IntelliMAD: Domain-Agnostic Framework for Model Anomaly Detection Algorithms Benchmarking and Fine-Tuning in Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Federated Learning (FL) has been increasingly adopted in a wide variety of industrial applications, creating a strong need for a development of novel tools to assist in transitioning from a model's purely theoretical investigation to FL introduction into practice. However, before FL can be fully integrated into practical applications, further research is essential to develop methodologies for training more robust and effective models, particularly those capable of performing reliably in real-world environments characterized by imperfect data – a task that requires professionals and researchers to utilize comprehensive frameworks for conducting these investigations. While existing FL frameworks offer platforms for model training in a federated manner, they have been primarily developed for Machine Learning (ML) model training on simplified and curated datasets, which narrows down the scope of their application to verification of the unrealistic setups without considering the possibility of model anomalies. The transition to real-world deployments possesses a need for a new approach to the evaluation of Model Anomaly Detection (MAD) capabilities within federated setting. Hence, we propose to extend existing tools with the *IntelliMAD*, an accessible, generic domain-agnostic framework for MAD algorithms benchmarking under various execution conditions. In addition, our tool incorporates the evaluation of FL robustness when operating with data quality variations that are introduced into the clients' datasets in a runtime, bringing experimental verification closer to real-world deployments. *IntelliMAD* framework further facilitates integration of MAD into real-world applications, allowing batch experiment configuration, verification and planning, FL performance visualization, and calculation of composite robustness scores. The novel functionality of our framework allows for a discovery of MAD parameters required to adjust the aggregation algorithm for particular deployments to achieve the desired level of robustness of the entire FL setup. We demonstrate the feasibility of our approach to the evaluation of MAD robustness and present findings from benchmarking several widely used aggregation algorithms in the image processing domain. We further investigate possible use cases for discovering suitable MAD algorithm parameters.

1 Introduction

Federated Learning (FL), introduced by Google in 2016 (9), has since been increasingly adopted in a wide variety of industrial applications, shifting from entirely theoretical developments to real-world deployments. Application domains for FL range from civil implementations for traffic and

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

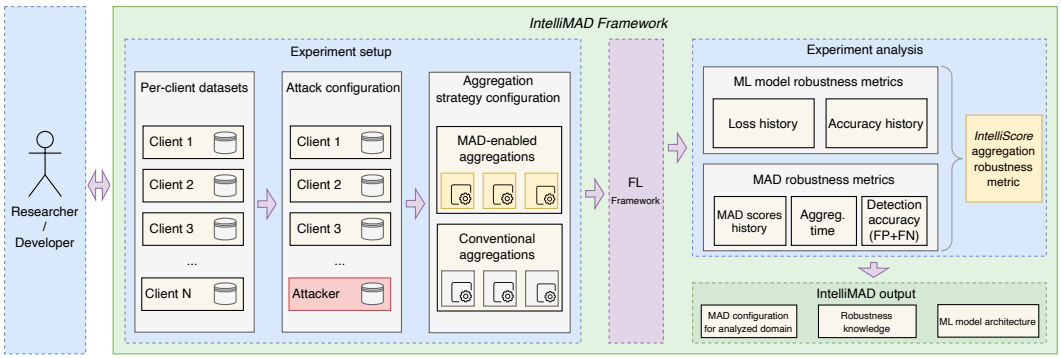

Figure 1: Logical View of IntelliMAD Framework

transportation systems (7; 4), medicine, social media, to military systems (6). This shift possesses a need for a new standard in FL benchmarking, as domains with more strict security regulations and data compliance policies, such as healthcare or finance, may require the development of applications and systems that adhere to these security standards and regulations, including requirements for the implementation of attack mitigation techniques and defense mechanisms that are embedded in the FL architecture.

