InteFL: Framework for AI-Assisted Design of Trustworthy and Efficient Federated Learning Applications

Dmitrii Korobeinikov*, Arnaldo Barea*, Leon Reznik*
, Raman Zatsarenko*, Sergei Chuprov†, and Angel Peredo†.
*Rochester Institute of Technology, Rochester, New York, USA
†University of Texas Rio Grande Valley, Edinburg, Texas, USA

dk9148@rit.edu, ajb6289@rit.edu, lrvcs@rit.edu, rz4983@rit.edu, sergei.chuprov@utrgv.edu, angel.peredo01@utrgv.edu

***Abstract*—Although Federated Learning (FL) enhances clients' security and privacy by retaining data locally, the decentralized nature of this paradigm exposes it to diverse malicious actions as well as technological and reliability issues that can degrade data quality, hinder convergence, and reduce global model accuracy. Accounting for these destabilizing factors is essential in FL design for its integration in robust and reliable practical applications. To address this need, we introduce *InteFL*, a software tool engineered for the systematic design and evaluation of efficient and secure FL. *InteFL* provides functionality for benchmarking of FL performance under diverse conditions, including static and temporally dynamic adversarial attacks, and supports configuring and fine-tuning of robust aggregation algorithms and their parameters. Built as an extension to existing FL tools, *InteFL* possesses a higher practicality by incorporating intelligent agents that enhance tool usability. We present several use cases demonstrating how our tool supports the search and investigation of more resilient aggregation techniques for diverse attack models, and how parameter fine-tuning accelerates convergence and improves the accuracy of resulting ML models.**

***Index Terms*—Federated learning, benchmarking suite, robustness assessment, adversarial attacks**

TABLE I
COMPARISON OF *InteFL* CAPABILITIES WITH OTHER FL FRAMEWORKS AND TOOLS

Framework	Capabilities[a]									
	Dynamic attacks[b]	Centralized config	Robust agg. evaluation	Experiment scheduling	Config-based dataset	Metrics visualization	AI-agent case study	Optimal config selection	HuggingFace integration[c]	Artifacts UI dashboard
Flower [3]	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
FedML [4]	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
EasyFL [5]	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗
FedEasy [6]	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗
ByzFL [7]	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
FedSecurity [8]	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗
***InteFL* (Ours)**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[a] Available out of the box.
[b] When the attack model and/or its intensity changes over the course of FL execution.

I. INTRODUCTION

Federated Learning (FL) is a decentralized approach to Machine Learning (ML) where the data used for model training remains local at data sources such as sensors, CCTV cameras, or medical equipment. Because FL distributes training across heterogeneous Edge devices, practitioners must account for reliability challenges typical of such deployments. These include degraded data fidelity from faulty sensors, intentional tampering by compromised participants, and variable network quality, all of which can undermine the trained model. Unstable network connectivity or denial-of-service attacks may interrupt communication and lead to intermittent FL client participation or data loss. User errors and intentional manipulation (e.g., data mislabeling) introduce semantic noise, while malfunctioning or outdated sensors or hardware faults degrade signal-to-noise ratio (SNR) in data samples and distort local model updates. In conventional FL with aggregation strategies such as FedAvg [1], the presence of malicious clients in collaborative training further exacerbates these issues, because compromised participants may gain access to global model updates. They can submit poisoned gradients yet continue to receive updated global models, potentially reconstructing sensitive information from benign contributors [2]. Collectively, these effects deteriorate model accuracy, hinder convergence and generalization, ultimately compromising security in FL applications. Designing of dependable FL requires careful consideration of these factors and evaluation of its performance under realistic conditions. While scientific evaluations of FL algorithms often rely on curated and pre-processed benchmarking datasets designed for controlled ML training and testing, such resources rarely reflect data heterogeneity and quality variations typical for real-world systems. Thus, they have limited abilities to research the FL dependability for further integration in real-world systems.

