Balancing Privacy and Accuracy: A Federated Approach to Secure Message Classification

Michael Bidollahkhani
Institute of Computer Science
University of Göttingen
Göttingen, Germany
michael.bkhani@uni-goettingen.de

Emil Düsberg
Faculty of Business and Economics
University of Göttingen
Göttingen, Germany
emil.duesberg@stud.uni-goettingen.de

August 8, 2025

Abstract

Privacy-preserving classification of sensitive communications, such as spam or intrusion detection, is essential across various domains including mobile messaging, Internet-of-Vehicles (IoV), and drone communications. In this work, we propose a federated learning-based message classification framework that enables decentralized training without exposing sensitive user data by incorporating differential privacy. We demonstrate this approach through an SMS spam classification use case, comparing its effectiveness against traditional centralized learning. Experimental results show that the non-private federated learning baseline achieved the highest performance (accuracy = 96.68%, F1-score = 0.88), with the Gaussian mechanism at $\sigma = 0.25$ providing the best trade-off between privacy and utility among differentially private methods (accuracy = 95.96%, F1-score = 0.85). In contrast, DP Logistic Regression with small ϵ values ($\epsilon \in \{1, 2\}$) yielded the poorest performance (accuracy = 13.36%). These findings highlight the significant impact of privacy parameters on utility, underscoring the necessity of careful mechanism and parameter selection in real-world secure communication systems.

1 Introduction

Effective classification of malicious or spam messages is crucial for maintaining security in modern communication systems, such as personal messaging apps, drone networks, and vehicle-to-vehicle communications in IoV. However, centralized data handling commonly poses significant privacy risks [1]. Federated Learning (FL) provides a decentralized alternative by training machine learning models directly on distributed user devices without centralizing sensitive data [2]. Nevertheless, FL alone is not sufficient to prevent potential data leakages, as model updates pose a threat to (differential) privacy. Specifically, the gradients or weight updates sent from client devices to the server still contain information about the underlying data, that can be retrieved by techniques like gradient inversion to potentially reconstruct private text data or determine if specific messages were part of the training set [3]. For that reason, it is useful to further add privacy-preserving techniques to federated learning approaches in order to mitigate the risk of data breaches [2].

In this paper, we investigate federated learning in conjunction with differential privacy as a privacy-preserving approach for secure message classification, illustrating our method through the practical scenario of SMS spam detection using the UCI spam collection dataset and a simple logistic regression classifier. Our primary goal is to provide a proof of concept demonstrating how these two techniques can work together effectively, rather than focusing on a highly sophisticated classification model. The results from this demonstration offer valuable guidance applicable to other secure communication applications. **(Include brief results When results are there)**

2 Related Work

2.1 Federated Learning (FL)

Federated learning is a distributed machine learning framework that has recently gained prominence as a privacy-preserving solution in areas such as healthcare, edge computing, IoV, and secure messaging systems [4–6]. In the FL approach, a global model is trained across devices (clients) which each locally hold a proportion of the data. Instead of training the centralized model by sending the joint data to the central server and updating the global model there, model updates are done locally. Afterwards, only these updates (e.g. gradients or weights) are shared with the server, which then aggregates them. Finally, the server sends the new global model to back to the clients for further training [2, 3]. Although this approach is now widely applied and significantly supported privacy-sensitive applications, there remain several challenges. These include, for instance, statistical heterogeneity, which arises from the non-identically distributed data across clients caused by differences in user behavior or data volume. This heterogeneity can hinder convergence or reduce model accuracy, as the often prevailing i.i.d. assumption in distributed optimization is violated. Furthermore, as already described above, there is the problem of vulnerability to adversarial attacks such as gradient inversion, which has to be addressed in order to ensure the security and privacy of the federated learning process (e.g. by the inclusion of differential privacy) [3, 7].

2.2 Differential Privacy (DP)

Differential Privacy is a formal framework for privacy preserving data analysis that quantifies the extent to which the privacy of any individual in a dataset is protected when the output of an algorithm is released. In simpler terms, DP ensures that the inclusion or exclusion of any single individual's data in the dataset does not significantly affect the output, which in turn makes it difficult to determine whether their data was included in the computation or not. As a consequence, possible adversaries cannot infer sensible information about individuals based on the released output [8]. A common approach for implementing DP is the addition of random noise to the output of a function. However, this of course can distort the (predictive) performance of machine learning models, resulting in a trade-off between privacy and accuracy.