One possible way to achieve a higher security is the integration of Model Anomaly Detection (MAD) algorithms on the aggregation server. These algorithms allow to detect and exclude anomalous models during the aggregation process. Numerous algorithms have been proposed over the past years (**?** 10; 3). Often MAD algorithms include parameters that need to be adjusted to fine-tune an algorithm to enhance the accuracy and robustness within a deployment domain. The process of investigation of required MAD algorithm parameters, as well as the scientific evaluation that is essential for the integration into real-world applications, is rarely straightforward due to the lack of frameworks that would allow a comprehensive MAD benchmarking under desired deployment conditions. Existing solutions are often narrowed down to tasks of either solely Machine Learning (ML) model training in a federated manner, or experiment results summarization and tracking. Moreover, they usually require extensive learning curve with manual setup, coding, and lack the visualization of metrics characterizing MAD performance.

Our contributions. We propose *IntelliMAD*, an accessible benchmarking domain-agnostic software framework that extends existing solutions that are designed for ML model training in a federated manner. Our framework implements a comprehensive and configurable robustness evaluation of FL aggregation strategies. It allows setting up the experiment plan through a centralized configuration file, to include client datasets, runtime data poisoning attacks and ML model training-related parameters. The centralized experiment configuration facilitates the experiment preparation process and allows experiment scheduling and parameters validation. With the assessment of several well-known aggregation strategies already integrated, our framework is scalable with respect to new aggregation strategies, dataset modalities, and domains.

We introduce the evaluation of a robust FL with MAD algorithm assessment and demonstrate the design workflow of experiments showcasing the feasibility of our approach. In our case example, we show how our MAD evaluation reveals the insufficiency of existing MAD algorithms and tools for real-world deployments. While theoretically robust, their setup process requires the knowledge that is not available in the real-world settings.

We present and discuss the results of a pilot usability study conducted with two groups of users. The first group used the Flower Framework to set up FL experiment execution, while the second group used `IntelliMAD` for the same task. The results demonstrate a significant reduction in the time required to set up experiment workflow when using our framework.

Table 1: Examples of existing frameworks and testbeds for model training in FL manner

Framework	Embedded Datasets	Runtime Attack	Aggreg. Strategies	MAD Robustness
FedML (FLamby)[a]	Fed-Camelyon16, Fed-LIDC-IDRI, Fed-IXI, Fed-TCGA-BRCA and others from medical imaging domain	✗	FedAvg, FedProx, Scaffold, Cyclic Learning, FedAdam, FedYogi, FedAdagrad	✗
Flower[b]	Available through flwr-datasets package	✗	FedAvg, FedMedian, FedYogi, FedAdagrad, FedAdam, QFedAvg, FedOpt	✗
TensorFlow Federated (TFF)[c]	EMNIST, StackOverflow, Shakespeare, CIFAR-100, others	✗	FedAvg, Secure Aggregation, Clipping, Zeroing, custom strategies	✗
FATE[d]	No built-in datasets; supports custom data loading	✗	FedAvg, SecureBoost, HeteroNN, FedProx, others	✗
PyTorch Lightning[e]	No native FL support; datasets via PyTorch	✗	Not applicable (used with external FL frameworks)	✗
IntelliMAD (ours)[f]	**FEMNIST, ITS, BloodMNIST, PneumoniaMNIST, Flair**	✓	**FedAvg, Bulyan, MultiKrum, Krum, TrimmedMean, Trust&Reputation, PID**	✓

[a] `https://fedml.ai` (5)
[b] `https://flower.ai` (2)
[c] `https://tensorflow.org/federated` (1)
[d] `https://github.com/FederatedAI/FATE` (8)
[e] `https://lightning.ai`
[f] `https://drive.google.com/file/d/1f6fXl86IAV1b15UKxT8XdqB2_VrW8SPU/view?usp=share_link`

2 Background

More complex setups, such as those found in domains that operate with highly sensitive and heavily regulated data, e.g., healthcare, finance, or governmental systems, introduce challenges related to data handling and processing. Additional security measures must be implemented, as data protection and privacy is paramount, particularly in the setups involving remote ML model communication. Clients supplying models can be physically or remotely compromised, leading to critical security vulnerabilities. In such scenarios, if an ML model is being trained on a compromised device, it may become poisoned, either through the manipulation with training data or model updates.

