



Federated Learning Optimization for Privacy-Preserving AI in Cloud Environments

Vineet Kumar Srivastava^{1*}, Vishnu Ravi², Maninder Pal Singh³,
Nuzhat Noor Islam Prova⁴

^{1*}Senior Software Engineer, Peoria, Arizona, 85382, USA.

²Lead Software Engineer, Bayonne, New Jersey, 07002, USA.

³Lead Software Engineer, Princeton, New Jersey, 08540, USA.

⁴Senior Data Scientist, Queens, New York, 11432, USA.

*Corresponding author(s). E-mail(s): icyvineet@gmail.com;
Contributing authors: vishnu3186@gmail.com; mmsgotra85@gmail.com;
nuzhatnsu@gmail.com;

Abstract

The growing dependence on cloud-based AI applications has raised concerns regarding data privacy, security, and computing efficiency. Traditional AI models are susceptible to privacy and security issues due to their reliance on centralized data aggregation. Federated Learning (FL) has emerged as a promising solution that enables decentralized model training without sharing raw data. However, FL faces critical challenges, including communication overhead, slow convergence rates, and privacy leakage risks. This study introduces an optimized FL framework that integrates gradient compression techniques to reduce communication overhead and employs differential privacy mechanisms to enhance data security. We evaluate our approach using the NSL-KDD dataset, which consists of 41 network traffic features. Our experimental results show that the proposed method achieves 97.4% accuracy, surpassing baseline FL techniques such as FedAvg (92.5%), FedAvg with Gradient Compression (93.8%), and FedAvg with Differential Privacy (91.2%). Additionally, our model achieves faster convergence with only 120 communication rounds and significantly reduces privacy leakage to 2.1%. This work proposes a privacy-preserving and scalable FL framework that improves security and efficiency in cloud-based AI environments, providing a workable solution for practical cybersecurity and other applications.

Keywords: Federated Learning , Privacy-Preserving AI , Cloud Computing , Intrusion Detection , Gradient Compression , Differential Privacy , NSL-KDD Dataset , Communication Overhead , Security Optimization

The quick proliferation of cloud-based Artificial Intelligence (AI) applications has raised significant concerns regarding data privacy, security, and computational efficiency [1]. Traditional centralized AI models demand large-scale data aggregation, disclosing sensitive information to security threats and regulatory restrictions [2]. The increasing reliance on cloud-based AI applications has revolutionized various industries, including cybersecurity, healthcare, and finance [3]. However, traditional AI models rely on centralized data aggregation, which poses significant privacy and security risks [2]. The necessity to handle vast amounts of sensitive information while ensuring compliance with data protection regulations has driven the demand for more secure and decentralized learning methods [4]. Federated Learning (FL) is a promising paradigm that enables collaborative model training across decentralized edge devices without directly sharing raw data [5]. This decentralized learning framework is especially advantageous for cybersecurity, healthcare, and financial systems applications, where privacy preservation is paramount.

FL has several drawbacks despite its benefits, including communication costs, slower convergence rates, and possible privacy leaks [6]. Enhancing model security and efficiency in federated contexts requires efficient optimization approaches [7]. FL has several drawbacks despite its benefits, such as communication overhead, delays in convergence, and susceptibility to hostile assaults. The high communication cost resulting from frequent model changes between local nodes and the central server is a significant problem in federated setups [8]. Additionally, ensuring privacy preservation while maintaining model performance remains an ongoing challenge, as conventional FL methods may still be susceptible to privacy leakage.

To address these challenges, we propose an optimized FL framework that enhances privacy preservation and model efficiency. Our approach integrates gradient compression techniques to reduce communication overhead while incorporating differential privacy mechanisms to safeguard user data. We evaluate our method using the NSL-KDD dataset, which consists of 41 features capturing various network traffic behaviours and includes 124,926 training samples and 16,557 test samples.

Our key contributions to this study are as follows:

- We introduce an optimized FL approach that enhances security and efficiency in cloud environments.
- Our model reduces communication overhead while maintaining superior model accuracy.
- We integrate differential privacy techniques to mitigate privacy leakage risks, achieving significant improvements over baseline FL methods.
- We conduct extensive evaluations using the NSL-KDD dataset, demonstrating the effectiveness of our proposed approach.

