

Original Research

Received 10 June 2026

Revised 20 July 2026

Accepted 02 Aug 2026

Natural Resources for Human
Health 2026, Vol. 6 (15s): 1280–1296

KEYWORDS:

Federated Learning, Data
Heterogeneity, Data Skew Index,
Differential Privacy, Adaptive
Differential Privacy, Byzantine
Robustness

Privacy-Preserving Stroke Prediction under Non-IID Federated Learning: Adaptive Differential Privacy and Data-Skew-Aware Aggregation

M. Srikanth Sagar¹, L Sridhara Rao²

¹Research Scholar, School of Engineering, Anurag University.

²School of Engineering, Anurag University.

ABSTRACT: Classification of stroke using federated healthcare data poses challenges because of privacy issues and variation in local distribution of data. FL offers collaboration of models training without the exchange of raw patient data but non-IID client data can lead to variation in sample size and class distribution. DP offers an additional privacy measure, but comes with the drawback of utility-privacy trade-off. In this research, a Data Skew-aware Adaptive Differential Privacy and Aggregation scheme is proposed for federated stroke classification. A composite Data Skew Index (DSI) that takes into account both the quantity skew and label-distribution skew is employed to derive the Gaussian DP noise and influence factor in the process of aggregation. Rather than discarding the clients with deviating updates, our approach uses soft influence adjustment, keeping all clients intact. Experiments are performed using the stroke classification data set with 12 non-IID clients and Gaussian Byzantine perturbation. Comparison is made between the proposed scheme and FedAvg, Krum, Bulyan, and Adaptive Differential Privacy through ablation study, attack sensitivity analysis, five seed study, and privacy-utility analysis. With five different seeds, the proposed scheme gets omniscient error rate

1. INTRODUCTION

The Federated Learning framework allows several clients to train a model collaboratively without the need to share their data. The clients train a local model and send the updates to the server for aggregation. One such framework is FedAvg [1].

Statistical heterogeneity is an important problem in FL. The training dataset size and class distribution may be heterogeneous across clients, thus leading to the lack of independence and identical distribution (non-IID). Such heterogeneity may have an effect on the model's performance in the training process, optimization, and global model performance [2, 3]. In our paper, we investigate two forms of heterogeneity: the quantity skew and label skew.

Differential Privacy (DP) ensures the protection of information from leaking while updating the model with scaled noise [4, 5]. But the higher the noise, the lower will be the utility of the model causing privacy-utility trade-off. Also, an equal amount of noise for all the clients is unjustified because their distribution of data is not similar.

Aggregation issue can also be associated with the aforementioned one. While FedAvg takes into account only the sample size of each client [1], Byzantine resistant methods such as Krum and Bulyan consider the weight of clients depending on the update relationships to mitigate the influence of any potentially malicious updates [6, 7]. However, a large difference in the update can also be due to the non-IID data.

In order to address the problems mentioned above, a joint Data Skew Index (DSI), which includes both data quantity skew and data label skew for each client, is introduced. The DSI will be used in adaptive Gaussian DP noise and aggregation weights.

The resulting framework follows:

Quantity Skew + Label Skew $\rightarrow$ DSI $\rightarrow$ ADP $\rightarrow$ Noise $\rightarrow$ Aggregation Awareness

Byzantine detection or client elimination is not part of the proposed model; Byzantine attacks are used rather as an adversarial test criterion.

The main contributions are:

1. The composite of Data Skew Index (DSI) that comprises the quantity skew and label skew.
2. Adaptive Differential Privacy Approach for Choosing the Gaussian Noise Using DSI for Each Client.
3. An aggregation method based on DSI that adjusts client influence according to data skewness without disregarding diverse clients.
4. Testing under controlled conditions on 12 clients, Gaussian Byzantine perturbation, baseline comparison, ablation testing, attack strength sensitivity, seed sensitivity, and analysis of privacy-utility trade-offs.

The remainder of this paper discusses the methodology for the related work, experiment evaluation, results, limitations, future work, and conclusions.

2. RELATED WORK

2.1 Federated Learning Heterogeneity

The concepts of statistical heterogeneity, differential privacy (DP), and Byzantine resilience have all been thoroughly investigated in FL. There could be variations in the clients depending on the data amount and the distribution of labels in their datasets. This is why there arise situations where there is no IID distribution of data [2, 3].

2.2 Differential Privacy and Byzantine Robustness

The concept of Differential Privacy has gained considerable popularity in the domain of preserving privacy of information which is stored in the form of FL model updates. It has recently been tried to optimize the trade-off between privacy and utility using noise injection techniques [5, 8]. Byzantine resilient aggregation methods are intended for tackling any manipulation done in the FL updates by the clients. Krum and Bulyan aggregating methods are great examples of reducing the influence of malicious updates [6, 7]. However, deviation of updates can also occur due to valid non-IID data and privacy interference.

2.3 Research Gap and Positioning

Quantity skew and label distribution skew have been studied as non-IID examples in previous works, and DP and Byzantine aggregation works have separately considered privacy and security. In this work, quantity skew and label skew are combined together as the Client-Level Data Skew Index (DSI) to denote both Client Level DP noise and aggregation effect. Neither Krum nor Bulyan has directly dealt with the detection and tolerance of Byzantine clients. In the proposed system, all clients are alive, and aggregation effect is slightly damped depending on their DSI values. Byzantine perturbation is taken into account only as an adversarial case in this work for validation.