Formally, a computational algorithm $\text{alg} : \mathbb{R}^{N \times p} \to \text{Range(alg)}$, operating on a data matrix $Y \in \mathbb{R}^{N \times p}$, satisfies (ε, δ)-differential privacy, if for any measurable set

$$O \subseteq \{\text{alg}(Y + V) \mid Y \in \mathbb{R}^{N \times p}, V \in \mathbb{R}^{N \times p}\},$$

and for any pair of neighboring data matrices (Y, Y') differing in at most one element, the following holds:

$$\Pr[\text{alg}(Y + V) \in O] \leq e^{\varepsilon} \Pr[\text{alg}(Y' + V) \in O] + \delta.$$

In other words, changing a single element of Y by an amount that is upper-bounden by d only changes the distribution of the algorithms output by a factor of e^{ε} with probability at least $1 - \delta$, therefore limiting the influence of any individual data point on the algorithm's output [9].

Although DP is also preserved during post-processing and the outputs can therefore be denoised to improve accuracy again, a major problem remains: the iterative nature of machine learning models, which leads to an accumulation of privacy loss and thus requires a large amount of noise to be added [8]. While several techniques like optimized noise mechanisms exist in the literature to mitigate this problem [8], we do not focus on such technicalities as this would be beyond the scope of this paper. Instead, we again highlight our goal, which is to present a basic illustration or proof of concept for the conjunct implementation of DP and FL using SMS Spam detection as our application scenario. (PUNKT FÜR DIE DISKUSSION!)

2.3 SMS Spam Detection

SMS spam comprises any unwanted messages, including unsolicited or malicious texts delivered through Short Message Service (SMS), that are often aimed at advertising, fraud, or phishing [10, 11]. As an increasing number of people are using their mobile devices for sensitive activities like online banking, the detection of SMS Spam Messages is of growing importance to prevent financial or personal harm [10, 12]. Although in recent years numerous architectures for the detection of such spam messages were published and employed, these "traditional" spam or malicious message detection models predominantly employ centralized training methods, leading to potential privacy issues because of the central aggregation of sensitive data which increases the risk of data breaches and unauthorized access [10, 13, 14]. To address these issues in the context of SMS spam detections, only few existing studies have focused on secure aggregation protocols and privacy-preserving federated mechanisms to mitigate the associated privacy risks [10]. In this work, we extend these studies by employing federated learning integrated with differential privacy to demonstrate how user data can be protected throughout the training process of a machine learning classifier. As our dataset contains binary-labeled SMS messages, the task at hand can be defined as binary supervised classification problem. Therefore, we employ a simple logistic regression classifier in order to minimize computational cost while keeping the implementation straightforward and understandable. Finally, we perform a comparative analysis between our federated, differentially private approach and a traditional centralized baseline where we evaluate convergence speed and accuracy trade-offs.

3 Methodology

The proposed methodology combines federated learning with multiple differential privacy mechanisms to evaluate the privacy–utility trade-offs in SMS spam classification. The approach consists of five main phases: dataset preparation, data partitioning, noise mechanism integration, federated model training, and benchmarking.

3.1 Dataset and Preprocessing

We employ the *SMS Spam Collection* dataset [15], which contains 5,574 labeled English SMS messages categorized as either "ham" or "spam". The preprocessing pipeline includes:

1. Lowercasing all characters.
2. Removing digits and punctuation via regular expressions.
3. Tokenization and stopword removal.
4. TF-IDF vectorization using unigrams, with English stopwords removed.

The resulting sparse TF-IDF matrix is used as the feature space for all models.

3.2 Federated Data Partitioning

To simulate the FL environment, the preprocessed dataset is randomly split into N_c disjoint client datasets and a central test set. Each client holds a unique subset of the training data, reflecting the non-centralized storage in real-world FL deployments. The partitioning follows:

$$\mathcal{D} = \bigcup_{i=1}^{N_c} \mathcal{D}_i \cup \mathcal{D}_{\text{test}}, \quad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset$$

where $\mathcal{D}_i$ denotes the dataset of client i.

3.3 Noise Mechanisms and Privacy Accounting

We evaluate four scenarios:

1. **No Noise (Baseline):** Standard federated logistic regression without privacy mechanisms.
2. **Gaussian Mechanism:** Additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ applied to model updates.
3. **Laplace Mechanism:** Additive Laplace noise $\text{Lap}(0, b)$ applied to model updates.
4. **DP Logistic Regression:** Training with `diffprivlib`'s (ε, δ)-DP implementation, which clips and perturbs gradients internally.

