Evaluating the Transferability of Data Poisoning Attacks across Centralized & Federated Learning Techniques

By Michael J. Mattarock II

B.A. in International Affairs and History, summa cum laude, January 2012, George Washington University
M.S. in Information Systems Technology, August 2017, George Washington University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 15, 2025

Praxis directed by

John Dargan
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Michael J Mattarock II has passed the Final Examination for the degree of Doctor of Engineering as of July 8, 2025. This is the final and approved form of the Praxis.

Evaluating the Transferability of Data Poisoning Attacks across Centralized & Federated Learning Techniques

Michael J. Mattarock II

Praxis Research Committee:

John Dargan, Professorial Lecturer in Engineering and Applied Science, Praxis Director

Ken Clements, Professorial Lecturer in Engineering and Applied Science, Committee Chair

Michael Darko, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Michael J. Mattarock II
All rights reserved

Dedication

The author wishes to dedicate this work to his parents, Michael and Dawna Mattarock, whose unwavering work ethic and constant encouragement inspired their son to reach higher and never stop striving for greatness.

Acknowledgements

The author desires to acknowledge his advisor for this praxis and course professors at the George Washington University who provided invaluable guidance and feedback for this research along his academic journey. A special thank you to all of the informal mentors, friends, and colleagues that offered their support at Carnegie Mellon University and the university for providing the financial means to complete this incredible degree program.

Abstract of Praxis

Evaluating the Transferability of Data Poisoning Attacks across Centralized & Federated Learning Techniques

This research investigated the main differences between Centralized Learning (CL) and Federated Learning (FL) in the context of data poisoning attacks focusing on how each architecture mitigated or stopped adversarial disruptions. Computer vision and classification tasks within machine learning models across diverse datasets was used. Evaluation around the performance of CL and FL under different levels of data poisoning was done to provide a comprehensive understanding of how learning technique affected overall vulnerability. Across the trials, special focus was paid to evaluating how model accuracy, precision, recall, cross entropy loss, and F1 score changed as the level of data poisoning increased with different datasets. Different models and datasets were tested under controlled poisoning situations to simulate real-world environments building a deeper understanding of the tension of data security and design in modern machine learning (ML) systems, largely classification tasks within computer vision.

The importance of this is shown by the growing adoption of FL as a way of enabling Distributed Artificial Intelligence (DAI), especially as organizations aim to expand AI integration while making data privacy stronger against adversarial threats. Key operational considerations driving this shift include managing data with differing classification or control levels, enforcing role-based data access, and maintaining performance consistency despite the increasing geographic dispersion of data sources (an inherent limitation of centralized architectures that we see today). In exploring both the strengths and weaknesses of each learning

technique, this research informs more secure and context-appropriate AI system design across varying use cases and industries.

Table of Contents

Dedication ..... *iv*

Acknowledgements ..... *v*

Abstract of Praxis ..... *vi*

List of Figures ..... *x*

List of Tables ..... *xi*

List of Acronyms ..... *xii*

Chapter 1 – Introduction *1*

1.1 Background 1

1.2 Centralized vs. Federated Learning: An Architectural Comparison 2

1.3 Proliferation of Data Poisoning Attacks 3

1.4 Distributed AI and Transferability Across Models 5

1.5 Problem Description and Research Differentiation 7

1.6 Thesis Statement 8

1.7 Research Questions and Hypotheses 9

1.8 Research Limitations 10

1.9 Organization of Praxis 10

Chapter 2 – Literature Review *12*

2.1 Introduction 12

2.2 Distributed AI 13

2.3 Proliferation of Data Poisoning Attacks 16

2.4 Attack Strategies in Centralized Learning 17

2.5 Federated Learning: A New Attack Landscape 18

2.6 Data Poisoning Transferability in Machine Learning 20

2.7 Attack Impact and Consequences 23

2.8 Use of Adversarial Training 24

2.9 Adversarial Training: A Comparative Perspective 26

2.10 Model Training and the Potential for Model Drift 27

2.11 Architectural Considerations 29

2.12 Differences in Attack Vectors 32

2.13 Defense Mechanisms 33

2.14 Performance within Use Cases **33**
2.15 Datasets and Propelling Research **34**
2.16 Use of Benchmarking and Metrics **36**
2.17 Identification of Data Poisoning Attacks **40**
2.18 Research Connection **42**
Chapter 3 – Methodology ***44***
3.1 Experimentation Overview **44**
3.2 Experimentation Approach: Dataset and Model Selection **46**
3.3 Centralized Learning with Data Poisoning **51**
3.4 Federated Learning Implementation **53**
3.5 Poisoning Conditions **57**
3.6 Performance Evaluation Metrics **58**
3.7 Comparison and Analysis **65**
3.8 Digital Dashboard Development **66**
3.9 Conclusion **68**
Chapter 4 – Results ***69***
4.1 Establishing Desired Outcomes **69**
4.2 Establishing Assessment Criteria **70**
4.3 Assessing Experimental Trial Outputs **71**
4.4 Comparing Results between Datasets and Learning Techniques **75**
4.5 Evaluating Overall Data Poisoning **81**
4.6 Resulting Architectural Considerations **84**
4.7 Assessing Wider Cyber Vulnerabilities **86**
4.8 Contributions to the AI Engineering Body of Knowledge **88**
4.9 Concluding Considerations **90**
Chapter 5 – Research Discussion ***92***
5.1 Summary of Research and Core Findings **92**
5.2 Research Implications **95**
5.3 Opportunities for Future Research **98**
5.4 Applying Research Novelty **99**
5.5 Conclusion **100**
Appendix ***102***
References ***108***

List of Figures

Figure 1 – 1: Operational View of CL and FL……7

Figure 3 – 1: CL and FL Example Operational Views……41

Figure 3 – 2: Experimental Trials Overview……42

Figure 3 – 3: Experimental Overview……43

Figure 3 – 4: MNIST Shallow CNN Model……47

Figure 3 – 5: ImageWoof Model……47

Figure 3 – 6: Example Flower Configuration with 5 Clients (Bo, Kan, 2023)……...50

Figure 3 – 7: FL Trial Parameters……51

Figure 3 – 8: FL Process Flow Chart……53

Figure 3 – 9: Example View of Dashboard……63

Figure 4 – 1: MNIST Overall Results……..67-68

Figure 4 – 2: ImageWoof Overall Results……69-70

Figure 4 – 3: MNIST Centralized Learning Experimental Trials……..72

Figure 4 – 4: MNIST Federated Learning Experimental Trials……73

Figure 4 – 5: ImageWoof Centralized Learning Experimental Trials……..74-75

Figure 4 – 6: ImageWoof Federated Learning Experimental Trials……76-77

Figure A – 1: Overall Cross-Entropy Loss of Xray14……..96

Figure A – 2: Overall Accuracy of Xray14……97

Figure A – 3: Overall ROC-AUC of Xray14……97

Figure A – 4: Xray14 Poisoned Levels for CL Trials……...98

Figure A – 5: Xray14 Poisoned Levels for FL Trials……99

List of Tables

Table 2 – 1: CL Architecture Versus FL Architecture……29

Table 3 – 1: MNIST Dataset……...45

Table 3 – 2: ImageWoof Dataset……46

Table 3 – 3: Performance Metrics Overview……59

Table 5 – 1: Research Hypotheses Overview……87

Table 5 – 2: Results of Research Hypotheses……89

Table 5 – 3: Hypothesis 1 Paired t-test ……90

Table 5 – 4: Hypothesis 2 Paired t-test ……90

Table 5 – 5: Hypothesis 3 Paired t-test ……91

List of Acronyms

AI	Artificial Intelligence
AIE BoK	Artificial Intelligence Body of Knowledge
AML	Adversarial Machine Learning
ASR	Attack Success Rate
AUC	Area Under the Curve
CJADC2	Combined Joint All-Domain Command and Control
CL	Centralized Learning
CNN	Convolutional Neural Network
DAI	Distributed Artificial Intelligence
DoD	Department of Defense
FGSM	Fast Gradient Sign Method
FL	Federated Learning
IID	Independent and Identically Distributed
IoT	Internet of Things
LAVAN	Low-Visibility Adversarial Network
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
NLP	Natural Language Processing
OT	Operational Technology
PGD	Projected Gradient Descent
PII	Personally Identifiable Information
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SOW	Statement of Work
T&E	Testing and Evaluation
UAS	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
U.S.	United States

