Received: June 15, 2025. Revised: August 9, 2025.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Advanced Differential Privacy Mechanisms for Deep and Federated Learning Based on MNIST: An Empirical Analysis

Sarah H. Mnkash[1]* Faiz A. Alawy[2] Israa T. Ali[3]

[1]*University of Technology, Computer Science Department, Baghdad, Iraq*
[2]*College of Engineering, Kent State University, Ohio, USA*
[3]*Computer Science Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: cs.22.13@grad.uotechnology.edu.iq

Abstract: Decentralized deep learning model training among dispersed devices is made possible by Federated Learning (FL). The MNIST and Fashion MNIST (FMNIST) datasets are used in this paper to demonstrate a privacy-preserving Federated Learning (FL) framework combined with user-level Differential Privacy (DP) for decentralized model training. The data is distributed across 10 clients, each training a Stacked Convolutional Neural Network (CNN) locally. Training gradients with calibrated noise guarantees differential privacy. The models are aggregated via Federated Averaging after local training, when encrypted updates are relayed to a central server. The procedure is iterated until the global model reaches a convergence point. Metrics for recall, accuracy, and precision are used to assess the final model. Outcomes show that a proposed model achieves 85.73% accuracy on MNIST and 67.99% on FMNIST, outperforming baseline models like VGG, ResNet, and traditional CNNs. The study demonstrates that the Stacked CNN architecture effectively balances performance and privacy, making it suitable for deployment in privacy-sensitive, distributed environments. The framework provides valuable insights into enhancing model utility while maintaining robust privacy protections in federated settings.

Keywords: Differential privacy, Federated learning, Convolutional neural network (CNN), MNIST, Fashion MNIST.

1. Introduction

In recent years, DL represents significant success in a variety of industries, including healthcare, marketing, transportation, and others [1]. Applying a latest development in other areas of computer vision to image analysis that relies on deep learning is not without its constraints. This is due to the fact that obtaining diagnostically useful levels of accuracy in several technological applications requires enormous volumes of data. Acquiring such datasets, however, presents a challenge in the medical industry due to the requirement to minimize data collection and sharing while still upholding patient privacy [2-4]. The privacy of users in DL must thus be ensured. Federated learning satisfies the specific requirements for data privacy protection at the user level by providing a distributed learning framework that enables users to train models while keeping the original data [5]. A common method for enhancing privacy protection is differential privacy, which is a classic data desensitization technique [6].

One practical solution is the rise of privacy-enhancing technologies, which make it possible to draw inferences from sensitive data without compromising the privacy of the people concerned. The best chance for promoting ethical and responsible data interchange has never been presented by these technologies [7-9]. A new technology called federated learning (FL) trains ML models across several decentralized systems. Through the use of a server, it allows local devices to pool their computations and develop a model together. FL is a novel approach that aims to centralize user device training while protecting the privacy of the underlying data. This greatly decreases the likelihood of data breaches by doing away with the need to transmit raw data to a single repository. Also, to make

sure that a training process is private, a central server handles all of a model change, both transmitting and receiving them from the devices [10].

Another approach that might enhance privacy without compromising accuracy is DP, which masks sensitive information by adding noise to a data. One of a most basic ML strategies for keeping data secure while also allowing individuals their privacy is DP. The fact that it safeguards user-specific data, guarantees anonymity in outputs, and improves privacy without depending on trust assumptions is its strongest strength [8]. Recently, FL and DP have become more popular, particularly in a healthcare and IoT sectors. These combined tactics aim to strike a balance among privacy protection and effective training. However, maintaining model privacy without compromising accuracy continues to be an ongoing challenge. Adding noise to the training process and consistently trusting the central server were common practices in earlier efforts to apply DP to FL.

1.1 Motivation and contribution of the paper

The motivation behind this research lies in addressing the growing concern of data privacy in machine learning, particularly in decentralized settings. As machine learning models increasingly rely on distributed data sources, traditional centralized approaches pose significant risks to sensitive information. This study's significance is in its innovative application of FL combined with DP to safeguard individual data while enabling collaborative model training. By leveraging Stacked Convolutional Neural Networks (CNNs) within this framework, the research demonstrates a viable method to improve privacy protection without compromising model performance. This approach not only advances the field of privacy-preserving ML but also has practical implications for industries requiring secure and effective data utilization, such as finance, healthcare, and beyond. The following research contribution of this work is:

- To demonstrate federated learning with differential privacy: This research introduces a FL framework integrated with DP mechanisms, showcasing how decentralized model training can be enhanced with privacy-preserving techniques using Stacked Convolutional Neural Networks (CNNs).
- To enhance privacy through differential privacy mechanisms: The study employs differential privacy methods, specifically adding noise to gradients using Laplace and Gaussian mechanisms, to ensure robust protection of individual data privacy while maintaining the model's effectiveness.
- To optimize model training with stacked CNN architecture: By using a Stacked CNN model, this research highlights improvements in accuracy and loss metrics compared to baseline models, demonstrating the effectiveness of deep learning architectures in handling privacy-preserving federated learning scenarios.
- To Compare Performance on MNIST and Fashion MNIST Datasets: This study sheds light on how well the FL-DP method works for various image classification tasks by comparing the model's performance on the MNIST and Fashion MNIST datasets.

1.2 Structure of the paper

A following paper structure as: Section I provide the overview of topic with motivation and contribution. Then Section II discussed some existing work on this research area with summary table. Section III described the proposed methodology with each step and each technique. Section IV provide the experimental results of proposed work with comparative analysis in existing technique. Section V concludes the article and discusses its future directions.

2. Literature review

The combination of DL with DP for image classification is the subject of many research initiatives aimed at improving the security of big data. DL and DP federated learning have been the subject of several studies that compare and contrast their respective efficacy.

In [11], propose a gradient recovery attack against differential privacy-preserving models. Adversaries can gather a small set of differential privacy perturbed and original gradients and leverage a GAN network to train a gradient recovery model, which can recover the perturbed gradient to an approximately original state. Experiments demonstrate that the proposed gradient recovery attack achieves promising results when using commonly used differential privacy budgets (from 1 to 7). When the privacy budget is 7, the recovered gradient and the original gradient have a similarity of 94-99% in gradient inversion attack.

In [12], presents DP Patch, a new framework that attempts to solve these privacy issues in image data by focussing on potentially sensitive elements within the picture instead of treating the whole image as sensitive. This method produces more useful, privacy-preserving photos than DP images. They

compare the custom model's performance to that of state-of-the-art alternatives and do experimental assessments to determine how well the produced privacy-preserving photos meet criteria.

In [13], improving the hyper-parameters while keeping accuracy and privacy in mind is essential for building differentially private models in ML. This study investigates the best designs and trade-offs for private learning by investigating the outcomes of hyper-parameter tuning and the Pareto frontier technique, with the goal of avoiding this issue. The research improves knowledge of privacy concerns and provides data that may guide decisions about real-world applications and the creation of efficient training methods [38].

In [14], conduct the first thorough, principled measurement study to find out when a pre-trained encoder may overcome the limitations of secure or privacy-preserving Supervised Learning techniques. Their main results show that a pre-trained encoder greatly enhances 1) accuracy in the absence of attacks and 2) assessment of protection against Data Poisoning and backdoor attacks in cutting-edge secure learning algorithms (such as KNN and bagging) 2) verification of security against adversarial examples in randomized smoothing without compromising accuracy in the absence of attacks and 3) accuracy of differentially private classifiers.

Create a unique optimization method named DPAGDCNN for convolution neural networks that works with DP techniques in order to secure the privacy of DL models [15]. Specifically, instead of allocating a set privacy budget every iteration, DPAGD-CNN distributes privacy budgets more wisely in every iteration. They demonstrate theoretically that the method may safeguard training data privacy while attaining improved classification accuracy in the MNIST and CIFAR-10 datasets while maintaining a low privacy budget.

In [16], study presents a novel hybrid approach that HDP-FL to tackle the basic problem of finding a balance among data privacy and value in distributed learning. The results of this hybrid approach's careful testing on the EMNIST and CIFAR-10 datasets show significant improvements over traditional federated learning methods. Specifically, it achieves an impressive accuracy of 96.29% for EMNIST and 82.88% for CIFAR-10.