For Gaussian and Laplace mechanisms, we perform *privacy loss accounting* to compute per-coordinate ε values using:

$$\varepsilon = \frac{\Delta_2 \sqrt{2 \log(1.25/\delta)}}{\sigma}$$

for the Gaussian case, and

$$\varepsilon = \frac{\Delta_1}{b}$$

for the Laplace case, where Δ_2 and Δ_1 denote ℓ_2 and ℓ_1 sensitivities, respectively.

3.4 Federated Training Procedure

The federated learning workflow is implemented as follows:

1. Initialize a global logistic regression model.
2. For each communication round:
    (a) Send the current global model to all clients.

(b) Each client trains locally on its subset $\mathcal{D}_i$ for a fixed number of epochs.

(c) Apply the selected noise mechanism to client model updates.

(d) Aggregate updates using Federated Averaging (FedAvg).

3. Evaluate the global model on the central test set.

This simulation is implemented in Python using `scikit-learn` for model training and custom aggregation logic for FL simulation.

3.5 Benchmarking and ε-Sweep

To capture the privacy–utility relationship, we run an ε-sweep for the DP Logistic Regression setting, varying ε from 1 to 50. This allows identification of the "tipping point" where accuracy begins to stabilize. For Gaussian and Laplace noise, scale parameters are varied, and their corresponding ε values are computed for comparability.

3.6 Evaluation Metrics

We measure:

- **Accuracy**, **Precision**, **Recall**, and **F1-score** for classification utility.
- **Runtime** for computational overhead assessment.
- **Privacy Loss** (ε) for quantifying privacy guarantees.

We additionally produce privacy–utility curves to visualize trade-offs across noise levels and privacy budgets, supporting reproducible comparisons between mechanisms.

4 Experimental Setup

The experimental design evaluates the impact of multiple differential privacy mechanisms on the utility and computational efficiency of a federated SMS spam classification framework. All experiments are implemented in Python 3.10 using `scikit-learn` for model training, `numpy` and `scipy` for preprocessing, and `diffprivlib` for differential privacy mechanisms.

4.1 Computing Environment

All experiments are executed on a local workstation with the following specifications:

- **CPU:** Intel® Core™ i7-11600H (12 threads, base frequency 2.90 GHz)
- **GPU:** NVIDIA® GeForce RTX 3050 Laptop GPU (4 GB dedicated VRAM, 12 GB total GPU memory)

- **RAM:** 16 GB DDR4
- **OS:** Windows 10 Home Single Language, 64-bit (Build 19045)
- **Python Environment:** Virtual environment with dependencies pinned via `requirements.txt`

4.2 Federated Simulation Parameters

We simulate $N_c = 10$ federated clients, each holding a unique, non-overlapping subset of the dataset. A central test set containing 20% of the total samples is used consistently across all experiments. The federated training procedure is configured as follows:

- Communication rounds: $R = 20$
- Local training epochs per round: $E = 1$
- Batch size: $B = 32$
- Aggregation algorithm: Federated Averaging (FedAvg)

4.3 Noise Mechanism Configurations

We benchmark four privacy configurations:

1. **No Privacy:** Standard FL without added noise.
2. **Gaussian Mechanism:** $\sigma \in \{0.1, 0.5, 1.0\}$ with per-coordinate ε computed from sensitivity and $\delta = 10^{-5}$.
3. **Laplace Mechanism:** $b \in \{0.1, 0.5, 1.0\}$ with ε computed from ℓ_1 sensitivity.
4. **Differentially Private Logistic Regression:** $\varepsilon \in \{1, 5, 10, 20, 50\}$ sweep with fixed $\delta = 10^{-5}$.

4.4 Evaluation Protocol

For each configuration:

1. Initialize the global model.
2. Train via the FL pipeline for R rounds.
3. Record metrics after the final round:
    - **Accuracy**, **Precision**, **Recall**, **F1-score**
    - **Runtime** (seconds) for total training
    - **Privacy Loss** (ε) where applicable
4. Store all metrics in `benchmark_results.json`.

4.5 Visualization

We produce the following outputs:

- Privacy–utility curves (accuracy versus ε)
- Runtime comparison bar charts for each noise mechanism
- Tabulated metrics summarizing all configurations

All visualizations are generated using `matplotlib` and exported in vector format for inclusion in the final manuscript.