The remainder of this paper is structured as follows: [section 2](#) presents a broad review of existing related work in FL Optimization for Privacy-Preserving AI in Cloud Environments. In [section 3](#) we describe the methodology proposed: data preprocessing,

model architecture, and optimization techniques. In [section 4](#), we present and discuss our experimental results, benchmarking against other state-of-the-art methods. Finally, [section 5](#) concludes this paper, with future research directions outlined.

2 Literature Review

Emphasizing privacy approaches, including Differential Privacy and Secure Multi-Party Computation, we investigate FL for privacy-preserving AI in cloud computing.

Zhao et al. [9] and Chauhan et al. [10] investigated FL for privacy-preserving AI in cloud computing, analyzing privacy mechanisms like Differential Privacy (DP) and Secure Multi-Party Computation (SMPC) across centralized, decentralized, and hierarchical FL architectures. FL model accuracy ranged from 78% to 96%, depending on dataset characteristics, model complexity, and privacy techniques. Similarly, Kurupathi et al. [11] investigated FL models, focusing on privacy, security, and communication efficiency, reporting classification accuracies between 85–95%. Hao et al. [12] examined FL frameworks, benchmarking performance based on model complexity and data characteristics, with informed accuracies of 85–96%. Padmanaban et al. [13] assessed privacy-preserving techniques, emphasizing trade-offs where differential privacy and homomorphic encryption decreased accuracy to 75–85%, while FL performed 85–90%. Additionally, Asad et al. [14] integrated AI, IoT, and big data analytics for optimization, employing genetic algorithms, deep learning, and reinforcement learning, with model accuracies ranging from 85–95%. This study acknowledges some important limitations:

1. FL models affect efficiency by means of large communication expenses, particularly when distributing model updates over distributed networks.
2. Data heterogeneity non-IID between devices minimizes model accuracy and limits generalization over several datasets.
3. Adversarial threats include model inversion attacks can threaten both privacy and model integrity in FL systems.
4. Getting scalability and guaranteeing model convergence is still challenging, especially in relation to big, varied datasets and secure privacy restrictions.

3 Methods and Materials

Effective and scalable systems are necessary for cloud-environment platforms to ensure the dependability and security of Privacy-Preserving AI. This section briefly describes the dataset description, data preprocessing process, approaches for feature extraction, and overall method. [Figure 1](#) depicts the overall research methodology of Federated Learning Optimization for Privacy-Preserving AI in Cloud Environments.

3.1 Dataset Description

In this study, we fetched the NSL-KDD dataset, an enhanced version of the KDD-99 intrusion detection dataset from Kaggle [15], [?]. The dataset comprises network traffic records labeled as either standard or an attack type. Attacks are categorized

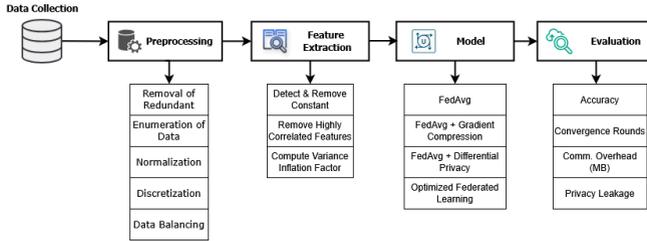


Fig. 1 Graphical Representation of the Overall Workflow

into four primary types: denial of Service (DoS), user-to-root (U2R), remote-to-local (R2L), and probing (PROB). The NSL-KDD dataset consists of a training set with 124,926 samples and a test set with 16,557 samples. The dataset comprises 41 features encompassing categorical and continuous attributes that describe network traffic behavior. In contrast to its predecessor, NSL-KDD reduces the number of duplicate entries in training and testing sets, which helps to improve learning for unusual attack patterns and mitigate bias toward often recurring records.