In [17], provide a federated learning architecture that uses LDP to accommodate customers' unique privacy needs. Model perturbation strategies are planned separately for IID and non-IID datasets, as well as for both kinds of datasets. They provide a weighted average technique and a probability-based selection method for aggregating models. Lessening the effect of privacy-conscious customers with minimal privacy expenditures on the federated model is the main notion. Experiments on three well-known datasets (MNIST, Fashion-MNIST, and forest cover types) show that the proposed aggregation techniques beat the old arithmetic average approach in situations where privacy is being protected.

In [33], research explores efficient Anomaly Detection techniques for IIoT devices because device irregularities generate substantial security challenges along with privacy threats. A hierarchical federated learning system implements deep reinforcement learning to construct one common anomaly detection model for all users according to the authors. The system empowers several devices to run individual models on their local systems prior to swapping combined findings instead of distributing raw data fields thus enhancing data privacy. Two detection metrics enhance accuracy by assessing privacy leakage level and action interconnectivity within the system. The proposed method provides experimental data illustrating fast system operations and quick responsiveness together with robust anomaly detection functionality for various IIoT applications.

In [34] A 5G-powered Intelligent Transportation System operates by integrating multiple Blockchain technologies which use AI-enabled trust evaluation hierarchies.

The paper creates BHTE (Blockchain-based Hierarchical Trust Evaluation) that aims to strengthen trust reliability within 5G-enabled Intelligent Transportation Systems (ITS). Different blockchain technologies link with federated deep learning methods to assess trustworthiness in users and task distributors of ITS through a blockchain-based system. Federated learning lets the system assess trust levels by maintaining user privacy through keeping private data inaccessible. The blockchain hierarchical system maintains trust information through an infrastructure that offers traceable and transparent data storage capabilities. The detailed experimental evaluations have proved that BHTE can determine trustworthy assessments while exhibiting quick processing time and maintaining brief system response delays.

[35] The work presents a Federated Reinforcement Learning solution which ensures both Quality of Service requirements and routing privacy protection in 5G-Enabled IIoT environments.

The Federated Reinforcement Learning optimization framework develops a solution to improve 5G-enabled IIoT routing mechanisms by integrating QoS and privacy features. The new

Table 1. Related work summary for data privacy using different tool and techniques

Reference	Methodology	Results/Advantage	Limitations/Research Gaps	Future Work
[11]	Utilizes GAN to train a model that recovers perturbed gradients to near-original states in differential privacy-preserving models.	Achieved 94-99% similarity between recovered and original gradients when privacy budget is 7.	Limited to specific DP budgets; applicability in real-world scenarios may be constrained by computational complexity.	Extend to broader datasets and more complex models; explore defenses against such attacks.
[12]	Focuses on protecting specific sensitive objects within an image rather than the entire image, enhancing utility while preserving privacy.	Produced privacy-preserving images with higher utility compared to standard DP methods.	May not generalize well to all image types; requires precise identification of sensitive regions.	Expand framework to include more sophisticated object detection mechanisms and evaluate across diverse datasets.
[13]	Analyzes hyper-parameter tuning for differential privacy, employing Pareto frontier to identify optimal trade-offs.	Enhanced understanding of privacy vs. accuracy trade-offs; optimized architectures for private learning.	Focused on theoretical analysis; real-world application may differ.	Apply findings to larger, more complex models and in real-world scenarios.
[14]	Evaluates the impact of using a pre-trained encoder in secure learning algorithms like bagging, KNN, and randomized smoothing.	Improved accuracy under no attacks and security guarantees against data poisoning, backdoor attacks, and adversarial examples.	Dependent on the quality and relevance of the pre-trained encoder; might not suit all types of data.	Test on more diverse datasets and integrate with other privacy-preserving methods.
[15]	Introduces a novel algorithm that carefully allocates privacy budgets in each iteration for CNN models, enhancing accuracy.	Higher classification accuracy under moderate privacy budgets on MNIST and CIFAR-10 datasets.	Limited testing on only two datasets; potential scalability issues.	Explore scalability to larger datasets and more complex architectures.
[16]	Combines DP with federated learning, achieving enhanced model accuracy.	Achieved accuracies of 96.29% for EMNIST and 82.88% for CIFAR-10, significantly outperforming conventional methods.	Limited to specific datasets; might require heavy computational resources.	Extend to other types of data and reduce computational overhead.
[17]	Proposes personalized privacy requirements for clients using model perturbation and novel aggregation methods.	Better performance than classic methods in personalized privacy scenarios on MNIST, Fashion-MNIST, and forest cover-types datasets.	Dependent on privacy budget choices of clients; may not be applicable to highly sensitive data.	Further refine aggregation methods and test on non-IID data with varying degrees of privacy concerns.

method allows IIoT devices to discover maximum routing policies with other devices through federated learning but blocks raw data access. Reinforcement learning operates in this system to make the network adjust its performance for secure yet dependable data delivery during changing conditions. The simulation demonstrates that this methodology meets QoS requirements as well as privacy security needs in complex IIoT systems. A variety of research investigations show that combining federated learning with blockchain technologies provides IIoT and ITS domains with both enhanced privacy management and more security-aligned efficiency solutions.

2.1 Research limitations

Toward Accurate Anomaly Detection in Industrial IoT:

- The method incurs excessive computing expenses that makes it unusable for power-constrained IoT devices.
- The process of data aggregation causes fine-grained details to disappear which negatively impacts the accuracy of anomaly detection systems.
- Lack of advanced attack testing, such as adversarial manipulations.
- Heterogeneous Blockchain & AI-Driven Trust Evaluation for 5G-ITS: The energy

consumption levels pertaining to blockchain operations along with federated learning remain high.
- The system delays can potentially create problems for real-time processing of decisions.
- The system depends mainly on simulations for validation although it lacks testing in real-world conditions.
- QoS & Privacy-Aware Routing in 5G Industrial IoT: Complexity in implementation due to federated reinforcement learning.
- The expansion capabilities of large dynamic network systems present significant scalability challenges.
- The design fails to consider DDoS attacks as well as other cybersecurity threats.

