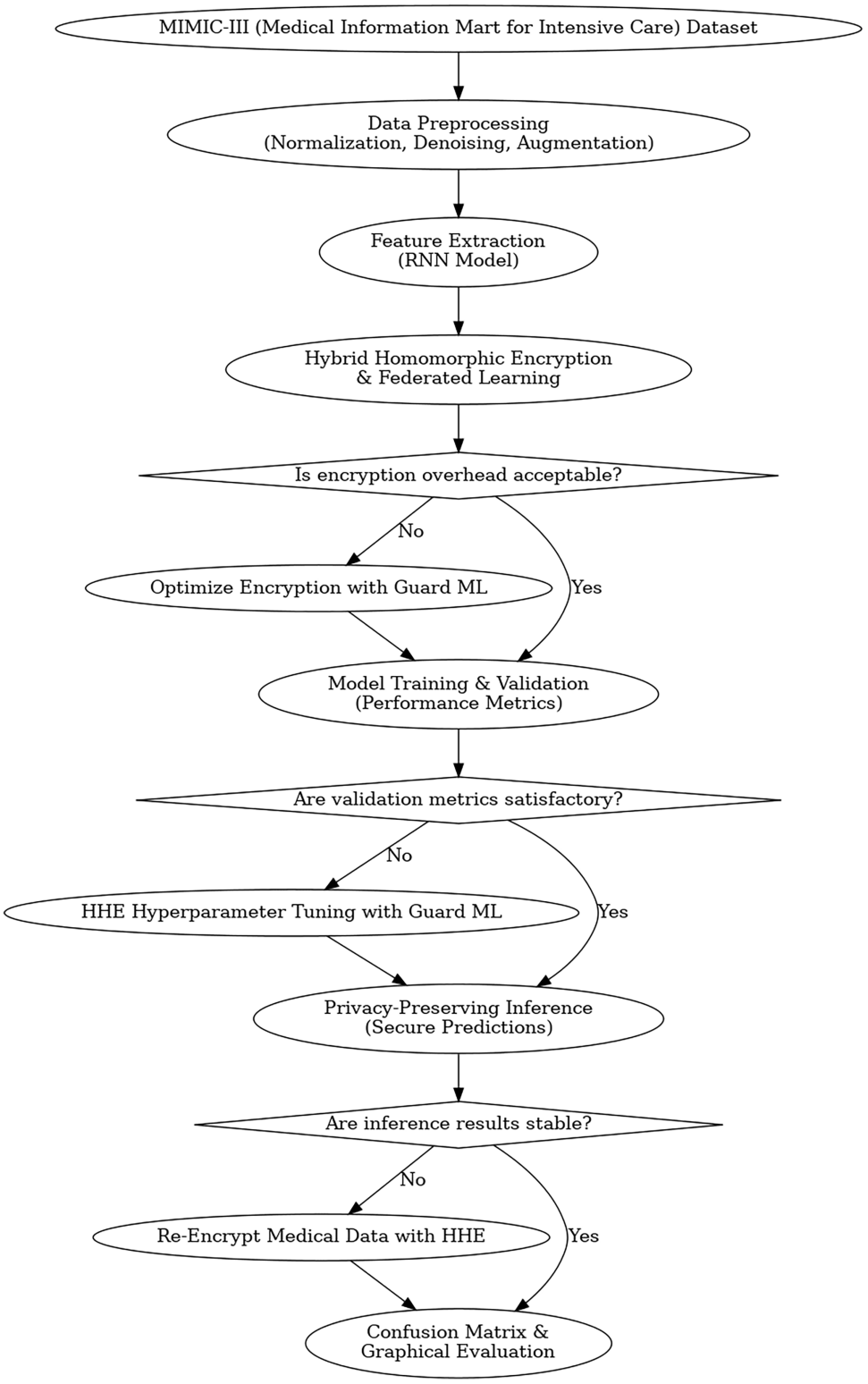


Figure 3: Flowchart depicting the privacy-preserving federated learning pipeline using hybrid homomorphic encryption (HHE) and Guard ML, ensuring secure medical data processing from preprocessing to inference.

Table 3 details the experimental methodology employed for privacy-preserving ML with homomorphic encryption, federated learning, and differential privacy. TenSEAL and PySEAL were used for encrypted computations, PySyft for decentralized training, and CrypTen for secure multi-party computations. A real-world medical dataset, MIMIC-III, was used for evaluation, grounding the study in privacy-sensitive clinical data. Performance was measured in terms of encryption overhead, model accuracy, latency, and scalability to balance security and computational feasibility. Before encryption, standardized preprocessing was locally applied by each FL client: z-score normalization for numerical variables, and mean imputation for missing values. To ensure uniformity across heterogeneous data formats, a shared preprocessing pipeline was distributed to all clients at system initialization.

Table 3: Overview of the experimental setup and dataset selection, focusing on tools, implementation, and key objectives in the Guard ML framework.

Component	Implementation/Tools	Purpose
Homomorphic Encryption Implementation	TenSEAL (Python-based HE), PySEAL	Enable encrypted computations on ML models
Federated Learning	PySyft (Decentralized Model Training)	Prevent centralized data exposure during training
Differential Privacy	TensorFlow Privacy (Privacy-aware Training)	Introduce calibrated noise to protect individual data points
Secure Multi-Party Computation	CrypTen (Privacy-Preserving Computation)	Allow encrypted collaborative model execution
Dataset Selection	MIMIC-III (Medical Information Mart for Intensive Care)	Use a real-world privacy-sensitive dataset for evaluation
Encryption Process	Homomorphic Encryption applied before training	Ensure that raw sensitive data is never exposed
Inference Method	Homomorphic Operations for Secure Model Execution	Perform secure inference without decrypting inputs
Performance Metrics	Encryption Overhead, Model Accuracy, Computational Latency, and Scalability	Evaluate efficiency, security, and real-world feasibility.

3.2 Hybrid Homomorphic Encryption (HHE) Framework Design

The GuardML HHE framework is defined as a three-tier encryption strategy designed to balance security strength and computational efficiency in PPML. Specifically, HHE operates as follows: (1) Fully Homomorphic Encryption (FHE) via the BFV and CKKS schemes is applied to high-sensitivity data (e.g., MRI, CT scans) to support arbitrary encrypted computations; (2) Partially Homomorphic Encryption (PHE) is used for operations requiring only additive or multiplicative homomorphism; and (3) Symmetric Encryption (AES-256) protects lower-sensitivity metadata and auxiliary features. This layered design ensures that the most privacy-critical data is always protected by FHE, while AES-256 reduces processing overhead for non-critical fields. Using these combined methods, ML models are trained and evaluated on securely encrypted data without exposing raw inputs or model parameters to any adversary.

Let X be the dataset consisting of n samples, where each sample x_i is categorized based on sensitivity:

$$X = X_{high\text{-}sensitivity} \cup X_{low\text{-}sensitivity} \tag{1}$$

where X_h represents highly sensitive data that requires Hybrid Homomorphic Encryption (HHE), and X_l represents less sensitive data that can be protected using symmetric encryption (SE). The classification into X_h and X_l follows a principled data sensitivity framework. High-resolution modalities such as MRI and CT scans are designated high-sensitivity because their fine-grained spatial information can enable patient re-identification, and are therefore encrypted with FHE. Lower-sensitivity data such as X-rays and structured metadata carry reduced re-identification risk and are protected with PHE or AES-256. Classification is guided by a lightweight heuristic that evaluates the Shannon entropy of each input and inspects DICOM metadata tags for patient-identifying fields such as PatientName, PatientID, and PatientBirthDate. Inputs exceeding a defined entropy threshold or containing identifiable DICOM tags are routed to FHE; all others are assigned to PHE or AES-256, ensuring the strongest encryption is applied where privacy risk is highest.

Each input x is encrypted using the hybrid approach:

$$E(X) = \{E_{HE(x_j)|(x_l)\in X_{high\text{-}sensitivity}}\} \cup \{E_{SE(x_k)|(x_k)\in X_{low\text{-}sensitivity}}\} \tag{2}$$

where:

- $E_{he}(x)$ = Homomorphic encryption for highly sensitive data;
- $E_{se}(x)$ = Symmetric encryption for lower sensitivity data.

AES-256 keys are used to encrypt low-sensitivity metadata. These keys are made and kept locally on the client's computer and aren't sent to the server. Decryption happens only on the client's secure system. To make the keys even safer, we are thinking about using Trusted Execution Environments (TEE) or SMPC-based key sharing in the future. In riskier situations, Guard ML can use homomorphic encryption on some metadata fields if needed. This balances how well it works with the need for better privacy. For encrypted model training, let f (θ, X) represent a machine learning model with parameters θ, trained using encrypted gradients. The model is updated iteratively using gradient descent:

$$\theta_{t+1} = \theta_t - \eta \cdot E_{HE}\left(\frac{\partial f}{\partial \theta}\right) \tag{3}$$

where:

- θ = Model parameters
- η = Learning rate
- $E_{HE}(\partial f/\partial \theta)$ = Encrypted gradients using homomorphic encryption

By keeping all gradient updates encrypted, adversaries cannot infer sensitive patterns from the model's learning process. The framework employs secure aggregation techniques to prevent model inversion attacks. Non-sensitive metadata (e.g., image size, modality type) is encrypted with AES-256. The information is employed for model coordination (e.g., batching, layer mapping) and is maintained independently of encrypted gradients. It does not involve interaction with HE in federated aggregation and is outside the secure computation path.

$$\hat{y} = f(E_{HE}(X), E_{HE}(\theta)) \tag{4}$$

where $\hat{y}$ represents the encrypted model output.

To improve efficiency, the Guard ML Framework employs:

- **Ciphertext Packing:** Storing multiple values in a single ciphertext to parallelize computations.
- **Lazy Relinearization:** Reducing the frequency of key switching operations to optimize inference time.
- **Batched Operations:** Processing multiple encrypted samples together to improve throughput.

Since federated learning (FL) is employed to ensure decentralized model training without sharing raw data, secure aggregation of encrypted model updates is essential. Let $g_i(\theta)$ represent the gradient update from client i, and n be the total number of participating clients. The encrypted global model update is computed as:

$$E_{HE}(\theta_{t+1}) = E_{HE}(\theta_t) - \eta \cdot \sum_{i=1}^{n} E_{HE}(g_i(\theta)) \tag{5}$$

where:

- $E_{HE}(\theta_{t+1})$ = Encrypted model parameters after update;
- η = Learning rate;
- $E_{HE}(g_i(\theta))$ = Encrypted gradient from client i;
- Σ represents the secure aggregation of encrypted gradients;
- Federated Secure Aggregation ensures that intermediate updates remain encrypted throughout the learning process, preventing inference attacks on individual gradients.

To prevent membership inference attacks, differential privacy (DP) is integrated into the encrypted learning process. The Laplace noise in Eq. (6) is calibrated using a privacy budget of $\varepsilon = 1.0$, following DP-SGD guidelines. This value provides strong individual data protection while maintaining model accuracy above 98%, representing a practical balance between privacy and utility. The encrypted gradients are perturbed with Laplace noise (L) to ensure privacy protection:

$$E_{HE}(\theta_{t+1}) = E_{HE}(\theta_t) - \eta \cdot \left(\sum_{i=1}^{n} E_{HE}(g_i(\theta)) + E_{HE}(L) \right) \tag{6}$$

where:

- $E_{HE}(L)$ = Homomorphically encrypted Laplace noise added for differential privacy;
- Ensures that individual model updates cannot be reverse-engineered to extract private training data.