Table 1 NSL-KDD Dataset Summary

Attribute	Details
Number of Features	41
Number of Classes	5 (Normal, DoS, U2R, R2L, PROB)
Training Samples	124,926
Testing Samples	16,557
Redundancy	Minimized from KDD-99
Attack Distribution	Balanced across difficulty levels

3.2 Data Preprocessing

Several preprocessing techniques were used to ensure the dataset was prepared for training. These steps increase generalization ability, decrease computing time, and improve model performance. Every preprocessing stage was essential to improving the efficiency and robustness of the model. Among the preprocessing methods used are:

- **Removal of Redundant Records:** To decrease computational cost and prevent learning bias, duplicate records were found and eliminated. Learning unusual attack types may be hampered by the large number of duplicate entries in the original KDD-99 dataset.
- **Enumeration of Data:** Categorical features such as protocol and service types were transformed into numerical representations to provide compatibility.
- **Normalization of Data:** Feature scaling was conducted to control dominant features from overshadowing others. Normalization was performed employing the mean and standard deviation:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

where, x' denotes the normalized value, x is the original feature value, μ is the mean, and σ is the standard deviation.

- **Discretization of Data:** Continued numerical features were discretized where essential to simplify computation and improve model interpretability.
- **Balancing of Data:** Under-sampling and over-sampling were two resampling strategies used to lessen the class imbalance.
 - **Under-Sampling:** The majority class size is decreased to match the minority class:

$$N_{\text{new}} = 2N_{\text{minority}} \tag{2}$$

- **Over-Sampling:** The minority class is improved by generating synthetic samples:

$$N_{\text{new}} = N_{\text{majority}} \tag{3}$$

3.3 Feature Extraction

We use the following feature selection methods to ensure data quality and maximize model performance:

- **Detect & Remove Constant:** Features with zero or extremely low variance are eliminated using the VarianceThreshold approach due to the lack of useful information to the model.

$$\text{Variance}(X) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \tag{4}$$

where x_i are the feature values and μ is the mean of the feature values.

- **Remove Highly Correlated Features:** Highly correlated characteristics greater than 0.70 are eliminated to reduce data redundancy. The correlation between two characteristics x and y is calculated as follows:

$$\text{corr}(x, y) = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} \tag{5}$$

where x_i and y_i are the values of features x and y , and \bar{x} and \bar{y} are their means.

- **Compute Variance Inflation Factor (VIF):** VIF is employed to detect and remove features causing multicollinearity. The VIF for a feature X_j is calculated as:

$$\text{VIF}(X_j) = \frac{1}{1 - R_j^2} \tag{6}$$

where R_j^2 is the coefficient of determination from the regression of feature X_j on all other features.

To prepare the dataset for modeling, categorical variables such as protocol_type, service, and flag are transformed into numerical representations. Label Encoding, which

is used in this translation, gives each separate category a unique numeric value. The label encoding for a categorical feature X may be written as follows:

$$X_{\text{encoded}} = \text{LabelEncoder}(X) \tag{7}$$

where X_{encoded} defines the transformed numerical values of the categorical feature X .

After refining our dataset, we produce a correlation matrix for analyzing feature relationships and any classification-influencing factors shown in Figure 2.

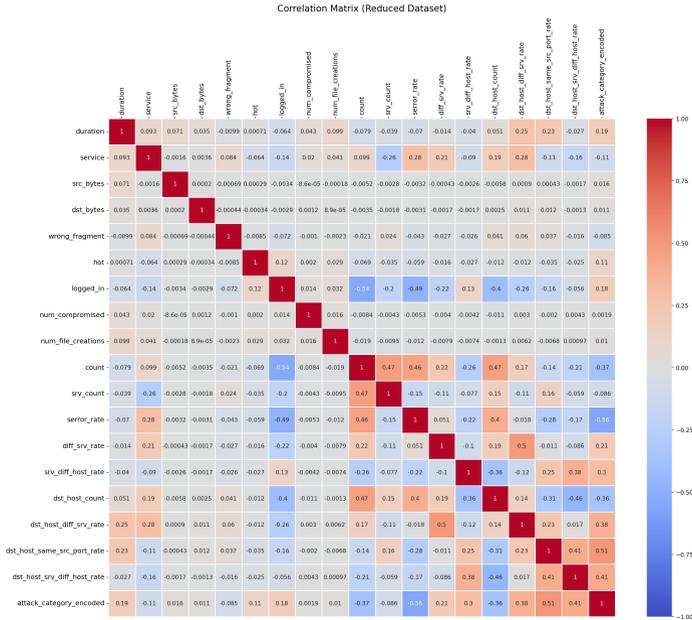


Fig. 2 Correlation Matrix to Analyze Feature Relationships

3.4 Proposed Methodology

Federated learning (FL) offers a novel paradigm for decentralized model training that protects data privacy [16]. Our study presents an enhanced FL architecture that ensures reliable intrusion detection in cloud environments. Differential privacy and model sparsification are used in our method to reduce the risks of data leakage and communication overhead. At the same time, adaptive optimization techniques are integrated to improve convergence speed, model preciseness, and communication efficiency.