This issue becomes even more significant in distributed learning settings, such as those employed in FL, where the training process is delegated across numerous client devices, many of which may have varying degrees of trustworthiness. The decentralized nature of FL makes it more difficult to monitor and secure all participating nodes, thereby increasing the risk of model poisoning and necessitating the implementation of robust, model-level defense mechanisms to preserve model integrity and ensure safe deployment in real-world applications. The development of such defense mechanisms necessitates significant research efforts that involve continuous and extensive experimentation with various datasets and data modalities.

Figure 2: Repetitive experimentation in FL research

2.1 Why is Experimentation Framework Required?

Both ML model training and research and development of additional security-related parameters and defense mechanisms for more sensitive setups are common use cases across researchers and software developers. Both communities face critical challenges when it comes to the implementation and assessment of these functionalities, as existing frameworks and testbeds are either inaccessible due to the extensive learning curve required to achieve desired results, or they are designed solely for ML model training and lack tracking and visualization of necessary metrics. A significant challenge lies in the necessity to understand and adapt to the specific usage patterns of each available framework. Implementing custom functionalities tailored to specialized needs often requires a steep learning curve. This barrier can negatively affect the pace of experimentation and increase the complexity of evaluating theoretical approaches, ultimately slowing down the progress of research.

Another layer of complexity is introduced when ML models need to be validated on diverse datasets, which is a typical task within experimental evaluation of theoretical foundations. The integration of each new dataset into an existing experimental workflow often requires additional effort, such as data preprocessing and adaptation. For the testing of various defense mechanisms, the functionality that would simulate various attack scenarios also needs to be implemented. This process is rarely straightforward and may significantly slow down the experimentation timeline. Additional important factor is the experiment planning and scalability. When verifying theoretical foundations for defense mechanisms, it is often necessary to conduct multiple lengthy experiments in a physically remote environment. In such cases, it is often necessary to vary one parameter at a time in a controlled manner in order to isolate the effect of each change and accurately assess its impact. Without a unified framework to manage, execute, and track these experimental variations, the process becomes error-prone, time-consuming, and difficult to reproduce.

2.2 Overview of Existing Solutions for Experimentation Management and ML Model Training

Existing frameworks are often narrowed down to a specific task domain. These existing solutions can be grouped based on the functionality they offer and the types of problems they are designed to address. Existing framework can be grouped to ML model training and execution frameworks, and experiment metric tracking and summarization frameworks.

Table 1 provides an overview of frameworks and testbeds commonly used for ML model training in a federated manner. Numerous frameworks are available that provide functionality solely for ML model training and performance tracking, and their capabilities are largely confined to managing ML model training under predefined parameters. Several existing solutions offer functionality for tracking and analyzing experimental results. Although many existing tools provide extensive functionalities within their domain, they they lack accessibility by research community, as they require extensive programming and setups to be applied for research needs. They often need to be combined with additional tools or custom implementations to fully support continuous experimentation and comprehensive metric acquisition, particularly in distributed or FL setups. The lack of a unified solution that covers FL-specific requirements, standardized metrics management, and out-of-the-box accessibility presents a gap that must be addressed to facilitate FL research and development.

2.3 Model Anomaly Detection Research Workflow

Secure and robust FL can be implemented with the developing and evaluation of centralized aggregation techniques that incorporate MAD. Such techniques aim to identify and exclude anomalous models from the FL training process, contributing to the robustness and reliability of FL systems.

When researching these algorithms, continuous experimental validation of theoretical hypotheses is a critical part of the workflow. A typical hypothesis lifecycle requires the integration of theoretical foundations with experimental evidence. Theoretical models often evolve based on experimental findings, forming a cycle where experimental results both validate and refine theory. This iterative process is depicted in Figure 2, which outlines the workflow for executing FL experiments in the context of FL security research.

During the MAD research, it is essential to track metrics related to the robustness of an algorithm. The robustness is determined by MAD capabilities to detect and exclude anomalous models from the centralized aggregation, while maintaining suitable accuracy of aggregated ML model. Another objective is the computational overhead which may be possessed by MAD algorithm on the aggregation server.