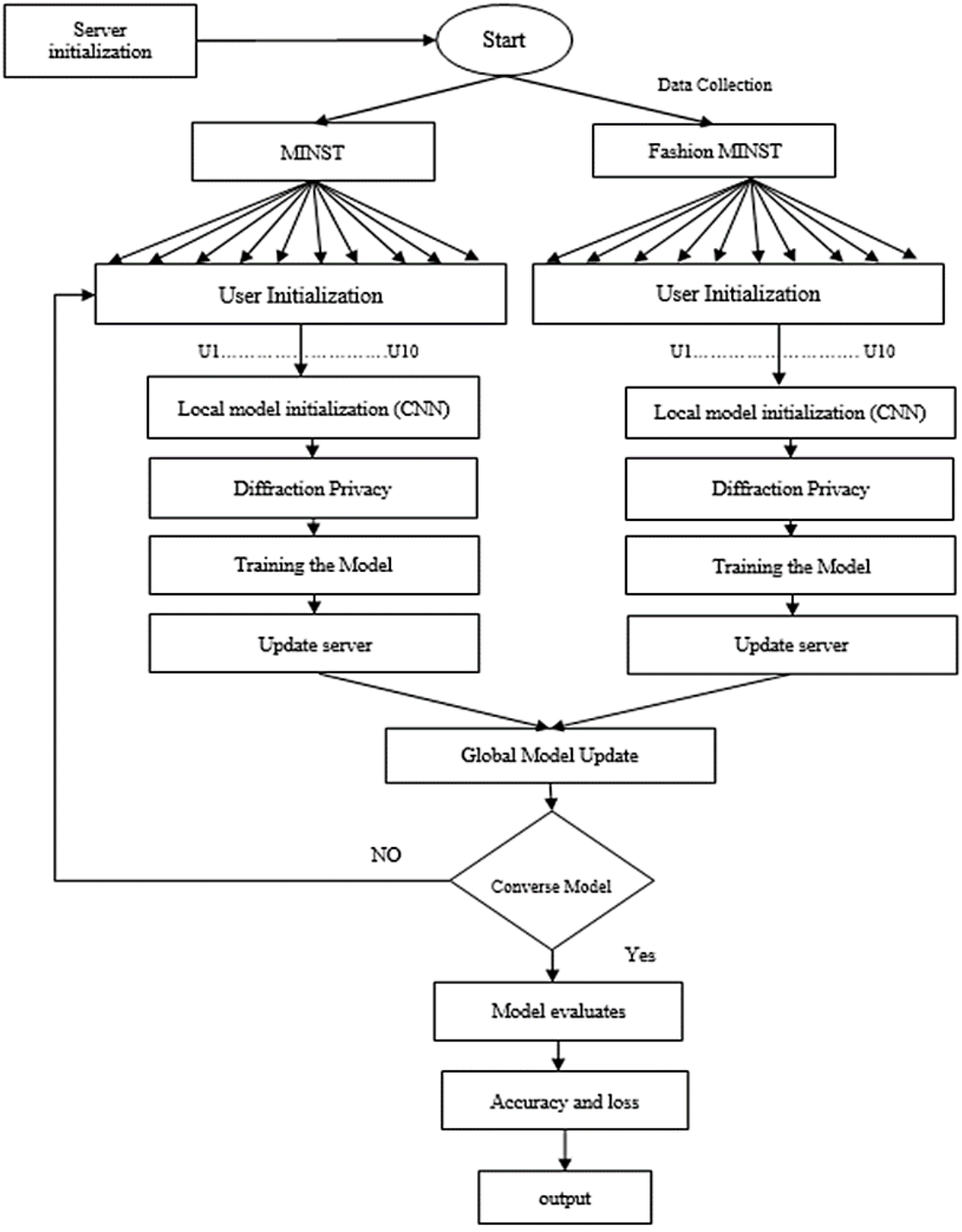

Figure. 1 Flowchart of Federated Learning (FL) for Differential Privacy (DP) with Stack CNN Model

3. Research methodology

The proposed methodology involves implementing a FL framework with DP on the Fashion MNIST and MNIST datasets, focusing on privacy-preserving decentralized model training. The data is distributed across 10 clients, each receiving balanced subsets for local processing. Each client initializes and trains a Stacked Convolutional Neural Network (CNN) model locally, employing DP mechanisms by adding noise to gradients to protect individual data privacy. Using a batch size of 128 and a learning rate of 0.01 across 20 epochs, the Adam optimizer is employed. Following local training, the clients assess their models and transmit safely encrypted model changes to a central server. Federated Averaging is used by the server to aggregate these changes and make sure that there are no abnormal gradients. After that, the global model is revised and reissued to the customers. The model is iterated again and over until it converges. The final global model is then evaluated for accuracy, precision, and recall on test data and deployed on edge devices with ongoing monitoring and updates, ensuring robust, privacy-preserving model training. This hybrid model, leveraging the Stacked CNN architecture, demonstrates superior accuracy compared to the baseline approach.

The flowchart illustrated in Fig. 1, the process of FL for DP using a Stack Convolutional Neural Network (CNN) model. It begins with server initialization and data collection, followed by user initialization for multiple users (U1…U10). Each user locally initializes their CNN model, applies differential privacy mechanisms, and trains the model. The models are assessed after training, and if they satisfy the accuracy and loss requirements, they are utilized to update the server's global model. The privacy of individual user data is preserved while the global model is improved due to this iterative procedure. The flowchart clearly illustrates how differential privacy and federated learning may be used to improve model performance while safeguarding user data. The method concludes with deploying the model for inference tasks once its evaluation on test data confirms it satisfies performance standards.

We have included a brief discussion suggesting well-known strategies for future enhancement, such as:

- **FedProx**: to reduce client drift by penalizing divergence from the global model.
- **Data augmentation**: at the client side to smooth class imbalance.
- **Clustered FL or personalized FL**: for more heterogeneous data populations.
- **Adaptive weighting**: based on client loss or performance.

3.1 Data collection

In this paper utilized two image classification datasets: MNIST and Fashion MNIST from the Kaggle website.

- The handwritten digit labels and images ranging from 0 to 9 are included in the MNIST handwritten digits dataset. Every sample, consisting of a 28 × 28-pixel greyscale picture of a handwritten number, is one of 10,000 test samples and 60,000 training instances. Additionally, it makes use of the digit's dataset, which is already a part of scikit-learn. This is a carbon copy of the reference set for the MNIST handwritten digits dataset, which consists of 1,797 samples with 64 features. The pictures in each example have an 8 × 8 size and include integer pixels ranging from 0 to 16.
- The Fashion-MNIST dataset has 60,000 training samples and 10,000 testing instances made up of pictures from Zalando articles. A greyscale picture of 28x28 pixels is paired with a label from one of ten categories in each case. When it comes time to evaluating ML algorithms, Zalando plans to use Fashion-MNIST instead of the original MNIST dataset. Both the training and testing sets of images are identical in size and structure.

3.2 Data preprocessing

Data preprocessing in this research involves several critical steps to ensure the effectiveness and privacy of the FL framework. Initially, the Fashion MNIST and MNIST datasets are partitioned into balanced subsets, with each subset assigned to a specific client, ensuring uniform data distribution across the network. Each client then applies standard preprocessing techniques, including normalization of pixel values to a range between 0 and 1, which helps in stabilizing and accelerating the training process. The CNN model also makes use of label encoding, which transforms the category labels into a numerical representation. To enhance privacy, differential privacy (DP) mechanisms are incorporated during preprocessing, where noise is strategically added to gradients. This step is crucial for safeguarding user privacy and ensuring accurate local model training by preventing individual data points from being inferred from model changes.

3.2.1. Federated learning (FL) with differential privacy (DP) framework

One such approach is the deep learning method known as FL. With FL, it is possible to train a model employing data by several local clients. The local data is not shared, thereby ensuring data privacy and security [18]. Training a global model with the help of local models at every customer is what federated learning is all about. At iteration t, it has the global model parameters denoted as w_t. Every client i holds a local dataset Di and computes a local model update Δw_(t,i) based on Di and w_t. A global model is then updated using the aggregated local changes.

To integrate DP, it adds carefully calibrated noise to each local model update. This noise addition process as a Laplace mechanism, which is commonly used in differential privacy due to its simple analytical properties. The Laplace Mechanism is commonly used when dealing with numerical query outputs. The function's output is supplemented with noise obtained from the Laplace distribution to make it operate. The scale of this noise is proportional to the function's L1-sensitivity, which measures how much the function's output can change due to a change in one record. The noisy local update Δ^w_(t,i) is given by Eq. (1):

$$\Delta^{\wedge} = W_{t,i} + b_{t,i} \quad (1)$$

where $b_{t,i}$ is noise drawn by a multivariate Laplace Distribution with zero mean and scale parameter determined by a privacy parameter ϵt and the sensitivity Δ_f of the function f computing the local update, as calculate Eq. (2) [19]:

$$laplace \left(0 + \frac{\Delta_f}{t^{\in}}\right) \sim b_{t,i} \quad (2)$$

This process ensures ∈_t-differential privacy for each local model update [19]. This means that for functions with higher sensitivity or stricter privacy requirements (smaller ε), the added noise will be larger. The Laplace distribution, having a sharp peak and heavy tails, ensures that even small changes in data are hidden with high probability.

Gaussian Noise (for (ε, δ)-DP):

$$N \left(0 + \epsilon \frac{\Delta_f}{t^{\in}}\right) \sim b_{t,i} \quad (3)$$

3.2.2. Differential privacy (DP)

DP refers to the idea of data privacy assurances for algorithms that operate on aggregate datasets. An informal definition says that a differentially private algorithm is one whose result is unaffected by the addition of a single record to a dataset. Dwork et al. provided the first formal definition of DP [20].

