

Privacy-Preserving Federated Learning with Adaptive Differential Privacy in Distributed Networks

GAURAV KUMAR SINHA¹

¹7319 LONGSPUR HOLLOW LANE KATY, TX 77493-4704, UNITED STATES.

Abstract

Federated learning lets many devices or institutions train a shared model without moving their raw data to a central location, yet the model updates they send can still reveal sensitive information about the data on which they were computed. Differential privacy offers a rigorous defense by clipping each update and adding calibrated random noise, but most deployments use a single fixed noise level and a single fixed clipping bound for every client, every round and every layer of the model. This is wasteful. It adds more noise than needed when updates are small and late in training, it ignores the fact that clients differ in how sensitive their data are, and it makes accuracy fall sharply when strong privacy is required. This paper proposes an adaptive differential privacy framework for federated learning in distributed networks. The framework adjusts the clipping bound to the observed size of updates, schedules the noise level across rounds, assigns personalized privacy budgets to clients, and accounts for the cumulative privacy loss of each client. It combines these mechanisms with update compression and with a hierarchical client, edge and cloud architecture that adds protection at two levels and reduces traffic to the cloud. In simulation with 1,000 clients, the adaptive framework reached 86.7% accuracy on a clothing-image task at a privacy loss parameter of 8, against 80.9% for fixed-noise differential privacy, and the gain was largest under strong privacy and severe data heterogeneity. Membership inference attack success fell from an area under the curve of 0.71 without protection to 0.53, communication per round fell by a factor of 25 with compression, and cloud-bound traffic fell by a factor of 25 with the hierarchical design. The results suggest that adapting privacy mechanisms to the state of training and to the client gives a better balance between privacy and utility than a fixed setting.

Keywords: Federated Learning, Differential Privacy, Adaptive Clipping, Privacy Budget, Hierarchical Federated Learning, Update Compression, Distributed Networks, Membership Inference.

1. Introduction

Much of the data that could improve machine learning models is sensitive and scattered across many owners. Hospitals hold patient records, banks hold transactions, phones hold messages and photographs, and vehicles, factories and homes hold streams of sensor readings. Regulations such as the European General Data Protection Regulation restrict how personal data can be collected and moved [27], and organizations are often unwilling to share their data even when the law allows it. Centralizing the data for training is therefore frequently impossible or undesirable.

Federated learning was proposed as an alternative. Instead of sending data to a server, each participant trains a copy of the model on its own data and sends only the resulting model update to a coordinator, which combines the updates into an improved global model that is returned to the participants [1]. The data never leave their owners, which is a significant gain for privacy. It is not, however, a guarantee of privacy. Research has shown that model updates and gradients can leak information about the data behind them. An observer can sometimes decide whether a particular record was used in training, and under some conditions can reconstruct training examples from shared gradients [9], [10]. The coordinator, or anyone who intercepts the updates, may therefore learn more than the participants intend to reveal.

Differential privacy provides a mathematically rigorous answer to this threat. Its central idea is that the output of a computation should change very little whether or not any single individual's data are included, so that an observer cannot confidently infer anything about that individual [2]. In practice, this is achieved by limiting the influence of each participant, through clipping, and by adding random noise whose size is calibrated to that limit. Applied to learning, this approach was shown to be workable for deep networks [3], and later adapted to federated training in a way that protects entire clients rather than single records [4], [5].

The price of differential privacy is accuracy. The noise that protects participants also disturbs learning, and the stronger the protection, the larger the loss. Most implementations choose one clipping bound and one noise level in advance and apply them uniformly. This design has three weaknesses. First, update sizes change greatly during training, so a bound that is right at the start is far too large later, and noise calibrated to an oversized bound swamps the signal. Second, the usefulness of noise differs across rounds, since early rounds learn coarse structure that tolerates noise while late rounds refine fine details that do not. Third, clients differ. Some hold highly sensitive records and need strong protection, while others hold less sensitive data and could contribute more accurately under a weaker setting, yet a uniform policy treats them alike and either over-protects some or under-protects others.

We therefore propose an adaptive differential privacy framework for federated learning in distributed networks. The contributions of this paper are as follows: (i) adaptive clipping and noise scheduling that track the size of updates and the stage of training; (ii) personalized privacy budgets with per-client accounting of cumulative privacy loss; (iii) a compression-aware design in which update compression and privacy noise are calibrated together; (iv) a hierarchical client, edge and cloud architecture that places protection at two levels and cuts traffic to the cloud; and (v) a simulation study that measures utility, privacy leakage, communication and fairness. The remainder of the paper is organized as follows. Section II reviews background, Section III discusses related work, Section IV presents the methodology, Section V describes the experimental setup, Section VI reports results, Section VII discusses implications and limitations, and Section VIII concludes.

2. LITERATURE REVIEW

A. Federated Learning

The basic federated learning workflow is shown in Fig. 1. It proceeds in rounds, each with three steps. In the first step, a central server initializes the model and sends it to the participating clients. In the second, each client trains the model on its own private data, shown in the figure by the database and padlock attached to each client, and prepares an update. In the third step, the clients send their local models or updates to the central server, which aggregates them into a new global model and starts the next round. The aggregation shown in the figure combines the contributions of all clients, usually weighted by the amount of data each holds [1]. Because only model parameters travel, the raw data remain with their owners.