Table 1. Overview of Relevant Studies and Knowledge Gap

Study	Focus	Gap Relevant to This Work
Liu et al. (2021) [2]	Non-IID FL	No joint quantity-label skew measure for adaptive decisions.
Suresh et al. (2020) [3]	Non-IID Learning	Joint use of heterogeneity types for DP and aggregation is limited.
Kumar et al. (2021) [4]	FL Privacy	No DSI-based adaptive privacy or aggregation.
He et al. (2020) [5]	DP in FL	No data-skew-based noise adaptation.

Study	Focus	Gap Relevant to This Work
Zhang et al. (2020) [6]	Adaptive DP	Does not jointly use skew for aggregation.
Pillutla et al. (2022) [7]	Krum	Does not model legitimate data heterogeneity.
Blanchard et al. (2017) [8]	Bulyan	Does not explicitly model client data skew.
Present Work	DSI-aware-DP+ aggregation	Combines quantity and label skew into DSI for adaptive DP noise and soft aggregation weighting.

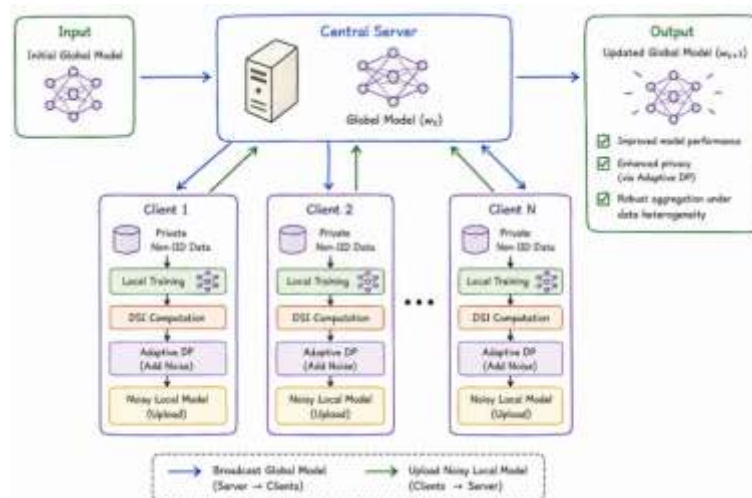
3. PROPOSED METHODOLOG

3.1. System Architecture

Such a model is designed for federated learning setups where client data statistics are different. As can be seen from Fig. 1 below, there is a server that acts as an intermediary in the federated learning setup while the real client data are situated on the local side. First, the global model is sent to the clients for the purpose of conducting local training on the non-IID data. Each client computes DSI and conducts ADP on the client level.

Distributed model averaging utilizes DSI aware aggregation for the local noisy models in order to obtain the updated global model. The updated global model is then distributed to all the involved clients in the subsequent communication phase.

Fig. 1: Architecture of the proposed DSI-aware Adaptive Differential Privacy and aggregation framework



3.2 Federated Learning Formulation

The local data available from client i can be described using Eq. (1):

$$D_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i} \quad (1)$$

where n_i denotes the number of training samples for client i . The total number of training samples for all clients is defined by Eq. (2):

$$N_{\text{total}} = \sum_{i=1}^N n_i \quad (2)$$

Consider the global model parameter w_t at iteration t . This model is broadcast to all the participating clients for the optimization of the client-side data. The updated local model can be given as Eq. (3):

$$w_{i,t}^{\text{local}} = \text{Train}(w_t, D_i) \quad (3)$$

This local model is perturbed via client-wise Gaussian noise level σ_i to yield $\tilde{w}_{i,t}$. The global model update is achieved by the weighted aggregation in Eq. (4):

$$w_{t+1} = \sum_{i=1}^N \omega_i \tilde{w}_{i,t} \quad (4)$$

Here, ω_i represents the normalized weighting factor. The condition for normalization is expressed by Eq. (5).

$$\sum_{i=1}^N \omega_i = 1 \quad (5)$$

Fig.1 displays the relationship of the server-client interaction in relation to local training, DSI computation, Adaptive-DP, noisy model swapping, and aggregation using DSI.

3.3 Controlled Non-IID Client Partitioning

The evaluation of the proposed framework with non-IID client distribution is done through dividing the dataset into partitions through the use of the Dirichlet-based non-IID method. In this regard, the Dirichlet distribution is set with α that is the measure of heterogeneity among the clients. For the last setting, there are 12 clients with the use of the Dirichlet distribution of $\alpha = 1.0$. The seed for the process is 2947.

Table 2: Final Federated Learning Configuration

Parameter	Value
Number of clients	12
Dirichlet parameter	1.0
Training samples	4,088
Test samples	1,022
Partition seed	2947
Communication rounds	10

The Table 2 shows that there are 12 clients and 10 iterations in the experimental setup of the Dirichlet parameter of $\alpha = 1.0$. It leads to a clustering which is determined by the variation in the number of samples from the clients and the classes. The total 4,088 training samples have been used only once.

3.4 Proposed Composite Data Skew Index

This methodology, which he proposes, utilizes a measure called Data Skew Index (DSI) at the client level that considers both kinds of statistical heterogeneity: skewness of quantity and skewness of labelling.