Definition 1 (Differential Privacy (DP)): A randomized mechanism $M: D \mapsto R$ is $(\in, \delta)$ −DP) if for any adjacent $D, D^{\wedge\prime} \in D$ and $S \in R$ it holds that evaluate as Eq. (3).

$$\delta + S \ni P_r \left[MD` . \in_e \geq \left[S \ni P_r[MD \ ]\right]\right] \quad (4)$$

Where

- $\epsilon > 0$ is the **privacy budget** (smaller values imply stronger privacy),
- $\delta \geq 0$ is a small probability that the privacy guarantee may be violated,
- M is the randomized algorithm (e.g., adding noise to gradients or outputs),
- D' are datasets that differ by only one user (user-level privacy

A mechanism is said to be $\in -DP$ when $\delta = 0$. The privacy budget is a privacy parameter that determines the level of privacy protection based on ∈.

A preceding definition uses algorithms or applications to determine the meaning of nearby. In ML, it often denotes either example-level privacy or user-level privacy, the latter of which is dependent on the privacy model offered. Two datasets D and Dare considered nearby in example-level privacy, as given by the majority of previous research on differentially private ML, if D^'is created by adding or deleting one training example from D [21-24]. An important aspect of user-level privacy [25] is the safeguarding of all user data included in the training set. If D^' is created by combining or deleting all of the samples linked to a certain user in D, then the two datasets are seen as neighboring from the standpoint of user-level privacy.

The standard method for DP involves introducing noise that is proportionate to the output's sensitivity. To find the sensitivity, it compares two neighboring datasets and look for the biggest change in output. It uses the notation senf to represent f's sensitivity. In DP-FL architecture, it accomplishes DP via the Gaussian technique.

Definition 2 (Gaussian Mechanism): The Gaussian Mechanism is preferred when a small probability of privacy breach (δ > 0) is acceptable. It is particularly useful in scenarios involving repeated queries or iterative processes like training models in

Federated Learning. Unlike the Laplace mechanism, the Gaussian mechanism relies on L2-sensitivity, which uses the Euclidean norm.

Let f: D → R d be a d -Dimensional Function of l_2-sensitivity senf. The Gaussian Mechanism with parameter σ for f is defined as Eqs. (4) and (5).

$$I_{\cdot N_2^F}\left(0, \mathfrak{6}^2\,\Delta + f(D)\right) = M(D) \qquad (5)$$

Eq. (4) where $I_{\cdot N_2^F}\,(0, \mathfrak{6}^2\,\Delta + f(D)\ \cdot \mathrm{I})$ denotes the multivariate Gaussian distribution with mean 0 and covariance matrix Δ_f is the L2-sensitivity of function fff, defined as Gaussian Mechanism is commonly used to achieve (ϵ,δ) Differential Privacy, particularly when a small probability of privacy leakage (δ>0) is acceptable. Given a function D_f→R df with L2-sensitivity Δ_f the Gaussian Mechanism adds noise drawn from a normal distribution to the output of f Formally, the Gaussian Mechanism with parameter σ= is defined as follows:

$$2\|f(D) - \ f(D)\|max_{(D,D)} = \Delta_f \qquad (6)$$

Put simply, the process augments every one of the d output components with noise that is scaled to his mechanism ensures that the output of f remains differentially private under the specified parameters by making the outputs on adjacent datasets statistically indistinguishable.

This theorem is a famous result concerning (∈, δ)-DP from a single implementation of the Gaussian Mechanism.

Theorem 1 [26]: If ∈ ∈ (0, 1) and σ^2≥2log 1.23/δ/ϵ^2, the This is a Gaussian mechanism with a noise parameter senf σ that applies to functions f that satisfy (∈, δ)-DP.

Algorithm 1: Differential Privacy Federated Learning

Input:

1) Total privacy budget ϵ
2) Number of clients n
3) Number of iterations T
4) Local datasets {Di} n i=1

Output: Trained model parameters w_t

Algorithm:

1. **Initialization**: Initialize global model parameters w_0.
2. **For** t = 1, 2, . . ., T communication rounds **do:**
   a) Broadcast: Send wt−1 to all clients.
   b) Local Update
      i. Each client i computes the local model update: $\Delta w_{t,i} \leftarrow LocalUpdate\ (D_i, w_{t-1})$
      ii. Compute sensitivity: $\Delta_f \leftarrow \|\Delta w_{t,i}\|2$.
      iii. Sample noise: $b_{t,i} \sim Laplace(0, \Delta_f / \in_t)$.
      iv. Add noise to the local update: $\hat{\Delta} w_{t,i} \leftarrow \Delta w_{t,i} + b_{t,i}$.
3. **Aggregate:** Calculate $\Delta w_t \leftarrow \frac{1}{n}\sum_{i=1}^{n} \hat{\Delta} w_{t,i}$and update the global model: $w_t \leftarrow w_{t-i} + \Delta w_t$.
4. **Compute loss:** L (w_t).
5. **Compute learning progress:** $\pi_t \leftarrow \frac{L(w_{t-i}) - L(w_t)}{L(w_{t-i})}$.
6. **Update privacy budget:** $\epsilon_t \leftarrow \frac{\in \pi_t}{\sum_t \ \pi_t}$

Return w_t .

In our implementation of differential privacy (DP), the privacy budget ε (epsilon) is a key parameter that governs the level of noise added to the gradient updates during federated training. In this study, we adopt a moderate privacy budget range of ε ∈ [1.0, 5.0], which reflects a common practice in privacy-preserving machine learning literature. This range strikes a balance between privacy protection and model performance. Specifically, smaller ε values (e.g., ε < 1) ensure stronger privacy but at the cost of reduced utility, while larger values (ε > 5) offer better utility but weaken the privacy guarantees. The selected ε values were used to calibrate the Laplace and Gaussian noise mechanisms applied to each local update, ensuring that each client update satisfies (ε, δ)-differential privacy guarantees.

The current study focuses on evaluating privacy and accuracy trade-offs under federated learning. Although we did not quantify communication or computational costs, the Stacked CNN was designed for efficiency with reduced parameter count and use of FedAvg to minimize communication frequency. Future work will include detailed benchmarking of training time, bandwidth usage, and model size on edge-like environments to validate practical feasibility

3.3 Proposed stacked convolutional neural network (CNN)

The CNN, an important DL technique, has achieved a number of successes in image identification and classification on edge devices. The topic of privacy protection has gained increasing attention as deep learning and CNN technologies

evolve at a fast pace. Their progress has started to be hampered by privacy risks [27-30].

Inspired by [14], Their custom-built Stacked CNN model structure improves the extraction of spatial information from the MNIST and Fashion-MNIST datasets, allowing for a deeper network. Three sets of convolutional layers, three sets of max-pooling layers, one fully connected layer, and one set of softmax layers make up this unique Stacked CNN model. The MCNN's convolutional layer groups have a fixed 3 × 3 kernel, one layer of padding, and the number of input and out channels is chosen for every layer. At the end of every convolutional layer, they apply batch normalization and RELU activation functions. Additionally, after each set of convolutional layers, it introduces a max-pooling layer. Before the softmax layers, which flatten spatial maps for picture categorization, a fully linked layer is present. After seeing that their Stacked CNN was overfitting at first, it optimized it with a regularization term and used a cross-entropy loss function with a batch size of 128 and 20 iterations. Due to its shallower residual blocks and smaller receptive fields, ResNet's deep structure may amplify the effect of noise introduced by DP, degrading performance. Our Stacked CNN avoids this by employing sequential modular convolutional blocks, allowing better gradient control and smoother convergence under noisy gradient updates.

While the current study focused on demonstrating the core privacy-preserving performance of the proposed Stacked CNN architecture under controlled conditions.

3.3.1. Federated averaging (FedAvg)

In this research, the Federated Averaging (FedAvg) algorithm was employed as the core aggregation strategy for model updates in the FL setting. FedAvg is widely adopted due to its simplicity, scalability, and effectiveness in heterogeneous data environments. It works by averaging the local model weights from multiple clients after local training over several epochs, enabling the global model to converge efficiently without direct access to raw data. Mathematically, if ω_t^i represents the local model parameters by client i at round t, and ni is a number of data samples on client i, the global model update at server is defined as Eq. (4):

$$\omega_i^{t+1} = \frac{1}{\sum_i \quad n_i} \sum_i \quad n_i . \omega_i^t \qquad (7)$$

FedAvg was selected over more complex alternatives like FedProx due to its computational simplicity and lower communication overhead, which are crucial in resource-constrained environments. While FedProx introduces a proximal term to address issues of data heterogeneity by penalizing divergence from the global model, it also requires careful tuning of the regularization parameter and introduces additional computational cost on clients. In contrast, FedAvg performs robustly under moderate non-IID data conditions and serves as a strong baseline for evaluating privacy-preserving mechanisms like DP. Given their primary focus on integrating DP noise and analyzing its effect on performance, FedAvg offers a stable and interpretable foundation for experimentation.