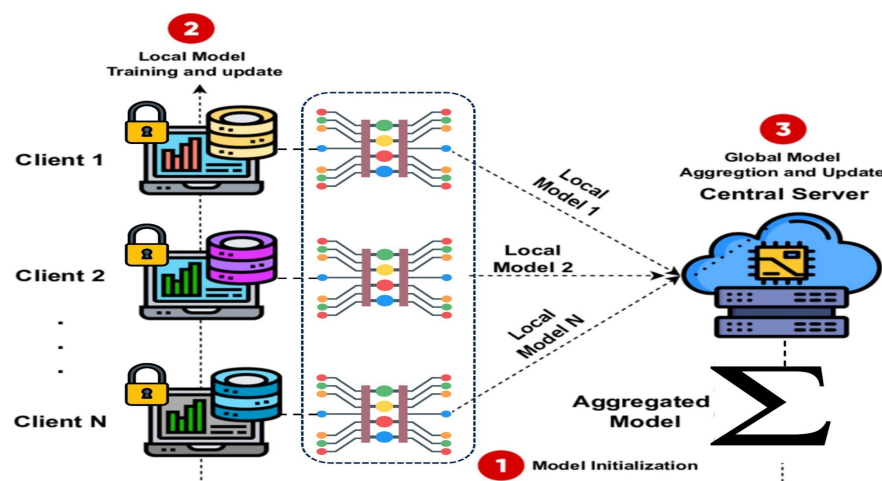


Fig. 1. Basic federated learning workflow: (1) model initialization, (2) local model training and update on private client data, and (3) global model aggregation and update at the central server.

Surveys describe the main challenges of this setting [11], [12]. Data are typically not identically distributed across clients, which slows convergence and can bias the global model toward some clients. Clients differ in computing power and connectivity, and only a fraction are available in any round. Communication is expensive, particularly for large models, and the updates themselves may expose private information. Methods such as proximal regularization address heterogeneity by limiting how far local models drift from the global one [13], while compression and sparsification address communication cost [16]-[18].

B. Privacy Threats in Federated Learning

Several attacks demonstrate that sharing updates is not safe by itself. Membership inference attacks try to determine whether a specific record was part of the training data, exploiting the fact that models behave slightly differently on examples they have seen [9]. Stronger formulations that calibrate the attack to each example show that earlier evaluations often underestimated leakage [28]. Gradient inversion attacks go further and attempt to reconstruct training inputs from the gradients that clients share, and they have succeeded for small batches [10]. Attacks that infer properties or class representatives of a participant's data from the updates they send have also been shown [22]. These threats can come from the server, from other participants or from an eavesdropper on the network, and defenses need to address more than one of these adversaries.

C. Differential Privacy and Its Use in Learning

Differential privacy formalizes the idea that a computation reveals little about any one individual. A mechanism is said to be differentially private if the probability of any output changes by only a bounded factor when one individual's data are added or removed, with a small allowance for failure. Two parameters describe the guarantee: a privacy loss parameter, usually called epsilon, whose smaller values mean stronger privacy, and a failure probability, usually called delta, that is set to a very small number [2]. The Gaussian mechanism achieves the guarantee by adding normally distributed noise scaled to the largest influence that any individual can have, and refined calibration methods reduce the noise needed for a given guarantee [23]. A key property is composition. Each time a mechanism is applied, privacy loss accumulates, and training involves many rounds, so the total loss must be tracked. Tighter accounting methods based on Rényi divergence provide much smaller bounds on the accumulated loss than naive composition [7].

For deep learning, clipping each example's gradient and adding noise to the sum made it possible to train networks with a formal guarantee [3]. In federated learning, the same idea was applied at the level of clients. Each client's update is clipped to a bound, the clipped updates are aggregated, and noise is added so that the presence or absence of a whole client cannot be inferred [4], [5]. Analyses of differentially private federated learning give convergence bounds that show how the noise level, the number of clients and the number of rounds interact [15]. The central difficulty, which motivates this paper, is that the noise needed for strong privacy slows learning and lowers accuracy.

D. Adaptive Mechanisms

Because a fixed clipping bound is hard to choose, adaptive clipping was proposed. The bound is adjusted during training so that it tracks a chosen quantile of the observed update sizes, which removes a sensitive hyperparameter and keeps the noise proportionate to the actual signal [6]. Adaptive approaches can also vary the noise over training, and the idea of personalized privacy, in which different individuals receive different guarantees according to their preferences, has a long history in the database literature [21]. In federated settings, personalized differential privacy has been explored for the Internet of Things, where participants have different privacy needs and resources [20]. These ideas are individually promising, but their combination with compression and a hierarchical network is less studied.

E. Compression and Hierarchical Architectures

Communication is a main cost of federated learning. Sparsification sends only the largest entries of an update [16], quantization represents values with a few bits [17], and error feedback stores the part that was discarded and adds it to a later update, which repairs much of the bias that compression introduces [18]. Frameworks that combine sparsification, quantization and

entropy coding achieve large reductions in communication volume [19]. Compression interacts with privacy, because it reduces the number of coordinates to which noise must be added but also changes the sensitivity of the update. Hierarchical federated learning adds a layer of edge servers between clients and the cloud, so that clients synchronize with a nearby edge server frequently and the edge servers communicate with the cloud less often, which reduces latency and wide-area traffic [14]. Fig. 2 shows this structure and contrasts it with the traditional design.