2.4 Evaluation of Aggregation Robustness

In this section we introduce our definition of a robust FL aggregation. In ideal case without any malicious clients participating FL process, the robust FL aggregation can be defined through the robustness of the resulting ML model in terms of loss and accuracy. However, for the case when malicious parties are present in the joint training process, we introduce another component of a robust aggregation: MAD robustness. MAD robustness can be defined as the accuracy of the malicious client exclusion during the training process with identical MAD settings under varying execution conditions, e.g., proportion of malicious clients or attack intensities. Thus, the robust aggregation consists of two following components: robustness of aggregated ML model, and robustness of MAD algorithm.

Depending on the specific task and domain, each component may carry different weight in defining robust aggregation. In edge cases, certain components may be entirely excluded from the calculation of overall aggregation robustness. Below, we describe the complete set of metrics that characterize both ML model robustness and MAD algorithm robustness. We then introduce *IntelliScore*, a composite metric calculated based on the selected set of ML model and/or MAD robustness metrics most relevant to a given setup.

The evaluation vector of a robust MAD algorithm is given by:

$$\mathbf{v}_m^{(T)} = \left[\mathbb{E}_{d,a}(\mathcal{L}_{m,d,a}^{(T)}),\ -\mathbb{E}_{d,a}(\mathcal{A}_{m,d,a}^{(T)}),\ \mathbb{E}_{d,a}(\Phi_{m,d,a}(T)),\ \mathbb{E}_{d,a}(\Theta_{m,d,a}(T)),\ \sigma^2_{\mathcal{L}_m^{(\infty)}},\ \sigma^2_{\mathcal{A}_m^{(\infty)}},\ \sigma^2_{\Phi_m},\ \sigma^2_{\Theta_m}\right]^T$$

The final *IntelliScore* that represents the robustness of a MAD algorithm:

$$S_m^{(T)} = 1/\left\|\mathbf{v}_m^{(T)}\right\|$$

Finally, the robust MAD algorithm $M^* \in \mathcal{M}$ is the algorithm that:

$$M^* = \lim_{T\to\infty} \arg\max_{M_m \in \mathcal{M}} S_m^{(T)}$$

The detailed description of all metrics of a robust MAD algorithm is provided in the Appendix.

In our definition, we emphasize the consistency of the MAD algorithm performance under changing execution conditions. In real-world setups, it is often hard to predict the types of attacks that may occur in the joint training process; therefore, the robust MAD algorithm should render consistent performance within the defined scope.

3 *IntelliMAD* Features Overview

The *IntelliMAD* offers out-of-the-box platform for FL experiment execution and management that requires zero code adjustments to set up and configure basic FL experiment benchmarking. It offers experiment planning and configuration verification, as well as tracking and visualization of all relevant metrics, including the assessment of MAD algorithms robustness. The framework allows to configure clients, datasets, and data poisoning attacks through experiment configuration file, requiring no scripting or coding. More detailed description of framework functionalities is provided below.

Enhanced User Experience: Configurable data poisoning in clients' datasets, visualization and storing of FL execution metrics and centralized experiment configuration and validation

For the purpose of MAD evaluation, *IntelliMAD* offers the functionality that allows to configure data poisoning attacks that will be performed in the runtime on client datasets. This is essential for robustness assessment of MAD algorithms, as it allows to render a plethora of execution environment setups that would mimic the variability of real-world conditions. This setup flexibility allows to achieve a more robust resulting MAD algorithm, as MAD fine-tuning and adjusting for particular setups is simplified.

Unlike many other frameworks dedicated solely for ML model training, our solution provides immersive visualization of all experiment metrics that characterize both the performance of federated ML model training, and attached MAD mechanisms. Besides plotting per-client and aggregated characteristics of ML model training, our solution keeps track and visualizes metrics that provide insight on the performance of MAD algorithm. Detailed description of supported metrics is provided in Appendix. For each experiment execution, all collected metrics are also saved into CSV files, facilitating further analysis of FL performance.