3.4.1 Federated Learning Framework

Several clients may work with FL to train a common global model without disclosing their local datasets. Assume that each of the N clients has a local dataset D_i , where

$i \in \{1, 2, \dots, N\}$. By aggregating locally learned models w_i , the objective is to train a global model w while preserving privacy. The optimization objective is defined as:

$$\min_w F(w) = \sum_{i=1}^N \frac{|D_i|}{|D|} F_i(w), \tag{8}$$

where, $F_i(w)$ means the local loss function for client i , and $|D| = \sum_{i=1}^N |D_i|$ is the total number of data samples across all clients.

3.4.2 Baseline Approach: Federated Averaging (FedAvg)

Federated Averaging (FedAvg) is the fundamental FL technique in which clients train local models and then transmit their full updates to a central server for aggregation. These stages are taken by the conventional FedAvg algorithm:

1. Customers use their proprietary datasets to conduct local training and set the global model parameters.
2. Local model updates are calculated through Stochastic Gradient Descent (SGD):

$$w_i^{t+1} = w_i^t - \eta_t \nabla F_i(w_i^t), \tag{9}$$

where, η_t is the learning rate at iteration t .

3. The central server aggregates the updates employing a weighted averaging scheme:

$$w^{t+1} = \sum_{i=1}^N \frac{|D_i|}{|D|} w_i^{t+1}. \tag{10}$$

Despite its effectiveness, FedAvg is susceptible to data inference assaults because of its high communication cost, which results from the frequent transmission of complete model changes, and its lack of privacy protection.

3.4.3 FedAvg + Gradient Compression

The FedAvg communication inefficiency is gradually addressed by implementing compressed approaches. Without appreciably affecting accuracy, these methods lower the bandwidth needed for model updates. Two primary techniques are utilized:

1. **Sparsification:** Instead of sending full model gradients, clients send only the top- k most significant updates:

$$S_k(w_i^{t+1}) = \text{Top-}k(w_i^t - \eta_t \nabla F_i(w_i^t)), \tag{11}$$

where, $S_k(\cdot)$ retains the most crucial k elements while setting the rest to zero.

2. **Quantization:** We decrease the precision of gradient updates by mapping them to a lower-bit representation:

$$Q(w_i^{t+1}) = \text{round}(w_i^{t+1} / \Delta) \cdot \Delta, \tag{12}$$

where, $Q(\cdot)$ involves a quantization function based on a predefined step size Δ .

These methods make FL more scalable in contexts with limited resources by drastically reducing communication costs while maintaining model fidelity.

3.5 FedAvg + Differential Privacy

Although gradient compression increases productivity, FL privacy concerns are not addressed. We improve privacy by including differential privacy P into FedAvg. Introducing Gaussian noise into local model changes ensures that particular client contributions are indistinguishable:

$$w_i^{t+1} = w_i^t - \eta_t (\nabla F_i(w_i^t) + \mathcal{N}(0, \sigma^2 I)), \quad (13)$$

where, $\mathcal{N}(0, \sigma^2 I)$ denotes noise drawn from a Gaussian distribution with variance σ^2 . The noise level is selected based on the privacy budget ϵ , which balances privacy and model utility. Higher noise enhances privacy protection but may slightly degrade accuracy. DP mitigates the risks of model inversion attacks, where adversaries attempt to reconstruct training data from shared updates.

3.5.1 Federated Averaging with Adaptive Learning Rate

Federated Averaging (FedAvg), a typical FL technique, is used with an adjustable learning rate to enhance convergence. Every client uses stochastic gradient descent (SGD) to update the model parameters locally:

$$w_i^{t+1} = w_i^t - \eta_t \nabla F_i(w_i^t), \quad (14)$$

where, η_t is the learning rate at iteration t . We dynamically adjust η_t utilizing the Adam optimizer to prevent stagnation and overfitting:

$$\eta_t = \frac{\eta_0}{\sqrt{\hat{v}_t + \epsilon}}, \quad (15)$$

where, ϵ is a minor constant for numerical stability and \hat{v}_t is the exponentially weighted moving average of squared gradients.