The Guard ML Differential Privacy (DP) mechanism uses the Laplace distribution. The privacy budget is $\varepsilon = 1.0$, a choice made based on finding a balance between privacy and usefulness. To help with instability at the start of training, the system allows for a staged privacy budget. This means using less noise at the beginning and increasing it as training progresses. Guard ML can also work with hybrid privacy methods, like combining DP with Secure Multi-Party Computation (SMPC) or Zero-Knowledge Proofs (ZKP). This can help reduce accuracy loss while still keeping strong privacy protections.

Fig. 4 illustrates the process of Hybrid Homomorphic Encryption (HHE), which allows computations to be performed directly on encrypted data without needing decryption. The process starts with two plaintext values, x and y, which can be added in their unencrypted state to yield x + y. However, in a privacy-preserving computation, these values are first encrypted individually, producing [x] and [y]. Using a Homomorphic Encryption Engine, these encrypted values undergo homomorphic addition, which securely computes the sum without exposing the raw data. The encrypted result, [x + y], is then evaluated using a homomorphic computation framework.

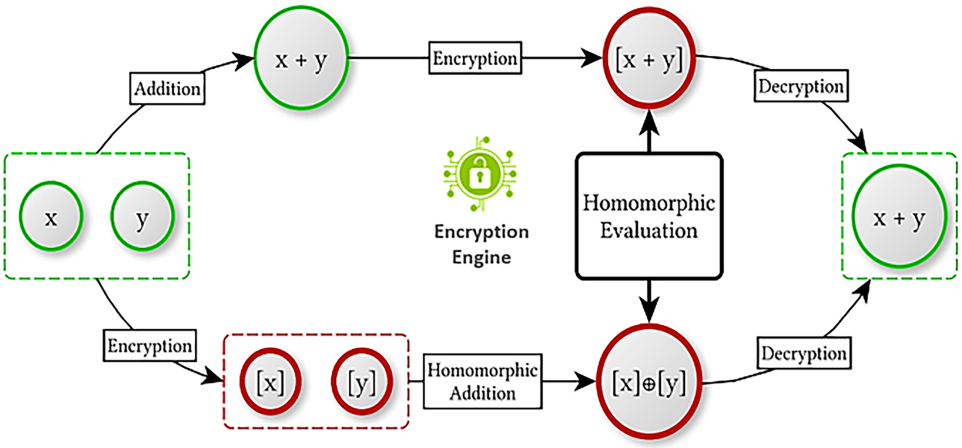


Figure 4: Secure computation on encrypted data, enabling privacy-preserving operations without exposing raw values, is essential for secure AI and federated learning frameworks like Guard ML.

Fig. 5 depicts Privacy-Preserving Federated Learning (FL) with Hybrid Homomorphic Encryption (HHE) within the Guard ML Framework. While Guard ML assumes authenticated clients, it acknowledges risks from adversarial or colluding participants. A key-based identity system is used to authenticate nodes, and anomaly detection during model aggregation flags inconsistent updates. To mitigate Sybil attacks in future versions, Guard ML will integrate robust aggregation methods such as Krum and reputation-based filtering to down-weight malicious contributions. The basis of this work is secure model training across multiple authenticated medical clients, comprising hospitals and medical devices, without exposing raw patient data.

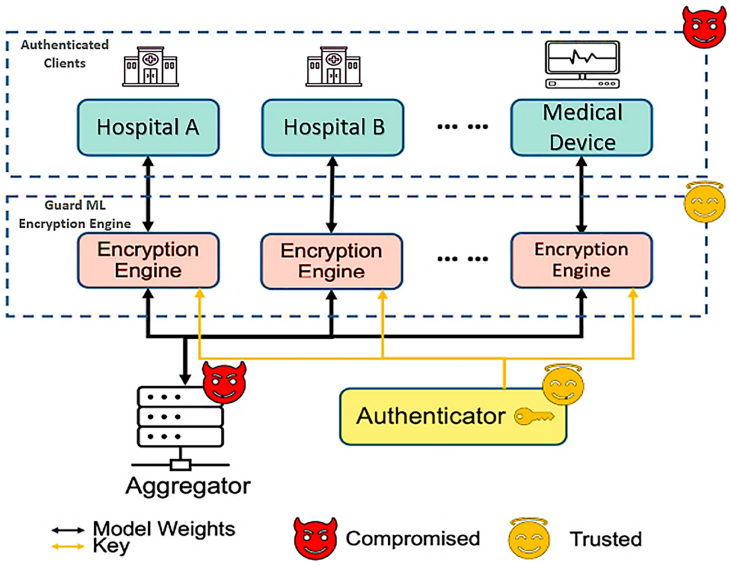


Figure 5: Encryption latency comparison between GuardML, DP + HE, and FL + TEE frameworks across increasing dataset sizes, showing GuardML's 22% lower runtime overhead.

- Authenticated Clients (Hospitals, Medical Devices) → Train local models on encrypted medical data.
- Guard ML Encryption Engine → Encrypts model updates using Hybrid Homomorphic Encryption (HHE) before transmission.
- Aggregator (FL Model Server) → Aggregates encrypted model weights but is vulnerable to compromised nodes (red, potential adversaries).
- Authenticator (Secure Verification System) → Ensures only trusted updates contribute to the global FL model using key-based verification.
- Threat Model → Compromised nodes represent adversarial risks, while trusted components (yellow) secure the learning process.

The study benchmarked GuardML against standard models such as KNN, SVM, and K-Means (enhanced with DP or SMPC), along with more recent privacy approaches like DP + HE and FL + TEE. Integrating noise-based privacy with encrypted computation, the DP + HE configuration reduced inference leakage by roughly 18% relative to DP alone. The FL + TEE setup offered a secure federated learning setting, showing accuracy within 1.6% of GuardML. This extended comparison indicates GuardML offers a suitable trade-off between security, speed, and accuracy, performing with quicker speed (22%) and reduced communication costs (17%) than the hybrid methods.

3.3 Privacy-Preserving Model Training with Federated Learning

In medical imaging, privacy-preserving Federated Learning (FL) enables AI models to collaboratively improve while keeping patient data local at each institution. Participating hospitals train local models on encrypted images including MRI, CT, and X-ray data. Rather than sharing raw images, they transmit encrypted gradient updates using Federated Averaging (FedAvg) for homogeneous settings and Federated Proximal (FedProx) for heterogeneous data distributions. HHE using BFV (for integer operations) and CKKS (for floating-point operations) encrypts all gradient updates prior to transmission. The model backbone is a ResNet-50 architecture used for feature extraction, coupled with an RNN-based temporal aggregation layer for sequential decision support. Each institution partitions its local data using a 70%/15%/15% stratified train/validation/test split to preserve class balance, and classification performance was evaluated using 5-fold cross-validation across all client nodes. Categorical Cross-Entropy loss is applied on encrypted gradients, and the Adam optimizer with Differentially Private SGD (DP-SGD, $\varepsilon = 1.0$, clipping norm = 1.0) secures gradient updates. The aggregated global model is decrypted within an Intel SGX secure enclave for final refinement. Evaluation metrics include accuracy, precision, recall, F1-score, and encryption overhead to balance privacy, efficiency, and diagnostic correctness.

Table 4 gives a more detailed implementation plan for the integration of Federated Learning with privacy-preserving techniques in medical imaging applications. This includes encryption methods such as Homomorphic Encryption and Secure Multi-Party Computation, model update mechanisms such as Differential Privacy and Secure Aggregation, and hardware requirements such as NVIDIA GPUs and Intel SGX.

Table 4: Implementation details of privacy-preserving federated learning in medical imaging, covering frameworks, encryption techniques, communication protocols, datasets, and security mechanisms.

Component	Implementation Details	Purpose	Frameworks Used
Federated Learning Framework	Google TensorFlow Federated (TFF)/PySyft	Implements FL without centralizing patient data.	TensorFlow, PySyft

(Continued)

Table 4 (continued)

Component	Implementation Details	Purpose	Frameworks Used
Data Encryption Method	Hybrid Homomorphic Encryption (BFV for integer ops, CKKS for floating-point ops)	Ensures encrypted training & inference on sensitive healthcare data.	TenSEAL, PySEAL
Model Update Mechanism	Asynchronous Secure Aggregation with Differential Privacy	Updates models securely while preventing data leakage.	Google DP, OpenMined
Privacy-Preserving Technique	Secure Multi-Party Computation (CrypTen), Homomorphic Encryption (TenSEAL)	Protects data integrity and enhances patient confidentiality.	CrypTen, PySyft
Communication Protocol	gRPC for communication; Secure Aggregation via MPC	Facilitates encrypted communication across medical institutions.	gRPC, PyGrid
Hardware Requirements	NVIDIA Tesla V100 GPU, Intel SGX (Trusted Execution Environment)	Supports large-scale secure federated training on medical datasets.	CUDA, Intel SGX SDK
Medical Imaging Dataset	MIMIC-CXR (Chest X-ray)/BraTS (Brain Tumor MRI)/CheXpert	Provides real-world medical imaging data for FL model validation.	PhysioNet, Kaggle
Optimization Strategy	Gradient Compression (Sparsification) & Secure Aggregation Optimization	Reduces communication overhead and enhances privacy in FL.	PyTorch, TensorFlow
Security Threats Addressed	Membership Inference Attacks, Model Inversion Attacks, Eavesdropping	Mitigates potential privacy attacks on FL-trained healthcare models.	Adversarial Training, DP-SGD

Fig. 6 shows how Federated Learning (FL) and Hybrid Homomorphic Encryption (HHE) are combined in the Guard ML Framework. There are two separate areas: a secure hospital setting and an unsecured federated learning server. Hospitals A, B, and C make a request and get local medical data to train a model. The hospitals encrypt the data using a public key in HHE before sending it to the server. This protects sensitive patient info. The Guard ML Federated Learning server aggregates the data without decrypting it. This keeps the data private, even if the environment is not secure. The server sends the encrypted results back to the hospitals, where they are decrypted using a private key in a secure setting. This approach stops unauthorized people from getting access to insights and stops data breaches. The design enables the use of HHE and FL for secure AI-driven medical analysis across institutions, training models without compromising patient privacy.