Additionally, all FL experiment parameters are configured through a centralized configuration file. This approach eliminates the need for direct code modifications, simplifying the adjustment of experiment parameters and ensuring consistency across different experimental runs. Before executing an experiment, the configuration is validated to prevent failures caused by incorrect settings.

Robustness Evaluation: Benchmarking of FL aggregation strategies Our framework supports the benchmarking of and against several well-known aggregation strategies, e.g., Krum, Multi-Krum, Trimmed Mean, FedAvg, and preserves the underlying functionality of FL model training framework to integrate new aggregation strategies while collecting their MAD-related performance metrics. It provides a uniformed benchmark for the robustness of MAD algorithms that can be configured and executed without manual code adjustments. Our solution expands the standard workflow of a conventional ML model training framework beyond the task of simply training ML models, but rather assessing the robustness of the entire setup, including security and algorithmic performance, using the robustness concept described in Section 2.4.

Experiment scheduling and scalability Our solution provides functionality for experiment planning through the configuration file, with the number of subsequent experiments limited only by available time and computational resources of the host machine. It enables the configuration and execution of multiple FL strategies in sequence, with configurable parameters between each run. In the case of multi-experiment plan execution, aggregated ML model metrics and MAD algorithm performance metrics collected during each dedicated experiment are plotted on the same graph, allowing to track the influence of each set of experiment parameters on the FL execution.

4 *IntelliMAD* Workflow

In our framework the process of the experiment execution can be logically divided into the phases described below.

Experiment Setup In the initial phase, the dataset is configured and loaded into the framework. The dataset must be pre-processed to conform to the specific file and folder structure required by `IntelliMAD`. Framework is supplied with datasets described in the Table 1. Based on the attack type and the number of malicious clients specified in the configuration file, client datasets are selectively poisoned and stored temporarily in the host file system. Optionally, these poisoned datasets can be preserved for later verification by enabling the corresponding option in the configuration file. The ML model type and architecture are selected based on the chosen dataset. The framework supports assigning the corresponding model architecture through a dataset keyword in the configuration file, allowing for flexibility and automation. Additionally, it is architecturally possible to assign different models by defining and specifying the ML model keyword in the configuration file, in the case if comparison is needed between different models on one dataset. Model training can be initiated either from scratch or by fine-tuning an existing pre-trained model. If training from scratch, the framework loads a model architecture suited to the selected dataset. In the case of fine-tuning, a pre-trained model is loaded and further employed during training. All other parameters required for experiment

execution, such as the number of clients, number of training rounds, and attack scenario, are parsed and validated from the centralized configuration file. In cases where the configuration is invalid, the experiment execution is stopped, and appropriate error messages are shown to inform the user of the specific misconfigurations that must be resolved.

Experiment Execution During this phase, the pre-processed datasets and configuration parameters from the setup phase are passed to the underlying third-party ML model training framework. This external framework is responsible for managing all model training tasks, including computing the loss and accuracy at each aggregation round or local epoch, and providing both intermediate model updates and the final trained model. Our implementation utilizes Flower Framework (2).

Experiment Metrics Processing *IntelliMAD* collects essential raw data from the underlying ML model training framework during the execution process. This raw data includes metrics such as the loss and accuracy values recorded at each aggregation round. Additionally, client participation history is collected. Specifically, whether or not a particular client's update was incorporated into the centralized model during aggregation round. This participation data is crucial for evaluating the efficiency and effectiveness of MAD algorithms at subsequent analysis stages. Performance-related metrics, such as algorithm execution times, are also captured while in this phase to provide insights into system-level behavior and computational demands. Once the training phase is completed within the ML model training framework, *IntelliMAD* proceeds to compute derivative metrics based on the collected raw data. These derived statistics include aggregated metrics such as average loss and accuracy across all rounds, as well as performance indicators specific to the MAD algorithms, such as F1 scores, precision, recall, and MAD-specific accuracy.