3. RELATED WORK

Work on private federated learning can be arranged into four threads: client-level differential privacy, adaptive control of clipping and noise, privacy with efficient communication, and system architecture. The first thread established that federated averaging can be made differentially private at the client level by clipping updates and adding noise to the aggregate, and that useful models can still be trained when many clients participate [4], [5]. A limitation noted in these studies is that the number of participating clients strongly influences the achievable accuracy, since noise is divided across more contributors when the client population is large. In small federations, such as a handful of hospitals, the cost of privacy is much higher.

The second thread concerns adaptivity. Adaptive clipping showed that the clipping bound can be learned privately from the data and that doing so achieves accuracy comparable to a carefully tuned fixed bound without the need for tuning [6]. Convergence analyses of private federated learning show that there is an optimal number of rounds for a given privacy budget, and that spending the budget uniformly is not optimal [15]. Personalized privacy extends adaptivity to participants, allowing different guarantees for different clients [20], [21]. Our work combines the three forms of adaptivity, over update size, over rounds and over clients, within one framework and studies how they interact.

The third thread asks how privacy and communication efficiency interact. Compression methods reduce the size of updates by large factors [16]-[19], and some studies suggest that the randomness of compression can itself contribute to privacy, although without a formal guarantee unless noise is added explicitly. Hybrid approaches combine differential privacy with cryptographic secure aggregation, so that the server sees only the sum of the updates and the noise needed can be reduced because each client's contribution is hidden [8], [24]. Our framework is compatible with secure aggregation and focuses on how to calibrate noise and compression together.

The fourth thread concerns architecture. Hierarchical designs with client, edge and cloud layers were proposed to reduce communication cost and latency [14]. They also change the privacy picture, since trust can be divided among levels, and noise can be added at the client to protect against the edge server and again at the edge to protect against the cloud. This has received less attention than the communication benefits.

Three open issues remain. First, most private federated systems use fixed settings that do not exploit the changing scale of updates over training or the differing needs of clients. Second, the joint design of compression and noise is rarely studied. Third, the privacy implications of hierarchical architectures are seldom examined in the same study as their communication benefits. This paper addresses all three.

4. METHODOLOGY

A. System Architecture

The framework supports both the traditional and the hierarchical organization of federated learning, shown in Fig. 2. In the traditional organization, shown on the left of the figure, a single parameter server, located in the cloud or at the edge, exchanges models directly with a pool of clients such as sensors, vehicles, phones, robots and computers. In the hierarchical organization, shown on the right, a cloud parameter server sits at the top, a layer of edge parameter servers sits beneath it, and each edge server manages its own pool of clients. Clients exchange updates with their edge server, and the edge servers exchange aggregated models with the cloud.

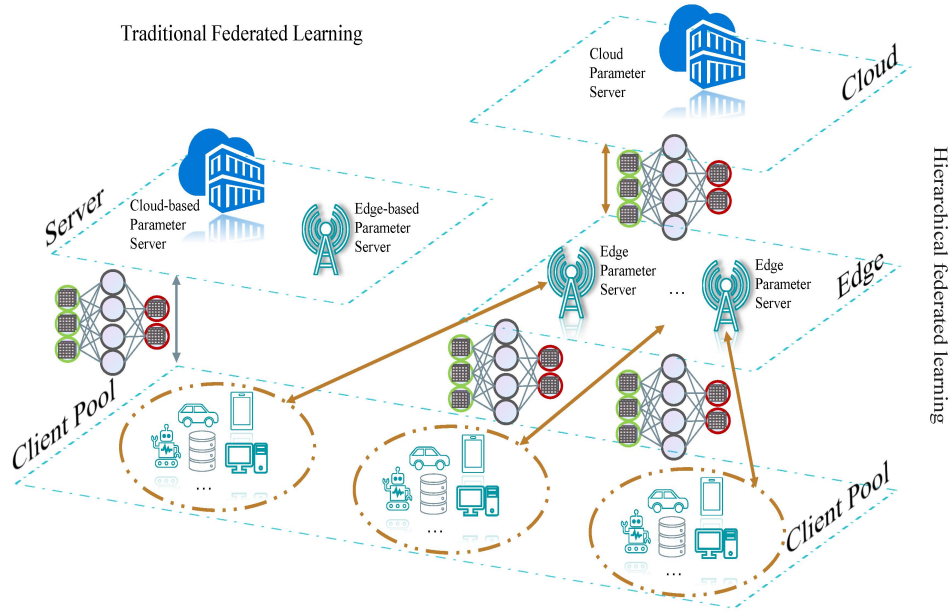


Fig. 2. Traditional federated learning with a single parameter server (left) and hierarchical federated learning with cloud, edge and client layers (right).

The hierarchical organization offers three benefits for this study. It shortens the distance traveled by most updates, since clients talk to a nearby edge server. It reduces cloud-bound traffic, since the cloud receives one aggregated update per edge server and not one per client. And it allows privacy to be applied in two stages, with protection at the client against the edge server and further protection at the edge against the cloud. The privacy mechanisms of the framework apply to both organizations, with the difference that, in the hierarchical case, an additional noise step takes place at each edge server before the aggregate is sent to the cloud.

B. Threat Model

We consider an honest-but-curious server, which follows the protocol but tries to learn as much as possible from the updates it receives, together with an external eavesdropper who may observe messages on the network. In the hierarchical case, both the edge servers and the cloud server are treated as curious. We also consider participants who may attempt membership inference or gradient inversion using the global model they receive. We do not consider attackers who corrupt the training process by sending malicious updates, which is a separate problem of robustness, and we assume that clients faithfully apply the privacy mechanism on their own devices. The goal of the framework is to bound what any of these observers can learn about the presence or content of any one client's data.