Chapter 1 – Introduction

1.1 Background

Machine learning (ML) and artificial intelligence (AI) technologies continue to evolve, with their role in transforming various operational activities becoming increasingly profound, yet the wider spread adoption also introduces significant security risks. This is especially true with the rise of data poisoning attacks, where adversaries tamper or disrupt training data to degrade the performance of a model or fully manipulate its outcomes. These attacks can have dire consequences, especially in high-stakes fields like national security and autonomous systems, where AI models are employed to automate decision-making processes in areas such as surveillance, autonomous systems, and cybersecurity. At the same time, current benchmark metrics such as accuracy and recall do not provide sufficient visibility when data poisoning occurs as their measures lack indication of significant tampering of models. For instance, an autonomous model that is subjected to a data poisoning attack may not show significant change in performance initially if a single attribute is changed (backward to forward); however this tampering has the potential for tremendous disruption on the performance of the system in operation.

In the current state there are two prominent approaches for model training, centralized and federated learning (CL and FL), each with its own advantages and challenges in preventing and mitigating data poisoning attacks among other considerations. Alongside these learning architectures, adversarial training has emerged as a critical method for hardening models against attack and manipulation. Understanding how these methods intersect and contribute to the security of ML systems is key to

addressing the research problem sought to be addressed in this praxis: assessing the transferability of data poisoning attacks across centralized and federated learning and evaluating the role of adversarial training in mitigating such attacks. It is the goal of this praxis to identify which learning technique (CL or FL) proves more resilient to data poisoning along with gauging the effectiveness of model training across different levels of attack.

1.2 Centralized vs. Federated Learning: An Architectural Comparison

CL and FL are two different approaches to training ML models with the first involving the gathering of data from multiple sources into a single, centralized server, where it is used to train one global model (common in ML operations today). This approach is typically viewed as standard and allows for full oversight of the training process from a single point, given the central authority has full access to the training data and can monitor / control every aspect of the model. In these systems, defense against data poisoning and other attacks can be tightly integrated with centralized control enabling a streamlined monitoring of data validation, anomaly detection, and if employed adversarial training. However, CL also presents significant risks, most notably the creation of a single point of failure when considering attack vulnerability. If an adversary can access the central server or model, they can inject poisoned data at this single point and compromise the overall performance across all system components due to a faulty model. This can be particularly dangerous in safety and mission critical applications, where timely decisions are often made based on predictive analytics and such a disruption can be catastrophic to an operation. We can see this through an afflicted CL

system in a defense network that could lead to misinterpretations of battlefield data or incorrect threat assessments, potentially jeopardizing human lives.

Following a much different approach to training, FL is a decentralized technique designed to address disparate data sources with multiple nodes across a wider system. In FL, the training process is distributed across these multiple clients, which can include edge devices, mobile phones, or institutional servers, each of which retains control over its local data. Instead of transferring raw data to a central server, each client trains its own local model and sends only the model updates (i.e., learned parameters) to a universal model, which then aggregates them. This architecture has a number of benefits including privacy and data segregation, especially in defense environments where sensitive data from multiple sources (i.e., different allied nations) must be protected and controlled. Further FL offers a built-in redundancy that is not observable in CL as networked nodes are able to back up one another (dependent on policies and configuration). This becomes paramount when one such node suffers an attack as can be the case with data poisoning, which theoretically could be easier to contain in a federated system where there is not reliance on a central model.

1.3 Proliferation of Data Poisoning Attacks

There is a rise of data poisoning attacks, one of the most pressing security concerns in modern machine learning, especially as ML becomes increasingly ubiquitous and applied to new environments with increasing responsibility. A data poisoning attack occurs when an adversary introduces malicious data into the training process with the intention of corrupting the resulting model, which can lead to incorrect predictions, biased outcomes, or even complete failure of the system (Chen, 2024). Research today

suggests the threat of data poisoning attacks is consistently growing with various actors testing their attack capabilities and monitoring response throughout industries and organizations. The overall impact of compromised systems can be disastrous with many organizations often never realizing an attack has occurred until it is far too late.

The central research goal of this praxis is to evaluate how data poisoning attacks influence and differ between centralized and federated learning techniques across different models and datasets. Data poisoning attacks take many different forms and include indiscriminate attacks that degrade the overall performance of a model to targeted attacks that aim to manipulate specific outputs or behaviors often to the advantage of an adversary. It is also worth mentioning that they can be unintentional as well in the case of accidental errors made during model training. As an example with an autonomous system, an adversary might launch a targeted poisoning attack against a model that controls autonomous drones, causing it to misidentify targets or fail to execute critical missions that would otherwise be accomplished or a developer mislabeling directional attributes throwing the vehicle off course. Similarly, an adversary could poison training data used in intelligence analysis models, skewing results of predictive analytics that inform strategic decisions.

There is a proliferation of data poisoning attacks fueled by various factors as AI has become more widespread and attackers seek new ways to exploit systems that are increasing completing essential tasks. The increasing complexity of ML models makes it difficult to verify the integrity of training data in many instances, especially in large-scale or real-time applications where accuracy is paramount and a model may contain billions of parameters that are not easily protected. The rise of open data and crowdsourced

datasets creates further opportunities for adversaries to inject poisoned data into public or semi-public training pipelines to unsuspecting users and at scale given the number of populations / organizations that can be reached. Further, the growing interconnectivity of AI systems across different sectors increases the risk of attack transferability, where an attack designed for one model could potentially compromise another model trained on a related or derived dataset.

Federated learning introduces new avenues for such attacks, as highlighted through it uses and operational focus on autonomous systems among other applications. In FL environments, where individual clients (such as remote sensors, autonomous vehicles, or defense command nodes) train local models, data poisoning attacks can occur without directly compromising the universal model. An adversary can manipulate the local training data on one client or node, poisoning its model updates before they ever reach the central server. These poisoned updates are then aggregated with legitimate updates from other clients, gradually corrupting the universal model over time while also creating disruption at the local model and degrade performance at both locations. The difficulty of detecting poisoned updates in federated systems highlights one of the reasons sophisticated defense mechanisms are needed that can identify and mitigate malicious behavior before it spreads.

1.4 Distributed AI and Transferability Across Models

Throughout this praxis the term transferability is used to assess the spread of adversarial influence across models, datasets and ML environments. As a working definition transferability refers to the ability of adversarially crafted poisoned data to degrade the performance or alter the behavior of a system, even if those models differ in

configuration, parameters, or training data. When an attacker poisons a dataset, they typically aim to insert malicious patterns or corruptions that exploit vulnerabilities in a specific model. Yet if the poisoned data affects other parts of the model or system not directly targeted during the attack, this demonstrates such transferability of an attack, as can be the case with multiple classes of a model being poisoned simultaneously when only one class was initially targeted. This scaling cybersecurity concern is of critical importance as a single poisoned data element has the potential to harm many classes or potentially models trained by different entities, even in the instance of different architectures (e.g. convolutional neural networks vs. transformers). Attackers do not need detailed knowledge of the target model, producing a generalization of attacks that are more feasible and dangerous.