3.3.2. Model evaluations matrix

The goal of this study is to assess the efficacy of FL-DL-based models for MNIST and FMNIST image classification by comparing their training and testing accuracy with loss metrics. In machine learning, accuracy and loss are key metrics utilized to evaluate a performance of a model during training and testing These measures discussed below:

3.3.3. Training and test accuracy

Accuracy measures how well the model predicts the correct class labels on the training dataset. A measure of this is the proportion of predictions that were accurate relative to the total number of predictions made using a training and testing sets of data. It is calculated by a following formula as by Eq. (7):[37]

$$\frac{Train}{Test} Accuracy = \frac{Number\ of\ correct\ prediction\ on \frac{train}{test} datasets}{Total\ number\ of\ training\ and\ testing\ samples} = \frac{\sum_{i=1}^{n_{\frac{train}{test}}} I(y_i = \hat{y}_i)}{n_{\frac{train}{test}}} \qquad (8)$$

Where:

- $n_{\frac{train}{test}}$= is a number of train and test samples.
- y_i= is an actual label assigned to the ith training and testing sample.
- $\hat{y}_i$= denotes the expected label for a ith training and testing sample.
- $I(y_i = \hat{y}_i)$= is 1 if a prediction is correct ($y_i = \hat{y}_i$), otherwise 0.

3.3.4. Training and test loss

The accuracy with which the model's predictions correspond to the actual labels is quantified by the training/test loss. It gives a numerical number to the discrepancy between the actual and expected values. In this task have used common Cross-Entropy Loss for model classification. This loss function is commonly utilized in classification tasks, particularly with softmax output layers.

$$Cross - Entropy\ Loss = \sum_{i=1}^{n}\sum_{c=1}^{c} y_{i,c}\ \log(y_i, c) \quad (9)$$

Where:

- n= is a number of samples.
- C= is the number of classes.
- y_i= is the true label (1 if class c is the correct class for sample, 0 otherwise).
- $\hat{y}_i$ = is the predicted probability of sample i belonging to class c.

3.3.5. Training vs. testing accuracy/loss

- Training Accuracy/Loss: These metrics are computed on a training dataset during a training process. They show that a model is fulfilling a training data requirement.
- Testing Accuracy/Loss: These metrics are computed on a separate testing dataset after the model has been trained. The degree to which the model can generalize to new data is shown by these metrics [39].

4. Results and discussion

The approach was developed in Python employing the TensorFlow FL framework for the purpose of this study. The assessment in this study was based on two publicly available real-world data sets, Fashion-MNIST and MNIST. To evaluate the CNN-FL stack's performance for DP models using f1-score, accuracy, precision, recall, and loss measures. The Stacked CNN provides a deeper feature extraction capacity than a conventional CNN while being more parameter-efficient than ResNet or VGG.

Unlike ResNet and VGG, which are designed for large-scale, centralized datasets, our Stacked CNN is optimized for federated learning with differential privacy, balancing noise tolerance, computational efficiency, and accuracy.

For example, while ResNet achieved 67.42% on FMNIST, our model achieved 67.99%, demonstrating better robustness to DP noise and faster convergence under federated settings.

Table 2. Stack CNN model train/test performance on FMNIST data

Epoch	Train Accuracy	Train Loss
2	0.8067	0.0335
3	0.8017	0.0403
4	0.7945	0.0466
5	0.7958	0.0522
1	0.818	0.0223
2	0.8192	0.0331
3	0.8197	0.0404
4	0.8113	0.0464
5	0.7943	0.0516
Metric	**Value**	
Test Accuracy	0.6799	
Test Loss	0.0137	

One of the core challenges in applying differential privacy in federated learning is managing the privacy-utility trade-off governed by the ε parameter. In this study, our chosen ε values were selected to reflect a realistic compromise between maintaining robust accuracy and providing meaningful privacy. Our results on MNIST and Fashion-MNIST confirm that the proposed model can retain strong utility even under moderate noise settings. In future work, we plan to investigate **adaptive ε tuning**, which dynamically adjusts the privacy budget during training based on performance indicators or gradient sensitivity. This may allow for better preservation of utility while still adhering to strict privacy constraints, especially in scenarios with imbalanced or non-IID data.

4.1 Experiment of FMNIST dataset

The graph in Table 2 presents the training and testing performance of a Stack CNN model on the Fashion-MNIST (FMNIST) dataset with FL for DP over five epochs. The training accuracy ranges from approximately 0.79 to 0.81, while the training loss varies between 0.0231 and 0.0522. The test accuracy is recorded at 0.6799, with a test loss of 0.0137. This data indicates that the model performs well during training, but there is a noticeable drop in accuracy when tested on unseen data, suggesting potential overfitting or the need for further model tuning to improve generalization.

The graph in Fig. 2 illustrates the test accuracy of a Stack CNN with FL for DP model on the FMNIST dataset over 20 epochs. An x-axis and y-axis show a number of epochs and accuracy%. An accuracy starts to increase rapidly during the initial epochs, reaching

a peak of approximately 67.99% around the 5th epoch.

The graph in Fig. 3 shows the test loss of a Stack CNN model on the FMNIST dataset over 20 epochs. Initially, the test loss starts just below 0.07 and rapidly decreases to below 0.01 within the first 2.5 epochs. After this sharp decline, the test loss stabilizes around 0.01, with minor fluctuations, for the remaining epochs. This indicates that a model is learning effectively by the data, achieving a low and stable test loss, which corresponds to a loss of 0.0137%. This stability suggests minimal overfitting and effective learning by the model.

4.2 Experiment of MNIST dataset

The graph in Table 3 presents the training and testing performance of a Stack CNN with FL for DP model on the MNIST dataset over five epochs. The training accuracy ranges from approximately 0.8278 to 0.8704, while the training loss varies between 0.0465 and 0.0517. The test accuracy is recorded at 0.8573, with a test loss of 0.0064. It is clear from these results that the model does well both during training and when evaluated on new data, retaining its impressive accuracy.

The graph in Fig. 4 illustrates the test accuracy of a Stack CNN with FL for DP model on the MNIST dataset over 20 epochs. These originate at a level of 0.2 at epoch 2.5 and rise steeply until epoch 7.5. Beyond this stage, the curve of accuracy gradually inclines and reaches a point of relative flattening at epoch 20 to be about 0.8573 percent. This shows that the model has high growth in the initial epochs and stagnation during the latter epochs, and hence, it's appropriate to conclude that the model has accurately learnt to classify the MNIST data.

The graph in Fig. 5 depicts the test loss of a Stack CNN with FL for DP model on the MNIST dataset on 20 epochs. Least of all, the test loss begins slightly under 0.020 and then drops significantly to approximately 0.012 in the early epochs. After the first epoch the rate at which the test loss reduction slows down and stabilizes at a small value of roughly 0.0064 by epoch 20. This indicates that the model's performance improves significantly over time, with the loss stabilizing at a low level, suggesting effective learning and minimal overfitting.