3.4.1. Quantity Skew

Let n_i be the number of training examples in client i , then the mean client size is

$$\bar{n} = (1/N) \sum_{i=1}^N n_i \quad (6)$$

The amount skew term is shown in Eq. (7):

$$QS_i = |n_i - \bar{n}| / N_{\text{total}} \quad (7)$$

As the client samples diverge from the mean of clients, the value of quantity-skew increases. Consequently, Eq. (7) gives us the variability in quantity of samples per client level.

3.4.2. Label-Distribution Skew

Let the global proportion of class c be denoted by $P(c)$, and that of client i be denoted by $P_i(c)$. The skewness in labels is calculated using Eq. (8):

$$LS_i = (1/2) \sum_{c=1}^C |P_i(c) - P(c)|, \quad (8)$$

where C denotes the number of classes targeted. The constant $1/2$ represents the total variation distance between the label distribution at client level and global level.

3.4.3. Composite DSI

Composite DSI is obtained from combining both quantity skew and label distribution skew as in Eq. (9):

$$DSI_i = (QS_i + LS_i) / 2. \quad (9)$$

Therefore, Eq. (9) constitutes an efficient client-side representation of statistical heterogeneity for the upcoming privacy and aggregation techniques. Note that the DSI value is not a measure of malice. This is because the client may have high DSI as a result of the local data being different from the global distribution for the right reasons.

3.5. DSI-Driven Adaptive Differential Privacy

DSI is used to generate the noise scale for Differential Privacy. The approach uses client-dependent noise scale rather than applying the same Gaussian noise to every client because of the skewed local data distribution.

Adaptive Noise Scale for Client i can be expressed as: Eq. (10)

$$\sigma_i = f(DSI_i), \quad (10)$$

where $f(\cdot)$ represents the monotonic mapping function employed by the suggested framework. The mapping fulfils.

$$DSI_i^{(a)} > DSI_i^{(b)} \Rightarrow \sigma_i^{(a)} > \sigma_i^{(b)}. \quad (11)$$

Thus, Eq. (11) defines the desired monotonic relation between the skew of client data and the noise scale being used. The Gaussian noise is drawn using Eq. (12):

$$\varepsilon_i \sim N(0, \sigma_i^2) \quad (12)$$

and added to the local model as shown in Eq. (13):

$$\tilde{w}_{i,t} = w_{i,t}^{\text{local}} + \varepsilon_i. \quad (13)$$

For the last 12-client system, the noise scaling factor value varies between 0.0109 and 0.0199 with a mean value of 0.0143.

3.6. DSI-Aware Client Aggregation

The Federated Averaging (FedAvg) method relies on the local sample size to determine the client's participation. The client weight is shown in Eq. (14):

$$\omega^{\text{FedAvg}}_i = n_i / \sum_{j=1}^N n_j. \quad (14)$$

Extension to sample-based weighting using the computed DSI value. Unnormalized DSI aware weight is denoted as Eq. (15):

$$\hat{\omega}_i = n_i / (1 + \lambda DSI_i) \quad (15)$$

where λ is the skew sensitivity parameter. Normalized contribution can be calculated using equation (16):

$$\omega_i = \hat{\omega}_i / \sum_{j=1}^N \hat{\omega}_j \quad (16)$$

Thus, the normalized weights meet the condition in Eq. (5). In case $\lambda > 0$, the connection between DSI and influence on aggregation can be written as

$$\text{DSI}_i \uparrow \Rightarrow \hat{\omega}_i \downarrow \quad (17)$$

As such, Eq. (17) ensures soft influence control and not hard client exclusion. No client is excluded from aggregation.

3.7. Complete Proposed Training Procedure

For every iteration, the server will communicate the global model to the client. The client will do local learning, determine the skewness of both data and labels, and calculate its DSI value. This noise scaling factor that is specific to the client in DP is calculated using the DSI value.

Algorithm 1. DSI-Aware Adaptive Differential Privacy and Aggregation

Require: Global model w_0 , client datasets $\{D_i\}_{i=1}^N$, rounds T , DSI parameter λ , $\sigma_{\min}$, $\sigma_{\max}$

Ensure: Final global model w_t

- 1: for $t = 1$ to T do
- 2: Server broadcasts w_t to participating clients.
- 3: for each client $i = 1, \dots, N$ do
- 4: Train locally on D_i and obtain $w_{i,t}^{\text{local}}$.
- 5: Compute quantity skew QS_i using Eq. (7).
- 6: Compute label skew LS_i using Eq. (8).
- 7: Compute DSI using Eq. (9).
- 8: Compute adaptive noise scale: $\sigma_i = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min})\text{DSI}_i$
- 9: Generate Gaussian noise: $\epsilon_i \sim N(0, \sigma_i^2)$
- 10: Perturb the local model: $\tilde{w}_{i,t} = w_{i,t}^{\text{local}} + \epsilon_i$
- 11: Compute DSI-aware contribution: $\hat{\omega}_i = n_i / (1 + \lambda \text{DSI}_i)$
- 12: end for
- 13: Normalize client weights: $\omega_i = \hat{\omega}_i / \sum_j \hat{\omega}_j$
- 14: Aggregate: $w_{t+1} = \sum_i \omega_i \tilde{w}_{i,t}$
- 15: end for
- return w_t

The new global model is obtained by aggregating the noisy local model. The entire process for the computation of DSI, Adaptive DP Perturbation, aggregation weight, and aggregation process is shown in Algorithm 1 below.