This research topic surrounding the spread and transferability of data poisoning attacks across centralized and federated learning is paramount as it addresses the fundamental challenges of securing distributed AI systems, which are rapidly expanding and necessary to fulfill operational requirements. This is critical as more operations employing ML move to decentralized architectures as is the case with capability deployments implemented by Combined Joint All-Domain Command and Control (CJADC2) in military applications as just one example. Understanding how these attacks transfer between different learning techniques will provide valuable insights into the design of more resilient systems, particularly in critical applications where failure is not an option and high levels of accuracy is required.

1.5 Problem Description and Research Differentiation

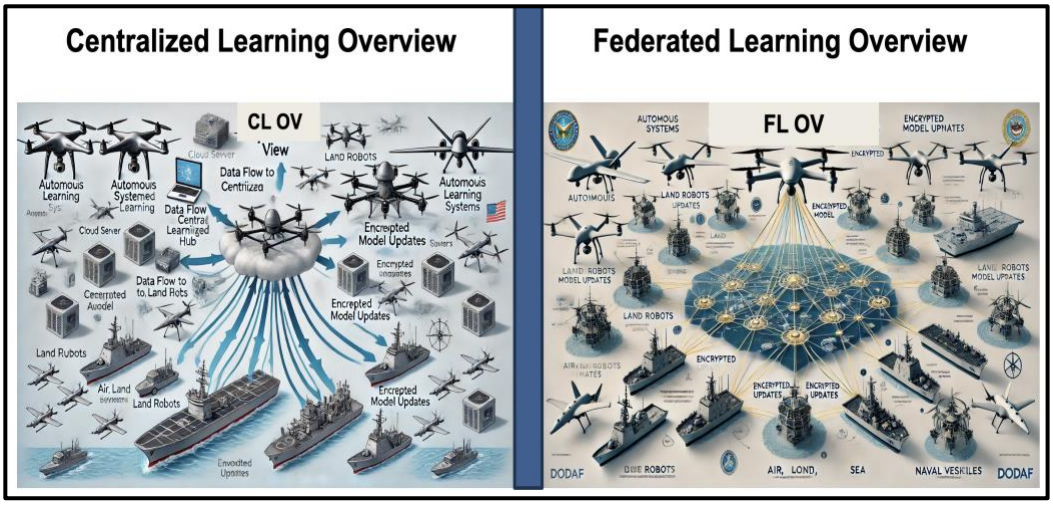
The key problem of this praxis is assessing how well data poisoning attacks can disrupt ML in CL and FL environments, identifying the potential for such attacks to spread, and evaluating architectural effectiveness as well as learning technique in defending against these attacks. The focus was on models used in computer vision, particularly with a nod to applications in classification. The objective of the experimental trials conducted in this praxis was to build a general foundation that is industry-agnostic, allowing for wider use of the results gleaned in varying applications across diverse organizations and multiple use cases.

The transferability of adversarial attacks is crucial for understanding the broader security landscape of ML models with particular focus here to transference across classes. CL, where data is aggregated in a single location, and FL, where models are trained across decentralized devices, offer different attack surfaces and different sources of vulnerability. Some use cases include autonomous systems like drones, vehicles, or other defense systems where knowledge is applied, given their strong reliance on ML models and sensitivity to compromised data. The differentiator with this research lies in its objective to not only evaluate the benefits and challenges of CL with FL, but also exploring the impact of data poisoning on distributed AI environments at different levels, which will only continue to grow as more widespread need of real-time/near real-time decision-making is required outside of standard compute environments. The results of this research will allow senior leaders across industries to make better decisions on how best to safeguard these changing environments to better protect their most critical assets.

Operational View: Scenario

The key question is which architecture better defends against the spread of data poisoning attacks. The main performance metrics are accuracy, precision, recall, cross entropy loss, and F1 score. Figure 1-1 below depicts a high level overview of the two distinct learning environments that will be explored in this research.

Figure 1-1: Operational Views of CL and FL



Operational View: Praxis Experimentation

- **Centralized Learning (CL):** A base trial with six data sources (same dataset), two of which are subject to data poisoning. Trials include assessing baseline performance, compromised performance, and performance with adversarial training.
- **Federated Learning (FL):** Similar trials conducted for FL, evaluating the same metrics across the same number of data sources.

1.6 Thesis Statement

Data poisoning attacks degrade model performance with model training technique (CL and FL) playing a significant role in the severity and spread of the influence of an

attack. Through evaluation of model training transferability, we can identify which learning technique is more effective in reducing the spread of attacks across datasets.

1.7 Research Questions and Hypotheses

The research questions and hypotheses of this praxis focused on the influence of data poisoning attacks on ML and how learning technique (CL and FL) played a role in reducing overall vulnerability of such attacks.

RQ1: How does model training affect susceptibility and transferability across Centralized and Federated Learning platforms?

RQ2: Which model training technique (CL / FL) better reduces the risk of attack spread across a platform?

RQ3: What correlations exist between architecture and learning technique to mitigate the influence of data poisoning attacks?

Similarly, three research hypotheses were developed based on current state research across these fields and influenced through the literature review.

H1: There is significant difference in the performance of models across CL and FL techniques at a foundational level.

H2: Federated Learning reduces the spread and impact of data poisoning attacks given its decentralized nature.

H3: When model training is completed upfront it is more effective in combating the influence of data poisoning attacks, which allow recognition of outliers, anomalies, and possible attack vectors.

1.8 Research Limitations

This research strived to contribute valuable insights into the security of ML models with a key strength in comparing the side-by-side analysis of CL and FL architectures under both normal and attacked (data poisoned) conditions. Architectural hardening and robust model training as a countermeasure were also informative, providing practical guidance on how to strengthen models against such attacks and how to effectively promote resiliency based on the models and datasets that were tested. While a number of performance metrics were used (Accuracy, Recall, precision, Cross Entropy Loss, F1 Score), it would be beneficial to further expand on additional metrics used to evaluate effectiveness (e.g., ROC-AUC) and how these may vary in poisoned versus unpoisoned datasets with other countermeasures employed like adversarial training (such additional metrics were employed in an additional use case as demonstrated in the Appendix with an appropriate dataset). Additional defenses like robust data aggregation methods in FL could also be explored as well as incorporation of more robust datasets, which require increased levels of compute power especially for FL experimentation. Further details on statistical analysis techniques (beyond regression) would add rigor to the experimental design.

1.9 Organization of Praxis

This praxis is organized into **five (5)** chapters:

Chapter 1 serves as the introduction providing an overview of the topic area the praxis is addressing, the problem statement, research motivation, primary objectives, research questions, limitations, and scope.

Chapter 2 is the literature review and explores the current state of the art across well-aligned publications related to the praxis topic, methodologies employed, and supporting subject matter. This body of research served as a foundational component from which the praxis expanded to new areas and evaluations.

Chapter 3 discusses the methodology employed to accomplish the experimental trials of the research and how different development environments were derived to evaluate centralized and federated learning as they encounter varying levels of data poisoning attacks.

Chapter 4 showcases the results of the praxis, which were produced as a result of carrying out the methodology identified in Chapter 3.

Chapter 5 summarizes the overall takeaways of the praxis, constraints of the research, and offers recommendations for future exploration of the primary components that were evaluated.