To address this gap, we introduce *InteFL*, a domain-agnostic framework that facilitates the design of dependable applica-

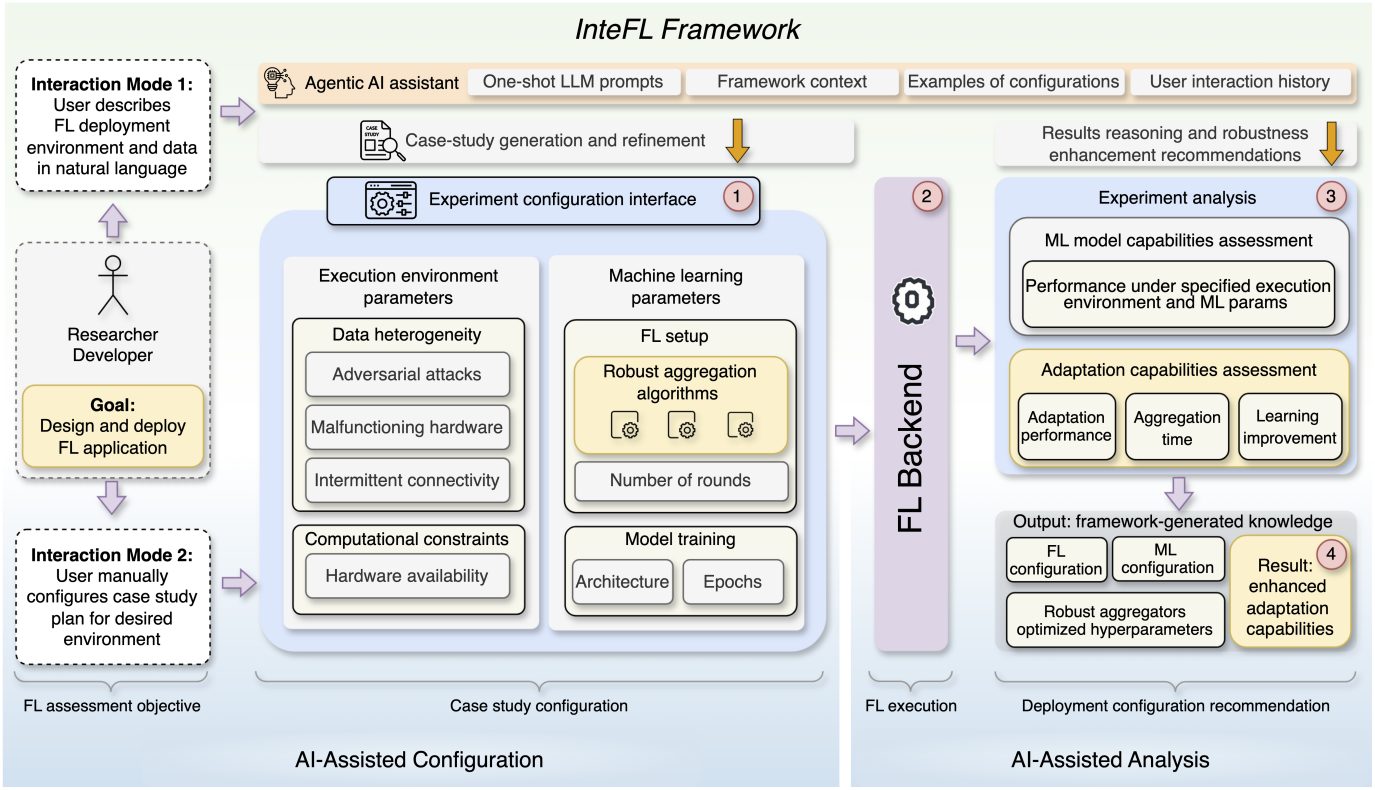

Fig. 1. *InteFL* framework enables the specification of FL execution environment parameters, including possible adversarial attacks, and training data quality variations. Based on the benchmarking results, framework provides the knowledge for FL deployment configuration that yields the highest dependability