3.8. Computational Complexity

The computation of aggregation weights that takes into account DSI is linear with respect to the number of participating clients. From Eq. (16), we note that the computation of aggregation weight will have a complexity of $\mathcal{O}(N)$. The Gaussian differential privacy perturbation depends on the number of model parameters P .

However, the approach fails to create a distance matrix between pairs of clients' updates. Consequently, unlike the distance-based Byzantine algorithms like Krum, DSI-based averaging does not need to conduct pairwise comparison of clients' updates.

4. EXPERIMENTAL SETUP

4.1. Dataset and Classification Task

In order to perform experiments, it is required to consider the federated learning of binary classification in this paper. The federated learning of binary classification is stated as the problem of detecting whether the stroke occurs or not using the dependent variable. There is a huge class imbalance in the dataset and that is why the evaluation of federated models cannot be done using the accuracy.

These data sets are then divided into training and test data sets. In this experiment, there are 4,088 training samples and 1,022 test samples. These samples are distributed to 12 federated clients in a controlled Dirichlet non-IID fashion as discussed in Section 3.3. The client partition generated is significantly dissimilar not only because of the number of local samples but also because of their distribution of classes. The number of local samples range between 166 and 750, and the number of positive stroke samples differs across all clients.

4.2. Federated Learning Setup

The setup for federated learning in this final experiment is presented in Table 3. Seeds 2947 of the fixed partition have been used in all experiments to make sure that consistent client-level non-IID structure is maintained throughout the experiment. This is done to ensure that any difference in the outcome of the experiment does not occur due to the distribution of client's data.

Table 3. Final Experimental Setup

Parameter	Setting
Dataset	Stroke classification
Training samples	4,088
Test samples	1,022
Number of clients	12
Dirichlet α	1.0
Partition seed	2947
Communication rounds	10
Main seed	42
Byzantine clients	C11, C12
Attack	Gaussian perturbation
Attack standard deviation (σ_{attack})	0.50
DSI λ	1.0

4.3. Client-Level Non-IID Configuration

In the final partition, there are 4,088 samples of training data distributed to 12 clients exactly once. There is no duplication or exclusion of any sample, and the distribution of the clients is given in Table 4 below.

Client-level distribution reflects both the difference in terms of quantities as well as the difference in the distribution of labels. For instance, the client C1 has 750 samples with 4 positive samples, and client C2 has 39 positive samples with a total of 172 samples. Additionally, client C6 has lesser amounts of local data compared to client C8; however, the percentage of positive samples is remarkably different from each other.

Table 4: Client 12-Partitioning Table

Client	Samples	Stroke 0	Stroke 1
C1	750	746	4
C2	172	133	39
C3	204	174	30
C4	388	372	16
C5	723	720	3
C6	166	163	3
C7	300	278	22
C8	487	472	15
C9	175	160	15
C10	348	342	6
C11	190	154	36
C12	185	175	10
Total	4088	3905	183

The distribution of samples per client represents both the heterogeneity in the sample quantity and the label distribution. For instance, the number of training samples for client C1 is 750 with 4 positive labels, while that for client C2 is 172 and has 39 positive labels. Furthermore, while client C6 has fewer local data than client C8, it still has a much different label distribution. Such a distribution helps ensure the required variance in quantity and label distribution.

4.4. Baseline and Comparative Methods

The suggested framework is analysed in comparison to traditional and Byzantine-resistant aggregators. The basic experimental configurations are listed below. The following techniques were analysed:

- E1: FedAvg – classic sampling-based federated averaging without Adaptive Differential Privacy and Byzantine noise injection.
- E2: FedAvg with Gaussian Attack - FedAvg with the standard Gaussian Byzantine parameter attack.
- E3: Krum + Gaussian Attack – Evaluation of Krum Byzantine resilient aggregation technique for Gaussian Attack.
- E4: Bulyan + Gaussian Attack – Same attack setting for evaluation of Bulyan’s resilient aggregation method.

- E5: Adaptive DP + FedAvg – Adaptive Differential Privacy along with traditional aggregation method FedAvg without any Byzantine attack.
- E6: DSPA+AdaptiveDP+GaussianAttack – the main suggested combination that consists of DSI-aware aggregation and Adaptive Differential Privacy in the context of Gaussian Byzantine attack.

All approaches utilize the same client partitions and principal experimental setup whenever possible. Krum and Bulyan represent well-known Byzantine fault-tolerant baselines for comparison with the suggested DSI-aware aggregation method [6, 7].

4.5. Byzantine Attack Model

In order to test the robustness of the system to adversarial attacks, a Gaussian parameter perturbation attack is launched on two chosen clients, C11 and C12. An attacked client's model receives the addition of Gaussian perturbation to its local model parameters before aggregation. If w_i represents the locally trained model parameters, then the attacked model is given by Eq. (18):

$$w_{i\text{attack}} = w_i + \eta_i, \quad (18)$$

where the disturbance is generated using Eq. (19):

$$\eta_i \sim N(0, \sigma^2 \text{ attack}) \quad (19)$$

The main experiments involve using the attack strength defined in Equation (20):

$$\sigma_{\text{attack}} = 0.50. \quad (20)$$

Other strength-of-attack experiments take their parameters from Eq. (21):

$$\sigma_{\text{attack}} \in \{0.25, 0.50, 0.75, 1.00\}. \quad (21)$$

These settings are used to evaluate the sensitivity of the aggregation methods to increasing Byzantine perturbation strength. The attack is applied only to the designated Byzantine clients. The remaining clients are treated as honest, although their updates may naturally differ because of heterogeneous local datasets.