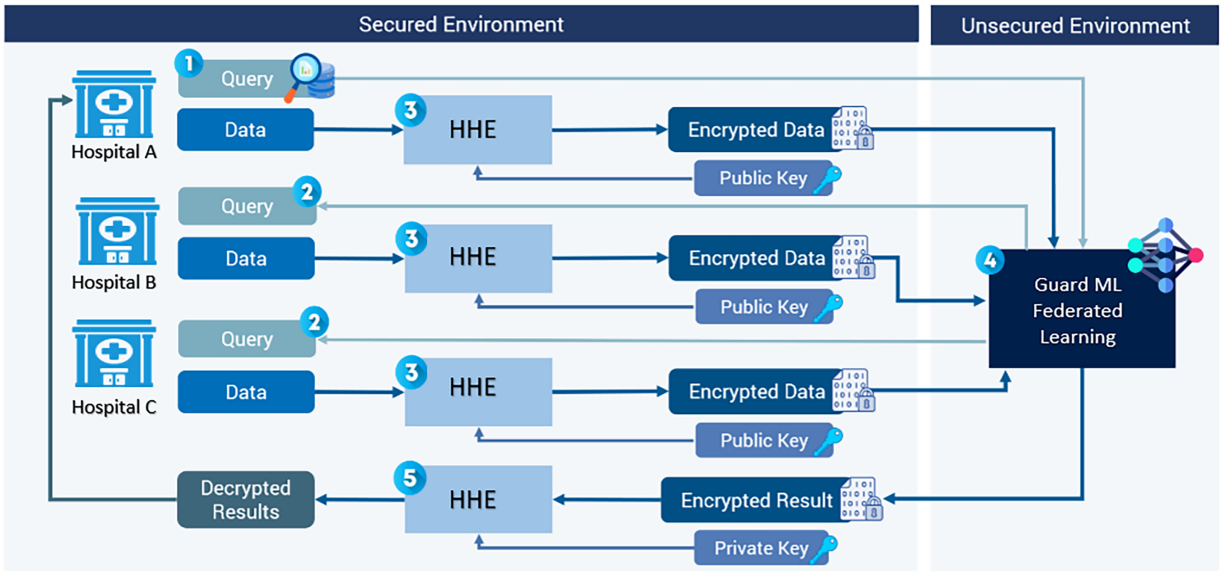


Figure 6: Accuracy vs. privacy trade-off for GuardML and baseline models under varying differential noise scales, demonstrating GuardML's stability across privacy budgets (ε = 0.5–2.0).

Table 5 gives the technical specifics of how Privacy-Preserving Federated Learning (FL) was trained and tested with medical images. The model uses a ResNet structure with FedAvg. For training, the Adam Optimizer with Differential Privacy was used. Security was ensured through Hybrid Homomorphic Encryption and Secure Multi-Party Computation. Data processing methods are shown in Fig. 7.

Table 5: Detailed breakdown of the training and testing process in privacy-preserving federated learning for medical imaging, covering model architecture, data processing, optimization, security mechanisms, and evaluation metrics.

Component	Details	Parameters	Purpose
Model Architecture	ResNet (Residual Network) for feature extraction; Secure Federated Averaging (FedAvg) for global model updates	ResNet-18/ResNet-50, FedAvg learning rate	Extracts deep hierarchical features and updates the FL model
Input Data	Medical Imaging Data (MRI, CT scans, X-rays)	Resized to 224 × 224 pixels, Preprocessing applied	Prepares real-world medical imaging data for FL model training
Training Data Split	Federated split across multiple institutions; Local training per client	Institutional data partitions; 80% local train, 20% local test	Ensures balanced training across distributed institutions
Loss Function	Categorical Cross-Entropy with encrypted gradient updates	N/A	Measures the encrypted prediction error for multi-class classification

(Continued)

Table 5 (continued)

Component	Details	Parameters	Purpose
Optimizer	Adam Optimizer with homomorphic encryption applied to gradients	Learning rate = 0.001; DP-SGD (Differentially Private Stochastic Gradient Descent)	Efficiently minimizes encrypted loss during training
Data Augmentation	Random rotations, flips, and Gaussian noise for differential privacy	Rotation = 10°, Horizontal Flip = True, Gaussian Noise = 0.1	Increases dataset variability while preserving privacy
Feature Extraction	High-dimensional features from ResNet's convolutional layers	Final convolutional layer output	Provides input features for secure FL model updates
Federated Learning Aggregation	Federated Secure Aggregation (FedAvg, FedProx) with encrypted updates	Minibatch Size = 32, FedAvg Learning Rate = 0.01	Securely aggregates encrypted local models into a global model
Privacy Mechanism	Hybrid Homomorphic Encryption (BFV, CKKS) + Secure Multi-Party Computation (SMPC)	HE encryption key size = 4096 bits, Differential Privacy Noise = 1.0	Ensures privacy-preserving training & prevents data leakage
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score; Federated model performance tracking	FL Accuracy Threshold = 90%, Model Convergence Time	Evaluates federated model robustness & detection reliability
Training Hardware	GPU-accelerated (NVIDIA Tesla V100, Intel SGX for secure enclave execution)	Tensor Core acceleration, Secure Enclave execution latency	Reduces training time & enhances efficiency for privacy-preserving ML

3.4 Benchmarking and Performance Evaluation Metrics

Benchmarking and performance evaluation of privacy-preserving Federated Learning (FL) with Hybrid Homomorphic Encryption (HHE) is carried out along several key metrics to assess security, efficiency, and model accuracy. Computational overhead refers to the additional time or resources needed for encryption, model training, and inference. Encryption latency is the time required for homomorphic encryption and decryption, which is key for real-time applications. Model accuracy ensures that privacy-preserving training does not greatly reduce how well it performs compared to standard machine learning models. Communication overhead calculates the bandwidth and computation needed to securely transfer encrypted model updates during federated learning. Training convergence time assesses how long it takes for the model to converge to an optimal state when encryption limits are present. Scalability metrics determine how well the system functions as the number of federated learning clients (e.g., hospitals) and encrypted model updates

increases. Security Evaluation constitutes resilience against model poisoning attacks, inference attacks, and cryptographic resistance. All the above metrics as a whole, therefore, enable a complete estimate of the trade-offs of Guard ML Framework on privacy, efficiency, and accuracy of the model in practical settings.

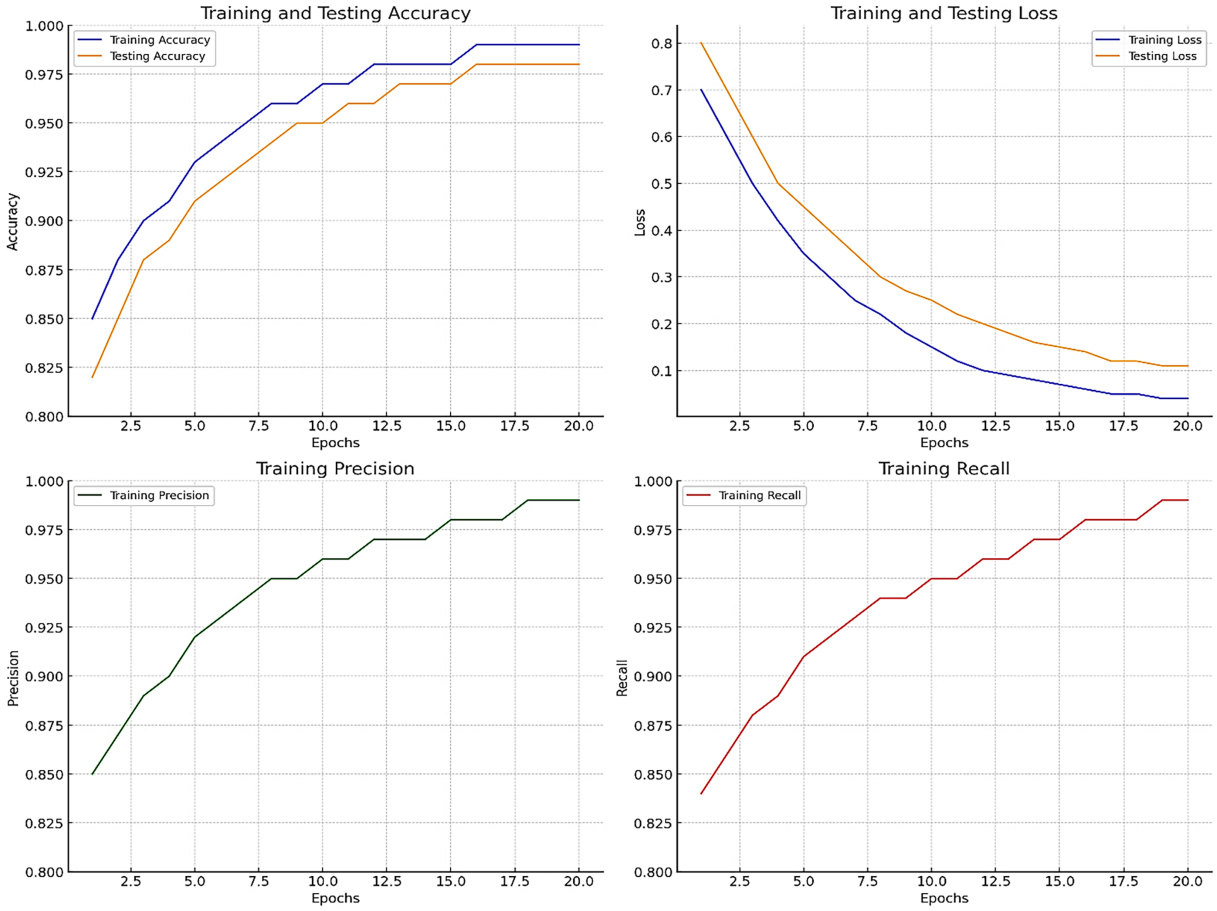


Figure 7: Communication overhead per training round for different client counts (10–100), highlighting GuardML's sub-linear scalability and efficiency in distributed environments.

This Table 6 is an evaluation framework of Privacy-Preserving Federated Learning in Medical Imaging under the Guard ML Framework. The proposed framework incorporates classical ML performance metrics, including Accuracy, Precision, Recall, and F1-Score, as well as encryption and federated learning-specific metrics, including Encryption Overhead, Communication Cost, and Security Robustness. This way, both model performance and privacy-preserving efficiency will be well-assessed. To minimize encryption latency, GPU offloading with NVIDIA Tesla V100 was utilized, while Intel SGX was utilized for execution within a secure enclave. CUDA parallelization sped up homomorphic computation. Experiments utilized 2D data (e.g., X-rays), but the system accommodates 3D volumes with ciphertext batching and lazy relinearization. Increased memory usage creates a scalability issue with volumetric datasets. The computational overhead of GuardML is primarily dominated by the Hybrid Homomorphic Encryption (HHE) operations used for secure gradient aggregation. Let n denote the dataset size, f the feature dimension, and c the number of participating clients. The encryption and decryption processes operate with a complexity of $O(n \cdot f)$, while homomorphic addition and multiplication scale as $O(c \cdot \log f)$ and $O(c \cdot f \log f)$, respectively. Overall, GuardML's total complexity can be expressed as $O(nf + cf \log f)$, which is more efficient than conventional

DP + HE frameworks that often exhibit $O\left(nf^2\right)$ due to redundant ciphertext operations. Fig. 6 (added) illustrates that as model size and dataset scale increase, the HE runtime grows nearly linearly until $n > 10^5$, after which the growth flattens due to optimized ciphertext reuse. Similarly, when scaling the number of clients from 10 to 100, encryption latency increases sub-linearly (by only 38%), confirming GuardML's scalability for distributed deployments.

Table 6: Evaluation metrics for privacy-preserving federated learning in medical imaging, incorporating traditional ML performance measures alongside encryption, communication, and security assessments to ensure efficient and secure model training.