tions by providing the FL research platform that simulates a variety of possible FL execution conditions, including dynamically performed adversarial attacks and data quality (DQ) variations. *InteFL* extends existing FL tools, such as Flower [3], that are intended for ML model training in a distributed manner, with modules that both enhance FL accessibility by ML researchers and developers, and introduce the functionality to research and benchmark dependable FL for further integration in real-world systems. *InteFL* includes experiment configuration and feedback modules that help researchers to intuitively specify all parameters of an FL execution, including expected data modalities and its quality variations, anticipated adversarial attacks, and a validation module ensuring correctness and experiment reproducibility. The analysis of case study results is further facilitated with the generated visual snapshots containing all artifacts collected during the execution. Furthermore, our tool integrates an AI agent that assists users in designing, verifying, and refining FL execution configurations through natural-language interaction, reducing the learning curve necessary to operate with the tool. *InteFL* monitors model convergence, anomaly detection performance, and FL training dynamics, stores this information in a per-experiment knowledge memory, and applies reasoning via an LLM agent to generate refined training strategies. This integration facilitates the discovery of FL parameters that yield greater performance within the assessed objective, which ultimately benefits the FL research and development community.

Our contributions are as follows:

(1) We present *InteFL*, a dependability-oriented FL framework that extends existing tools with systematic support for evaluating interactions between attacks, defenses, and deployment constraints. Unlike other frameworks that focus primarily on algorithmic prototyping, *InteFL* targets researchers and practitioners designing trustworthy FL systems, enabling controlled simulation of dynamic adversarial behaviors, temporal DQ variations, heterogeneous client participation, and resource constraints. By implementing these capabilities atop available FL backends, the framework bridges the gap between isolated studies and realistic, deployment-driven evaluation.

(2) We demonstrate how *InteFL* reduces the effort required to plan, configure, and execute complex FL experiments by providing integrated experiment orchestration, attack scheduling, and results visualization. Through representative case studies and evaluation results, we show how the framework supports comparative robustness analysis across attack types and intensities, making it more effective than existing platforms for studying how and under which conditions FL systems fail or remain resilient.

Finally, **(3)** We introduce a context-aware AI agent embedded into the framework that further differentiates *InteFL* from prior FL tools. The agent translates natural-language user goals into validated experiment configurations, analyzes

collected metrics and logs, and provides recommendations for improving FL dependability. This capability lowers the barrier to entry for non-experts and accelerates experimental workflows, enabling a broader class of users to conduct dependable FL research without deep prior expertise.

II. Existing Tools

In FL research, aspects such as adversarial attacks, data modalities, the ML task, the number of participating clients, and the available computational resources influence the security and the performance of resulting ML models. Adversarial attacks can affect both the communication network's quality of service and the quality of client-supplied models. The architecture of the chosen ML model must align with available computational resources; local client data may be noisy or corrupted in the case of outdated or malfunctioning sensors.

To counter these vulnerabilities and challenges, numerous robust aggregation algorithms have been proposed. These algorithms leverage statistical outlier detection or distance-based indicators to identify anomalous client updates [9]–[13]. Although many such algorithms render adequate performance in benchmarking on curated datasets, their behavior depends critically on a set of tunable hyperparameters, e.g., thresholds, clipping constants, statistical quantiles, or weighting factors. They determine trade-offs between robustness and convergence. Because data heterogeneity, model size, and attack patterns differ across domains, no single parameter setting generalizes well.

Several open-source FL platforms now exist, including Flower [3], FedML [4], EasyFL [5], FedEasy [6], ByzFL [7], and FedSecurity [8], each offering programmatic abstractions for orchestrating distributed training. However, some of these tools prioritize algorithm prototyping and scalability benchmarks over systematic security evaluation, leaving practitioners without integrated support for stress-testing deployments against realistic threat models. Others, while providing a comprehensive benchmarking suite, employ complicated configuration and setup that complicates their application in the research community. Flower offers portable simulation across diverse hardware; FedML focuses on production-grade scaling and IoT integration; EasyFL minimizes boilerplate through high-level abstractions; and ByzFL centers on evaluating defenses against Byzantine faults. Despite these strengths, none unifies adversarial testing, configuration validation, and AI-guided analysis into a single reproducible pipeline.