4.6. Evaluation Metrics

Performance of each federated model is measured on the basis of accuracy, precision, recall, F1 score, ROC-AUC, sensitivity, and specificity. TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

Formula for accuracy:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}). \quad (22)$$

Precision is defined as:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (23)$$

Recall, which is equivalent to Sensitivity for the positive class, is given by:

$$\text{Recall} = \text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}). \quad (24)$$

The F1-score is calculated as:

$$\text{F1} = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}). \quad (25)$$

Specificity is defined as:

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}). \quad (26)$$

Because the stroke classification problem is strongly imbalanced, F1-score, Recall, and ROC-AUC are emphasized in addition to Accuracy. Accuracy can remain high when a classifier predominantly predicts the majority class; therefore, minority-class-sensitive measures provide a more informative assessment of classification performance.

4.7. Repeated-Seed Validation

To check the stability of results, the critical attacks on configuration E2, E3, E4, and E6 have been tested with five independent experiment seeds. The set of experiment seeds is $S = \{42, 123, 456, 789, 1000\}$. The set of clients is kept constant with partition seed 2947, whereas the experiment seed changes in each run. Mean and sample standard deviation in five runs will be compared.

4.8. Ablation Study

A1: Adaptive Dynamic Programming & DSI Weighting – both suggested elements are active.

- Adaptive DP: DSI-Aware Aggregation is turned off in favor of Adaptive Differential Privacy only.
- A3: DSI Weighting Only – The DSI-based aggregation process is kept intact without the Adaptive Differential Privacy. In the ablation studies, the impact of individual elements can be measured rather than that of the entire system as a whole.

4.9. Attack-Strength Sensitivity Analysis

To assess the robustness of the tested approaches, the standard deviation of the Gaussian attacks is varied. Four attack types are analyzed:

$$\sigma_{\text{attack}} \in \{0.25, 0.50, 0.75, 1.00\}$$

In this way, for each level of attacks, the optimal communication round is determined based on F1-score, which is consistent with the main method of evaluation used in this paper. F1-score, Accuracy, Recall, and ROC-AUC measures are then compared among the aggregation approaches.

4.10. Privacy–Utility Analysis

The privacy versus utility characteristic of the newly designed ADP scheme is analyzed through a scaling of the client-based adaptive noise values to the baseline DSI design.

Five scales of multiplicative noise are examined:

$$s_{\sigma} \in \{0.50, 0.75, 1.00, 1.25, 1.50\}$$

In case of a client having base adaptive noise scale σ_i , scaled noise is represented as

$$\sigma_{\text{scaled } i} = s_{\sigma} \sigma_i.$$

The Byzantine attack is disabled for this analysis so that changes in predictive performance can primarily be attributed to variations in Adaptive Differential Privacy noise rather than changes in attack strength. Accuracy, Recall, F1-score, and ROC-AUC are evaluated at each noise scale. The resulting analysis is presented in Section 5.7.

4.11. Implementation Details

The framework under discussion is implemented in Python with PyTorch. The lightweight fully-connected neural network, which is called SATNet in the implementation process, serves as the classifier both locally and globally. The network is composed of three fully connected layers with the following dimensions.

$$d_{\text{in}} \rightarrow 64 \rightarrow 32 \rightarrow 2.$$

The ReLU activation function is used after the first two layers, while the dropout function with probability $p = 0.2$ is used after each hidden layer. The same local optimization setup is used by all clients in order to have a fair basis of comparison between the different federated learning techniques that have been evaluated. The Adam optimizer is employed with a learning rate of 0.001. The number of local epochs per client is set to five with batch size equal to 32. Since the data is imbalanced, a class-weighted cross entropy loss function is used. The class-weighted loss function is given by:

$$L = - \sum_{c=1}^C \alpha_c y_c \log(p_c).$$

Where “C” is the number of classes, y_c is the target variable value for class “c”, p_c is the predicted probability of class “c”, and α_c the corresponding class coefficient. In the main federated learning experiment, the number of clients is 12, the value of the Dirichlet concentration parameter α is 1.0, and the seed for client sampling is constant at 2947. The total number of communication rounds in the experiment is 10. The optimal communication round in terms of performance is selected based on the F1-score. In the main adversarial configuration, Byzantine clients are C11 and C12, while the standard deviation of the Gaussian attack is 0.50. The primary implementation design is shown in Table 5.

Table 5: Implementation and Training Configuration

Parameter	Value
Framework	Python / PyTorch
Model	SATNet
Hidden layers	64, 32
Output classes	2
Activation	ReLU
Dropout probability	0.2
Optimizer	Adam
Learning rate	0.001
Local epochs	5
Batch size	32
Loss function	Weighted Cross-Entropy
Clients	12
Dirichlet α	1.0
Partition seed	2947
Communication rounds	10
Primary experiment seed	42
Model selection metric	F1-score
Byzantine clients	C11, C12
Gaussian attack STD	0.50

Table 6: Principal Single-Seed Experimental Results

Method	Round	Accuracy	Precision	Recall	F1	AUC
FedAvg	8	89.24	19.39	38.00	25.68	79.88
FedAvg + Gaussian	10	88.94	19.42	40.00	26.14	79.60

Method	Round	Accuracy	Precision	Recall	F1	AUC
Krum + Gaussian	2	91.49	24.66	36.00	29.27	79.77
Bulyan + Gaussian	3	88.16	22.05	56.00	31.64	82.20
Adaptive DP + FedAvg	10	88.65	19.44	42.00	26.58	79.70
Proposed	7	92.56	26.79	30.00	28.30	79.09

5. RESULTS AND DISCUSSION

5.1. Main Comparative Evaluation

The main comparison is among the FedAvg, Krum, Bulyan, adaptive differential privacy (DP) with FedAvg, and the suggested method for DSI-aware aggregation using adaptive DP. In the case of the major adversarial attack setting, there are 12 clients out of which 2 are Byzantine clients (C11 and C12), and $\sigma_{\text{attack}} = 0.50$.