C. Privacy Pipeline at the Client

Fig. 3 shows the pipeline through which an update passes before it leaves a local node. After local training, the update first goes through compression, which reduces its size. It then passes through a differential privacy stage within the protected region marked in the figure, and only then is it sent to the central server. The figure shows two differential privacy modules that are connected to each other by a two-way link. One module is drawn with a bell-shaped noise distribution, which represents the addition of random noise, and the other is drawn with a vertical double arrow, which we interpret as the adaptive adjustment of the privacy level. The link between them indicates that the noise calibration and the adaptive control exchange information. In our framework, the second module monitors the size of updates and the accumulated privacy loss and tells the first how much noise to add and how tightly to clip.

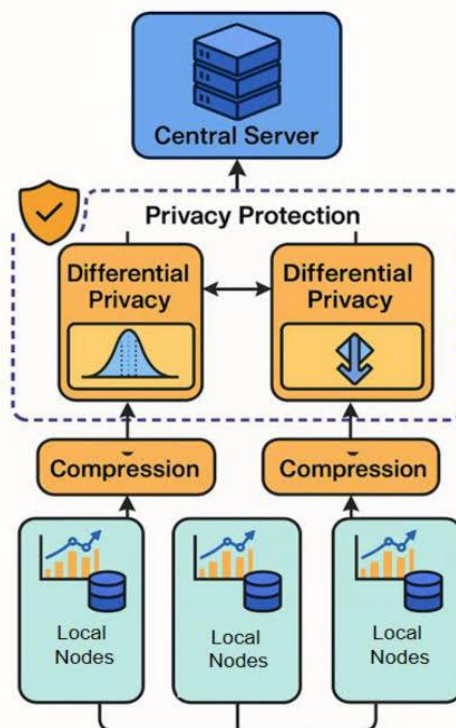


Fig. 3. Privacy-preserving pipeline: local nodes compress their updates, which then pass through the differential privacy modules inside the privacy protection boundary before reaching the central server.

Four operations take place at each participating client in each round. First, the client trains locally and forms its update as the difference between its new and its old model. Second, it clips the update so that its overall size does not exceed the current clipping bound, which limits the influence that one client can have. Third, it compresses the clipped update by keeping only its largest entries and representing them with a small number of bits, with the discarded part stored for error feedback in a later round [18]. Fourth, it adds random noise whose level is set by the current noise multiplier and the clipping bound. Placing compression before noise means that noise is added only to the coordinates that are transmitted, which reduces the total amount of noise for a given guarantee. Clipping before compression ensures that the sensitivity of the compressed update is bounded by the clipping bound.

D. Adaptive Clipping

The clipping bound is not fixed but follows the data. After each round, each client reports one privacy-protected bit that tells the server whether its update was smaller than the current bound. The server collects these bits, with noise added to protect them, estimates the share of clients whose updates fall below the bound and moves the bound up or down so that this share approaches a target quantile, typically the median or a somewhat higher value [6]. If most updates are smaller than the bound, the bound shrinks and the noise, which is proportional to the bound, shrinks with it. If most updates exceed the bound, the bound grows so that useful signal is not discarded by excessive clipping. This tracking matters because update sizes in practice fall considerably as training approaches convergence, and a fixed bound set for early rounds would be many times too large late in training.

E. Noise Scheduling Across Rounds

The noise multiplier is also not constant. Early in training, when the model is far from a good solution, updates are large and point in a consistent direction, so that they tolerate considerable noise. Later, the updates are smaller, more varied and more sensitive to noise. The framework therefore begins with a higher noise multiplier and lowers it according to a schedule that is governed by the training signal, namely the decline of the validation loss and the size of the updates. The total privacy budget is fixed in advance, and the schedule is constrained so that the accumulated privacy loss at the end of training does not exceed it. In effect, the budget is spent unevenly, with less spent where the model is insensitive to noise and more where it is sensitive.

The accounting of accumulated loss uses a Rényi accountant that tracks the combined effect of the noise multiplier, the sampling rate and the number of rounds [7], so the schedule can be changed from round to round while the guarantee remains valid.

F. Personalized Privacy Budgets

Clients are assigned to privacy tiers according to the sensitivity of their data and their own preferences. A client with highly sensitive records, such as a hospital with identifiable patient data, receives a smaller budget and therefore more noise than a client with less sensitive data. Each client maintains its own account of cumulative privacy loss, computed from the noise it has added and the number of rounds in which it has participated, and it stops participating once its budget is exhausted. The server weights contributions to offset the unequal noise, giving more weight to updates that carry less noise, in the manner of personalized differential privacy [20], [21]. This design ensures that no client's stated limit is exceeded, which a uniform policy cannot guarantee when clients have different limits. It also allows clients with lax requirements to contribute more accurate updates.

G. Hierarchical Protection

In the hierarchical organization, privacy operates in two stages. At the client, the pipeline described above protects the update against the edge server. At the edge server, the updates of the clients in its pool are aggregated, and the server clips and adds a further, smaller amount of noise to the aggregate before sending it to the cloud, which protects against a curious cloud server and against an observer on the wide-area network. Because the edge aggregate is a sum of many clients, the extra noise needed for the cloud-level guarantee is small relative to the signal. Where available, secure aggregation can be applied within each edge pool, so that the edge server sees only the sum of the updates [8], and the client-level noise can then be reduced. The two-stage design divides trust, since neither the edge nor the cloud alone sees an update that is protected only by the other's noise.