III. Intelligent Framework for Robust Federated Learning

A. AI Agent for Research and Development

Our framework provides interfaces for case study design and execution that are designed to accommodate users with various levels of experience and knowledge of benchmarking configuration specifics. Experienced users may craft experiment specifications by hand, retaining granular control over every parameter while the framework's schema-based validator catches inconsistencies before execution begins. Even if an error is introduced, the validation mechanism prevents the execution of an invalid case study.

Newcomers benefit from a conversational interface: they describe their deployment scenario in plain English, and the system translates this description into a validated JSON configuration via a one-shot LLM prompt augmented with framework documentation and historical examples. Each prompt bundles the framework's API reference, excerpts from the validation schema, a curated set of working configurations, and any past experiments tied to the requesting user, grounding the model's output in project-specific constraints.

Once a case study has been executed, users can analyze the generated plots and collected metrics either manually through the UI dashboard or with AI assistance. When the case study design was produced through the AI agent, *InteFL* performs an additional reasoning step after experiment execution. Upon completion, the agent receives a structured summary of loss curves, exclusion logs, and timing data alongside the user's original objective. It then synthesizes a recommendation, ranking aggregation strategies, flagging parameter sensitivities, and highlighting trade-offs between robustness and throughput.

B. InteFL Execution Stages

InteFL organizes each study into four phases (Figure 1): **(1)** the user specifies a configuration specifying clients, attacks, and aggregation rules; **(2)** the Flower backend executes federated rounds and logs metrics; **(3)** post-hoc analysis quantifies defense efficacy; and **(4)** the system distills actionable deployment guidance from the collected evidence.

The experiment configuration interface (Figure 1, (1)) defines both the execution environment and ML parameters. Users encode anticipated operational stresses, e.g., non-IID partitions, coordinated poisoning, which the framework materializes as concrete perturbations during training. ML parameters encompass FL setup and model training settings. The robust aggregation algorithm and its hyperparameters can be selected based on theoretical recommendations for the given environment, e.g., anticipated anomaly severity. Model architecture and training parameters, such as local client epochs, are linked to environment constraints like available compute. The configuration is then validated by strict logic to prevent misconfiguration. After configuration, model training executes in the FL backend (Figure 1, (2)). *InteFL* uses Flower for distributed training and collects performance metrics including training and validation loss and accuracy. The framework also tracks whether the chosen robust aggregation algorithm successfully excluded anomalous clients at each round. Upon completion, all metrics pass to the analysis stage.

During post-experiment analysis (Figure 1, (3)), *InteFL* evaluates defense quality through exclusion accuracy, which estimates how reliably the aggregator identifies compromised updates, and aggregation time per round. These metrics reveal whether a strategy acceptably balances security and efficiency, while evidence of improved learning further validates robust aggregation. The final phase distills these observations into actionable guidance: recommended aggregator choices,

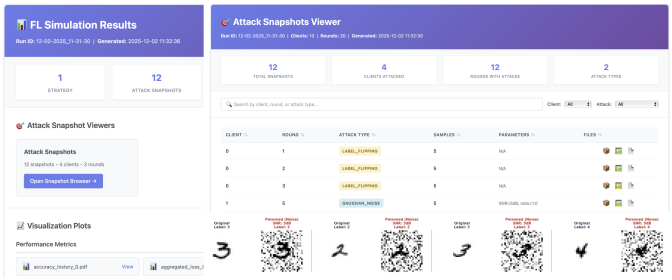

Fig. 2. *InteFL* generates html report for every executed case study. Report includes the JSON configuration for reproducibility, plots, CSV files with collected metrics, and attack snapshots. In Section IV we demonstrate select artifacts and discuss how they help to design FL applications