Chapter 2 – Literature Review

2.1 Introduction

As explored in Chapter 1, data poisoning attacks can be viewed as a vector of adversarial strategies aimed at corrupting ML models by injecting misleading (incorrect) data during model training and producing inaccurate system outputs that can disrupt decision-making dependent on the model (and its foundational data) for a variety of reasons. These attacks, while somewhat new, are a significant and growing threat, posing serious risks to the integrity and security of AI systems across industries. We earlier discussed that ML systems are continually leveraged in critical domains such as defense, medicine, and autonomous vehicles, making the consequences of successful data poisoning attacks even more severe, especially in these identified industries. The expanded use of publicly available and open data sources, which consistently lack appropriate security safeguards, along with failure to conduct testing and evaluation after data selection contribute to adversary success. One recent study revealed that "the emergence of AI and its reliance on massive amounts of data has enabled adversaries to discover new ways to compromise models by introducing tainted data during the training phase" (Qin, 2023). As more mission critical applications use these models and depend on publicly available datasets and sources coupled with the growing complexity of ML and the volume of data cultivate an urgency for creating the defense strategies required to reduce the risks of data poisoning.

These kinds of attacks inhibit the use of ML in important applications, especially in settings where operational success depends on data integrity and model durability in variable environments. As we examine data poisoning and its potential, targeted

poisoning and indiscriminate poisoning are the two basic categories into which it can be viewed as an attack strategy. Here targeted poisoning seeks to disrupt / degrade the performance of a model on specific, attacker-chosen inputs, while indiscriminate poisoning seeks to decrease the overall model performance (more of a macro-effect). In both cases, adversaries exploit vulnerabilities in the data collection and training processes to manipulate model outputs and tamper with model performance. Backdoor attacks are one type of targeted poisoning attacking that can be employed. Ind doing so, a model may act normally on most inputs but then misclassify inputs that contain a specific trigger / data attribute that was set by the attacker. Ozdayi's (2021) study demonstrated that pixelating photos used by object detectors in targeted poisoning attacks can incorporate triggers into FL systems as an example. Adversaries can breach individual clients these distributed environments and poison data directly into the local nodes in FL systems, while in CL environments, such tainting usually entails altering the data pipeline before it ever reaches the central server, where training takes place.

2.2 Distributed AI

The use of AI has brought significant interest in decentralized architectures, which signal a transition from traditional, centralized models that gather, process, and train data at a single location. Global operations place data and infrastructure in different parts of the world and sometimes it is not practical to bring the data to the AI for processing (as is the case in CL). Centralized systems by and large, while common in ML often face challenges related to scalability, privacy, and security as they can become bottlenecks when handling large datasets or sensitive information as well as causing latency as operations are geographically spread. CL poses multiple risks, especially a

single point of failure, while also delaying the availability of model outputs in disparate environments where continuous connectivity is not guaranteed. Distributed AI (DAI) can help these challenges and offers a more scalable alternative by placing computational tasks across multiple nodes or clients on the edge with training occurring locally (versus centrally). DAI is usually paired with FL and can integrate blockchain, and/or multi-agent systems, especially where data segregation is important or there is a fast need of ML outputs. The shift to distributed environments is motivated by scalability (decentralized systems are better equipped to manage greater data volumes or movement) and task complexity through parallel processing that can occur across connected nodes. Redundancy is another factor as one failing node or client does not jeopardize all components of a system in this kind of configuration. DAI also helps data privacy and security by allowing data to remain local, which reduces the risks associated with central data training on a single server. In FL models, only parameters are shared between nodes / clients, preserving data privacy and mitigating the risk of data breaches as raw data is not sent centrally.

Distributed architectures can help address vulnerabilities in ML systems as seen in the argument that centralized AI systems are particularly susceptible to adversarial attacks, "machine learning algorithms used in centralized systems become prime targets for adversarial attacks given their redundancy and overuse" (Xie, et al., 2024). The decentralized nature of DAI helps to mitigate these risks, making it a more secure alternative for a number of different applications. DAI also supports modern AI needs, particularly in edge computing and blockchain integration where FL allows local training and ensuring the privacy of raw data. In many ways DAI and FL go hand in hand in

working together to achieve these benefits. DAI also shares a close relationship with edge computing, which helps speed up processing and enable immediate data analysis at the source, while also potentially using blockchain to establish a decentralized system of trust to ensure secure interactions. FL offers a promising path forward as it limits the need to transfer raw data, especially important in sensitive fields like healthcare and finance, where data control is critical (Shejwalkar, 2021).

There are many benefits of DAI when its use is aligned appropriately to operational and mission need dictated by the teams using it. It enhances privacy protection by offering multiple data streams, mitigates risk of single points of failure, and enhances overall system scalability and efficiency by distributing tasks across multiple nodes. Atul Kumar and Sameep Mehta highlight that "resilient architectures are essential to maintaining the integrity of machine learning-based systems, which are increasingly vulnerable to targeted attacks" (Kumar & Mehta, 2017). DAI can offer additional protection against these many different threats, while also delivering enhanced capabilities including real-time processing of data (versus latency delays in transmission to a central server). DAI is not without its challenges, especially when considering its compute and configuration costs that usually outpace those of centralized architectures.

These benefits do come at a price with communication overhead between nodes being expensive, especially when they are dispersed far away geographically requiring more coordination and synchronization across distributed clients. Initial development and ongoing sustainment costs are much more expensive compared to centralized systems due to a more complex architecture that needs more infrastructure with multiple training locations. Similarly establishing and maintaining trust between nodes requires more

security mechanisms, which can introduce additional complexity and cost as well. Current research suggests that adversarial attacks can exploit weaknesses in distributed architectures, emphasizing the need for stronger security protocols. "While machine learning has become ubiquitous, distributed systems are still vulnerable to new forms of adversarial evasion attacks" (Glickman, 2021). Benefits and costs exist in tension requiring organizations to properly prioritize their needs given their business or mission needs.

2.3 Proliferation of Data Poisoning Attacks

The use of AI in organizations continues to grow and data poisoning has become more common, which has made the security landscape surrounding ML systems more complicated to manage. These systems can provide profitable returns due to ML becoming more common in mission critical sectors (healthcare, finance, and autonomous systems) adversaries have likewise increased their offensive efforts. General safety and operational effectiveness, are threatened by the proliferation of data poisoning, not just the integrity of AI models Recent IEEE research, mentioned "as machine learning becomes widely deployed in sensitive applications, data poisoning attacks, where adversaries manipulate training data to degrade model performance, are an emerging threat" (Aljanabi, 2023). Malicious data that impairs the model's accuracy during the training phase can have potentially profound effects in real-world situations.

The increasing use of evolving technologies and autonomous decision-making systems without proper cybersecurity measures being implemented further exacerbates susceptibility to the influence of data poisoning. This "rapid proliferation of computing

systems that contain smart mechanisms, such as auto decision making and self-learning mechanisms, have become more common in daily life" (Yerlikaya, 2022). As a result, these systems are now more exposed to adversarial manipulation. In centralized learning models, where all data is aggregated in a single location, attackers can more easily access and corrupt the training data. FL, while decentralized in its approach, is not immune to data poisoning either. In federated systems, individual devices train local models and only share updates, yet if an attacker controls or compromises one or more local nodes, poisoned updates can still impact the performance of the universal model through regular updating.