H. Evaluation Metrics

We report the following measures. Utility is measured by test accuracy of the global model, and for the clinical task by the area under the receiver operating characteristic curve. The privacy guarantee is reported by the final privacy loss parameter at a fixed small failure probability. Empirical leakage is measured by the success of a membership inference attack, summarized as the area under the curve, where a value of 0.5 means that the attacker does no better than chance and a value of 1 means perfect inference. Communication is measured as the size of the update sent per round and the volume reaching the cloud. Fairness is measured by the accuracy of the worst-performing client, and compliance by the share of clients whose own privacy limit was exceeded. Efficiency is measured by the number of rounds needed to reach a target accuracy and the latency of a round.

5. EXPERIMENTAL SETUP

A. Simulated Network

All experiments were run in simulation with 1,000 clients. In the hierarchical configuration, the clients were divided into 40 pools of 25, each served by one edge server, and the edge servers connected to a single cloud server, as in Fig. 2. In each round, five percent of the clients were sampled to participate. Local training used a few epochs of stochastic gradient descent, and training ran for 200 rounds. The privacy guarantee was computed with a Rényi accountant at a failure probability of one in one hundred thousand, and the main comparisons used a privacy loss parameter of 8, with additional experiments at values from 1 to 16.

B. Tasks and Data

Three tasks were used. The first is classification of handwritten digit images [25]. The second is classification of clothing images, a harder task of similar format [25], which is the main benchmark. The third is a clinical prediction task on tabular patient records distributed across simulated hospitals, evaluated with the area under the curve. Data were partitioned across clients to be non-identical, using a skewed assignment of labels that follows the standard

approach for studying heterogeneity [26], with mild, moderate and severe skew levels. The moderate level was the default. Each client held on average about sixty training examples in the image tasks and a few hundred records in the clinical task.

C. Models and Baselines

The image models were small convolutional networks with about 1.2 million parameters, and the clinical model was a small fully connected network. We compared seven configurations: federated averaging without privacy; fixed-noise differential privacy at a privacy loss parameter of 8 and at 2; differential privacy with adaptive clipping only; the proposed adaptive framework with noise scheduling and personalized budgets; the adaptive framework with compression; and the hierarchical adaptive framework with compression. Compression kept the largest four percent of entries and quantized them to a few bits, which reduced the size of the update by a factor of about 25.

D. Attacks and Evaluation Protocol

Membership inference was evaluated using a calibrated attack of the kind that compares the model's behavior on a target example with its behavior on similar examples that were not used in training [28], and it was run against the final global model, using one thousand member and one thousand non-member examples. Empirical attacks provide a lower bound on leakage and do not replace the formal guarantee [29]. Each experiment was repeated with five random seeds, and differences smaller than the seed-to-seed variation of about 0.4 percentage points in accuracy are not interpreted. Privacy tiers for personalized budgets were assigned randomly, with twenty percent of clients in a strict tier, whose budget was half the default, sixty percent in a standard tier and twenty percent in a relaxed tier, whose budget was double the default.

TABLE I. Experimental Configuration

Component	Setting
Clients	1,000 simulated, 40 pools of 25 in the hierarchical case
Participation per round	5% of clients, 200 rounds
Privacy accounting	Rényi accountant, failure probability of one in 100,000
Main privacy level	Privacy loss parameter of 8, range 1 to 16 studied
Tasks	Handwritten digits, clothing images (main), clinical records
Data skew	Mild, moderate (default) and severe label skew
Compression	Top 4% of entries, quantized, error feedback, about 25 times smaller
Privacy tiers	20% strict, 60% standard, 20% relaxed
Repetitions	Five seeds, means reported

6. RESULTS

A. Overall Comparison

Table II compares the configurations on the clothing-image task. Federated averaging without privacy reached 89.6% accuracy but was the most vulnerable, with a membership inference area under the curve of 0.71. Fixed-noise differential privacy at a privacy loss parameter of 8 cut the attack to 0.54, close to chance, but accuracy fell to 80.9%, and at a privacy loss parameter of 2 it fell to 70.2%. Adaptive clipping alone recovered nearly three points, to 83.8%. The full adaptive framework at the same privacy loss parameter of 8 reached 86.7%, which is 5.8 points above fixed-noise differential privacy and less than three points below the unprotected model, while the attack area under the curve was 0.53. With compression, accuracy was 86.0% and the update shrank from 4.8 megabytes to 0.19 megabytes. The hierarchical variant with compression reached 86.9%.

TABLE II. Comparison of Configurations on the Clothing-Image Task

Configuration	Privacy Loss Parameter	Accuracy (%)	Membership Inference AUC	Update Size per Round (MB)
Federated averaging, no privacy	None	89.6	0.71	4.8
Fixed-noise differential privacy	8	80.9	0.54	4.8
Fixed-noise differential privacy	2	70.2	0.51	4.8
Adaptive clipping only	8	83.8	0.54	4.8
Adaptive framework	8	86.7	0.53	4.8
Adaptive framework with compression	8	86.0	0.53	0.19
Hierarchical adaptive framework with compression	8	86.9	0.53	0.19

The two most important observations are that privacy protection brought attack success close to chance in every protected configuration, so the differences between them are in utility and cost and not in empirical leakage, and that the adaptive design recovered most of the accuracy lost to fixed noise. The empirical attack gives only a lower bound on leakage, so the formal guarantee, not the attack result, is the basis of the privacy claim.