Single-seed results have been presented in Table 6. The best F1-score communication round for each approach is presented. According to Table 6, the proposed approach outperforms other approaches in terms of Accuracy (92.56%) and Precision (26.79%).

Bulyan has the highest Recall and F1-score values, which are 56.00% and 31.64%, respectively. In this case, the suggested algorithm has the highest Accuracy in comparison to other algorithms in this single seed experiment, whereas Bulyan is better than the rest in terms of Recall and F1-scores.

5.2. Robustness to Increasing Gaussian Attack Strength

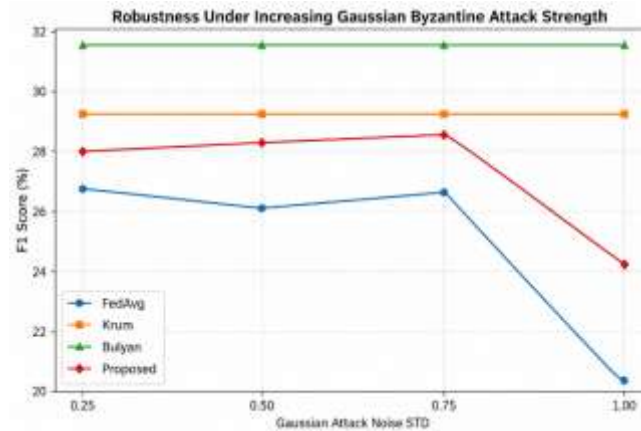
The F1-scores for Gaussian attack standard deviations 0.25, 0.50, 0.75, and 1.00 have been calculated as discussed in Section 4.9. These are presented in Table 7.

Table 7: F1-Score Under Increasing Gaussian Attack Strength

Attack STD	FedAvg	Krum	Bulyan	Proposed
0.25	26.75	29.27	31.64	28.00
0.50	26.14	29.27	31.64	28.30
0.75	26.67	29.27	31.64	28.57
1.00	20.47	29.27	31.64	24.37

As evident from Table 7 below, FedAvg drops sharply when subjected to the highest strength of the attack, whereas the F1-score of the proposed algorithm is above 28% even when the STD of the attack is 0.75, and when it reaches 1.00, the F1-score is 24.37%. Robustness tendency is demonstrated in Fig. 2

Fig. 2: F1-score under increasing Gaussian Byzantine attack strength



5.3. Ablation Study

Individual impacts of Adaptive DP and DSI aware aggregation are analyzed by considering three different settings. The findings are provided in Table 8.

From Table 8, Adaptive DP Only performs best in terms of Accuracy (92.76%), while the combination of the two approaches enhances F1 from 27.45% to 28.30%. On the other hand, DSI Weighting Only performs best in terms of Recall (40.00%).

Table 8: Ablation Study Under Gaussian Attack ($\sigma_{\text{attack}} = 0.50$)

Configuration	Round	Accuracy	Recall	F1
Adaptive DP + DSI Weighting	7	92.56	30.00	28.30
Adaptive DP Only	7	92.76	28.00	27.45
DSI Weighting Only	10	88.94	40.00	26.14

It is clear from the results that adaptive DP and DSI-aware weightings influence various facets of the models' performance. It is thus clear that their integration yields a more balanced setup than the ideal in all metrics.

5.4. Repeated-Seed Validation

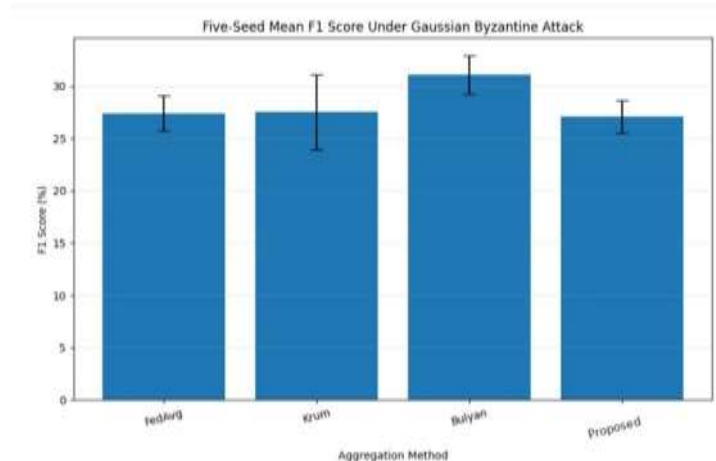
The methods under attack were assessed by means of the five seeds mentioned in Section 4.7. While the client partition remained constant, the random seed changed. Mean values along with the sample standard deviation are presented in Table 9.