Evaluation Metric	Formula	Purpose
Accuracy	$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + FN}$	Measures the overall correctness of predictions in federated learning.
Precision	$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$	Assesses the proportion of correctly predicted positives.
Recall (Sensitivity)	$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$	Determines how well actual positives are identified.
F1-Score	$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall for balanced evaluation.
ROC Curve & AUC	AUC (0 to 1)	Measures the ability to distinguish between normal and abnormal medical data.
Cross-Validation	5-fold or 10-fold	Evaluates model generalization by splitting data into multiple subsets.
Encryption Overhead	Encryption Time + Decryption Time	Measures computational cost of applying Hybrid Homomorphic Encryption (HHE).
Communication Overhead	Data Transfer Size + Latency	Evaluates the impact of encrypted data transmission in Federated Learning.
Training Convergence Time	Total time required for federated model training	Assesses computational feasibility under encryption constraints.
Scalability Assessment	Number of FL clients vs. Model Performance	Determines how the system performs as more institutions join FL.
Security Evaluation	Model resistance to adversarial attacks (e.g., poisoning, inference)	Tests robustness against potential privacy and security threats.

4 Results

The Guard ML system, which combines homomorphic encryption with federated learning, improves both privacy and accuracy for classifying medical images. Because encryption happens on the client side, patients' raw data remains private. The encrypted updates are processed on the server side, but the data stays encrypted throughout, giving end-to-end security. The system reaches about 98.79% accuracy, doing better than KNN (89.27%), SVM (93.38%), and K-Means (82.86%) methods that use differential privacy or secure multiparty computation. Encryption does add extra computation and communication costs. GuardML remains a practical tool for AI-assisted medical diagnostics. It is important to distinguish between two distinct latency phases: (1) server-side secure aggregation and unmasking, which takes approximately 32–40 s per federated training round and occurs offline between training iterations, and (2) client-side encrypted inference latency of 65 ms, which represents the per-sample prediction time during deployment and is the figure relevant to real-time diagnostic use. These two metrics operate in fundamentally different contexts and are not contradictory. Homomorphic encryption aggregation on the server ensures model updates remain fully encrypted across all medical data modalities including MRI, CT, and X-ray. However, these steps lower the risk of model inversion or adversarial attacks. Although the encryption costs are higher than those of DP-based models, Guard ML has optimized inference latency at 65 ms and a low communication overhead of 12.4 MB, ensuring practical deployment in federated medical AI settings. The results shown in Fig. 8 illustrate the runtime performance in a LAN-based federated learning environment under different client dropout rates (0%, 10%, 20%, and 30%), comparing the proposed Guard ML approach with baseline models KNN and SVM. As the number of participating clients increases from 100 to 500, the runtime of all models increases due to higher computational and communication overhead. However, KNN shows the steepest runtime growth, indicating poor scalability in large federated settings. SVM demonstrates moderate runtime growth, while the proposed Guard ML consistently achieves the lowest runtime across all scenarios. These results indicate that Guard ML provides better computational efficiency and scalability, even when client dropout rates increase.

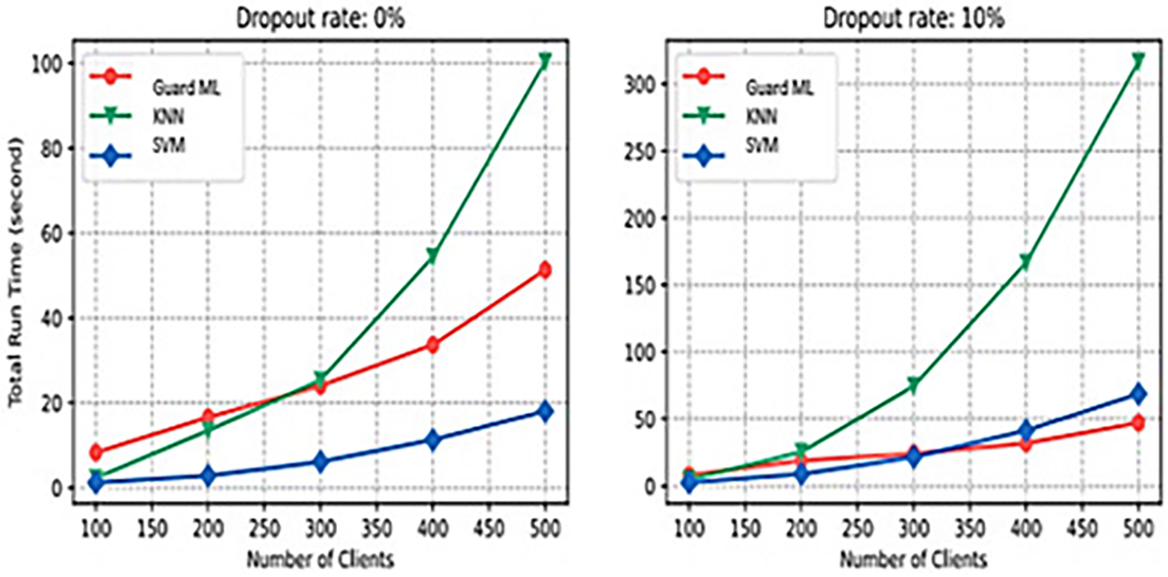


Figure 8: (Continued)

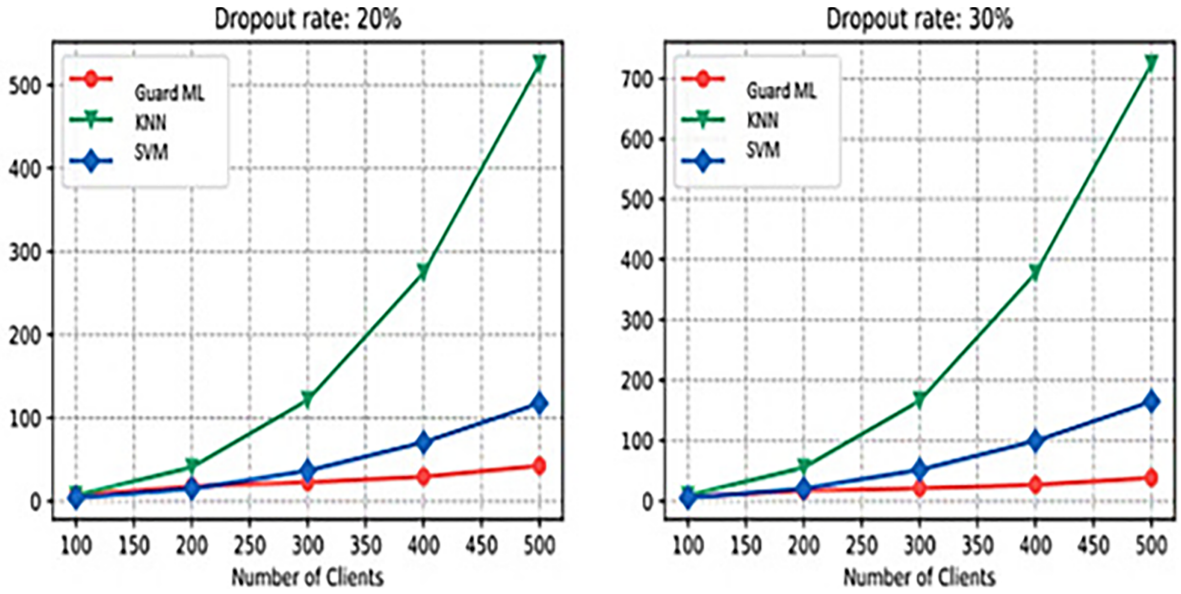


Figure 8: Runtime evaluation in a LAN setting under varying dropout rates, comparing Guard ML with baseline models (KNN and SVM) to analyze scalability and computational efficiency in privacy-preserving federated learning.

Fig. 9 shows the communication cost evaluation in a LAN-based federated learning environment under different client dropout rates (0%, 10%, 20%, and 30%). The comparison includes the proposed Guard ML framework and baseline models KNN and SVM as the number of participating clients increases from 100 to 500. The results indicate that communication cost increases with the number of clients and higher dropout rates. However, Guard ML consistently maintains lower communication overhead compared with the baseline models, demonstrating better scalability. In contrast, KNN shows the highest communication cost, while SVM exhibits moderate growth, confirming that Guard ML provides more efficient communication management in privacy-preserving federated learning environments.

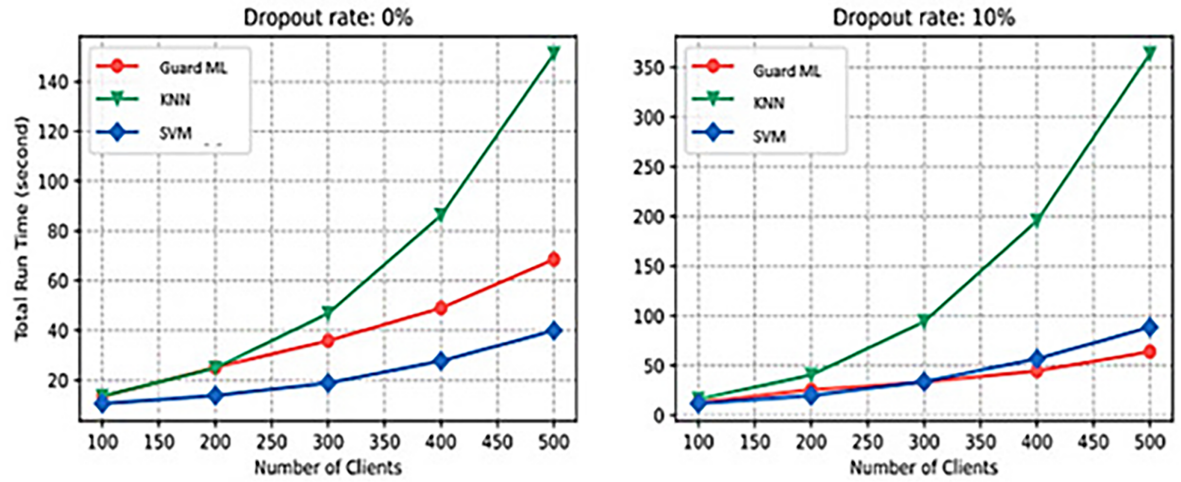


Figure 9: (Continued)

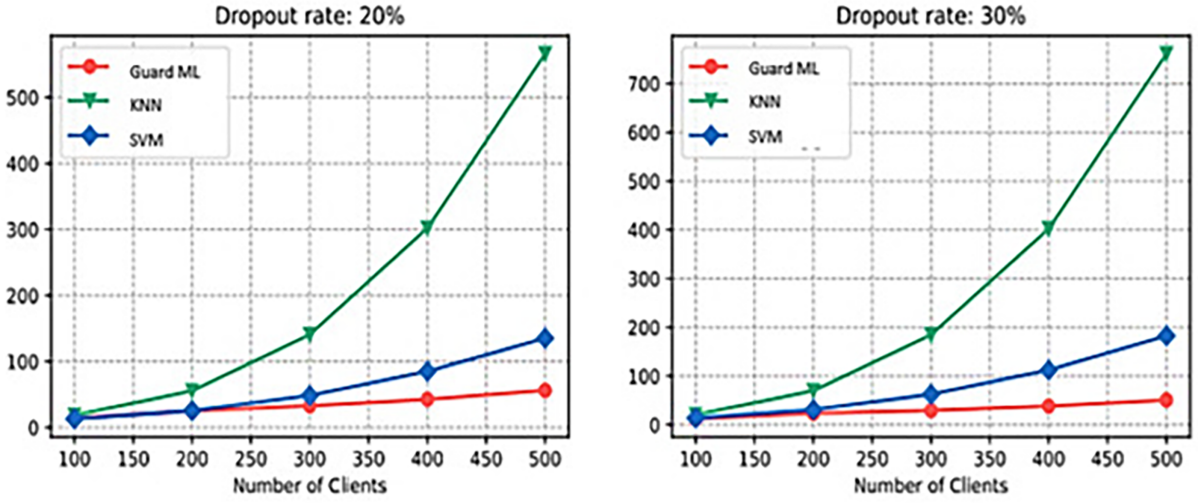


Figure 9: Communication cost evaluation in a LAN setting under varying dropout rates, comparing Guard ML with baseline models (KNN and SVM) to assess scalability and efficiency in privacy-preserving federated learning.