B. Privacy-Utility Trade-Off

Table III shows accuracy over a range of privacy levels. The advantage of the adaptive framework was largest at strong privacy. At a privacy loss parameter of 1 it reached 71.8%, against 62.4% for fixed noise, a gain of 9.4 points. The gain was 8.4 points at 2, 6.1 at 4, 5.8 at 8 and 3.8 at 16. In other words, the adaptive framework with a privacy loss parameter of 2 achieved roughly the accuracy that fixed noise needed a parameter of about 5 to reach, which corresponds to a much stronger guarantee for the same utility. This is the intended effect of tracking the update size and scheduling the noise, since those mechanisms matter most when the noise is large.

TABLE III. Accuracy (%) versus Privacy Loss Parameter on the Clothing-Image Task

Privacy Loss Parameter	Fixed-Noise Differential Privacy	Adaptive Framework	Gain (points)
1	62.4	71.8	9.4
2	70.2	78.6	8.4
4	77.1	83.2	6.1
8	80.9	86.7	5.8
16	84.5	88.3	3.8

C. Results Across Tasks

Table IV shows that the pattern holds across the three tasks at a privacy loss parameter of 8. On the digit task, which is easy, the cost of privacy was small, and the adaptive framework came within one point of the unprotected model. On the clothing task the adaptive framework recovered most of the loss. On the clinical task, where each simulated hospital holds relatively few records, the cost of fixed-noise privacy was highest, with the area under the curve falling from 0.874 to 0.801, and the adaptive framework reached 0.846. Small federations suffer more from privacy noise because there are fewer contributors across which the noise can average out, which is why adaptivity matters most there.

TABLE IV. Results Across Tasks at a Privacy Loss Parameter of 8

Task (Primary Metric)	No Privacy	Fixed-Noise Differential Privacy	Adaptive Framework
Handwritten digits (accuracy, %)	99.0	96.1	98.1
Clothing images (accuracy, %)	89.6	80.9	86.7
Clinical records (area under the curve)	0.874	0.801	0.846

D. Data Heterogeneity

Table V shows the effect of label skew on the clothing task at a privacy loss parameter of 8. Skew reduced accuracy for both approaches, since non-identical data slow convergence and increase the variation between updates. The adaptive framework retained its advantage and widened it as skew increased, with gains of 4.5 points under mild skew, 5.8 under moderate skew and 7.8 under severe skew. A likely reason is that heterogeneous updates vary widely in size, so that a single fixed bound is poorly matched to many of them, while adaptive clipping follows the distribution of sizes.

TABLE V. Accuracy (%) under Different Levels of Label Skew

Label Skew	Fixed-Noise Differential Privacy	Adaptive Framework	Gain (points)
Mild	83.4	87.9	4.5
Moderate	80.9	86.7	5.8
Severe	74.2	82.0	7.8

E. Hierarchical versus Traditional Organization

Table VI compares the traditional organization, with a single cloud parameter server, and the hierarchical organization of Fig. 2, both with the compressed adaptive framework. Accuracy was slightly higher in the hierarchical case, 86.9% against 86.0%, partly because edge aggregation over a pool of similar nearby clients reduced the variance of the updates before the additional noise was applied. Round latency fell from 9.4 seconds to 5.1 seconds because most communication stayed within the edge. The cloud-bound traffic per round fell from 190 megabytes to 7.6 megabytes, a factor of 25, since the cloud received 40 aggregates and not 1,000 individual updates. The hierarchical design also provides two-stage protection, so that a compromised cloud server sees only aggregates that carry their own noise.

TABLE VI. Traditional versus Hierarchical Organization with Compression

Organization	Accuracy (%)	Round Latency (s)	Cloud-Bound Traffic per Round (MB)	Protection Stages
Traditional, cloud parameter server	86.0	9.4	190	One
Hierarchical, edge and cloud	86.9	5.1	7.6	Two

F. Ablation Study

Table VII removes one element at a time from the compressed adaptive framework. Removing adaptive clipping lowered accuracy to 82.5% and the accuracy of the worst client to 72.1%, which makes it the most important single element. Removing the noise schedule reduced accuracy to 83.4%. Replacing personalized budgets with a single uniform budget changed accuracy very little, 85.4%, but caused 18% of clients to exceed the limit that they had stated,

showing that personalization is needed for compliance and not for accuracy. Compressing without error feedback lowered accuracy to 83.9%, which confirms that the bias of compression must be corrected when it is combined with noise.

TABLE VII. Ablation of the Compressed Adaptive Framework

Variant	Accuracy (%)	Worst-Client Accuracy (%)	Clients Exceeding Own Budget (%)
Full framework	86.0	78.3	0
Without adaptive clipping	82.5	72.1	0
Without noise schedule	83.4	73.5	0
Uniform budget instead of personalized	85.4	77.9	18
Compression without error feedback	83.9	75.2	0

G. Convergence and Fairness

The adaptive framework also converged faster. To reach 80% accuracy on the clothing task, it needed about 62 rounds, against about 118 rounds for fixed-noise privacy, which would have meant a much larger privacy cost per unit of progress. The accuracy of the worst client, a measure of fairness, was 78.3% for the full framework, against 70.4% for fixed-noise privacy, so the gains were not obtained at the expense of the clients with the most unusual data. Clients in the strict privacy tier, which received the most noise, had somewhat lower local accuracy than those in the relaxed tier, by about two points, which is the expected consequence of a stronger guarantee.