2.4 Attack Strategies in Centralized Learning

The two learning techniques explored in this praxis are CL and FL and it is worth thinking about how adversaries may attack them separately before addressing mitigation. Traditional centralized ML systems can have a smaller attack surface generally but with concentrated data (all collected and stored in one place) which can make spread easier for larger attacks or absorb tainted data from smaller ones. Adversaries usually gain access to a centralized repository to complete a poisoning attack in CL architectures since that is where the training occurs. These attacks can be subtle, like simply flipping labels on a small subset of training data, or more complex, involving the injection of entirely fabricated data to throw the model off balance and corrupt its results. Label-flipping attacks, in which the labels of training samples are intentionally altered, are one of the most common poisoning techniques given their ease of manipulation. The attacks fool the model making it learn improper relationships between features and labels like corrupting

binary classification tasks, where flipping labels of certain data points can lead to significant degradation in model performance common in CL attacks (Shejwalkar et al., 2021).

Adding noisy or corrupt data to a training dataset is another popular tactic and example within CL that enables an attacker to trick the decision boundary of a model with varying levels of success. In mission critical applications such as those in autonomous systems or healthcare, minor changes in the training data can result in severely inaccurate predictions with terrible repercussions. Attackers complete more effective attacks in centralized settings due to having direct access to the entire dataset (they are fully aware of the data distribution) and better familiarity with the known configuration. This centrality does come with the benefit of having superior visibility and control measures compared to decentralized ones, which make it simpler to identify such attacks. These are just some of the challenges more unique to CL over its federated compatriot that we will now review.

2.5 Federated Learning: A New Attack Landscape

FL uses a decentralized approach to model training, with multiple clients collaboratively training a universal model without sharing their raw data (Guo, 2021). This approach improves data segregation, but also opens new avenues for adversaries to launch data poisoning attacks through a new attack landscape. Attackers can compromise individual clients and submit poisoned model updates to the central aggregator through local nodes or data sources. This poses vulnerabilities at both the local nodes as well as the central aggregator. Given that the central server lacks access to the clients' raw data, it is challenging to verify the integrity of these updates across connected devices, which

creates an inherent reliance on the incoming data feeds to the universal aggregator. Data poisoning attacks in FL are a critical security threat given this dependency, where adversaries carefully craft poisoned updates that degrade the performance of the universal model while bypassing possible detection mechanisms (Yueqi Xie et al. 2024). Yet the larger the federated network, the less influence an individual node has on that central aggregator as averaged trends are typically the reconciling output.

A unique challenge in FL (different from CL) is the heterogeneity of data across clients as not all may be dealing with the same kind or states of data. Each client may possess data from different distributions or altogether dissimilar data, making it difficult to differentiate between malicious updates (pushed by an adversary) and benign, naturally divergent updates occurring organically. For instance, a client with an outlier distribution of data like a medical institution serving a rare patient demographic, may produce model updates that deviate significantly from the global average (heavily healthy patients versus a population of sick ones). When this occurs, aggressive filtering or anomaly detection may mistakenly flag these honest updates as adversarial (also true of adversarial training). This can open the door for misidentification of Byzantine attacks, where some clients intentionally send incorrect information to corrupt the overall model. Byzantine-resilient aggregation techniques, such as Krum or Trimmed Mean (described in the next section), can mitigate these attacks by filtering out the most suspicious updates before they influence the global model. These defenses while not foolproof struggle to balance robustness against performance degradation, especially when a significant portion of clients are compromised. At the same time, depending upon the situation and data network, an attack may not be occurring at all given divergence of data.

2.6 Data Poisoning Transferability in Machine Learning

Data poisoning attacks degrade the integrity of operations by inserting tainted data into training sets jeopardizing the reliability and accuracy of ML models, as addressed previously. When evaluating these attacks it is important to think about the scale of their impact and how widespread corrupted data can travel. Transferability is particularly concerning as spread allows damaging data to affect many systems or components with corrupted data negatively impacting different kinds of models and architectures, including those that are separate from the original target(s). It is vital that we couple cybersecurity concepts that seek to silo this transferability and provide trusted evaluation methods for sufficiently hardening systems, if we are to maintain the integrity of ML across industries and applications. This is where again constant testing and evaluation is necessary and can no longer be viewed as one time action (which we will observe in the results section).

Transferability in data poisoning attacks is not restricted to the initial target model by any means. Poisoned data often generalizes its harmful effects to other models or classes with distinct training algorithms. Newer research has explored the study of clean-label poisoning, where seemingly benign data is subtly altered to mislead models during training (Yueqi Xie et al. 2024). The attacks in this research and others succeeded without needing access to the target model parameters, which made them highly transferable and dangerous. Additional research has shown that leveraging features like alignment and uniformity in model training increased attack success across multiple models posed by transferable poisoning strategies (Liu et al. 2023). These strategies of

wide fallout are becoming more commonplace as adversaries seek to disrupt more and more populations and uses versus smaller ones.

Transferability is of even greater concern with the decentralized training technique of FL given the connectedness of various nodes or devices across the system. The training remains at the client's location with regular transfers to the central server acting as aggregator and source for new feeds to potentially disrupt local training. At a very basic level, think of sensors observing the sky where the majority of time each node observes a blue sky. The aggregator or universal model shares this as a 'truth' to each client, the sky is blue, while not focusing on the individual attributes that are being collected. The sharing of updates to the central server does create a central point of weakness that should not be overlooked though. This point was reinforced in recent studies where poisoned updates can affect the global model which can affect all of the participating clients (Jiang et al. 2023) for example. The cross-client effect shows security risks associated with transferability in distributed settings; though, as we have identified, the upside of the distributed layout usually outweighs the risks with FL approached adopted for a variety of business or mission needs.

Effectively countering these threats and others requires well thought-out approaches to assess and quantify transferability while looking at the interconnectedness of data environments. Mitigation strategies tend to evaluate the ability of poisoned data to move across different classes and tasks to understand its potential for disruption. There are metrics like the Attack Success Rate (ASR) that quantify accuracy drops in models exposed to poisoned data compared to those trained on clean data to better gauge changes. Gradient similarity analysis between clean and poisoned data can also measure

the extent to which poisoning perturbations are able to spread. To defend against Byzantine attacks in distributed learning environments as an example, robust aggregation mechanisms such as Krum and Trimmed Mean have been proposed and studied extensively. Using a previously defined term, a Byzantine attack refers to a scenario in which one or more participating clients behave arbitrarily or maliciously, either due to software faults, compromised devices, or intentional adversarial behavior, to disrupt or corrupt the global learning process. These attacks are very challenging because the malicious updates can be crafted to appear statistically plausible while still degrading model performance. In FL, Krum can mitigate this type of threat by selecting a single client update closest (in Euclidean distance) to the majority of other updates, under the assumption that most clients are honest. Although this can be effective in small-scale settings, the computational cost grows with more dimensions and client count, hurting its ability to scale. With this challenge in mind, Trimmed Mean, which aggregates updates by removing a fixed number of extreme values from each dimension and then averaging remaining values is another tactic we can use. This coordinate filtering process delivers robustness against a bounded number of outliers while maintaining efficiency, yet is vulnerable to sophisticated attacks if the adversaries collude across dimensions. Both Krum and Trimmed Mean are effective Byzantine-resilient algorithms with their success dependent on the number of honest clients, local data distributions and the sophistication of the adversary.

A combination of different cyber measures working together is needed to squash the impact of data poisoning attacks as ML models are increasingly more data during training. Data governance is very important through the development of model training

especially paired with techniques like adversarial training and other recognition of intentional corruption. Before the model is hurt, anomaly detection techniques can assist in locating and removing questionable data entries. Similarly other benchmark datasets can be used to validate models on a regular basis and help guarantee early poisoning effect identification when used appropriately. This is where ongoing test and evaluation is really important given how dynamic ML models are (one-time testing is not effective) and will be discussed more in-depth in Chapter 5. Overall, risk mitigation in ML environments requires a number of considerations in order to effectively protect systems and incorporate a number of (previously isolated) components in engineering practices.