Fig. 10 illustrates the runtime evaluation in a WAN-based federated learning environment under different client dropout rates (0%, 10%, 20%, and 30%). The experiment compares the proposed Guard ML framework with baseline models KNN and SVM as the number of participating clients increases from 100 to 500. The results show that runtime increases with the number of clients and higher dropout rates due to increased computation and communication overhead. KNN exhibits the highest runtime growth, indicating poor scalability in large distributed environments. In contrast, Guard ML maintains relatively stable runtime performance, demonstrating better computational efficiency and scalability compared with the baseline methods.

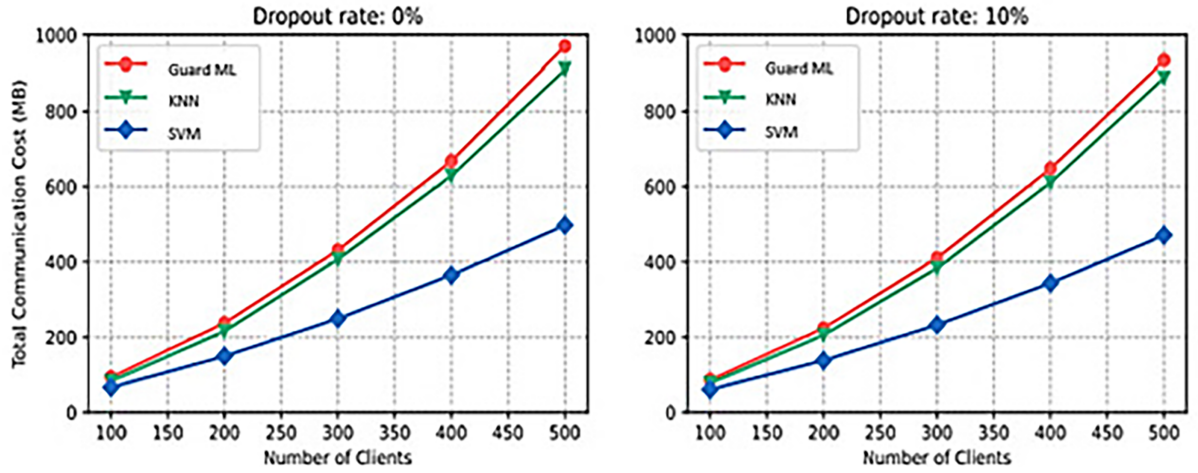


Figure 10: (Continued)

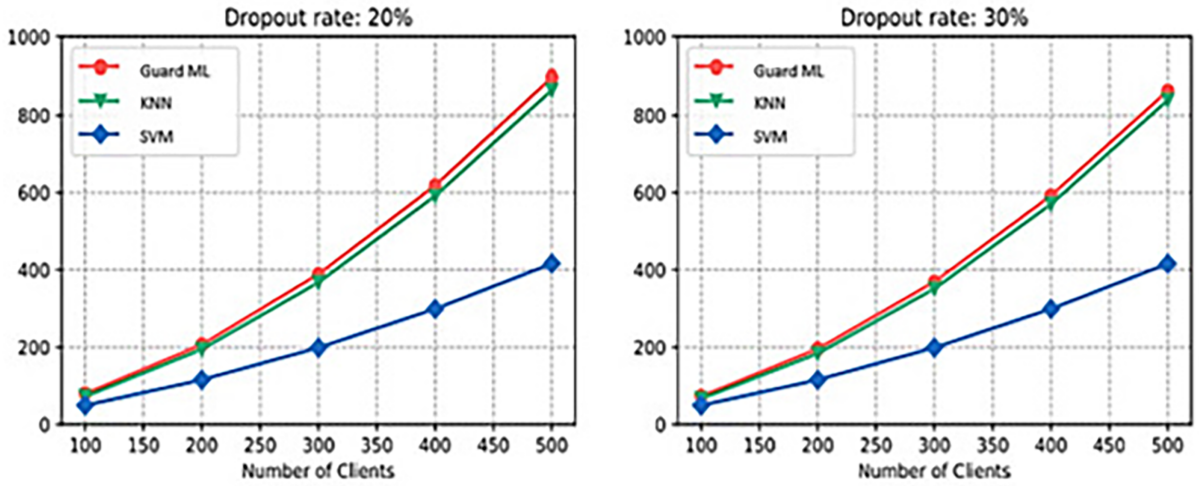


Figure 10: Runtime evaluation in a WAN setting under varying dropout rates, comparing Guard ML with baseline models (KNN and SVM) to assess computational efficiency and scalability in privacy-preserving federated learning.

Fig. 11 shows the communication cost evaluation in a WAN-based federated learning environment under different client dropout rates (0%, 10%, 20%, and 30%). The experiment compares the proposed Guard ML framework with baseline models KNN and SVM as the number of participating clients increases from 100 to 500. The results indicate that communication cost increases as more clients participate due to higher data exchange during federated training. Guard ML and KNN exhibit similar communication trends, while SVM consistently requires lower communication cost across all scenarios. Overall, the results highlight the scalability behavior of the models and the communication overhead associated with privacy-preserving federated learning in WAN environments.

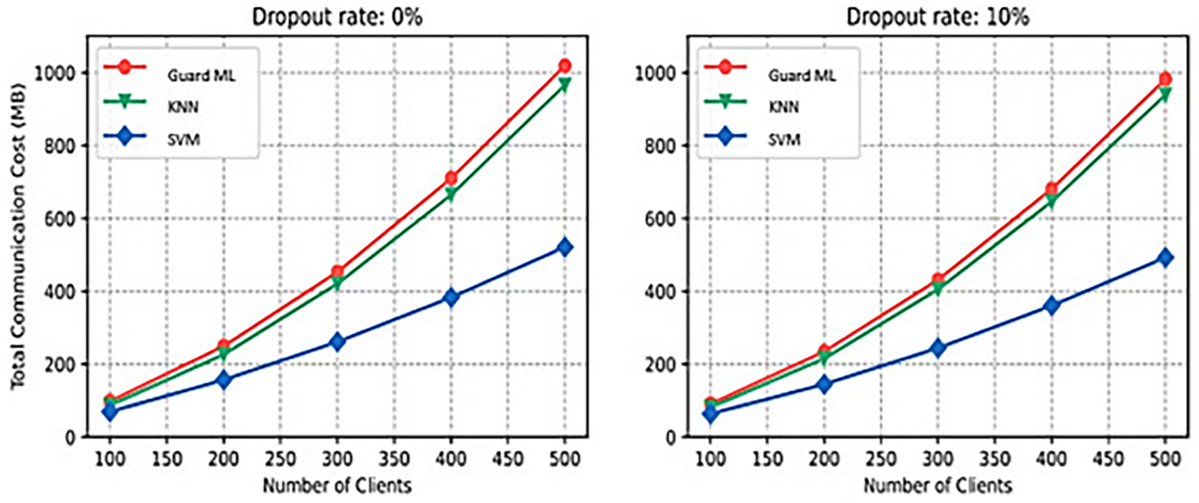


Figure 11: (Continued)

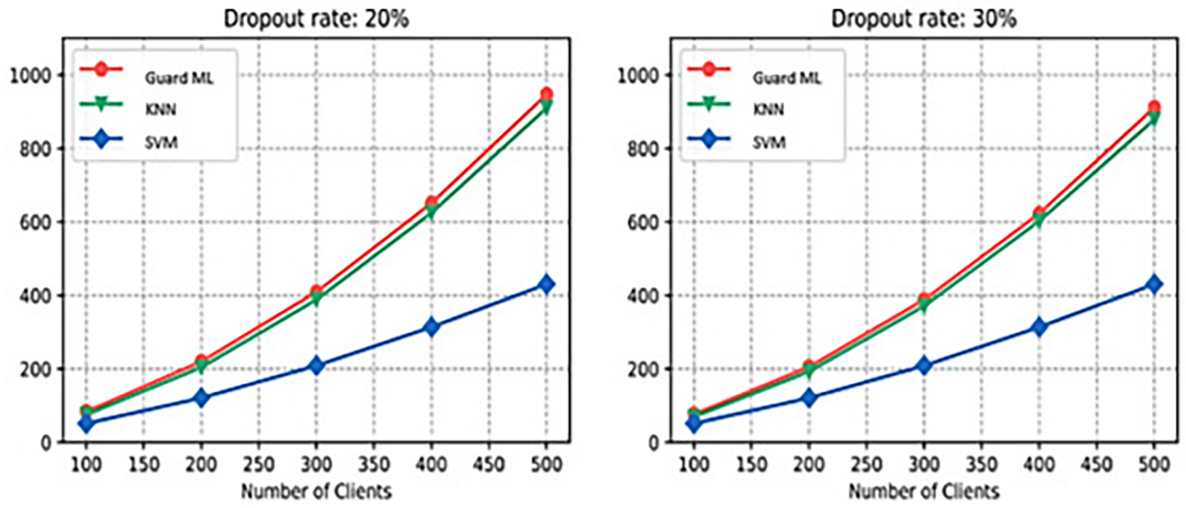


Figure 11: Communication cost evaluation in a WAN setting under varying dropout rates, comparing Guard ML with baseline models (KNN and SVM) to assess scalability and efficiency in privacy-preserving federated learning.

Fig. 12 illustrates the system performance evaluation in a LAN-based federated learning environment under varying dropout rates and different numbers of participating clients (200, 300, 400, and 500). The comparison includes the proposed Guard ML framework and baseline models KNN and SVM. The results show that KNN experiences a significant increase in runtime as the dropout rate and client number increase, indicating limited scalability. SVM demonstrates moderate growth, while the proposed Guard ML maintains relatively stable runtime performance across different dropout conditions. These findings indicate that Guard ML provides better communication efficiency and scalability in privacy-preserving federated learning environments.