7. DISCUSSION

The results support the view that a fixed privacy mechanism is inefficient for federated learning and that adaptation recovers much of the lost utility. The gain did not come from weakening the guarantee, since every configuration was compared at the same privacy loss parameter, but from spending the budget more intelligently. Tracking the size of updates reduced noise that was proportional to an oversized bound, scheduling concentrated the privacy budget where the model was least sensitive to noise and personalization matched the protection to what each client required. The ablation indicates that adaptive clipping is the most valuable single component, in agreement with earlier findings [6], while the remaining components add smaller but real gains.

The finding that the benefit was largest when privacy was strongest, when data were most heterogeneous and when federations were small is encouraging, since those are the conditions of practical interest. Hospitals and other institutions with strict requirements typically operate in precisely this regime. In large populations of phones, where the noise is spread across thousands of clients, fixed-noise methods already perform acceptably, and the benefit of adaptation is smaller. The framework is therefore most valuable for settings in which privacy is expensive.

The communication and hierarchy results show that privacy need not come at a high cost in bandwidth. Compression by a factor of 25 cost less than one point of accuracy when it was combined with error feedback, and adding noise only to the transmitted entries kept the total noise lower than it would have been for the full update. The hierarchical organization reduced cloud-bound traffic by a similar factor and halved latency, and its two-stage protection spreads trust among levels. In deployments in which edge servers belong to different operators from the cloud, this property could be as important as the savings in traffic.

Several limitations should be noted. The study is based on simulation, with simulated clients and a modest model size, and real networks involve stragglers, dropouts and heterogeneous hardware that may change the results. The formal guarantee depends on correct

implementation of clipping, noise generation and accounting on every client, and on the assumption that clients apply the mechanism honestly, which is not enforced by the framework. Empirical attacks measure only some forms of leakage, and a low attack success does not prove privacy. Adaptive clipping and scheduling consume a small part of the privacy budget themselves, which we account for but which becomes more significant at very small budgets. Personalized budgets require clients to state their needs honestly, and assigning tiers raises policy questions about fairness, since clients with strict budgets contribute less accurate updates and may therefore benefit less from the global model. The experiments covered image and tabular tasks, and results may differ for text or very large models. Finally, privacy of the learned model does not address other threats, such as poisoning of the training process.

These findings suggest practical guidance. Practitioners should track update sizes and adapt clipping rather than fixing a bound, should plan the privacy budget over the whole training run and not spend it uniformly, and should maintain per-client accounting whenever clients have different requirements. Compression should be combined with error feedback and applied before noise. Where a three-layer network is available, edge aggregation should be used both to cut traffic and to add a second layer of protection, and secure aggregation should be used where possible to reduce the noise that each client must add.

8. CONCLUSION

This paper presented an adaptive differential privacy framework for federated learning in distributed networks. The framework adapts the clipping bound to the observed size of updates, schedules noise over training rounds, assigns personalized privacy budgets with per-client accounting, calibrates noise together with update compression and supports a hierarchical client, edge and cloud organization with two stages of protection. In simulation with 1,000 clients, the framework reached 86.7% accuracy on a clothing-image task at a privacy loss parameter of 8, against 80.9% for fixed-noise differential privacy, and its advantage rose to more than nine points at the strongest privacy level and to nearly eight points under severe data heterogeneity. Membership inference attack success fell from 0.71 to 0.53, compression reduced communication per round by a factor of 25 at a cost of less than one point of accuracy, and the hierarchical organization reduced cloud-bound traffic by the same factor while halving round latency. An ablation showed that adaptive clipping was the most valuable single element and that personalized budgets were necessary for compliance with clients' stated limits.

Future work will proceed in four directions. The first is to deploy the framework on real networks with stragglers and dropouts and on larger models, including language models. The second is to combine it with secure aggregation and robust aggregation, so that privacy and resistance to malicious updates are addressed together. The third is to learn the noise schedule and the clipping target automatically, for example by treating them as a control problem with the privacy budget as a constraint. The fourth is to study the fairness implications of personalized privacy and to design mechanisms that compensate clients who accept stronger protection.

References

1. Pawan Kumar Singh, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5713>
2. Pawan Kumar Singh, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5712>
3. A. Mahendra Vardhan and S. Sridhar, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: <https://doi.org/10.1109/ICAC3N56670.2022.10074458>