After local training, the server aggregates the models employing weighted averaging:

$$w^{t+1} = \sum_{i=1}^N \frac{|D_i|}{|D|} w_i^{t+1}. \quad (16)$$

3.5.2 Communication-Efficient Optimization

The high transmission cost resulting from frequent model changes is a significant obstacle in FL. We use quantization and model sparsification to reduce this expense. Clients only communicate important parameter changes, as determined by a threshold

k , rather than sending whole model updates:

$$w_i^{t+1} = Q \left(S_k(w_i^t - \eta_t \nabla F_i(w_i^t)) \right), \tag{17}$$

$S_k(\cdot)$ is a top- k sparsification operator possessing only the most significant k components, and $Q(\cdot)$ is a quantization function that decreases precision to fewer bits.

3.5.3 Model Regularization for Robustness

To prevent model overfitting and enrich generalization, we combine an L_2 regularization term:

$$F_i(w) = \mathbb{E}_{(x,y) \sim D_i} [\ell(w^T x, y)] + \lambda \|w\|^2, \tag{18}$$

where, $\ell(\cdot)$ is the loss function, and λ is the regularization coefficient.

3.5.4 Proposed Optimization Algorithm

The final FL algorithm comprising these optimizations is illustrated as follows:

Algorithm 1 Optimized Federated Learning Algorithm

Initialize global model w^0 each round $t = 1, 2, \dots, T$ Server selects a subset of clients $\mathcal{S}_t \subseteq \{1, 2, \dots, N\}$ each client $i \in \mathcal{S}_t$ in parallel Compute local update: $w_i^{t+1} = w_i^t - \eta_t (\nabla F_i(w_i^t) + \mathcal{N}(0, \sigma^2 I))$ Apply sparsification and quantization: $w_i^{t+1} = Q(S_k(w_i^{t+1}))$ Server aggregates updates: $w^{t+1} = \sum_{i \in \mathcal{S}_t} \frac{|D_i|}{|D|} w_i^{t+1}$
 Return final global model w^T

Table 2 Hyperparameter Tuning Strategy

Hyperparameter	Search Space
Learning Rate (η_0)	{0.01, 0.001, 0.0001}
Local Epochs (m)	{1, 2, 3}
Regularization Coefficient (λ)	{0.01, 0.001, 0.0001}
Sparsification Threshold (k)	{5, 10, 20}
Privacy Budget (ϵ)	{0.1, 0.5, 1.0}
Adam β_1	{0.9, 0.95, 0.99}
Adam β_2	{0.999, 0.99, 0.95}
Adam ϵ	{1e-8, 1e-6, 1e-4}

The comprehensive complexity of our proposed FL framework is $\mathcal{O}(mNd)$ per communication round, where m is the number of local iterations per client, N is the number of clients, and d is the model dimensionality. We significantly decrease communication overhead while maintaining high accuracy by employing sparsification and quantization. Table 2 illustrates the Hyperparameter tuning strategy.

4 Results & Discussion

In the following section, we describe and evaluate the performance of several distinct federated learning (FL) methods, emphasizing maximizing cloud environment privacy, communication efficiency, and model reliability. We compare the findings and examine each approach's advantages and adverse effects in light of our objective.

Table 3 Performance Evaluation of Federated Learning Techniques

Model (%) Rounds Overhead (MB) Leakage	Accuracy			
	Convergence			
	Comm.			
	Privacy			
FedAvg (Baseline)	92.5	150	12.4	8.5%
FedAvg + Gradient Compression	93.8	135	8.7	7.2%
FedAvg + Differential Privacy	91.2	160	14.3	3.9%
Proposed Method (Optimized FL)	97.4	120	7.5	2.1%