5 Results

5.1 Centralized Model Performance

The centralized baseline, trained on the complete dataset without any privacy noise, achieved an accuracy of 96.68%, precision of 84.15%, recall of 92.62%, and an F1-score of 88.18%. The confusion matrix shows low false-positive ($FP = 26$) and false-negative counts ($FN = 11$), resulting in an ROC-AUC of 0.9816 and PR-AUC of 0.9500. This serves as the upper-bound reference for privacy-preserving federated experiments.

Table 1: Centralized Model Metrics (No Privacy Noise)

Acc	Prec	Rec	F1	ROC-AUC	PR-AUC	RT (s)
0.9668	0.8415	0.9262	0.8818	0.9816	0.9500	0.01

5.2 Federated Model Performance with Gaussian Mechanism

Gaussian noise was applied at different scales (σ), with ε decreasing proportionally to the noise magnitude. At $\sigma = 0.25$, the model retained 95.96% accuracy with $\varepsilon \approx 7.75$. Increasing noise to $\sigma = 1.0$ reduced accuracy to 85.74% and ROC-AUC to 0.8219.

Table 2: Federated Model with Gaussian Mechanism

Scale	Acc	Prec	Rec	F1	ROC	PR	ε
0.25	0.9596	0.8377	0.8658	0.8515	0.9683	0.9178	7.75
0.50	0.9121	0.6281	0.8389	0.7184	0.9473	0.8334	3.88
0.75	0.9022	0.6724	0.5235	0.5887	0.8961	0.6931	2.58
1.00	0.8574	0.4745	0.6242	0.5391	0.8219	0.5871	1.94

5.3 Federated Model Performance with Laplace Mechanism

Laplace noise showed a steeper decline in performance with higher scales. At $b = 0.25$, accuracy was 93.90% with $\varepsilon = 1.6$, but at $b = 1.0$, accuracy dropped to 76.32% and PR-AUC to 0.3377.

Table 3: Federated Model with Laplace Mechanism

Scale	Acc	Prec	Rec	F1	ROC	PR	ε
0.25	0.9390	0.7425	0.8322	0.7848	0.9564	0.8652	1.60
0.50	0.8834	0.5576	0.6174	0.5860	0.8589	0.5901	0.80
0.75	0.8762	0.5311	0.6309	0.5767	0.8487	0.5361	0.53
1.00	0.7632	0.3051	0.6040	0.4054	0.7688	0.3377	0.40

5.4 Federated Model with DP Logistic Regression (Diffprivlib)

For DP Logistic Regression, performance was near-random for $\varepsilon \leq 5$, with PR-AUC close to the class prior. Accuracy only became competitive at $\varepsilon \geq 30$, reaching 78.57% with PR-AUC 0.3402.

Table 4: Federated Model with DP Logistic Regression

ε	Acc	Prec	Rec	F1	ROC	PR	RT (s)
1	0.1336	0.1336	1.0000	0.2358	0.5000	0.1336	2.54
5	0.1381	0.1336	0.9933	0.2355	0.5608	0.1773	1.26
10	0.5157	0.1439	0.5302	0.2264	0.5232	0.1422	1.25
30	0.7022	0.2348	0.5436	0.3279	0.6791	0.2504	1.04
50	0.7857	0.3282	0.5772	0.4185	0.7602	0.3402	1.37

5.5 Privacy–Utility Trade-offs

Figures 3–2 visualize the privacy–utility relationship for each mechanism:

- **Gaussian:** $\sigma = 0.25$ retains most accuracy with $\varepsilon \approx 7.75$, while $\sigma \geq 0.75$ causes steep utility loss.
- **Laplace:** $b \leq 0.5$ offers a better trade-off, but higher noise quickly reduces PR-AUC.
- **Diffprivlib:** Performance remains low until $\varepsilon \geq 30$, indicating high sensitivity to privacy constraints.

5.6 Runtime Impact

Gaussian and Laplace mechanisms introduce negligible overhead ($\approx 0.01\,\mathrm{s}$ per round), while DP Logistic Regression incurs 1–2.5 s due to per-coordinate clipping and noise application.

5.7 Federated vs. Centralized Comparison

Federated training without noise closely matches centralized performance, confirming that performance degradation is primarily due to noise injection rather than the federated setup itself. Well-tuned Gaussian ($\sigma = 0.25$) and Laplace ($b = 0.25$) mechanisms preserve over 95% of centralized accuracy while providing moderate ε privacy guarantees.