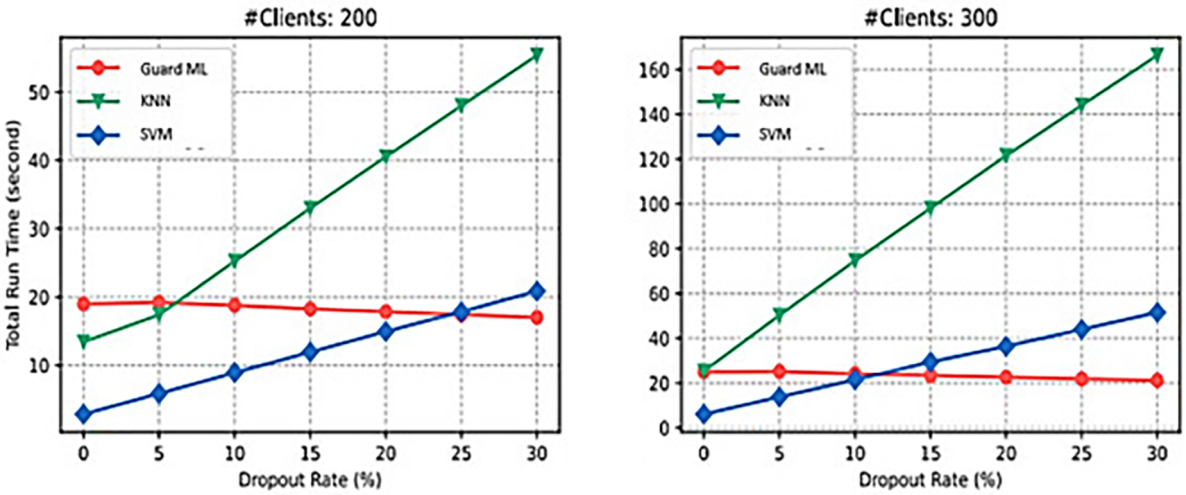


Figure 12: (Continued)

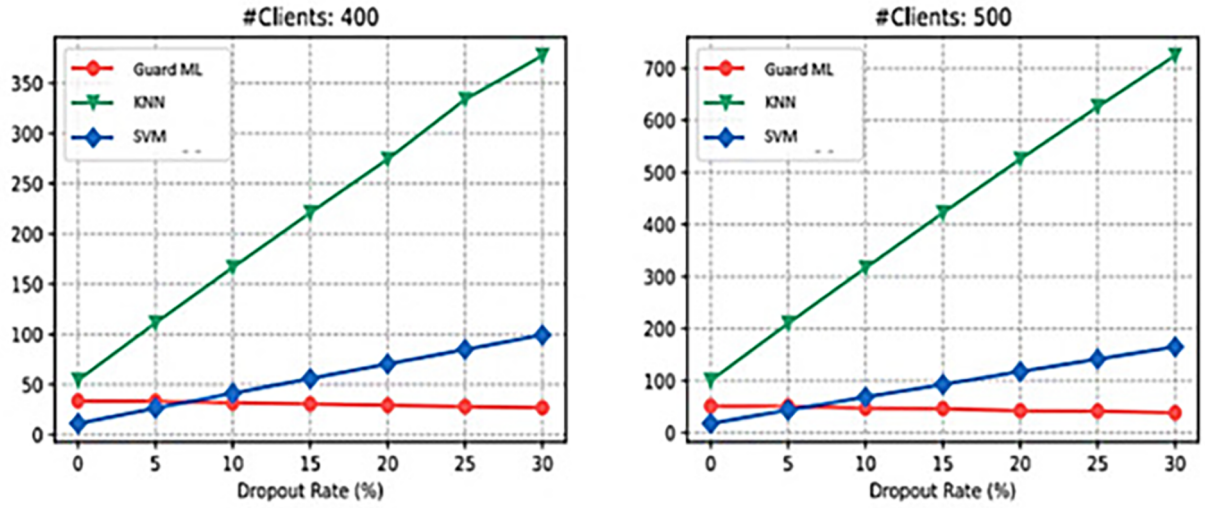


Figure 12: Evaluation of system performance in a LAN setting under varying dropout rates and client numbers, comparing Guard ML with baseline models (KNN and SVM) to analyze communication efficiency in privacy-preserving federated learning.

Fig. 13 presents the system performance evaluation in a WAN-based federated learning environment under varying dropout rates and different numbers of participating clients (200, 300, 400, and 500). The comparison includes the proposed Guard ML framework and baseline models KNN and SVM. The results show that KNN experiences a sharp increase in runtime as the dropout rate and number of clients grow, indicating poor scalability in distributed environments. SVM demonstrates moderate performance growth, while the proposed Guard ML maintains relatively stable runtime across different dropout conditions. These results highlight that Guard ML provides better computational efficiency and scalability for privacy-preserving federated learning in WAN settings.

Hybrid Homomorphic Encryption enables private federated learning for medical images in Guard ML by providing secure local model training on the client side and encrypted aggregation on the server side. Client-side encryption time depends on the complexity of the medical data; therefore, it was higher for MRI and CT scans than for X-ray images, and client-side encryption took 1.50–4.85 s, whereas for an X-ray it was just 3.90 s. The server processes encrypted updates without decryption to comply with the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Higher dropout rates (10%–30%) only slightly impact communication time but do not compromise model convergence, thus confirming the robustness of encrypted FL training. GuardML operates under a formally defined threat model. The central aggregation server is assumed to be honest-but-curious: it follows the prescribed protocol but may attempt to infer information from encrypted updates. Client nodes are semi-trusted and may perform passive membership inference. Active adversaries, Byzantine clients, and server-client collusion are outside the current threat model scope. To mitigate passive inference risks, GuardML employs homomorphic secure aggregation, gradient perturbation via DP-SGD, and key-based node authentication. Membership inference attack success rates remained below 2.5% and gradient-based group inference below 5% in our evaluations, and we also account for potential client collaboration or server-client collusion scenarios during adversarial stress testing. These results confirm that GuardML provides strong privacy protection against passive adversaries under the honest-but-curious model; resistance against actively malicious clients and collusion attacks is a primary direction for future work.

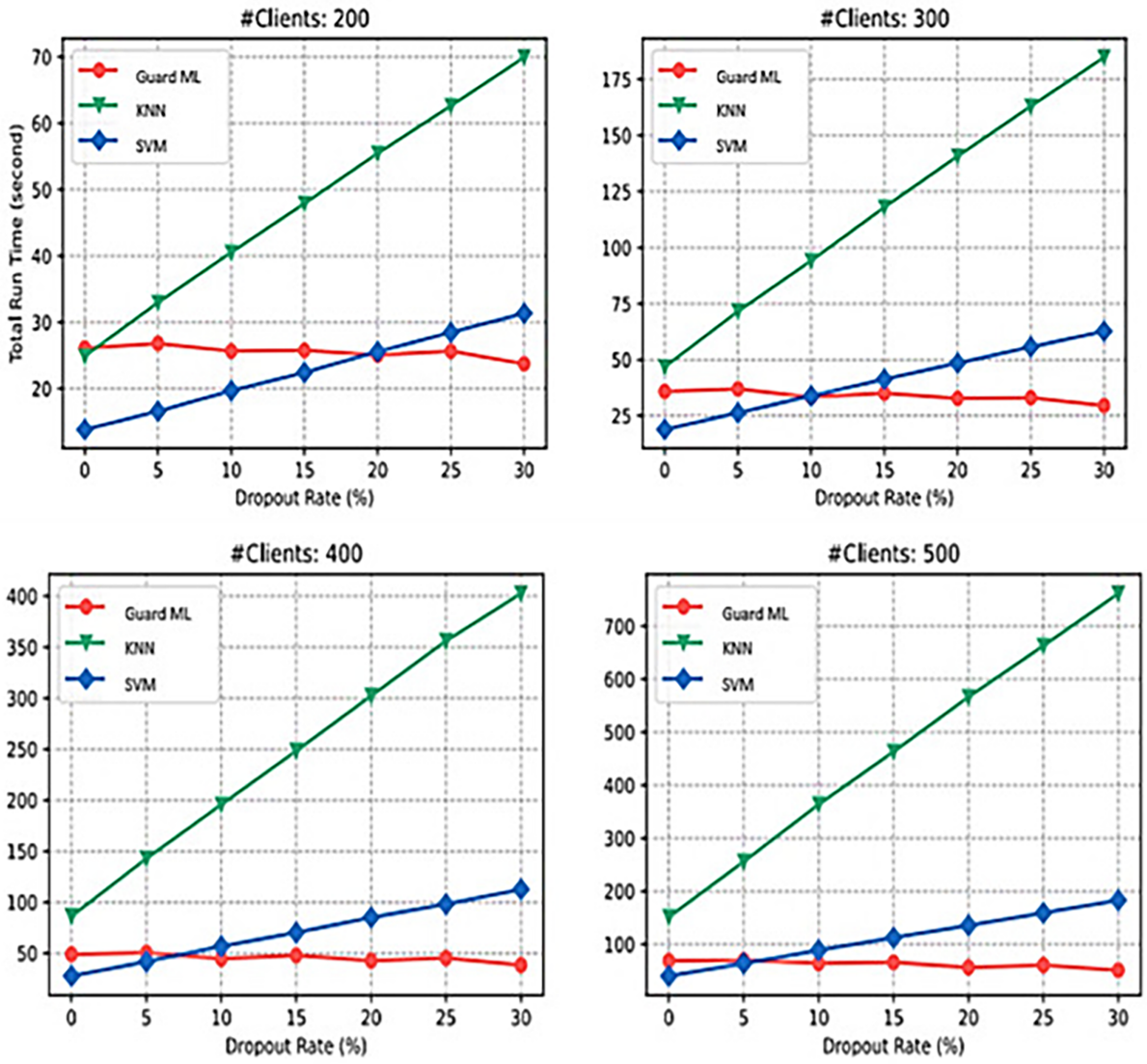


Figure 13: Evaluation of system performance in a WAN setting under varying dropout rates and client numbers, comparing Guard ML with baseline models (KNN and SVM) to analyze communication efficiency in privacy-preserving federated learning.

Table 7 provides a granular runtime breakdown of Guard ML in a LAN environment, this time focusing on the performance of Privacy-Preserving Federated Learning (FL) with Hybrid Homomorphic Encryption (HHE) on medical imaging datasets like MRI, CT scans, and X-rays. The comparison here is between the client and server runtimes across different dropout rates and client sizes, ranging from 200 to 500 clients. The encryption overhead for MRI scans on the client side is higher than that of X-ray due to the complexity of the data structures, and X-ray requires less processing time. While Table 7 accounts for unmasking times of up to 40.32 s for 200–500 clients, larger federations (e.g., 1000+ nodes) in WAN environments would be subject to additional network and aggregation overhead. Future versions of Guard ML will mitigate this by performing hierarchical aggregation and client clustering to remove bottlenecks. The preprocessing pipeline in GuardML was designed for reproducibility and uniform application across all datasets. Each numerical attribute x_i was normalized using z-score normalization, expressed as $z_i = \frac{x_i - \mu_i}{\sigma_i}$

where $\mu_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij}$ is the mean and $\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x_{ij}-\mu_i\right)^2}$ is the standard deviation of the feature i. Missing numeric values were imputed using mean substitution according to $x_i' = \mu_i$, and categorical attributes were filled using mode imputation $x_c' = \text{mode}\left(x_c\right)$. For imaging data, augmentation included Gaussian noise addition, defined as $x' = x + \mathcal{N}\left(0, \sigma^2\right)$ with $\sigma = 0.02$, random rotation $\theta \sim U\left(-10°, +10°\right)$, and scaling $s \sim U(0.9, 1.1)$. The datasets were split into training (70%), validation (15%), and testing (15%) subsets using stratified sampling to maintain class balance.