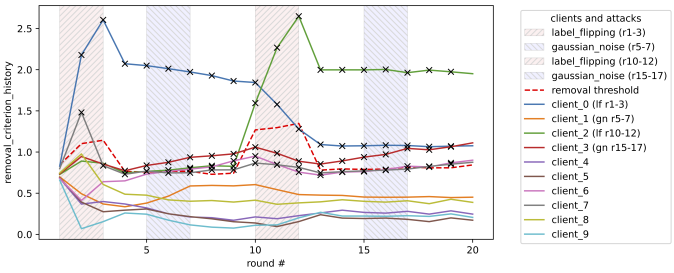

Fig. 3. *InteFL* generates informative plots that enable visual assessment of the FL performance. Here we demonstrate the exact PDF generated by the framework for the dynamics of client removal. Legend specifies the schedule of attacks, where clients 0 to 4 performed label flipping and gaussian noise attacks at select rounds which are also indicated with hatching. Client exclusion marks illustrate which clients were excluded

hyperparameter sensitivity ranges, and trade-off summaries informing production deployment (Figure 1, (4)).

C. UI Dashboard for Experiment Artifacts

InteFL integrates a UI dashboard that summarizes the results of case study execution in an accessible html page that is generated as a separate file for every executed case study. Artifacts of every experiment include plots, CSV files with all collected metrics, JSON configuration of the experiment that allows the reproducibility of the results, and attack snapshots that provide additional insights into how exactly the attacks were performed during the execution. The example of the generated dashboard is demonstrated in Figure 2.

D. Integrated Attack Models

InteFL implements six adversarial attack types organized into two categories: *data poisoning* and *model poisoning*, as described in Table II. Data Poisoning Attacks modify training data before local model updates are computed, and include label flipping, gaussian noise and token replacement. Model poisoning attacks include weight spiking, gradient scaling and byzantine perturbation. These attacks can be applied *statically* throughout FL execution or *dynamically* at specified aggregation rounds, with support for attack stacking where multiple attack types are performed by the same client simultaneously.

IV. *InteFL* IN PRACTICE

In this section, we demonstrate how our framework can be applied to address diverse real-world needs for FL research and development. In Sections IV-A and IV-B we describe real use-cases when *InteFL* was used for FL research and development; in Section IV-C we demonstrate the example of the interaction with our tool through the AI agent, as described in Section III-A.

A. Case Study I: Performance of Robust Aggregations under Dynamic Attacks with Image Classification Benchmarking Dataset

In this case study, we aim to determine the FL dependability under dynamic data poisoning attacks, particularly the ability of the robust aggregations to react to the changing environment, and the impact on the convergence of the global model. We utilize the numerical subset of the FEMNIST benchmarking dataset with IID partition on the image classification task with CNNs. We configure the FL execution to perform dynamic data positioning attacks on select clients. The results generated by *InteFL* showcase the performance of all assessed aggregation strategies along with the impact on the convergence of the resulting ML model. Figure 3 illustrates the dynamics of the MADE-PI [15] robust aggregation strategy threshold and client exclusion.

B. Case Study II: Assessment of Robust Aggregation Optimal Hyperparameters Selection when Operating with GPT-2 LLM on Medical Texts

In this case study, we demonstrate how *InteFL* can be employed to investigate the dependability of robust aggregation such as MADE-PI under the varied intensity of adversarial attack. We assess the GPT-2 [30] token classification model in a federated setting with label flipping attack executed by the varied proportion of participating clients, and track the performance of the aggregated ML model. We perform the label flipping attack in the MedMentions ST21pv dataset by randomly reassigning named entity recognition labels to incorrect tags. The results illustrating the degradation of the performance of the aggregated ML model are demonstrated in Table IV. The results demonstrate two practical use-cases: (a) establishing a baseline for ML model performance, (b) ML model performance improvement when anomalies are not removed (Apply Removal: ✗), and when they are attempted to be removed by the aggregation strategy (Apply Removal: ✓). Results suggest that MADE-PI is most effective when the fraction of malicious clients is small to moderate, where it removes most attackers with minimal collateral damage. As the intensity of attack increases, the defense still helps but cannot render the performance of the resulting ML model similar to baseline without either missing some attackers or sacrificing more benign clients.