Final Artifacts Storing Following the computation of all derivative metrics, *IntelliMAD* compiles and formats the results. This includes generating CSV files that encapsulate both the raw and processed numerical data, and producing visual plots that illustrate observed trends. This process allows to achieve reciprocal goals. On the one hand, it allows researchers and developers to configure all experiment parameters directly through *IntelliMAD* without needing to modify the underlying codebase, enabling straightforward setup and execution of experiments. Since the experiment configuration includes the MAD parameters, such process allows to discover the parameters ost suitable for a particular dataset and execution environment. On the other hand, this workflow provides the possibility to extend the framework to new domains by following the existing integration architecture.

5 Framework Evaluation and Discussion

5.1 Metrics Variance

The intuition behind the reasoning for our robustness score calculation implementation stems from the observation that less consistent and accurate anomalous models exclusion from the aggregation leads to greater variance of metrics in-between experiment executions. We demonstrate the variance of loss and accuracy metrics by defining and executing a series of experiments with the fixed proportion of malicious clients in the system. Experiments were executed on the subset of FEMNIST dataset, which consists of handwritten digits. The round at which the MAD algorithm was engaged was varied between executions, ranging from not excluding malicious clients at all, to excluding them closer to the end of experimental executions. This study showcases that the timely exclusion of anomalous models leads to smoother loss and accuracy dynamics. All experiments were conducted on the machine equipped with Apple M1 Pro CPU with 16GB RAM. The Fig. 3 demonstrates the results.

5.2 FL MAD Robustness

As was discussed in Sec. 2.4, a robust MAD algorithm can be described as the vector consisting of metrics that are to be selected based on the desired objective. If the goal for the resulting MAD setup is to provide the most accurate ML model, the vector can consist only of model-related metrics, such as loss and accuracy. If the security needs to be considered as well, metrics such as MAD

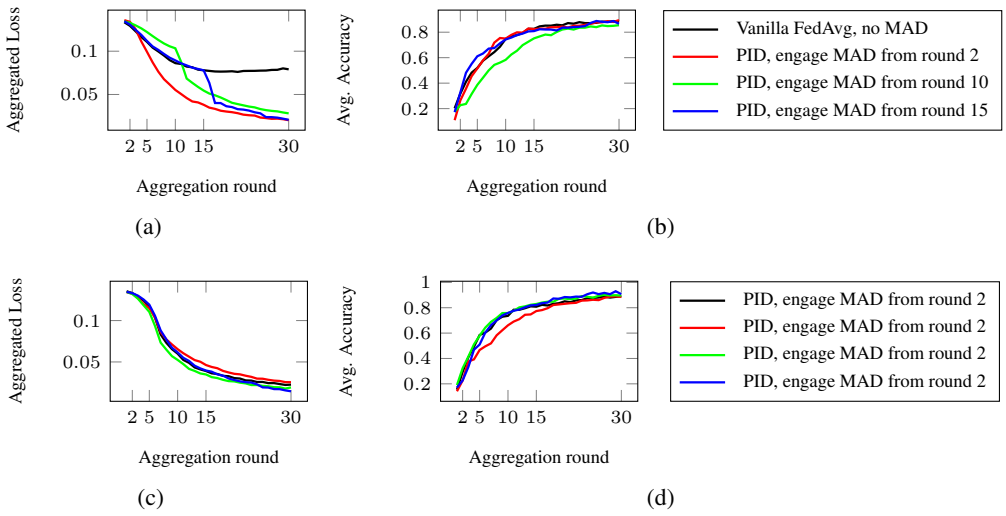

Figure 3: Demonstration of greater loss (a), and accuracy (b) variance when excluding clients on different aggregation rounds; and lesser variance – (c), (d) – when excluding at the same aggregation round