2.7 Attack Impact and Consequences

The consequences of a successful data poisoning attack can be huge and devastating especially in high risk areas like healthcare or autonomous systems where a compromised model can lead to life threatening outcomes. A poisoned model used for medical diagnosis for example may misclassify benign conditions as malignant and patients will undergo unnecessary and invasive treatments and huge emotional and financial burden (Sun et al., 2023). Malignant conditions may be missed and life saving interventions and treatment will be delayed with the same disastrous outcome. We can also think of other use cases like the autonomous driving industry which is at risk from poisoned models that can misinterpret sensor data and cause vehicles to make dangerous decisions like not recognizing pedestrians or misjudging road conditions. These real world examples reveal the increasing vulnerability that ML operations across sectors can have and signifies the need for more robust cybersecurity practices. As attackers develop more sophisticated methods, targeted poisoning is a concern, where specific individuals

or groups can be discriminated against by biased decision-making models without knowing it (Szyller, 2023). These attacks can exploit societal biases, spreading problems rather than containing them across populations, another output of data poisoning and its detrimental nature of spreading.

FL enables platforms to be designed with improved privacy facilitated in decentralized manner, which has its own set of vulnerabilities. The distributed nature of these systems makes it harder to track and isolate the source of a poisoning attack when one does occur. Attackers are able to hide in many clients, introduce malicious updates that degrade the global model performance and evade detection (Liang et al., 2023). The privacy requirements offered by FL make traditional security measures like data validation and anomaly detection impossible to implement, requiring the combination of technological advancements and cross-industry collaboration to be used in mitigating the impact of data poisoning. Robust defense mechanisms, including better monitoring and adaptive security, are key to protecting ML models from these ubiquitous threats.

2.8 Use of Adversarial Training

Adversarial training is a newer way to improve the security of ML against attacks like data poisoning, but still has needed development. As of the writing of this praxis, consistent results of this training remain lackluster with only some applications demonstrating reliability. It involves training models on adversarial examples or inputs designed to fool the model along with clean data to teach the models to recognize and defend against such attacks. In theory it's great and has been used in sensitive areas like finance, healthcare and autonomous systems where model integrity is crucial but needs more work especially in those areas. As ML is driving real world decisions the ability to

withstand adversarial attacks is key to accuracy, fairness and safety in automated processes and more robust training is needed (Zhang et al., 2021).

One of the key reasons adversarial training requires more research is the increasing sophistication of attacks targeting ML systems and the limited mechanisms to identify such attacks. As adversaries develop more advanced strategies there is some additional terminology that is important to know. Clean-label attacks occur when an adversary uses correctly labeled data to make the poisoned inputs appear legitimate to human reviewers and automated data validation processes while model-agnostic attacks are techniques that do not rely on knowledge of a specific model architecture, parameters, or training procedure. Given these advanced tactics traditional defense mechanisms are no longer sufficient alone in protecting ML. Adversarial training tries to equip models to be resilient against these threats by boosting their ability to generalize both benign and adversarial conditions. When models train with adversarial examples they can better detect and mitigate input manipulations to help ensure outputs remain accurate and trustworthy. Adversarial training has been tested in environments where models are running alone and seeing different data inputs. We can see this in healthcare, where it can help protect diagnostic models from subtle data manipulations that could otherwise lead to misclassifications (though it does come at the cost of training and performance inefficiencies).

Adversarial training often focuses on specific types of attacks and leaves models vulnerable to new attacks as it's hard to anticipate unknown attack strategies. The trade off between robustness and accuracy makes deployment tricky as models trained to be robust sometimes perform poorly on clean data (Guo, et al., 2022). So it's a business or

mission decision of security vs accurate output which needs to be decided by individual organizations, yet it should not be a decision of either or but rather how to balance both tenets given operational need and goals. Recent advances are trying to address these challenges through better optimization and adaptive training protocols. Techniques like transferable adversarial training help balance the trade-off between robustness and computational efficiency so models can generalize defenses across different attacks (Wang et al., 2024). As more work is done on this approach it seems possible that it will have a greater role in protecting ML systems from new threats.

2.9 Adversarial Training: A Comparative Perspective

When comparing adversarial training in centralized vs federated learning, several key differences in their approach to robustness and model security become clear. In centralized learning adversarial training is more effective because the central server has access to the entire dataset. With full data access you can generate adversarial examples that cover the whole data variability, so you can train the model more thoroughly. As a result centralized systems can be more robust to adversarial attacks because the model is trained on adversarial examples that cover the whole spectrum of potential attacks across the entire dataset. This central control allows for consistent adversarial training procedures that can be tailored to be robust to different adversarial scenarios.

On the other hand, adversarial training in FL is more fragmented and has its own set of challenges due to the decentralized nature of the data. Each client in a federated system is independent and generates adversarial examples only based on its local data. As a result, the global model may not get adversarial training fully as the adversarial examples from individual clients may not capture the full diversity of the data across all

the clients. This fragmented approach can lead to a less robust model especially when the local data distributions vary significantly between clients, a condition known as data heterogeneity. Further, the privacy preserving nature of FL adds more constraints given the lack of centralization in training. Privacy protocols like differential privacy limit the sharing of sensitive data between clients and the central server which makes it difficult to do adversarial training comprehensively across all the data in the system.

Despite all this, federated adversarial training has shown to be useful in making decentralized systems more robust and attack resistant. Researchers across industries are exploring techniques like secure aggregation, where the central server can aggregate model updates from clients without compromising individual data privacy and differentially private adversarial training, where adversarial examples are generated in a way that preserves client privacy and doesn't eliminate the benefits of FL. These techniques try to overcome the limitations of adversarial training in federated environments by making the system attack resistant without violating client privacy.

Recent research shows this is happening with adversarial training across different learning methods. "Federated adversarial training faces challenges in achieving the same level of robustness as centralized models, but innovations like secure aggregation can help improve security in federated settings" (Huang et al., 2023). Another study also stresses the need for privacy preserving mechanisms in federated systems, "differentially private adversarial training can strengthen model resilience while keeping client privacy" (Shejwalkar & Houmansadr, 2021). As FL grows, we need to address these challenges to make decentralized AI in real world applications secure and robust.

2.10 Model Training and the Potential for Model Drift

The training of ML models involves optimizing parameters over many exposures to labeled data, with the goal of minimizing error in predictive or classification tasks, which will be a focus of this praxis. Ideally, this process assumes that both the data distribution and underlying relationships between features and labels remain relatively stable across training and deployment phases. However, the real-world application of ML systems, especially in dynamic or adversarial environments, introduces vulnerabilities such as model drift and data poisoning, which can degrade model performance and undermine trust in autonomous decision-making systems, which must be kept in check and ideally quantifiable. Here model drift, refers to the degradation in predictive accuracy resulting from changes in the data distribution over time. This phenomenon may be benign (e.g., due to seasonality or evolving user behavior) or malicious, particularly when adversaries introduce subtle perturbations into the training data, such as the case could be from a data poisoning attack. As previously mentioned, data poisoning attacks deliberately inject manipulated or mislabeled data into the training pipeline, thereby biasing the learned model in favor of the attacker's objectives (Goldblum et al., 2022). Poisoned inputs can cause model drift not just by distorting decision boundaries but also by eroding generalization capability, particularly in federated or continual learning systems where data is decentralized and largely unverified (Shejwalkar et al., 2021).