C. Case Study III: AI-Assisted Experiment Design

This scenario illustrates how practitioners can leverage *InteFL*'s AI agent to translate high-level deployment goals into concrete, benchmarked configurations. We utilize the

TABLE II
OVERVIEW OF *InteFL* BENCHMARKING FUNCTIONALITIES FOR EVALUATING FL DEPENDABILITY

Aggregation strategies	**ML model performance metrics**	**Robust aggregation metrics**	**Attack models** [a]	**Available datasets**
FedAvg, Trimmed Mean / Median [9], [13], Krum, Multi-Krum [14], Bulyan, RFA [12], Trust & Reputation, MADE-PI [15]	Per-client and aggregated accuracy and loss, classification F1-score, comparison of aggregated metrics across executed strategies	Anomaly detection performance (FP, FN, TP, TN and derivatives), per-client distance to the centroid, aggregation time	*Data poisoning:* label flipping, Gaussian noise with target SNR, token replacement; *Model poisoning:* weight spiking [16], gradient scaling [17], Byzantine perturbation	*Image:* FEMNIST [18], MedMNIST [19], FLAIR [20], ITS [21], NLST [22]; *Text:* MedMentions [23], MedQuad [24], Financial PhraseBank [25], LexGLUE [26], PubMed-MEDLINE [27]

[a] Attacks can be applied either statically during the whole course of FL execution, or dynamically at select aggregation rounds or ranges of aggregation rounds. Combinations of attacks are supported, too.

TABLE III
DEMONSTRATION OF AN INTERACTION WITH *InteFL* THROUGH AI AGENT. WITH A HIGH-LEVEL USER DESCRIPTION, *InteFL* GENERATES A CASE STUDY PLAN, EXECUTES IT IN THE FL BACKEND, COLLECTS METRICS, AND REASONS BASED ON OBSERVED FL DYNAMICS

Case study generation based on natural-language description and results reasoning

Interaction step	**Purpose in *InteFL* workflow**	**User input**	**AI agent reasoning[a] [28]**	***InteFL* config[b]**
Case study request	User provides a natural-language FL scenario description. If information is missing, *InteFL* infers reasonable defaults, possibly, with the provided documentation.	*"We want to deploy FL into a hospital. The facility has older equipment. We plan to collect and label data locally, however, we have medical college students who can mislabel medical samples, so we'd like to mitigate these possible mistakes. We'd like to know which robust aggregation we should use to render the most dependable model."*		
Case study configuration	Framework generates an executable case study configuration (dataset, model, aggregation, clients, rounds, attack / data quality settings).			*Compare Trimmed-Mean, RFA, Multi-Krum, Bulyan, MADE-PI, and Trust&Reputation under persistent label noise (label_flipping) with theoretic thresholds with 30% flip fraction and 20% malicious clients to identify the most robust aggregation for mislabeling scenarios. Use PneumoniaMNIST (part of MedMNIST) dataset, 10 clients and 20 aggregation rounds.*
AI agent reasoning	Agent explains the reasoning for the generated case study configuration, accounting for the initial user request.		*" [...] simulating mislabeling (30% label flips) across 10 clients over 20 rounds and evaluating how well each method maintains accuracy, stability, and sensible client filtering. By running identical simulations that differ only in the aggregation method, we will generate side-by-side performance curves and recommend the most dependable strategy for deployment."*	
Deployment recommendation[c]	After the FL is executed, system interprets experiment metrics and suggests FL configuration that yields highest dependability.	MADE-PI as the primary choice (highest removal F1 at 96%, stable accuracy and small aggregation time on CPU), Bulyan as an alternative with good precision at 80%, avoid Trimmed Mean and RFA due to substantial false-positives that can lead to loss of the clinical training data.		