4. Ronakkumar shah , Cognitive Process Intelligence: Integrating Generative AI and Automated Workflow Discovery, Vol. 33 No. 08 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5676>
5. Ronakkumar shah, research Trends in Smart Workflow Automation: A Bibliometric Analysis of Emerging Themes, Home / Archives / Vol. 33 No. 05 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5675>
6. Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI <https://doi.org/10.30574/ijusra.2022.5.2.0077>
7. Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: <https://doi.org/10.30574/ijusra.2021.2.1.0047>
8. Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: <https://doi.org/10.32628/CSEIT1936413>
9. Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: <https://doi.org/10.32628/CSEIT192142>
10. Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI: <https://doi.org/10.32996/jcsts.2020.2.2.4>
11. A. Srivastava, "Securing PHI in cloud-based healthcare systems: Challenges, frameworks, and future directions," J. Comput. Anal. Appl., vol. 33, no. 2, Aug. 2024. [Online]. Available: <https://www.eudoxuspress.com/index.php/pub/article/view/4349/3190>
12. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: <https://ijcnis.org/index.php/ijcnis/article/view/8682>
13. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4347/3189>
14. A. Srivastava, "A unified framework for secure cloud data management: Integrating governance, metadata intelligence, and privacy controls," J. Inf. Syst. Eng. Manage., vol. 9, no. 2, Apr. 2024. [Online]. Available: https://www.jisem-journal.com/download/45_A_Unified_Framework.pdf
15. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4603>
16. Abdul Aziz. "Intelligent Health Analytics: Mastering AI in Healthcare Data". Acta Scientiae, vol. 25, no. 4, Aug. 2024, pp. 544-55, <https://www.periodicos.ulbra.org/index.php/acta/article/view/687>. <https://doi.org/10.5281/zenodo.21699886>
17. Event-Aware Multi-Layer Storage Risk Forecasting for Oracle Database Estates Using HAPF, Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 10(4), 2024 ,10.22399/ijcesen.5183
18. Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, 10.52710/cfs.1031
19. Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,10.22399/ijcesen.5382
20. An Automated Risk Scoring Framework for SQL Execution Plan Analysis and Performance Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep),Publication year 2021, Volume 2021, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/970>
21. Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/969>

22. Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS), Publication year 2023, Volume 2023, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/971>
23. Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, <https://jcreview.com/archives/volume-6/issue-1/13380>
24. Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5053>
25. Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5054>
26. Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. Power System Protection and Control, 52(1), 38–45. <https://pspac.info/index.php/dlbh/article/view/342>
27. Venumadhav Vavilala, Shankar Balla (2024, May). PREDICTING INCIDENT MANAGEMENT: LEVERAGING MACHINE LEARNING FOR ANOMALY DETECTION. Power System Protection and Control Scopus Q1 Journal. PSPC. <https://pspac.info/index.php/dlbh/article/view/270>
28. Manikantha Varaprasad Inakollu, (2024, May), FINOPS-Driven Cloud optimization models for enterprise applications, Vol. 52 No. 2 (2024): April-June 2024, 99-110, Power System Protection and Control, ISSN-1674-3415, URL: <https://pspac.info/index.php/dlbh/article/view/283>, DOI: <https://doi.org/10.46121/pspc.52.2.10>
29. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. <https://doi.org/10.5281/zenodo.13325989>
30. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. <https://dx.doi.org/10.21275/SR24724151733>, <https://www.ijsr.net/getabstract.php?paperid=SR24724151733>
31. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. <https://doi.org/10.5281/zenodo.13348695>
32. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. <https://doi.org/10.5281/zenodo.13325845>
33. Jayanth Para, 6G Internet Technology Cyber Threat Notification & Alert System, Vol. 52 No. 4 (2024): October-December 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/170>, DOI: <https://doi.org/10.46121/pspc.52.4.9>
34. Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/171>, DOI: <https://doi.org/10.46121/pspc.51.4.3>
35. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/294>
36. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/364>
37. Kaleshwar Aryasomayajula, Micro services: "A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): April-June 2023, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/374>

38. Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/359>
39. Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/362>
40. Vikas Suresh Bagora, SECURE EMBEDDED NETWORKING ARCHITECTURES USING TRUSTED EXECUTION ENVIRONMENTS IN BROADBAND GATEWAYS, Vol. 52 No. 2 (2024): April-June 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/363>
41. Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices, DOI: <https://doi.org/10.63282/3050-9246.IJETCSIT-V11I4P104>, Vol. 1 No. 4 (2020), <https://ijetsit.org/index.php/ijetsit/article/view/108>
42. Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI: <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105>, <https://ijaibdcms.org/index.php/ijaibdcms/article/view/78>
43. Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA, <https://eudoxuspress.com/index.php/pub/article/view/4118>
44. Rakshitha Kiran P., Samitha Khaiyum, Aparna R. Palandye, Apoorva S. D., Arjun R., Akshaya S. A., Annapoorna S., Leveraging LLaMA3 and LangChain for Rapid AI Application Development, Vol. 20 No. 10s (2024), <https://doi.org/10.52783/jes.5539>, <https://journal.esrgroups.org/jes/article/view/5539>
45. Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews, Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, December 31, 2020, Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, <https://ijrps.com/v11si4/3683>
46. Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu, Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , November 18, 2020, Vol. 11, Issue 1, Pages 1437–1446
DOI: 10.26452/ijrps.v11iSPL1.3686, <https://ijrps.com/v11si1/2911>
47. Ankit Pathak, Leveraging AI for Better Data Quality and Insights, Volume 7 Issue 3 (2025), <https://doi.org/10.32996/jcsts.2025.7.2.19>, <https://al-kindipublisher.com/index.php/jcsts/article/view/9345>
48. Uma Rajeswari Batchu, Bharati Reddi, Joshna Rani Surapaneni, Aruna Jangam, Sunil Misra, Anthony Addlagatta & Prakasham Reddy Shetty, Inhibition of Xanthine Oxidase by 1-O-Methyl Chrysophanol, A Hydroxyanthraquinone Isolated from Amycolatopsis Thermoflava ICTA 103, Published: 13 December 2025, Volume 66, pages 1049–1068 (2026)
<https://link.springer.com/article/10.1007/s12088-025-01534-4>