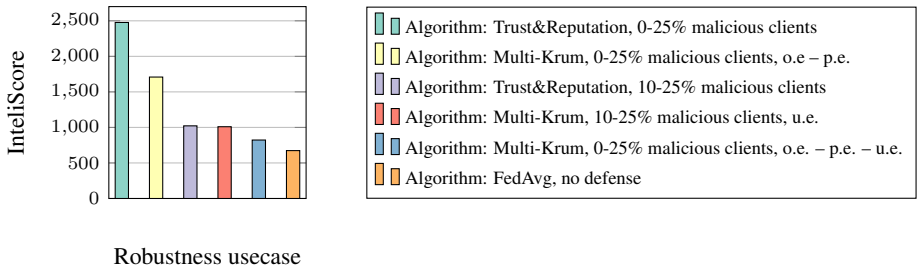

Figure 4: Comparison of robustness scores across different *robustness usecases*. Each bar represents *robustness usecase*, which consists of six FL executions with the varied proportion of malicious clients with specified MAD algorithm. Since Multi-Krum requires to configure the estimate of malicious clients in the system, various cases were tested with under estimation, precise estimation, and over estimation of malicious clients.

client exclusion accuracy can be included into the vector. If performance is also the objective for the evaluation, calculation time can be incorporated.

Six sets of experiments were conducted on BloodMNIST (11) dataset. Each set of experiments is referred to as *robustness usecase*. Each *robustness usecase* can consist of a desired number of FL experiment executions with the select parameter altering in-between experiments. Here, the number of malicious clients was modified in-between executions within a *robustness usecase* in order to test the ability of a MAD algorithm to adequately exclude all malicious clients under varying attack intensities. The evaluation vector consisted of loss and accuracy, which reflects the influence of the MAD algorithm ability to remove malicious clients on the performance of the aggregated model. The results are demonstrated in Fig. 5.2.

IntelliMAD allows to create a composite metric that enables the assessment of FL robustness in different execution conditions. These results demonstrate that Multi-Krum fails to render adequate performance in the cases that include underestimation of the number of malicious clients in the system; in such situations, the algorithm doesn't exclude all malicious parties, which results in the perturbations in the training loss and accuracy, which is reflected in the lower robustness score.

Extension to LLM Domain In order to demonstrate the possibility to extend our framework to a new domain, we conducted an LLM study, which can be found in Appendix.

5.3 User Experience

In order to evaluate to which extent the framework can enhance the user experience, we conducted a pilot study on two groups of users. The first group user Flower Framework to implement an FL setup from scratch, while the second group used *IntelliMAD* framework to execute FL experiments and extend the framework functionality to new domains. We collected the user feedback on the time that was required to set up and execute experiments in Flower, versus the time that was required to set up and run experiments in *IntelliMAD*. As collected feedback suggests, it took only 15 and 30 minutes for users to set up the framework and execute a sample experiment, which is about *60 times faster* than setting up an FL framework from scratch. Plot depicting collected user feedback can be found in Appendix.

6 Limitations

One of the limitations of our framework is that it requires underlying ML model training framework to work with, as is reflected on the logical view in the Figure 2. While designed, studied and supplied with underlying Flower Framework, our solution extends existing ML model training functionality with centralized experiment configuration, metrics processing, aggregation robustness evaluation, and results visualization. In order to integrate *IntelliMAD* with other ML model training frameworks, additional configuration is required.

The integration with third-party ML model training possesses a limitation related to implementation of aggregation strategies. The possibility to implement a new aggregation strategy, as well as the implementation guidelines, depends entirely upon the availability of custom aggregation strategies in the underlying ML model training framework. Another limitation arises from the nature of the datasets integration. While *IntelliMAD* is supplied with several datasets representing the image processing domain, extension to other domains or imaging datasets requires datasets pre-processing and manual integration. While such a design introduces certain level of additional complexity, we don't see it as being a major limitation, as the dataset pre-processing is an essential step in task involving ML model training.