[a] Each cell illustrates the reasoning stored in the case study configuration file. The generated configuration is a JSON file that includes all ML, data heterogeneity, and FL parameters, as shown in Figure 1.
[b] For readability, we present a natural-language summary of key configuration aspects that best reflect the user request.
[c] Simplified example of interpreting experimental results into a deployment recommendation.

AI agent embedded into our framework to generate case study design based on user-provided description of the target deployment environment. Particular parameters of the study are distilled by underlying LLM based on the framework documentation and user-provided goal. Finally, the framework-collected metrics are analyzed by the AI agent to provide deployment recommendations. The process of the AI-assisted case-study generation and refinement is demonstrated in Table III.

V. CONCLUSION

Building trustworthy FL systems now demands navigating a growing landscape of attack vectors, defensive algorithms, and deployment constraints, yet existing platforms offer little guidance on how these factors interact in practice. In this work, we developed and presented *InteFL*, an accessible framework for research and development of dependable FL applications that addresses these challenges. The tool functionalities such as simulation of dynamic adversarial data and model poisoning attacks, experiment configuration assistance and results visualization enhance the experience when researching and developing FL applications. By layering dependability-focused modules atop commodity FL backends, *InteFL* bridges the gap between controlled algorithm research and production-ready deployment. An embedded AI agent lowers the barrier to entry, guiding newcomers from natural-language requirements to validated experiment configurations without requiring deep FL expertise. The evaluation results illustrate the significant decrease in the time required to plan and execute FL experiments.

ACKNOWLEDGMENT

This research was supported in part by the US National Science Foundation (award 2321652). Also, we want to thank RIT students Tyler Black, Chris Soravilla and Yash Gulab Rathod for their help in planning and conducting the framework empirical study.

REFERENCES

[1] 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*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282.

TABLE IV
DOCUMENT-LEVEL NER PERFORMANCE. (A) ESTABLISHING A BASELINE; (B) PERFORMANCE UNDER ATTACKS WITH MADE-PI

(a) Establishing a baseline

Model	**F1**	**Precision**	**Recall**
TaggerOne [23]	0.548	0.536	0.561
UMLS Low-Resource Model [29]	0.655	0.693	0.621
GPT-2 Federated (Our results)	0.713	0.827	0.627

(b) The impact of various attack intensities on the ML model performance

Num. Poisoned Clients	**Apply Removal**	**F1**	**Prec.**	**Rec.**
1	✗	0.673	0.818	0.571
1	✓	0.709	0.846	0.610
2	✗	0.628	0.821	0.508
2	✓	0.718	0.833	0.631
3	✗	0.566	0.830	0.429
3	✓	0.544	0.855	0.399
4	✗	0.428	0.889	0.281
4	✓	0.463	0.862	0.317
5	✗	0.446	0.891	0.297
5	✓	0.314	0.878	0.191

[2] Z. Wang, J. Lee, and Q. Lei, "Reconstructing training data from model gradient, provably," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, F. Ruiz, J. Dy, and J.-W. van de Meent, Eds., vol. 206. PMLR, 25–27 Apr 2023, pp. 6595–6612.

[3] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. d. Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," no. arXiv:2007.14390, Mar. 2022, accessed: 2026-03-20. [Online]. Available: http://arxiv.org/abs/2007.14390

[4] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, X. Zhu, J. Wang, L. Shen, P. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, and S. Avestimehr, "Fedml: A research library and benchmark for federated machine learning," 2020, accessed: 2026-03-20. [Online]. Available: https://arxiv.org/abs/2007.13518

[5] W. Zhuang, X. Gan, Y. Wen, and S. Zhang, "Easyfl: A low-code federated learning platform for dummies," *IEEE Internet of Things Journal*, vol. 9, no. 15, p. 13740–13754, Aug. 2022.