Modern attacks exploit the assumptions made during training, like independent and identically distributed data, to create training points that maximize model deviation with minimal detection (Ma et al., 2022). So the poisoned model will behave normally under standard metrics but fail catastrophically under targeted inputs, a risk that’s especially high in safety-critical domains like healthcare and autonomous systems. Drift

from poisoning is not a post-hoc problem but a problem to be considered during model development through robust training paradigms, anomaly detection and continual validation (Paudice et al., 2018; Shen et al., 2021). As ML systems scale and integrate with real-time, distributed environments the dual threats of drift and poisoning require new frameworks for adversarial robustness especially in FL. Recent research has focused on hybrid defense strategies like robust aggregation in FL and spectral signature analysis for poison detection to combat covert drift induced by poisoned data (Cao et al., 2023). Ultimately securing the training process against both natural drift and adversarial manipulation is key to the reliability and ethics of AI.

2.11 Architectural Considerations

Each learning method / architecture has its pros and cons and environmental factors play a big role in selection and deployment. In centralized learning, data is collected and stored in one place, so it's easier to implement security measures. The central server has full visibility into the dataset and can apply data validation, anomaly detection and adversarial training uniformly. This centralization of the functions allows for more consistent model training and easier detection of poisoned data or adversarial intrusion into the performance of the model (more compact attack surface). Centralized architectures also create a single point of failure though. When a central server is compromised, the whole system is quickly at risk and attackers can target the single data repository and manipulate large parts of the dataset to poison the overall model. This vulnerability is especially concerning in industries where data security is critical like those previously mentioned (healthcare, finance and autonomous systems among others).

In contrast, FL distributes the training process across multiple clients, each of which maintains control over its local data. This technique is expanding in adoption given the need to synthesize large amounts of disparate data across repositories, geographies, and user bases quickly as data-informed decision-making becomes an expectation in most organizations. The decentralized approach enhances privacy (raw data never exits the local device nodes), which can be customized based on environment or operational need. This can be helpful in instances where information cannot be shared among wider user groups due to controls (e.g. PII, security classification) or the connectivity to a system of systems component. However, FL also introduces new challenges in terms of security and robustness for a working environment. Each client can be independently compromised, and the central or universal server only receives model updates, not the underlying data in raw form. This lack of visibility makes it difficult to detect poisoned updates or adversarial attacks as readily as can be done in a CL style architecture.

Table 2-1: CL Architecture versus FL Architecture

Aspect	Centralized Learning	Federated Learning
Data Location	Data is collected and stored in a central server.	Data remains distributed across multiple devices or nodes.
Data Privacy	Higher risk of privacy issues due to centralized data storage.	Improved privacy, as raw data never leaves the local devices.
Data Aggregation	Data is aggregated and processed in a central location.	Only model updates (e.g., gradients) are shared, not the raw data.
Communication Overhead	Minimal communication overhead, as all data is in one place.	High communication overhead, as frequent model updates are sent between clients and the server.
Robustness to Failure	Centralized point of failure; if the central server fails, the whole system can be compromised.	Decentralized, so the failure of individual clients does not affect the overall system.
System Scalability	Scalability can be limited by the central server's capacity and bandwidth.	More scalable, as data processing is distributed across multiple clients.
Adversarial Training	Easier to implement comprehensive adversarial training as the entire dataset is accessible.	Adversarial training is fragmented; each client generates adversarial examples based on local data, which may reduce model robustness.
Coordination Complexity	Low, as all training occurs centrally.	High, as the server must coordinate model updates from many clients.
Computation Power	Relies on the central server's computational resources.	Leverages the computational power of all participating devices.
Security Risks	Central server is a high-value target for attacks.	Distributed system reduces the impact of attacks on a single point, but is still vulnerable to poisoning attacks from compromised clients.
Data Heterogeneity	Assumes a more homogeneous dataset, as all data is centralized.	Clients may have heterogeneous data distributions, which can lead to challenges in model convergence.
Model Personalization	More challenging to personalize models for specific user needs.	Easier to personalize models for individual users based on their local data.

The table above reveals some of the primary differences and benefits between the two architectural approaches as well as offering considerations in addressing operational objectives and environmental considerations (Idika, et al., 2021).

2.12 Differences in Attack Vectors

The attack vectors are different in centralized and federated learning systems due to the location of where training occurs and the raw data resides. In CL the main attack vector is the central server or dataset located in a single spot. Before a dataset is ever used for testing or training, an attacker can contaminate it by adding wrong data at the server. Although data validation and monitoring can help reduce this risk it has influence particularly in settings where data is gathered from unreliable sources or has abnormal distributions (i.e. anomalies in values). In FL the attack surface is broader due to the decentralized nature of the system. Attackers can compromise individual clients and submit poisoned model updates to the universal model. These poisoned updates can degrade the global model's performance without the central server ever seeing the underlying data that is hurting it. This makes it harder to detect and mitigate such attacks as the central server has to rely on aggregation techniques to filter out the malicious contributions sent by the clients.

Returning to the previously defined Byzantine attacks, in which malevolent clients sent incorrect model updates to impede training, we can see their greater impact on FL over CL systems. Researchers have proposed a number of strategies to address these attacks, including Trimmed Mean, which takes the median of client updates to lessen the impact of poisoned updates, and Krum, which discards outlier changes. But these are not foolproof and often require a trade off between robustness and model performance. FL also has another challenge in the possibility of accidental or unintentional data poisoning given its distributed nature. At the local device or node level, teams working on data handling can mislabel training data, skew or imbalance data

(causing overfitting/underfitting), apply inconsistent preprocessing pipelines or use outdated datasets as some examples.

2.13 Defense Mechanisms

Following on from the different attack vectors we looked at in the previous section, defenses need to align to these different architectures. In CL, defenses usually revolve around controlling data quality at the source its single point. The centralized structure makes it easier to enforce security since all data is within a single entity (Yang et al., 2023). Outlier detection, robust loss functions and influence analysis are all ways to identify and reduce the impact of malicious samples in the data. These defenses benefit from having access to the entire dataset, so better statistical scrutiny and batch level verification. You can also see data sanitization pipelines that filter out suspicious entries before model training even begins.

The defenses for FL take a different approach since its privacy preserving and distributed nature. Since training happens across multiple clients, some of which may operate independently and maliciously, the server has no visibility to the raw data. In response, mechanisms focus on model update validation like those we mentioned earlier (Krum and Trimmed Mean) to suppress poisoned gradients by aggregating only the most reliable updates. These techniques discard malicious updates while preserving legitimate client updates (Shejwalkar & Houmansadr, 2021). Other approaches that can be considered are anomaly detection on update patterns or client reputation scoring to isolate and mitigate any adversarial behavior. This preserves data privacy and makes the global model more robust against data poisoning (Ghosh et al., 2024).

The defense strategies in CL and FL balance visibility and autonomy, true for both architectures. Centralized systems have more control over data integrity but at the cost of scalability and privacy, while federated systems have to rely on clever aggregation and anomaly detection to be resilient against compromised inputs. These differences matter a lot in real world applications like uncrewed platforms (drones or autonomous/robotic vehicles) where models need to learn from distributed sensor driven data. Data poisoning is not just a theoretical concern but a practical necessity in these missions.

2.14 Performance within Use Cases

Uncrewed systems, like unmanned aerial systems (UAS) and unmanned ground vehicles (UGVs), are changing many industries. This includes but not limited to surveillance, agriculture, disaster relief and defense operations. At the heart of these systems is unsupervised machine learning which allows them to interpret, move through and respond to their environment with minimal human input. Especially when combined with federated learning models, unsupervised learning enables these platforms to operate autonomously, absorb insights from surroundings and change behavior in real-time without labeled datasets. This leads to more autonomy, smarter decision making and operational efficiency. The sections below look at targeted performance applications of unsupervised learning in both aerial and ground based uncrewed systems, the benefits, limitations and the emerging opportunities in each domain.