7 Conclusion

In this work, we introduced *IntelliMAD*, a novel FL framework that extends existing tools to make FL research and development significantly more accessible by automating complex procedures and introducing novel functionality, allowing easy and flexible setup of experiments and conditions, and providing automatic recording and visualization of results without manual intervention, thus ensuring that results are reported automatically and presented without manual intervention, setting our solution apart from existing FL frameworks. *IntelliMAD* substantially lowers the required level of expertise, prior knowledge, and efforts for users wishing to conduct research and securely integrate FL into real-life applications – an advancement we demonstrate by measuring the reduced average time users spent performing their research and development tasks with our framework. *IntelliMAD* specifically empowers users to conduct in-depth research on real-life datasets, which might include data anomalies and shifts within FL environments and to systematically analyze the developed application performance as well as its robustness to possible anomalies. A major advantage of our contribution is the presentation and integration of a novel methodology for calculating the robustness of MAD algorithms, allowing for comprehensive experiments and comparative analysis of MAD algorithm robustness across diverse domains and datasets. *IntelliMAD* provides a powerful platform that significantly aids the community of researchers and practitioners across various domains in advancing their research and development efforts, streamlining the path towards creating more resilient, trustworthy, and practically deployable FL systems. Its design has been evaluated on various datasets in diverse application domains.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore,

D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow, Large-scale machine learning on heterogeneous systems," Nov. 2015.

[2] 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, arXiv:2007.14390. [Online]. Available: http://arxiv.org/abs/2007.14390

[3] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Byzantine-tolerant machine learning," no. arXiv:1703.02757, Mar. 2017, arXiv:1703.02757 [cs]. [Online]. Available: http://arxiv.org/abs/1703.02757

[4] S. Chuprov, R. Zatsarenko, D. Korobeinikov, and L. Reznik, "Robust training on the edge: Federated vs. transfer learning for computer vision in intelligent transportation systems," in *2024 IEEE World AI IoT Congress (AIIoT)*, 2024, pp. 172–178.

[5] 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," no. arXiv:2007.13518, Nov. 2020, arXiv:2007.13518. [Online]. Available: http://arxiv.org/abs/2007.13518

[6] S. Jha, A. Roy, A. Cobb, A. Berenbeim, and N. D. Bastian, "Challenges and opportunities in neuro-symbolic composition of foundation models," in *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*, 2023, pp. 156–161.

[7] D. Korobeinikov, S. Chuprov, R. Zatsarenko, and L. Reznik, "Federated learning robustness on real world data in intelligent transportation systems," Jun. 2024. [Online]. Available: https://par.nsf.gov/biblio/10532530-federated-learning-robustness-real-world-data-intelligent-transportation-systems

[8] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection."

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and y. A. Blaise Agüera, "Communication-efficient learning of deep networks from decentralized data," 2023. [Online]. Available: https://arxiv.org/abs/1602.05629

[10] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," no. arXiv:1802.07927, Jul. 2018, arXiv:1802.07927 [stat]. [Online]. Available: http://arxiv.org/abs/1802.07927

[11] 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, p. 41, Jan. 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions include the proposal of *IntelliMAD* framework along with the theoretical definition of MAD robustness. The paper presents both the framework and the conducted study to support these claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation of the proposed framework and the scope of it's application.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper doesn't include theoretical analysis. We provide the formal definition of a robust MAD algorithm which can be used as a foundation for FL robustness evaluation and can be adjusted by framework users for a particular needs and domain.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: configurations for all experiments along with the code and execution instructions are provided in the Google Drive folder anonymously. The proposed framework is designed in such a way that full experiment configuration is saved after every experiment execution, which allows to reproduce the exact results by providing the saved configuration as the framework input.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Upon paper publication, we will update the paper with the GitHub repository containing the code of our framework along with all necessary instructions to execute experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is presented to a necessary level of detail in the paper. The full configuration for all executed experiments is included with the code submission provided in Google Drive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We don't report error bars as this would require additional experiments, which is computationally expensive within FL, specifically in image processing domain and LLMs.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The characteristics of the machine used to execute experiments within image processing domain are described in Sec. 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The work discusses the positive impact through facilitation of a more secure and robust FL for real-world deployments.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release any data or models that have high risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All corresponding are included with the code licence.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets are included along with the code and documentation on Google Drive.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: The paper discusses the results that were collected during a pilot study conducted during academic activities. Questions and details can be provided upon request.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The paper discusses the results that were collected during a pilot study conducted during academic activities.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.