[6] M. Kundroo, G. Haider, N. Khoa, A. W. Mamond, and T. Kim, "Fedeasy : Federated learning with ease," *SoftwareX*, vol. 31, p. 102276, 2025.

[7] M. González, R. Guerraoui, R. Pinot, G. Rizk, J. Stephan, and F. Taïani, "Byzfl: Research framework for robust federated learning," 2025, accessed: 2026-03-20. [Online]. Available: https://arxiv.org/abs/2505.24802

[8] S. Han, B. Buyukates, Z. Hu, H. Jin, W. Jin, L. Sun, X. Wang, W. Wu, C. Xie, Y. Yao, K. Zhang, Q. Zhang, Y. Zhang, C. Joe-Wong, S. Avestimehr, and C. He, "Fedsecurity: A benchmark for attacks and defenses in federated learning and federated llms," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2024.

[9] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[10] X. Cao, J. Ren, J. Li, H. Zhang, Y. Cheng, and Y. Jia, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2021, pp. 16–28.

[11] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating Sybils in Federated Learning Poisoning," Jul. 2020, accessed: 2026-03-20. [Online]. Available: http://arxiv.org/abs/1808.04866

[12] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust Aggregation for Federated Learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022, accessed: 2026-03-20. [Online]. Available: http://arxiv.org/abs/1912.13445

[13] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[14] E.-M. El-Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 3521–3530.

[15] R. Zatsarenko, S. Chuprov, D. Korobeinikov, L. Reznik, and A. Peña, "Made-pi: Framework for effective anomaly detection in federated learning applications [accepted for publication]," in *International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2025.

[16] Y. Xie, M. Fang, and N. Z. Gong, "Model poisoning attacks to federated learning via multi-round consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 15 454–15 463.

[17] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 7232–7241.

[18] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," 2019, accessed: 2026-03-20. [Online]. Available: https://arxiv.org/abs/1812.01097

[19] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, Jan. 2023.

[20] A. Garioud, N. Gonthier, L. Landrieu, A. D. Wit, M. Valette, M. Poupée, S. Giordano, and B. Wattrelos, "Flair: a country-scale land cover semantic segmentation dataset from multi-source optical imagery," 2023, accessed: 2026-03-20. [Online]. Available: https://arxiv.org/abs/2310.13336

[21] S. Chuprov, L. Reznik, and G. Grigoryan, "Study on network importance for ml end application robustness," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 6627–6632.

[22] P. F. Pinsky, T. R. Church, G. Izmirlian, and B. S. Kramer, "The national lung screening trial: Results stratified by demographics, smoking history, and lung cancer histology," *Cancer*, vol. 119, no. 22, p. 3976–3983, Nov. 2013.

[23] S. Mohan and D. Li, "Medmentions: A large biomedical corpus annotated with umls concepts," 2019, accessed: 2026-03-20. [Online]. Available: https://arxiv.org/abs/1902.09476

[24] A. Ben Abacha and D. Demner-Fushman, "A question-entailment approach to question answering," *BMC Bioinform.*, vol. 20, no. 1, pp. 511:1–511:23, 2019.

[25] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.

[26] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras, "Lexglue: A benchmark dataset for legal language understanding in english," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022, pp. 4310–4330.

[27] C. Zakka, "Pubmed medline abstracts dataset," HuggingFace Datasets, 2023, accessed: 2026-03-20. [Online]. Available: https://huggingface.co/datasets/cyrilzakka/pubmed-medline

[28] OpenAI, "Openai api response," Generated via OpenAI API, 2025, model: GPT-5. [Online]. Available: https://openai.com/api/

[29] S. Mohan, R. Angell, N. Monath, and A. McCallum, "Low resource recognition and linking of biomedical concepts from a large ontology," in *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ser. BCB '21. ACM, Aug. 2021, p. 1–10.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.