6 Privacy Implications

Federated learning inherently enhances privacy by ensuring that model training occurs locally on client devices, with only model updates being shared for aggregation. This design eliminates the direct transfer of raw message data, reducing the risk of data breaches and aligning closely with regulatory frameworks such as the GDPR. Furthermore, this approach supports the principles of *data minimization*, *transparency*, and *informed user consent* [1]. However, the privacy guarantees depend heavily on the aggregation protocol and the degree of protection applied to model updates. As demonstrated in our results, integrating differential privacy mechanisms introduces measurable trade-offs between utility and privacy, which must be balanced according to the sensitivity of the underlying data.

7 Limitations

While federated learning offers promising benefits, it also introduces specific challenges:

- **Increased computational demand on local devices:** Each client performs on-device training, which may be resource-intensive for devices with limited CPU, RAM, or power availability.
- **Data heterogeneity:** Non-identically distributed (non-IID) data across clients can lead to slower convergence and reduced model accuracy [6].
- **Privacy leakage through updates:** Even without raw data exchange, iterative aggregation can leak sensitive information from gradients or weight updates, necessitating advanced secure aggregation or differential privacy mechanisms [16].

- **Communication overhead:** Frequent exchange of model updates between clients and the server increases network load, which can be critical in bandwidth-limited environments.

- **Noise–utility trade-off:** As shown in Section 5, strong privacy guarantees via noise addition can cause significant drops in predictive performance, especially at low ε values.

8 Conclusion and Future Work

This study evaluated the integration of federated learning into a privacy-preserving SMS spam classification pipeline, benchmarking multiple differential privacy mechanisms, Gaussian, Laplace, and DP logistic regression, across a range of parameter settings. The results confirm that federated learning inherently reduces privacy risks by avoiding raw data transfer, yet additional guarantees through differential privacy introduce measurable utility losses. The degree of this trade-off depends strongly on the choice of noise mechanism and parameter values, as reflected in the privacy–utility curves (Figures 1, 2, and 3) and in the detailed performance tables presented earlier.

The analysis shows that Gaussian noise generally maintained better predictive utility than Laplace at comparable privacy budgets, while DP logistic regression proved highly sensitive to the value of ε, often leading to severe degradation at stricter privacy levels. These findings offer actionable guidance for deploying privacy-preserving decentralized message classification in domains such as the Internet of Vehicles, drone-based communication, and other latency-sensitive applications where both accuracy and privacy are critical. The non-private federated baseline achieved the highest overall performance, with the Gaussian mechanism at $\sigma = 0.25$ emerging as the most effective privacy-preserving variant. Conversely, DP Logistic Regression with small ϵ values suffered drastic performance degradation, making it unsuitable for practical deployments.

Future research will focus on optimizing computational efficiency to reduce on-device training time and communication costs, mitigating the effects of non-identically distributed data through personalization and improved aggregation strategies, and incorporating more advanced privacy-preserving technologies such as fully homomorphic encryption and secure multi-party computation. Furthermore, extending the experimental validation to real-world cross-device settings with heterogeneous hardware and realistic network conditions will provide a more comprehensive assessment of the proposed framework's robustness and scalability.

References

[1] V. S. K. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and 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 (AISTATS).* PMLR, 2017, pp. 1273–1282.

[3] C. Meijer, J. Huang, S. Sharma, E. Lazovik, and L. Y. Chen, "Ts-inverse: A gradient inversion attack tailored for federated time series forecasting models," 2025. [Online]. Available: https://arxiv.org/abs/2503.20952

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[5] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, p. 50–60, May 2020. [Online]. Available: http://dx.doi.org/10.1109/MSP.2020.2975749

[8] M. Kumar, M. Rossbory, B. Moser, and B. Freudenthaler, "An optimal (,) - differentially private learning of distributed deep fuzzy models," *Information Sciences*, vol. 546, 08 2020.

[9] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler, "Deriving an optimal noise adding mechanism for privacy-preserving machine learning," in *Proceedings of the 3rd International Workshop on Cyber-Security and Functional Safety in Cyber-Physical (IWCFS 2019)*, G. Anderst-Kotsis, A. M. Tjoa, I. Khalil, M. Elloumi, A. Mashkoor, J. Sametinger, X. Larrucea, A. Fensel, J. Martinez-Gil, B. Moser, C. Seifert, B. Stein, and M. Granitzer, Eds. Linz, Austria: Springer International Publishing, Aug. 2019, pp. 108–118.