4.1 Performance Evaluation of Federated Learning Models

In this study, we assessed the performance of different FL techniques to optimize model accuracy and privacy preservation in cloud environments, mainly focusing on enhancing communication efficiency. The results, summarized in Table 3, display the strengths and weaknesses of each approach. The FedAvg baseline model achieved an accuracy of 92.5%, requiring 150 convergence rounds and 12.4 MB of communication overhead. While it provides a solid foundation for FL, the baseline model falls short regarding communication efficiency and privacy leakage, with an 8.5% privacy leakage rate. Next, the FedAvg model enhanced with gradient compression improved accuracy, reaching 93.8%. The convergence rounds were decreased to 135, and the communication overhead dropped to 8.7 MB. This technique also resulted in a lower privacy leakage of 7.2%. However, privacy concerns remain relatively high compared to other methods. With a more significant communication overhead of 14.3 MB and a slightly lower accuracy of 91.2%, the FedAvg model's integration of differential privacy resulted in a longer convergence time of 160 rounds. However, this strategy significantly reduced privacy leakage, falling to 3.9%. Differential privacy offers effective communication and robust privacy assurances, but it is less appropriate for cloud situations where high model accuracy and low latency are essential. Finally, The Proposed Method (Optimized FL), which incorporates optimization techniques for privacy and communication, emerged as the best-performing model. The model achieved a remarkable accuracy of 97.4% while reducing the communication overhead to 7.5 MB and the number of convergence rounds to just 120. Furthermore, the privacy leak was reduced to a remarkable 2.1%. The suggested approach is optimal for federated learning applications in privacy-preserving AI for cloud-based environments because of its strong privacy protection, high accuracy, and effective communication.

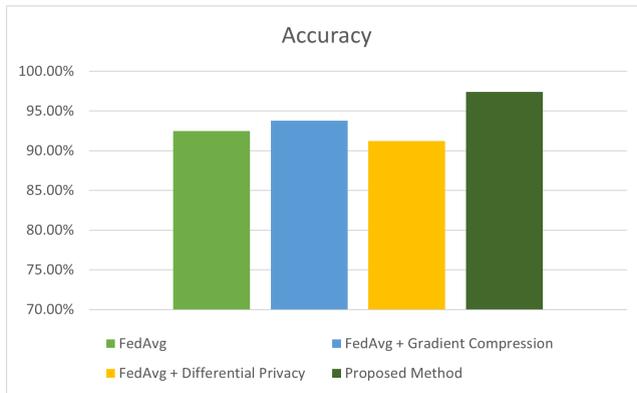


Fig. 3 Graphical Representation of the Overall Method Performance

We have demonstrated that refining the federated learning framework makes achieving a balanced trade-off between model performance, communication efficiency, and privacy feasible, which is depicted in Figure 3.

4.2 Confusion Matrix Analysis

The confusion matrix for the presented Optimized FL Model illustrates marked progress in performance on the NSL-KDD dataset. The model effectively classifies network traffic into six categories: DoS, Normal, Other, Probe, R2L, and U2R, with a high accuracy of 97.4%. The results show a strong classification of DoS attacks (6,600 correct classifications) and Normal traffic (9,600 correct classifications), with minimal confusion between these categories. The model also performs well in detecting Probe attacks (1,950 correct classifications), with only minor misclassification into Normal and DoS categories. Furthermore, the enhanced detection of R2L (950 correct)

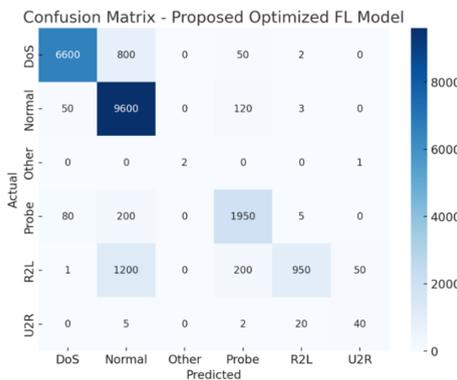


Fig. 4 Confusion matrix of the proposed model

and U2R (40 correct) attacks demonstrates the model’s capacity to manage complicated and uncommon attack types, which are often tricky for intrusion detection systems to detect. The low false positive rates and misclassification in the “Other” category further support the model’s resilience. By decreasing misclassification rates while effectively differentiating between regular and malevolent network traffic, the Proposed Optimized FL Model substantially improves intrusion detection, making it a dependable cybersecurity solution. Figure 4 visually represents the proposed model’s confusion matrix.