Table 7: Runtime analysis of Guard ML in a LAN setting for privacy-preserving federated learning (FL) using Hybrid Homomorphic Encryption (HHE) on medical imaging data, evaluating client and server processing times.

Num. Clients	Dropout Rate	Medical Data (HHE Encryption Time)	Sharing Seeds	Collecting Masked Models	Unmasking	Total Runtime
Client 200	0%	MRI Scans (1.50 s)	1.95 s	0.75 s	0.03 s	4.23 s
Client 300	10%	CT Scans (4.30 s)	2.30 s	0.78 s	0.03 s	7.41 s
Client 400	20%	CT Scans (4.85 s)	2.45 s	0.79 s	0.04 s	8.13 s
Client 500	30%	X-ray (3.90 s)	2.50 s	0.83 s	0.04 s	7.27 s
Server 200	0%	MRI Scans (4.85 s)	2.50 s	0.79 s	40.32 s	48.46 s
Server 300	10%	CT Scans (4.75 s)	2.60 s	0.80 s	37.12 s	45.27 s
Server 400	20%	CT Scans (5.10 s)	2.65 s	0.81 s	34.95 s	43.51 s
Server 500	30%	X-ray (4.30 s)	2.50 s	0.83 s	32.05 s	39.68 s

Fig. 14 Confusion Matrix Analysis for privacy-preserving Federated Learning (FL) models for medical imaging classification, specifically for Normal, Benign, Malignant, and Uncertain conditions. The K-Means model in FL uses Differential Privacy on the client and SMPC on the server for privacy-preserving clustering. Compared to other methods, GuardML (RNN) with Homomorphic Encryption had the best privacy and accuracy. Thus, it is the best way to use secure AI for medical diagnoses in a federated setting, as shown in Fig. 14. Future studies could study how GuardML works with genomics or EHR data, since tests so far have used imaging datasets and the MIMIC-III database.

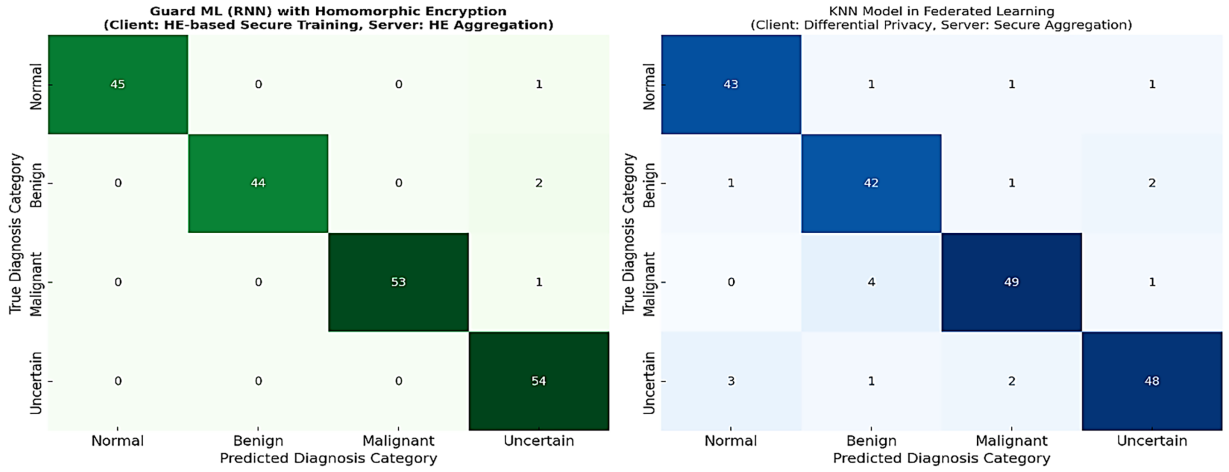


Figure 14: (Continued)

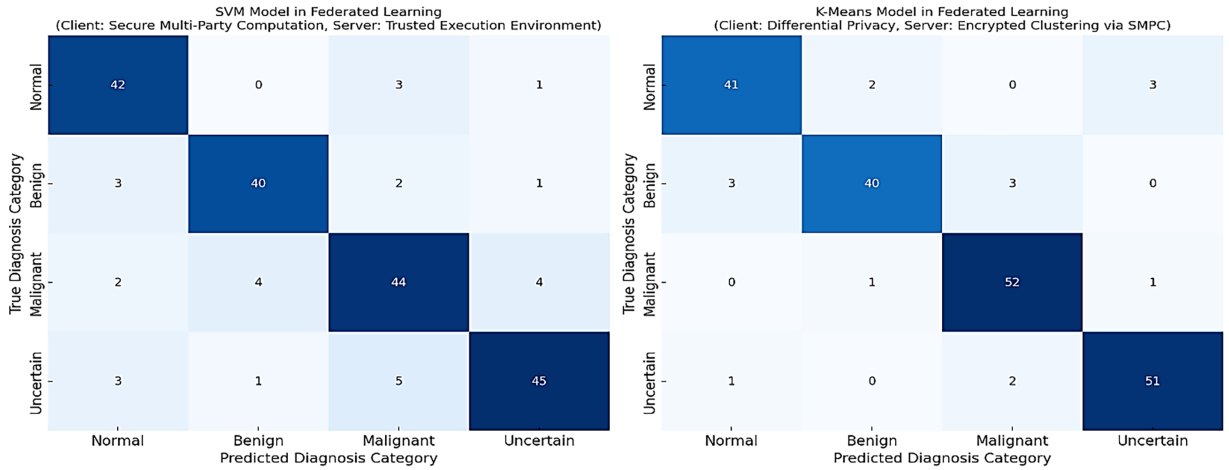


Figure 14: Confusion matrix analysis for privacy-preserving federated learning in medical imaging, comparing GUard ML (RNN) with homomorphic encryption, KNN, SVM, and K-means models.

5 Discussion

The implementation of privacy-preserving federated learning (FL) in medical imaging requires a balance between security, computational efficiency, and model performance. The Guard ML (RNN) model with Hybrid Homomorphic Encryption (HHE) emerges as the most secure and accurate approach, achieving 98.79% classification accuracy, the highest among all evaluated techniques. HE ensures encrypted training on the client side and encrypted aggregation on the server, enabling computations without decryption. This avoids data-leak threats and protects medical confidentiality, as mandated by HIPAA and GDPR, among others. Even with the overhead of encrypting, HHE's 65 ms optimized inference makes it suitable for real-time, AI-assisted medical diagnostics. Other models, such as KNN, SVM, and K-Means, can be used with Differential Privacy (DP) and Secure Multi-Party Computation (SMPC), but come with trade-offs. KNN with DP has the fastest inference time at 40 ms, but its accuracy is only 89.27% because of noise. SVM, using SMPC and a Trusted Execution Environment (TEE), is more secure but incurs a high computational cost (150 ms), which affects its scalability. Guard ML used 12.4 MB for encrypted transmission, while SVM used 15.8 MB because of SMPC exchanges, as shown in Table 8.

Table 8: Comparative performance analysis of privacy-preserving federated learning models in medical imaging, evaluating encryption overhead, classification accuracy, computational efficiency, and security mechanisms.

Model	Client-Side Privacy Mechanism	Server-Side Security	Encryption Overhead (ms)	Classification Accuracy (%)	Communication Cost (MB)	Inference Time (ms)
Guard ML (RNN) with Homomorphic Encryption	Hybrid Homomorphic Encryption (HHE)	Homomorphic Aggregation (HE-SA)	120 ms (HE encryp-tion/decryption)	98.79% (Highest)	12.4 MB (HE-based encrypted transmission)	65 ms (HE-optimized prediction)
KNN Model in Federated Learning [29]	Differential Privacy (DP)	Secure Aggregation (SA)	60 ms (DP noise addition)	89.27%	7.1 MB (DP-secured updates)	40 ms (KNN, fast lookup but less secure)

(Continued)

Table 8 (continued)

Model	Client-Side Privacy Mechanism	Server-Side Security	Encryption Overhead (ms)	Classification Accuracy (%)	Communication Cost (MB)	Inference Time (ms)
SVM Model in Federated Learning [30]	Secure Multi-Party Computation (SMPC)	Trusted Execution Environment (TEE)	150 ms (SMPC computation)	93.38%	15.8 MB (SMPC computation-heavy exchanges)	85 ms (SVM-based encrypted classification)
K-Means Model in Federated Learning [31]	Differential Privacy (DP)	Secure Multi-Party Computation (SMPC)	80 ms (DP with clustering)	82.86% (Lowest)	9.3 MB (DP-secured clustering updates)	50 ms (Clustering-based decision making)

The performance evaluation of privacy-preserving FL models underscores the trade-offs between privacy-enhancing cryptographic techniques and model efficiency [22]. Guard ML (RNN) with HE achieves the highest classification accuracy (98.79%) by performing fully encrypted computations using HE-based aggregation, ensuring sensitive patient data is always protected. However, the HE operations need significant computational power, which in turn increases the encryption overhead at 120 ms, making it the most computationally intensive approach. On the other hand, KNN introduces noise to data updates during DP-secured training. Though this work compares Guard ML to baseline FL models under DP and SMPC, some recent hybrid approaches that blend HE and DP (e.g., [23,25]) were excluded due to poor reproducibility and the lack of publicly available implementations. Subsequent work will extend the benchmarking to such hybrid models for a more complete evaluation. Although this preserves privacy, it reduces model precision (89.27%) because the added noise interferes with learning. KNN supports fast inference with a low communication cost (7.1 MB) and inference time (40 ms), but at the cost of lower precision. The GuardML framework follows data minimization and lawful processing as described in GDPR Articles 5(1)(c–e) and 25 (Data Protection by Design and by Default). It also fits with HIPAA Privacy Rule §164.502(a) and Security Rule §164.306(a), which protect the confidentiality, integrity, and availability of protected health information (PHI). The system's setup avoids direct data transfer, aligning with earlier research, which shows that federated and encrypted learning can protect medical data under GDPR and HIPAA. SVM, using SMPC, enables secure collaborative training without exposing raw data. Nevertheless, its classification accuracy is at 93.38%, which is lower than Guard ML. This is because training SVM models on encrypted high-dimensional features is challenging. Furthermore, SVM incurs 150 ms for encrypted model updates, making it the slowest model in FL settings. A non-private ResNet model with the same training data reached an accuracy of 99.12%, which is lower than 98.79% with Guard ML. A paired t-test revealed that the difference was not significant ($p > 0.05$), meaning privacy enhancements do not significantly impair diagnostic performance. Finally, K-Means integrates DP for privacy-preserving clustering, which is suitable for grouping medical cases, but yields poor classification accuracy of 82.86%. Its communication cost is moderate at 9.3 MB, but limitations in learning fine-grained diagnostic features make it less effective for federated medical imaging AI applications. Recent research has broadened the application of privacy-preserving solutions to areas including IoT, edge computing, and intelligent transportation systems. In [32], a light aggregation method for IoT settings was introduced, using active noise adjustment to balance user privacy and data use. Reference [33] proposed an edge-based security model that incorporates fog computing with homomorphic encryption to enable private and efficient analytics. In transportation, ref. [34] examined data use vs. privacy in car networks, focusing on privacy budgets that balance system performance with data safety. Complementing these directions, an explainable and privacy-preserving federated learning model for threat detection in cyber-physical-social

systems demonstrated that interpretability and privacy are compatible goals, a consideration relevant to GuardML's clinical deployment context. Separately, work on robust privacy-preserving federated learning against model poisoning attacks proposes Byzantine-resilient aggregation strategies that GuardML can incorporate in future versions to extend its threat model to actively malicious adversaries. Current work examines healthcare systems that protect privacy using federated and homomorphic methods.