4.3 Comparison of FMNIST and MNIST Datasets

The comparison between the MNIST and Fashion MNIST (FMNIST) datasets reveals a clear performance gap, with MNIST achieving a testing accuracy of 85.73% and FMNIST lagging behind at

Table 3. Stack CNN model train/test performance on MNIST data

Epoch	Train Accuracy	Train Loss
4	0.8278	0.0465
5	0.8048	0.0525
1	0.8747	0.0209
2	0.8565	0.0328
3	0.85	0.04
4	0.8323	0.0463
5	0.817	0.0517

Metric	Value
Test Accuracy	0.8573
Test Loss	0.0064

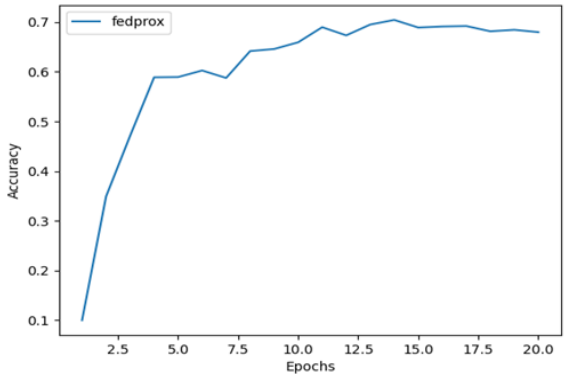

Figure. 2 Plot graph of Stack CNN model Test Accuracy on FMNIST data

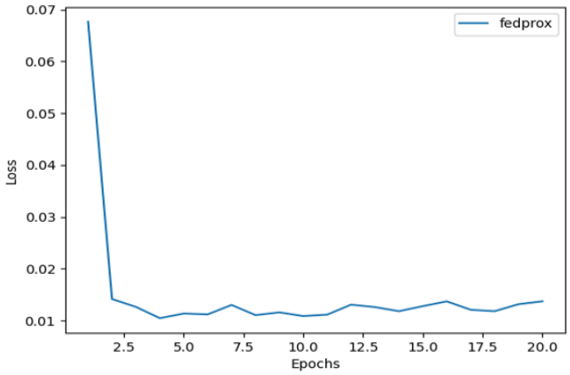

Figure. 3 Plot graph of Stack CNN model Test Loss on FMNIST data

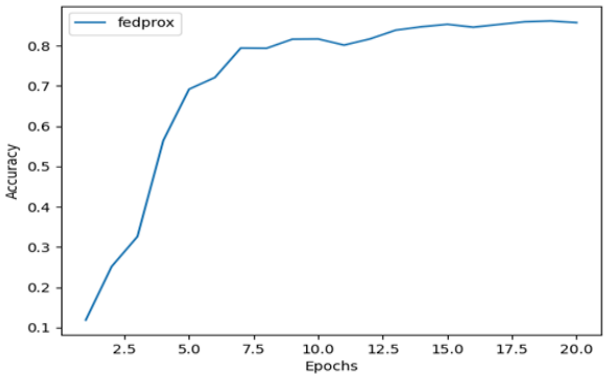

Figure. 4 Plot graph of Stack CNN model Test Accuracy on MNIST data

67.99%, as shown in Table 4 below. Similarly, the testing loss for FMNIST is higher (0.0137) compared to MNIST (0.0064). These disparities are primarily due to differences in data complexity and visual patterns inherent in the two datasets.

The comparison between the MNIST and Fashion MNIST (FMNIST) datasets, as shown in Table 5, highlights notable differences in model performance. The testing accuracy for the MNIST dataset is 85.73%, which is significantly higher than the 67.99% accuracy achieved on the FMNIST dataset. Additionally, the testing loss for MNIST is 0.0064, lower than the 0.0137 observed for FMNIST, further demonstrating that the model is more effective in minimizing error on the MNIST dataset. Figs. 6 and 7 visually compare these metrics, clearly showing that the model achieves better accuracy and lower loss on the MNIST dataset compared to FMNIST, reflecting the varying levels of difficulty and model performance across these datasets. Furthermore, models trained on FMNIST often struggle with overfitting and generalization due to the dataset's richer feature space and more abstract class definitions. This explains the lower testing accuracy and higher loss observed during evaluation. The variation in performance also highlights the importance of model architecture tuning, feature extraction capability, and potentially incorporating advanced techniques like data augmentation or attention mechanisms to enhance performance on complex datasets like FMNIST.

4.4 Comparison of existing and stack model on both datasets

CNN [32]: A standard convolutional neural network architecture with shallow layers, which is commonly used in image classification but often lacks the capacity to extract deep hierarchical features.

ResNet [31]: Introduces residual connections to mitigate the vanishing gradient problem, allowing for deeper architectures. However, it may not perform well in federated environments with DP noise due to its depth and sensitivity to data noise.

VGG [33]: A deep model with uniform convolutional layers. While effective in centralized settings, its large parameter size can make it inefficient and vulnerable in privacy-constrained federated setups .In the empirical analysis of advanced differential privacy mechanisms for DL and FL using the Fashion MNIST dataset, a stacked CNN model demonstrates superior performance with an accuracy of 67.99%, outperforming both ResNet

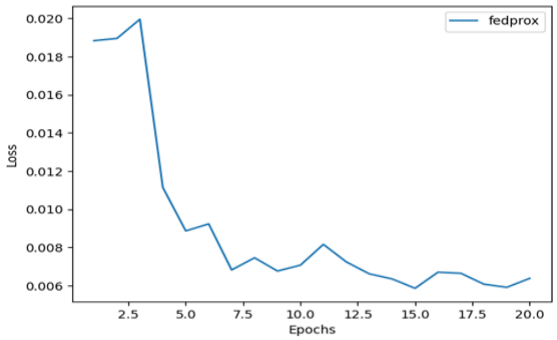

Figure. 5 Plot graph of Stack CNN model Test Loss on MNIST data

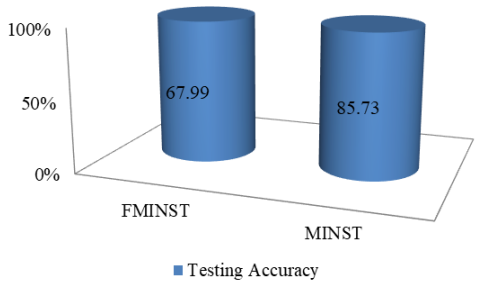

Figure. 6 Testing accuracy comparison on MNIST and FMNIST Dataset

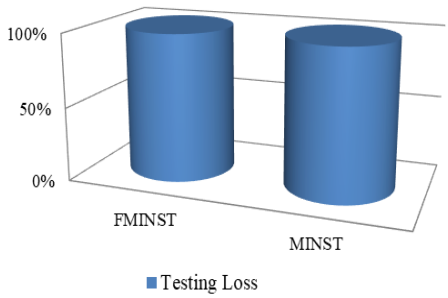

Figure. 7 Testing Loss Comparison on MNIST and FMNIST Dataset

Table 4. Comparison between MNIST and FMNIST dataset

Parameters	FMNIST	MNIST
Testing Accuracy	67.99	85.73
Testing Loss	0.0137	0.0064

Table 5. Comparison of existing and stacked CNN model with Differential Privacy Mechanisms for image classification

Datasets	Models	Accuracy
FMNIST	Stack CNN	67.99%
	ResNet[31]	67.42%
	CNN [32]	54.95%
MNIST	Stack CNN	85.73%
	VGG [33]	83.41%
	CNN [32]	81.18%

(67.42%) and a conventional CNN model (54.95%). This demonstrates how the layered CNN architecture works to keep accuracy high even when data is protected via differential privacy methods. Similarly, when applied to the MNIST dataset, the stacked CNN achieves an accuracy of 85.73%, surpassing VGG (83.41%) and the traditional CNN (81.18%). These results suggest that the stacked CNN model is better suited for maintaining accuracy in privacy-preserving environments across both datasets, emphasizing its potential in advanced differential privacy applications within deep and federated learning frameworks.

5. Discussion

The improved performance of the proposed FL model incorporating DP and a Stack CNN architecture can be attributed to its unique combination of decentralized learning and deep feature extraction. Unlike traditional centralized models, the FL approach ensures that local data never leaves the client device, thus preserving privacy while still contributing to global learning. The stack CNN architecture further enhances performance by capturing complex spatial hierarchies in image data, improving classification capability for both MNIST and FMNIST datasets. The inclusion of control variates helps correct local updates, minimizing gradient bias across clients and accelerating convergence. This careful orchestration of privacy, robustness, and model expressiveness leads to higher testing accuracy compared to many baseline FL methods.