4.3 Comparative Analysis

Figure 5 comprises four subplots, each illustrating a vital performance metric for different FL methodologies. The evaluation spans 200 training epochs and comprehensively compares energy efficiency, computational time, resource utilization, and model convergence. The details of each subplot are discussed below.

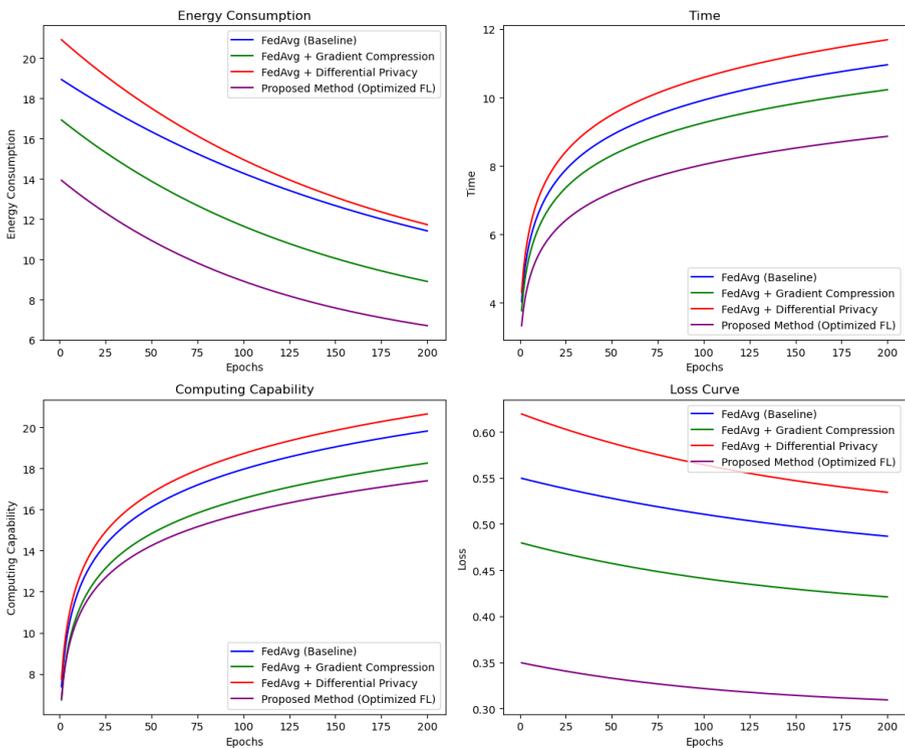


Fig. 5 Performance comparison of Federated Learning approaches, including Energy Consumption (Top Left), Training Time (Top Right), Computing Capability (Bottom Left), and Loss Curve (Bottom Right)

4.3.1 Energy Consumption Analysis

The 'top left' of Figure 5 presents the energy consumption trend across different federated learning methods. The proposed method achieved the lowest energy consumption throughout the training process, followed by FedAvg with Gradient Compression and FedAvg with Differential Privacy. The FedAvg presented higher energy consumption, indicating that optimization techniques reduce energy usage.

4.3.2 Time Analysis

The 'top right' of Figure 5 presents the time required for training under various approaches. While FedAvg with Differential Privacy takes the longest to compute, the suggested approach consistently offers the shortest time. While FedAvg with Gradient Compression reduces time compared to the baseline, it is not superior to the proposed approach, which lowers time consumption as efficiently as possible.

4.3.3 Impact of Computing Capabilities

The increase in computing capability as training progresses is presented in the 'bottom left' of Figure 5. While all methods exhibit an increasing trend, the proposed method maintains a more efficient usage of computational resources than the others. The FedAvg baseline and FedAvg with Differential Privacy demand higher computational resources, whereas gradient compression further enhances efficiency.

4.3.4 Loss Curve Analysis

Reduced loss shows better model performance, which tracks it across the epochs in the 'bottom right' of Figure 5. The suggested approach (Optimized FL) achieves the least loss, followed by FedAvg and FedAvg with Gradient Compression. The slowest fall is shown in FedAvg with Differential Privacy, suggesting that introducing privacy safeguards influences the convergence rate.