Table 9: Five-Seed Mean $\pm$ Standard Deviation Results

Method	Accuracy	Recall	F1	ROC-AUC
FedAvg + Gaussian	89.94 $\pm$ 3.47	38.80 $\pm$ 13.75	27.40 $\pm$ 1.65	80.05 $\pm$ 1.23
Krum + Gaussian	85.13 $\pm$ 7.22	54.40 $\pm$ 15.32	27.50 $\pm$ 3.59	81.12 $\pm$ 1.02
Bulyan + Gaussian	89.78 $\pm$ 3.04	46.80 $\pm$ 12.70	31.07 $\pm$ 1.88	82.64 $\pm$ 0.90
Proposed	90.18 $\pm$ 1.64	37.20 $\pm$ 5.76	27.10 $\pm$ 1.55	80.22 $\pm$ 1.50

As seen in Table 9, the proposed method attains the maximum average Accuracy score of $90.18 \pm 1.64\%$ and the lowest Accuracy score standard deviation. Bulyan scores the maximum average F1-score and ROC-AUC scores of $31.07 \pm 1.88\%$ and $82.64 \pm 0.90\%$ respectively. Hence, the proposed method attains excellent and consistent Accuracy, while Bulyan continues to have an edge in F1-score and ROC-AUC. Fig. 3 shows the five-seed F1-score comparison.

Fig. 3: Average F1 score and standard deviation for five random seeds.



5.5. Statistical Significance Analysis

Paired comparison of the F1 scores was done using the Wilcoxon signed-rank test, while the paired t-test was used for supplementary testing. Bonferroni correction was done for the three paired comparisons done on the suggested approach. The outcomes are presented in Table 10.

Table 10: Statistical Analysis for F1-Score Comparison

Comparison	Mean $\Delta F1$	Wilcoxon p	Corrected p	Cohen's dz
Proposed vs. E2	-0.30	1.0000	1.0000	-0.133
Proposed vs. E3	-0.39	0.6250	1.0000	-0.090
Proposed vs. E4	-3.97	0.0625	0.1875	-2.831

The F1 values of the new method and FedAvg and Krum show no statistically significant difference after Bonferroni correction in Table 10. The comparison of the new method and Bulyan yields a Wilcoxon test p-value of 0.0625, yet the p-value of the corrected test is 0.1875 and is larger than 0.05. It should be noted that five seeds are available for testing here.

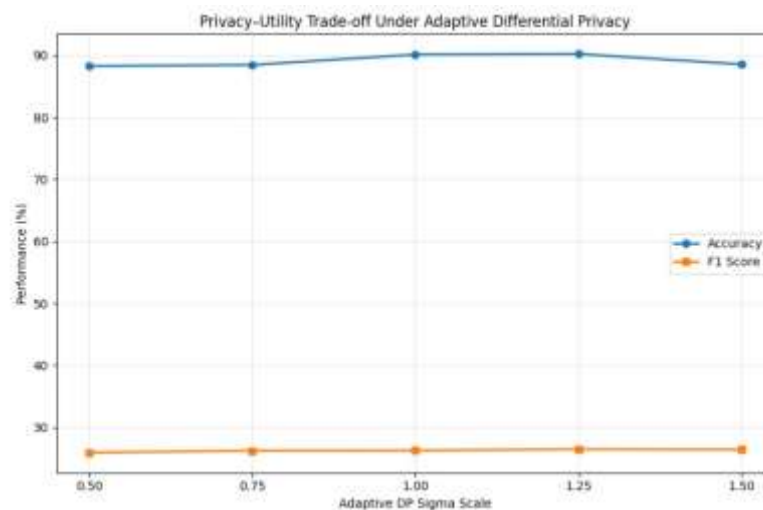
5.6. Client-Level Adaptive Privacy Behaviour

The proposed Adaptive DP approach applies different noise scaling factors to individual clients based on their DSI. The corresponding figures are provided in Table 11. Based on Table 11, the noise scaling factor is found to be between 0.0109 and 0.0199, with an average value of 0.0143 and standard deviation of 0.0029. Pearson and Spearman correlation coefficients between DSI and noise scaling factor are equal to 1.0000. Therefore, the implemented approach meets the intended monotonic relationship: Clients with high DSI have larger noise scaling factors. The relationship is demonstrated in Fig. 4

Table 11: Client-Level DSI and Adaptive Noise Scale

Client	DSI	σ_i
C1	0.0717	0.0165
C2	0.1097	0.0199
C3	0.0659	0.0159
C4	0.0095	0.0109
C5	0.0690	0.0162
C6	0.0367	0.0133
C7	0.0173	0.0116
C8	0.0268	0.0124
C9	0.0388	0.0135
C10	0.0166	0.0115
C11	0.0888	0.0180
C12	0.0217	0.0120

Fig. 4: Privacy–utility trade-off under different Adaptive Differential Privacy noise scales



5.7. Privacy–Utility Trade-off

The privacy-utility test measures the effect of applying five multiplicative scaling factors to the noise levels of the clients. The Byzantine attack is turned off to measure the effect of scaling noise levels alone. The outcomes are summarized in Table 12.