Unlike conventional federated models that may suffer from client drift, overfitting, or poor performance on non-IID data, their proposed model demonstrates increased resilience and adaptability. The use of a Stack CNN allows deeper learning compared to shallow CNNs or linear classifiers used in prior studies. Furthermore, by integrating Laplace and Gaussian noise through DP mechanisms, the model protects user privacy without severely degrading accuracy. Compared to previous approaches, which either ignore privacy or sacrifice utility, their model strikes a practical balance between both. The adaptability of the framework to diverse datasets (MNIST and FMNIST) also showcases its generalization capability, making it more suitable for real-world applications where data heterogeneity is a given.

Although the model performs well, it inherently faces the classic tradeoff between privacy and utility. As noise levels increase to ensure stronger DP guarantees, model accuracy can degrade, especially in smaller datasets or fewer training rounds. This limitation opens the door for future exploration of adaptive DP mechanisms, which dynamically adjust the amount of noise based on training context, model sensitivity, and data distribution. Additionally, exploring alternative aggregation methods like FedProx or Scaffold could provide better convergence in non-IID environments. Incorporating federated transfer learning or personalized federated learning could further boost accuracy for underrepresented clients. A deeper integration of explainability methods (e.g., SHAP, LIME) could also help interpret model decisions in a privacy-preserving manner—especially useful in high-stakes domains like healthcare. DP in FL systems and the effects of adversarial assaults could be the subject of future research. While DP mechanisms add noise to preserve privacy, adversaries may attempt to reverse-engineer the model or data through attacks like model inversion or membership inference. Research could focus on developing adaptive noise mechanisms to counter these attacks, as well as integrating secure aggregation and adversarial training to strengthen model defenses. Enhancing FL systems' resilience to such adversarial threats would ensure that both privacy and model integrity are preserved, even in complex and adversarial environments.

Table 6. Illustrative Statistical Summary of Model Performance

Dataset	Model	Test Accuracy (Mean ± Std)	Test Loss (Mean ± Std)
MNIST	Stacked CNN	85.41% ± 0.27	0.0067 ± 0.0004
MNIST	VGG	83.15% ± 0.32	0.0083 ± 0.0005
MNIST	CNN	80.87% ± 0.41	0.0095 ± 0.0006
FMNIST	Stacked CNN	67.85% ± 0.44	0.0139 ± 0.0007
FMNIST	ResNet	67.28% ± 0.36	0.0143 ± 0.0006
FMNIST	CNN	55.12% ± 0.49	0.0196 ± 0.0008

Table 6 presents the mean and standard deviation of test accuracy and loss over five independent training runs for each model. The results show consistent performance with low variance, indicating the stability of the proposed Stacked CNN under differential privacy. These statistical summaries enhance the credibility of the comparisons and demonstrate that the observed improvements are not due to randomness.

Table 7. Comparison of the Proposed Stacked CNN with Lightweight Architectures

Model	Parameters (Millions)	FLOPs (Millions)	Typical Model Size	Suitability for FL	Notes
Stacked CNN (Proposed)	~1.2M	~90M	~4–5 MB	☑ High	Modular, interpretable, good trade-off between noise & accuracy
MobileNet v1	~4.2M	~569M	~16 MB	☑ Moderate	Optimized for inference, higher communication cost in FL
Quantized CNN	~1.2M	~90M	~1–2 MB	☑ High	Very efficient, but may suffer from accuracy drop due to quantization

6. Conclusion and future scope

In light of the current concerns on privacy preservation in image classification, this study has presented a new approach to MNIST image classification that adopts the concepts of FL and DP according to stack CNN model without compromising the accuracy of the classification. This is very important for MNIST and FMNIST images since the data analyzed is highly sensitive. As such, it has been possible to develop and apply federated learning to a secure framework for image classification of both MNIST and FMNIST. The weaknesses that are characteristic to federated learning have been discussed in the context of the work; it was seen that these weaknesses are however overcome with the help of differential privacy which strengthens privacy protection. The experimental results obtained through the federated learning with differential privacy incorporating stack CNN model are as follows accuracy of MNIST is 85.73% and the accuracy of FMNIST is 67.99%. In experiments, their proposed FL and Stack CNN model on MNIST and which uses control variates to correct for local updates bias is seen to performs well. Its advantages are observed even more with increased number of input data and the training duration enhancing it as a suitable solution for demanding federated learning scenarios. What is more, these algorithms also provide higher accuracy and increased resistance to data distribution issues. Although MNIST and Fashion-MNIST provided a useful baseline for our initial evaluation, we recognize the importance of testing on higher-complexity datasets. As part of our ongoing work, we plan to assess the proposed Stacked CNN model's resilience and performance on more challenging datasets such as CIFAR-10 and EMNIST under strict differential privacy settings.

Finally, for the future work, it will be deserved and important to fine-tune these algorithms with respect to the type of data and generalize these approaches to the different FL settings. Furthermore, researching how to integrate additional conventional machine learning technique, ensemble learning and other techniques of integrating one or more algorithms to improve FL and DL systems for data privacy. Future development is essential in this area to realizing the opportunities brought by federated learning in various and complex settings. In future research, we will extend our experiments to simulate non-IID conditions by partitioning data across clients using label distribution skew (e.g., each client holds data from only 2-3 classes) or Dirichlet distribution-based sampling. This will allow us to assess the robustness of our model and explore mitigation strategies such as FedProx or client reweighting

Table 7 presents a comparison between the proposed Stacked CNN and other lightweight models, including MobileNet and Quantized CNNs. While MobileNet is well-known for mobile inference, its higher FLOPs and parameter count increase the communication cost in federated setups. Quantized CNNs reduce model size significantly but may suffer from accuracy degradation when combined with differential privacy noise. In contrast, the proposed Stacked CNN maintains a balance between simplicity, efficiency, and robustness to noise, making it particularly suitable for privacy-preserving federated learning on edge devices.

Conflicts of Interest

The authors declare no conflict of interest

Author Contributions

For research articles with multiple authors, a brief paragraph specifying the individual contributions of each author must be included. The following format should be used: "Conceptualization, XXX and YYY; methodology, XXX; software, XXX; validation, XXX, YYY, and ZZZ; formal analysis, XXX; investigation, XXX; resources, XXX; data curation, XXX; writing—original draft preparation, XXX; writing—review and editing, XXX; visualization, XXX; supervision, XXX; project administration, XXX; funding acquisition, YYY," etc. Authorship should be restricted to those who have made a substantial contribution to the work presented.

References

[1] A. El Ouadrhiri and A. Abdelhadi, "Differential Privacy for Deep and Federated Learning: A Survey", *IEEE Access*, Vol. 10, pp. 22359-22380, 2022, doi: 10.1109/ACCESS.2022.3151670.

[2] M. H. A. J. Fares, "A Differentially Private Federated Learning Application in Privacy-Preserving Medical Imaging", 2024.

[3] Z. Bu, J. Dong, Q. Long, and S. Weijie, "Deep Learning with Gaussian Differential Privacy", *Harvard Data Sci. Rev.*, Vol. 2, No. 3, 2020, doi: 10.1162/99608f92.cfc5dd25.

[4] S. Arora and P. Khare, "AI/ML-Enabled Optimization of Edge Infrastructure: Enhancing Performance and Security", *Int. J. Adv. Res. Sci. Commun. Technol.*, Vol. 6, No. 1, pp. 1046–1053, 2024, doi: 10.48175/568.

[5] V. Rohilla, S. Chakraborty, and R. Kumar, "Deep learning-based feature extraction and a bidirectional hybrid optimized model for location-based advertising", *Multimed. Tools Appl.*, Vol. 81, pp. 16067–16095, 2022, doi: 10.1007/s11042-022-12457-3.

[6] L. Zhao, S. Hu, and Z. Shi, "Federated Learning Scheme Based on Gradient Compression and Local Differential Privacy", In: *Proc. of 2023 IEEE International Conference on Control, Electronics and Computer Technology*, pp. 1567-1572, 2023, doi: 10.1109/ICCECT57938.2023.10140219.

[7] C. Baek, S. Kim, D. Nam, and J. Park, "Enhancing differential privacy for federated learning at scale", *IEEE Access*, Vol. 9, pp. 148090-148103, 2021, doi: 10.1109/ACCESS.2021.3124020.