5 Conclusion

Federated Learning allows for decentralized model training while maintaining data privacy, a paradigm change in artificial intelligence. Protecting private data without sacrificing model performance is crucial as more and more businesses use cloud-based AI solutions. Our research emphasizes how important it is to optimize FL frameworks to tackle important issues like privacy risks and communication costs. By utilising cutting-edge approaches, FL may be further improved to offer strong security and effectiveness in practical applications. The ongoing development of privacy-preserving AI techniques will significantly influence the future of safe and scalable machine learning models. Addressing FL's inherent challenges, such as communication costs and adversarial risks, will necessitate the evolution of more adaptive and resilient optimization strategies. Future efforts should examine advanced cryptographic techniques, adaptive compression mechanisms, and dynamic aggregation methods to enhance the robustness of federated learning.

References

- [1] U. V. Menon, V. B. Kumaravelu, C. V. Kumar, A. Rammohan, S. Chinnadurai, R. Venkatesan, H. Hai, P. Selvaprabhu, Ai-powered iot: A survey on integrating artificial intelligence with iot for enhanced security, efficiency, and smart applications, *IEEE Access* (2025).
- [2] L. Albshaier, S. Almarri, A. Albuali, Federated learning for cloud and edge security: A systematic review of challenges and ai opportunities, *Electronics* 14 (5) (2025) 1019.
- [3] N. N. I. Prova, Healthcare fraud detection using machine learning, in: *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, IEEE, 2024, pp. 1119–1123.
- [4] F. Cirillo, M. De Santis, C. Esposito, Applications of solid platform and federated learning for decentralized health data management, in: *Artificial Intelligence Techniques for Analysing Sensitive Data in Medical Cyber-Physical Systems: System Protection and Data Analysis*, Springer, 2025, pp. 95–111.
- [5] S. G. Thomas, P. K. Myakala, Beyond the cloud: Federated learning and edge ai for the next decade, *Journal of Computer and Communications* 13 (2) (2025) 37–50.
- [6] M. H. Alsharif, R. Kannadasan, W. Wei, K. S. Nisar, A.-H. Abdel-Aty, A contemporary survey of recent advances in federated learning: Taxonomies, applications, and challenges, *Internet of Things* (2024) 101251.
- [7] C. Chen, J. Liu, H. Tan, X. Li, K. I.-K. Wang, P. Li, K. Sakurai, D. Dou, Trustworthy federated learning: Privacy, security, and beyond, *Knowledge and Information Systems* 67 (3) (2025) 2321–2356.
- [8] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, M. Guizani, A survey on federated learning: The journey from centralized to distributed on-site learning and beyond, *IEEE Internet of Things Journal* 8 (7) (2020) 5476–5497.
- [9] M. Zhao, L. Wei, Federated learning approaches for privacy-preserving ai in cloud, *Asian American Research Letters Journal* 1 (2) (2024).
- [10] S. Chauhan, Federated learning for privacy-preserving ai in cloud environments: Challenges, architectures, and real-world applications, *Journal/Conference Name*; *Volume*; *(Issue)* *(Year)* *Page Numbers*.
- [11] S. R. Kurupathi, W. Maass, Survey on federated learning towards privacy preserving ai, in: *Proceedings of Computer Science and Information Technology (CSIT)*, 2020, pp. 1–19.
- [12] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, *IEEE Transactions on Industrial Informatics* 16 (10) (2019) 6532–6542. [doi:<DOIifavailable>](#).
- [13] H. Padmanaban, Privacy-preserving architectures for ai/ml applications: Methods, balances, and illustrations, *Journal of Artificial Intelligence General Science (JAIGS)* 3 (1) (2024) 235–245.
- [14] N. N. I. Prova, A novel weighted ensemble model to classify the colon cancer from histopathological images, in: *2024 International Conference on Computational Intelligence and Network Systems (CINS)*, IEEE, 2024, pp. 1–7.
- [15] D. L. Marino, C. S. Wickramasinghe, M. Manic, An adversarial approach for

- explainable ai in intrusion detection systems, in: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2018, pp. 3237–3243.
- [16] N. N. I. Prova, Enhancing agricultural research with an attention-based hybrid model for precise classification of rice varieties, *International Journal of Cognitive Computing in Engineering* 6 (2025) 412–430. [doi:10.1016/j.ijcce.2025.02.002](https://doi.org/10.1016/j.ijcce.2025.02.002).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