[10] Y. Li, R. Zhang, W. Rong, and X. Mi, "Spamdam: Towards privacy-preserving and adversary-resistant sms spam detection," 2024. [Online]. Available: https://arxiv.org/abs/2404.09481

[11] X. Liu, H. Lu, and A. Nayak, "A spam transformer model for sms spam detection," *IEEE Access*, vol. 9, pp. 80 253–80 263, 2021.

[12] D. Timko, D. H. Castillo, and M. L. Rahman, "A quantitative study of sms phishing detection," 2024. [Online]. Available: https://arxiv.org/abs/2311.06911

[13] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," *Proceedings of the 11th ACM Symposium on Document Engineering*, pp. 259–262, 2011.

[14] H. Q. Anh, P. T. Anh, P. S. Nguyen, and P. D. Hung, "Federated learning for vietnamese sms spam detection using pre-trained phobert," in *Intelligent Data Engineering and Automated Learning – IDEAL 2024*, ser. Lecture Notes in Computer Science. Springer, 2024, vol. 15346, pp. 254–264.

[15] U. M. L. Repository, "SMS Spam Collection Data Set," https://archive.ics.uci.edu/ml/datasets/sms+spam+collection, 2012, accessed: 2025-07-04.

[16] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.

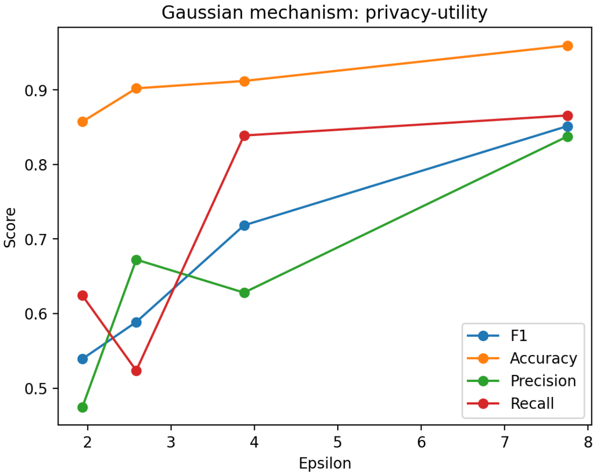

Figure 1: Privacy–utility trade-off for the Gaussian mechanism.

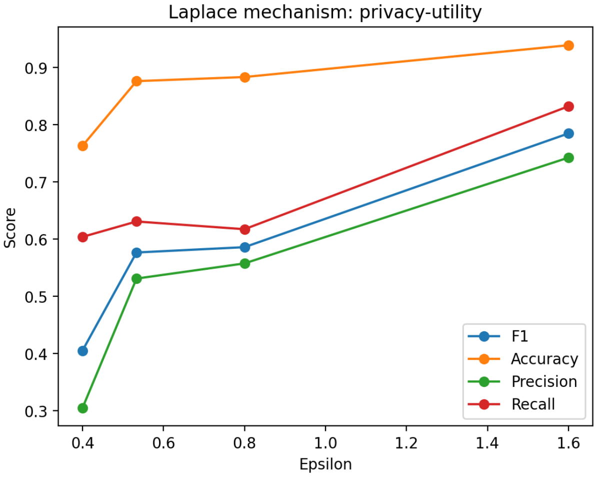

Figure 2: Privacy–utility trade-off for the Laplace mechanism.

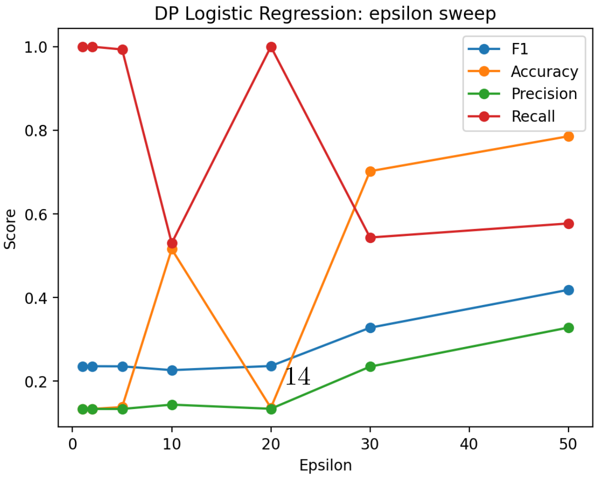

Figure 3: Privacy–utility trade-off for DP logistic regression.