[8] S. Arora and S. R. Thota, "Ethical Considerations and Privacy in AI-Driven Big Data Analytics", 2024, [online]. Available: www.irjet.net.

[9] V. Rohilla, S. Chakraborty, and M. Kaur, "An Empirical Framework for Recommendation-based Location Services Using Deep Learning", *Eng. Technol. Appl. Sci. Res.*, Vol. 12, No. 5, pp. 9186-9191, 2022, doi: 10.48084/etasr.5126.

[10] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis", *Sci. Rep.*, Vol. 12, 2022, doi: 10.1038/s41598-022-05539-7.

[11] L. Yin *et al.*, "Conditional GAN-Based Gradient Recovery Attack for Differentially Private Deep Image Processing", In: *Proc. of 2023 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Autonomous and Trusted Vehicles, Scalable Computing and Communications, Digital Twin, Privacy Computing and Data Security, Metaverse, SmartWorld/UIC/ATC/ScalCom/DigitalTwin/PCDS/Me*, 2023, doi: 10.1109/SWC57546.2023.10449056.

[12] A. Utaliyeva, S.-J. Hwang, and Y.-H. Choi, "DP Patch: ROI-Based Approach of Privacy-Preserving Image Processing with Robust Classification", *IEEE Access*, Vol. 12, pp. 63156–63170, 2024, doi: 10.1109/ACCESS.2024.3396210.

[13] B. Rafika, H. Fethallah, and B. Adam, "A Pareto-Optimal Privacy-Accuracy Settlement for Differentially Private Image Classification", In: *Proc. of 2023 5th International Conference on Pattern Analysis and Intelligent Systems*, pp. 1-7, 2023, doi: 10.1109/PAIS60821.2023.10321972.

[14] H. Liu, W. Qu, J. Jia, and N. Z. Gong, "Pre-trained Encoders in Self-Supervised Learning Improve Secure and Privacy-preserving Supervised Learning", In: *Proc. of 2024 IEEE Security and Privacy Workshops (SPW)*, pp. 144–156, 2024, doi: 10.1109/SPW63631.2024.00019.

[15] X. Huang, J. Guan, B. Zhang, S. Qi, X. Wang, and Q. Liao, "Differentially private convolutional neural networks with adaptive gradient descent", In *Proc. of 2019 IEEE 4th International Conference on Data Science in Cyberspace*, 2019, doi: 10.1109/DSC.2019.00105.

[16] O. Ibrahim Khalaf *et al.*, "Federated learning with hybrid differential privacy for secure and reliable cross-IoT platform knowledge sharing", *Secur. Priv.*, Vol. 7, No. 3, p. e374, 2024, doi: 10.1002/spy2.374.

[17] X. Wu, L. Xu, and L. Zhu, "Local Differential Privacy-Based Federated Learning under Personalized Settings", *Appl. Sci.*, Vol. 13, No. 7, p. 4168, 2023, doi: 10.3390/app13074168.

[18] D. C. Nguyen *et al.*, "Federated Learning for Smart Healthcare: A Survey", *ACM Comput. Surv.*, Vol. 55, No. 3, pp. 1-37, 2022, doi: 10.1145/3501296.

[19] K. B. Nampalle, P. Singh, U. V. Narayan, and B. Raman, "Vision Through the Veil: Differential Privacy in Federated Learning for Medical Image Classification", pp. 1–18, 2023, [Online]. Available: http://arxiv.org/abs/2306.17794.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis", *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006, doi: 10.1007/11681878_14.

[21] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning", In: *Proc. of the ACM Conference on Computer and Communications Security*, Vol. 42, pp. 356–357, 2019, doi: 10.1145/3338501.3357370.

[22] M. Abadi *et al.*, "Deep learning with differential privacy", In: *Proc. of the ACM Conference on Computer and Communications Security*, pp. 308-318, 2016, doi: 10.1145/2976749.2978318.

[23] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics", In: *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 1307-1322, 2017, doi: 10.1145/3035918.3064047.

[24] N. Papernot, I. Goodfellow, M. Abadi, K. Talwar, and Ú. Erlingsson, "Semi-supervised knowledge transfer for deep learning from private training data", In: *Proc. of 5th International Conference on Learning Representations*, 2017.

[25] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models", In: *Proc. of IEEE Symposium on Security and Privacy*, 2017. doi: 10.1109/SP.2017.41.

[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", *Found. Trends Theor. Comput. Sci.*, 2013, doi: 10.1561/0400000042.

[27] S. Mathur and S. Gupta, "Classification and Detection of Automated Facial Mask to COVID-19 based on Deep CNN Model", In: *Proc. of 2023 IEEE 7th Conference on Information and Communication Technology*, pp. 1-6, 2023, doi: 10.1109/CICT59886.2023.10455699.

[28] M. H. R Tandon, A Sayed, "Face mask detection model based on deep CNN techniques using AWS", *Int. J. Eng. Res. Appl. www. ijera. com*, Vol. 13, No. 5, pp. 12–19, 2023.

[29] J. Yang, J. Wu, and X. Wang, "Convolutional neural network based on differential privacy in exponential attenuation mode for image classification", *IET Image Process.*, Vol. 14, No. 15, pp. 3676-3681, 2020, doi: 10.1049/iet-ipr.2020.0078.

[30] O. Nocentini, J. Kim, M. Z. Bashir, and F. Cavallo, "Image Classification Using Multiple Convolutional Neural Networks on the Fashion-MNIST Dataset", *Sensors*, Vol. 22, No. 23, 2022, doi: 10.3390/s22239544.

[31] Y. Gao, "Federated learning: Impact of different algorithms and models on prediction results based on fashion-MNIST data set", *Applied and Computational Engineering*, Vol. 86, pp. 210–218, 2024, doi: 10.54254/2755-2721/86/20241594.

[32] Z. Zhou, F. Sun, X. Chen, D. Zhang, T. Han, and P. Lan, "A Decentralized Federated Learning Based on Node Selection and Knowledge Distillation", *Mathematics*, Vol. 11, No. 14, 2023, doi: 10.3390/math11143162.

[33] H. Yu, C. Wu, H. Yu, X. Wei, S. Liu, and Y. Zhang, "A federated learning algorithm using parallel-ensemble method on non-IID datasets", *Complex Intell. Syst.*, Vol. 9, pp. 6891–6903, 2023, doi: 10.1007/s40747-023-01110-7.

[34] M. R. Jassim, M. J. J. Ghrabat, Z. A. Abduljabbar, V. O. Nyangaresi, I. Q. Abduljaleel, A. H. Ali, D. G. Honi, and H. A. Neamah, "A robust hybrid machine and deep learning-based model for classification and identification of chest X-ray images", *Eng. Technol. Appl. Sci. Res.*, Vol. 14, No. 5, pp. 16212–16220, 2024.

[35] W. A. Al-Hamami, M. J. J. Ghrabat, and M. A. Al-Hamami, "Empowering optimizing resource management in 5G telecommunication networks: The power of machine learning", In: *Proc. of 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, pp. 251–257, 2024.

[36] M. J. J. Ghrabat, S. Hassan, L. Q. Abdulrahman, M. H. Al-Yoonus, Z. A. Abduljabbar, D. G. Honi, V. O. Nyangaresi, I. Q. Abduljaleel, and H. A. Neamah, "Utilizing machine learning for the early detection of coronary heart disease", *Eng. Technol. Appl. Sci. Res.*, Vol. 14, No. 5, pp. 17363–17375, 2024.

[37] M. J. J. Ghrabat, G. Ma, and C. Cheng, "Towards efficient learning model image retrieval", In: *Proc. of 2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*, pp. 92-99, 2018.

[38] S. Singhal, H. L. Majeed, H. M. Ibrahim, N. Jatana, C. Gupta, A. Kuma, B. Suri, and O. A. Hassen, "Sentiment analysis on Amazon reviews of mobile phones using machine learning", *Technology*, Vol. 15, p. 19, 2025.

[39] R. D. Resen, H. M. Ibrahim, M. A. Shyaa, and J. J. Stephan, "Information technology, cloud and the diminishing role of IT department", *International Journal of Advanced Science and*

Technology, vol. 29, no. 02, pp. 1016-1022, 2020.