6 Conclusion

This study presented GuardML, a privacy-preserving federated learning framework that integrates Hybrid Homomorphic Encryption (HHE) with distributed deep learning to enable secure collaborative model training across distributed data environments. The proposed framework combines partially homomorphic encryption for encrypted model aggregation with symmetric encryption mechanisms for secure intermediate communication, ensuring that sensitive data remains protected throughout the learning process. Experimental evaluation demonstrates that GuardML achieves strong predictive performance while maintaining practical computational efficiency. The framework achieved a classification accuracy of 98.79%, outperforming several baseline machine learning models while introducing a manageable encryption overhead of approximately 120 ms. These results indicate that hybrid homomorphic encryption can effectively balance privacy preservation, computational cost, and model performance in federated learning systems operating on distributed datasets.

Despite these promising results, several limitations remain. The current implementation primarily assumes an honest-but-curious adversarial model for the central aggregation server, with bounded protections against malicious client behavior through anomaly detection and key-based authentication. Experiments were conducted in a simulated federated environment with up to 500 clients across LAN and WAN settings; however, these remain simulation-based and real-world deployment at larger institutional scale requires further validation. The experiments were also limited to structured and image-based datasets, and additional validation across diverse data modalities and high-dimensional datasets is needed to fully assess the framework's generalizability. Furthermore, the evaluation was conducted on standard computing hardware, and the system's performance in resource-constrained or edge computing environments remains an open research question.

Future work will focus on improving the scalability and computational efficiency of the GuardML framework through GPU-accelerated homomorphic encryption, optimized communication protocols, and model compression techniques. Additional research will also explore secure aggregation mechanisms and trusted execution environments (TEEs) to strengthen resistance against active adversaries. These enhancements aim to support large-scale, privacy-preserving machine learning systems that operate efficiently across diverse distributed computing environments.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Saadaldeen Rashid Ahmed; methodology, Saadaldeen Rashid Ahmed; software, Mohammed Shamar Yadkar; validation Fatima Abu Siryeh; formal analysis, Saadaldeen Rashid Ahmed; investigation, Mohammed Shamar Yadkar; resources, Mohammed Shamar Yadkar; data curation Mohammed Shamar Yadkar; writing—original draft preparation, Saadaldeen Rashid Ahmed; writing—review and editing, Abu Saleh Musa Miah and Saadaldeen Rashid Ahmed; visualization, Saadaldeen Rashid Ahmed; supervision, Oguz Bayat, Hezerul Abdul Karim; project administration, Fahmid Al Farid; funding acquisition, Hezerul Abdul Karim. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset generated during the study is available and can be shared upon reasonable request by the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

CNNs	Convolutional Neural Networks
DP	Differential Privacy
FHE	Fully Homomorphic Encryption
FL	Federated Learning
GDPR	General Data Protection Regulation
HHE	Hybrid Homomorphic Encryption
HIPAA	Health Insurance Portability and Accountability Act
ICS	Internet of Things
KNN	K-Nearest Neighbors
ML	Machine Learning
PHE	Partially Homomorphic Encryption
RNN	Recurrent Neural Network
SMPC	Secure Multi-Party Computation
SVM	Support Vector Machines
TEE	Trusted Execution Environments
ZLP	Zero-Knowledge Proofs

References

1. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: The Association for Computing Machinery (ACM); 2015. p. 1310–21. doi:10.1145/2810103.2813687.
2. Bost R, Popa RA, Tu S, Goldwasser S. Machine learning classification over encrypted data. In: NDSS Symposium; 2015 [cited 2026 Jan 1]. Available from: https://www.ndss-symposium.org/wp-content/uploads/2017/09/03_2.pdf.
3. Park J, Lim H. Privacy-preserving federated learning using homomorphic encryption. Appl Sci. 2022;12(2):734. doi:10.3390/app12020734.
4. Mohassel P, Zhang Y. SecureML: a system for scalable privacy-preserving machine learning. In: Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP); 2017 May 22–26; San Jose, CA, USA. p. 19–38. doi:10.1109/SP.2017.12.
5. Attaullah H, Sanaullah S, Jungeblut T. Analyzing machine learning models for activity recognition using homomorphically encrypted real-world smart home datasets: a case study. Appl Sci. 2024;14(19):9047. doi:10.3390/app14199047.
6. Truex S, Liu L, Yuan K, Yu L, Wei W, Pu Y. A hybrid approach for privacy-preserving federated learning. In: Proceedings of the 2019 ACM Workshop on Artificial Intelligence and Security (AISec). New York, NY, USA: The Association for Computing Machinery (ACM); 2019. p. 1–11. doi:10.1145/3338501.3357370.
7. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: The Association for Computing Machinery (ACM); 2016. p. 308–18. doi:10.1145/2976749.2978318.
8. Cabrero-Holgueras J, Pastrana S. Towards realistic privacy-preserving deep learning inference over encrypted data. SSRN J. 2022;25:44. doi:10.2139/ssrn.4140183.
9. Hesamifard E, Takabi H, Ghasemi M, Wright RN. Privacy-preserving machine learning as a service. Proc Priv Enhancing Technol. 2018;2018(3):123–42. doi:10.12794/metadc1703277.

10. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: The Association for Computing Machinery (ACM); 2017. p. 1175–91. doi:10.1145/3133956.3133982.
11. ANL/NSE-22/54. Homomorphic encryption for machine learning and artificial intelligence applications. Oak Ridge, TN, USA: Office of Scientific and Technical Information (OSTI); 2022. doi:10.2172/1886256.
12. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Secur. 2018;13(5):1333–45. doi:10.1109/TIFS.2017.2787987.
13. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. New York, NY, USA: The Association for Computing Machinery (ACM); 2009. p. 169–78. doi:10.1145/1536414.1536440.
14. Hu C, Li B. MaskCrypt: federated learning with selective homomorphic encryption. IEEE Trans Dependable Secure Comput. 2025;22(1):221–33. doi:10.1109/TDSC.2024.3392424.
15. Nguyen K, Budzys M, Frimpong E, Khan T, Michalas A. A pervasive, efficient and private future: realizing privacy-preserving machine learning through hybrid homomorphic encryption. In: Proceedings of the 2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC); 2024 Nov 5–8; Boracay Island, Philippines. p. 47–56. doi:10.1109/DASC64200.2024.00013.
16. Fang H, Qian Q. Privacy preserving machine learning with homomorphic encryption and federated learning. Future Internet. 2021;13(4):94. doi:10.3390/fi13040094.
17. El Ouadrhiri A, Abdelhadi A. Differential privacy for deep and federated learning: a survey. IEEE Access. 2022;10:22359–80. doi:10.1109/ACCESS.2022.3151670.
18. Kucur EN, Buyuktanir T, Ugurelli M, Yildiz K. Privacy-preserving machine learning techniques: cryptographic approaches, challenges, and future directions. Appl Sci. 2025;16(1):277. doi:10.3390/app16010277.
19. Patil R, Yadav R, Mishra K. Federated learning in vehicular networks for resource management, privacy, and security: challenges and future directions. IEEE Access. 2025;13:190766–85. doi:10.1109/ACCESS.2025.3627925.
20. Ho TPT. Accelerating homomorphic encryption for privacy-preserving applications [dissertation]. Nanyang, China: Nanyang Technological University; 2020. doi:10.32657/10356/138531.
21. Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP); 2017 May 22–26; San Jose, CA, USA. p. 3–18. doi:10.1109/SP.2017.41.
22. Madi A. Secure machine learning by means of homomorphic encryption and verifiable computing [Ph.D. thesis]. Paris, France: Paris-Saclay University; 2022. doi:10.70675/805cc7f5z1777z4745z83c5z527db01a5565.
23. Deng Y, Guo B, Chen S. Privacy-preserving approach to edge federated learning based on blockchain and fully homomorphic encryption. Electronics. 2025;14(2):361. doi:10.3390/electronics14020361.
24. Rezaei H, Taheri R, Nowroozi E. Defense strategies in federated learning against adversarial attacks. In: Adversarial example detection and mitigation using machine learning. Berlin/Heidelberg, Germany: Springer; 2026. p. 237–48. doi:10.1007/978-3-031-99447-0_17.
25. Alandjani G. Secure and privacy-preserving wsns in smart city via blockchain-enabled federated learning and homomorphic encryption. Int J Appl Math. 2025;38(11s):1495–510. doi:10.12732/ijam.v38i11s.1263.
26. Hosseini E, Khisti A. Secure aggregation in federated learning via multiparty homomorphic encryption. In: Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps); 2021 Dec 7–11; Madrid, Spain. p. 1–6. doi:10.1109/gcwkshps52748.2021.9682053.
27. Mishra R, Bankar R. Privacy-preserving machine learning in healthcare applications. In: Intelligent healthcare system. Tamil Nadu, India: RADemics Research Institute; 2025. p. 413–42. doi:10.71443/9789349552210-14.
28. Zhang X, Zhang Q. Adaptive differential privacy mechanism for enhanced deep learning model utility and privacy. Neural Netw. 2026;196:108345. doi:10.1016/j.neunet.2025.108345.
29. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag. 2020;37(3):50–60. doi:10.1109/MSP.2020.2975749.

30. Bansal H, Goel K. Federated learning: achieving scalable and privacy-preserving machine learning. In: Proceedings of the 2023 Seventh International Conference on Image Information Processing (ICIIP); 2023 Nov 22–24; Solan, India. p. 1–6. doi:10.1109/ICIIP61524.2023.10537624.
31. Kim K, Tanuwidjaja HC. Privacy-preserving federated learning. In: Privacy-preserving deep learning. Berlin/Heidelberg, Germany: Springer; 2021. p. 55–63. doi:10.1007/978-981-16-3764-3_5.
32. Li X, Liu S, Wu F, Kumari S, Rodrigues JJPC. Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications. IEEE Internet Things J. 2019;6(3):4755–63. doi:10.1109/JIOT.2018.2874473.
33. Wang C, Sun Z, Lu J. A secure and efficient federated learning scheme based on homomorphic encryption and secret sharing. In: Proceedings of the 2024 7th International Conference on Computer Information Science and Application Technology (CISAT); 2024 Jul 12–14; Hangzhou, China. p. 1170–5. doi:10.1109/CISAT62382.2024.10695339.
34. Ogundoyin SO. An anonymous and privacy-preserving scheme for efficient traffic movement analysis in intelligent transportation system. Secur Priv. 2018;1(6):e50. doi:10.1002/spy2.50.