Table 12: Privacy–Utility Trade-off Under Different Adaptive Noise Scales

Scale	Mean σ	Round	Accuracy	Recall	F1
0.50×	0.0071	10	88.26	42.00	25.93
0.75×	0.0107	10	88.45	42.00	26.25
1.00×	0.0143	8	90.12	36.00	26.28
1.25×	0.0179	8	90.22	36.00	26.47
1.50×	0.0214	10	88.55	42.00	26.42

The utility shown by Table 12 for all the noise scales considered is relatively stable. Accuracy varies between 88.26% and 90.22%, while F1-score varies between 25.93% and 26.47%. The maximum Accuracy and F1-score values are observed at the 1.25× scale. Scaling the value further to 1.50× does not affect either Accuracy or F1-score positively. In this experiment, we are conducting the noise-scale sensitivity test and the privacy-utility test. The sigma values presented here do not formally prove ϵ or δ differential privacy guarantees due to various formal requirements

5.8. Overall Discussion

These findings clearly illustrate the applicability of the proposed framework for finding the right balance between the three aspects mentioned above. First of all, the suggested method produces the highest Mean Accuracy when five seeds are used ($90.18 \pm 1.64\%$), as well as the highest Accuracy among single seeds (92.56%). At the same time, Bulyan is superior in mean F1-score and ROC-AUC.

The performance of the suggested algorithm in case of a stronger Gaussian attack is impressive on a medium level, despite the deterioration of the F1-score at the maximal perturbation strength level. As for the ablation study, it indicates that the combination of Adaptive DP with DSI-aware weighting exceeds only Adaptive DP by the F1-score metric.

Also, it can be seen that the client-level experiment is showing that the DSI is the reason behind the adaptive noise scaling in the proposed approach. In addition to this, the privacy-utility experiment shows that there is a certain range of utility for various noise. Therefore, it can be stated from the experimental results that the proposed framework is achieving its important objective as the data skew of the clients is being used as input in order to control the privacy perturbation and aggregation weights.

6. LIMITATIONS AND FUTURE WORK

6.1. Limitations

In the case of the current study, the performance of our model has been evaluated on a dataset of stroke classification with a federated learning setup consisting of 12 clients. In the case of the adversarial attack, Gaussian Byzantine attacks have been used, however, our proposed framework focuses on the data skew aware privacy and aggregation rather than identifying Byzantine clients.

6.2. Future Work

Future work will study the efficacy of the proposed framework on large scale data sets as well as on different types of clients. Privacy budget accounting, attack models, and noise mapping techniques from DSI to noise can be used to improve the security of DSI protocol. Another exciting area of future research could involve the integration of data skew and Byzantine attacks.

7. CONCLUSION

In this work, a Data Skew Aware Adaptive Differential Privacy and Aggregation (DSADPA) framework has been proposed for federated learning using heterogeneous data. In the DSADPA framework that has been proposed in this paper, the data skew index (DSI) is used at the client level to measure data skewness across clients. This framework uses the data skew information

in the process of adapting the differential privacy noise level and aggregation effect. Unlike frameworks that use only the update deviation information in order to preserve participating clients and data heterogeneity in the aggregation process, the DSADPA framework can do both.

From the results on stroke classification dataset, it can be seen that the proposed approach performs competitively for both non-IID and Gaussian Byzantine attacks. From the experiments carried out with different seeds, the proposed approach gives the highest Accuracy scores, but Bulyan gives better F1-score and ROC-AUC scores for repeated seeds.

As seen from the results, the integration of adaptive differential privacy together with DSI-based weighting would be able to produce an acceptable trade-off between predictive accuracy and client-wise privacy adaptation. As can be concluded from the experimental results, incorporating data skewness in the process of privacy adaptation and aggregation appears to be an interesting approach to privacy-aware FL.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54, PMLR, 2017, pp. 1273–1282.
- [2] Q. Li, Y. Diao, Q. Chen, X. Zhang, Federated learning on non-iid data: A survey, *Neurocomputing* 465 (2021) 371–390.
- [3] D. M. Jimenez-Gutierrez, D. Solans, M. Heikkilä, A. Vitaletti, N. Kourtellis, A. Anagnostopoulos, I. Chatzigiannakis, Non-iid data in federated learning: A systematic review with taxonomy, metrics, methods, frameworks and future directions, *IEEE Communications Surveys & Tutorials* (2024).
- [4] H. B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning differentially private recurrent language models, in: *International Conference on Learning Representations*, 2018.
- [5] J. Fu, Y. Hong, X. Ling, L. Wang, X. Ran, Z. Sun, W. H. Wang, Z. Chen, Y. Cao, Differentially private federated learning: A systematic review (2024). arXiv:2405.08299.
- [6] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries.
- [7] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries Byzantine tolerant, in: *Advances in Neural Information Processing Systems*, Vol. 30, 2017.
- [8] E.-M. El-Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in byzantium, in: *35th International Conference on Machine Learning*, Vol. 80, PMLR, 2018, pp. 3521–3530.
- [9] S. Zhang, J. Huang, P. Li, C. Liang, Differentially private federated learning with local momentum updates and gradients filtering, *Information Sciences* 680 (2024) 120960. doi:10.1016/j.ins.2024.120960.
- [10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L.D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Harchaoui, S. Harchaoui, X. He, L. He, Z. Li, Y. Li, N. Papernot, R. Ramasubramanian, A. Scaman, S. M. A. Sen, A. T. Suresh, F. Tramer, S. Trifonov, Z. Xu, Z. Zhang, et al., Advances and open problems in federated learning, *Foundations and Trends in Machine Learning* 14 (1–2) (2021) 1–210.