Many operational scenarios UAS are deployed for tasks like aerial surveillance, disaster monitoring, environmental scanning and search and rescue missions. These airborne agents rely heavily on unsupervised learning to chart paths, make decisions and

navigate through landscapes that may be dangerous or unpredictable. One of the most critical capabilities in such cases is autonomous path planning, which allows drones to fly through hostile or inaccessible areas, reducing risk to human operators. Using onboard tools like LiDAR or visual cameras, drones use unsupervised algorithms that help interpret spatial layouts, guide them through cluttered or hazardous zones. The ability to learn from past missions and integrate environmental cues gives these systems the adaptability to avoid obstacles and flight optimization (Lyu et al., 2022).

In real-time operations UAS are also used in missions like infrastructure inspection or post-disaster surveillance. In these cases drones often perform object detection on the fly. Unsupervised models give them the ability to detect and categorize objects like buildings, vehicles or debris and even detect anomalies without labeled training data (Zhang et al., 2021). A great example of this was during the 2023 earthquakes in Turkey and Syria. The xView2 model developed at Carnegie Mellon University was deployed to assess structural damage and support search-and-rescue operations (Software Engineering Institute, 2025).On ground UGVs extend similar learning capabilities across a very different terrain. These systems operate in urban streets, isolated rural areas, remote frontiers and even contested or GPS-denied environments. Again unsupervised learning is key. Whether navigating rough terrain or driving through an urban alley, UGVs use sensor fusion – often through LiDAR and radar – to detect changes in the environment, adapt driving strategies and steer clear of obstacles (Anwar et al., 2021). Unlike UAS which typically fly in open airspace, UGVs often face moving or unpredictable ground level obstacles. In response unsupervised

learning helps them detect pedestrians, vehicles or debris by clustering objects based on shape, motion and other observed features (He, et al., 2022).

The benefits get even more pronounced when federated learning (FL) is involved. With FL these systems can train collaboratively across distributed environments without having to send data back to a central location. This is especially important in scenarios where connectivity is poor or data privacy needs to be maintained (McMahan et al., 2017). In agriculture for example FL-enabled UGVs can detect patterns like weed growth or crop disease in different regions, share model updates while keeping farm data local. Across all these domains a delicate balance must be struck. Performance must be reliable under pressure. At the same time systems must protect privacy and resist adversarial attacks. Nowhere is this more critical than in high-stakes environments like military reconnaissance or healthcare where mission success depends on secure autonomous intelligence.

2.15 Datasets and Propelling Research

Adversarial machine learning (AML), the field of hardening ML models against malicious interference and in some cases crafting the attacks themselves, relies on data! Not just any data, but high quality datasets with both natural examples and examples deliberately manipulated to deceive the model. This is especially important for training (both CL and FL) where we want to teach models not only to perform well in general but to withstand targeted adversarial manipulation. While conventional datasets lay the groundwork, adversarial examples are essential for stress testing models under attack conditions. As AML evolves so does the need for datasets that cover a wide range of

threats. Despite this need, getting suitable datasets is an ongoing problem not just in AML but across other ML applications as well.

Researchers usually turn to familiar benchmarks like MNIST, CIFAR-10 and ImageNet (some of which are used in this praxis), which have become the standard due to their availability and acceptance. For example Nguyen, et al. (2023) used MNIST and CIFAR-10 to show how neural networks can be tricked by slight perturbations. Since then these datasets have been reused in many follow-up studies. While valuable, these benchmarks are limited. Their structure, content and simplicity don't reflect the complexity of real world data or the diversity of adversarial attacks.

To bridge that gap, more focused datasets are being developed. Variants like Adversarial MNIST and Adversarial CIFAR-10 extend the original datasets by adding adversarial examples generated using methods like FGSM (Fast Gradient Sign Method) and PGD (Projected Gradient Descent). These updated datasets have really helped models better protect against varying types of attacks. Similarly, the dataset from the NIPS 2017 Adversarial Attacks and Defenses Competition brought together adversarial samples and pre-trained models to test controlled but more challenging conditions. While vision data dominates this space, other areas are now coming into focus like natural language processing (NLP). Adversarial datasets like TextFooler and HotFlip are being used to test model weaknesses by making subtle word level changes that shift sentence meaning while keeping the text readable to humans. These kinds of datasets are helping to uncover flaws in language models as these models become more integrated into daily life. The increased focus on language based systems has accelerated the effort to understand and defend against manipulation in this space with specialized datasets

emerging. New datasets like MAAD-FACE were created to test facial recognition systems against adversarial attacks, a space where misclassification can have real world consequences. LAVAN (Low-Visibility Adversarial Network) was introduced to test adversarial attacks in autonomous driving, specifically those perturbations that are subtle to humans but disruptive to machines. These efforts show a growing awareness: model defense strategies must be built on datasets that capture adversarial richness and variety. "The lack of diverse and real-world adversarial datasets hinders the development of more general defense mechanisms" (Tramer, et al. 2020). Without better data building truly robust models is a tough problem, especially for this area that is already under developed.

2.16 Use of Benchmarking and Metrics

Evaluating ML models, especially in centralized and federated learning, and adversarial training requires rigorous benchmarks and metrics. These are necessary to measure performance, scalability, security and robustness of models in different environments. Centralized and federated learning have different architectures so their evaluation metrics reflect the specific challenges of each, especially in adversarial attacks.

In CL, traditional metrics (i.e. accuracy, precision, recall) definitely matter, but recent research has expanded to more comprehensive evaluations like robustness against adversarial attacks and assessing the context of model performance. CL models are tested on very large datasets like ImageNet for image recognition or COCO for object detection. These benchmarks give a good measure of a model's performance in terms of accuracy, generalization and computational efficiency (Ma, et al., 2022). FL introduces new

challenges that require new metrics. Beyond accuracy, federated learning models are evaluated on communication efficiency, number of communication rounds and bytes transmitted between clients and the server (Idika, et al., 2021). Metrics that capture data heterogeneity like client-side model performance and client drift are important because of the decentralized and non-iid (independent and identically distributed) nature of data in FL (Li et al., 2023). Fairness is also an important metric in FL. Since client devices have varying data distributions, ensuring consistent model performance across clients is key (Mohri et al., 2020). To capture these, specialized datasets like LEAF, which has heterogeneous data from mobile devices, have become the standard for benchmarking federated learning models (Caldas et al., 2020).

Adversarial training to improve model robustness against adversarial attacks is measured by specific robustness metrics. Robust accuracy is one of the metrics to measure a model's performance on adversarially perturbed inputs. ASR is used to measure how vulnerable a model is to different types of adversarial attacks like Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attacks (Ma et al., 2023). Datasets designed for adversarial training like Adversarial CIFAR-10 and Adversarial ImageNet (as previously identified) are now essential to measure model robustness. These datasets test the ability of adversarially trained models to withstand perturbations and provide insights into the trade-off between robustness and standard accuracy. The AutoAttack benchmark is another widely used tool to measure model robustness under multiple attack strategies (Croce & Hein, 2020).

Centralized, federated and adversarial learning models need to be evaluated with a range of metrics specific to each system. Centralized learning uses traditional metrics and

large datasets, federated learning needs metrics for communication efficiency, fairness and robustness in decentralized environments. Adversarial training needs robust accuracy and attack success rates to see how well models can defend against attacks. With datasets like LEAF and benchmarks like RobustBench emerging, we can now evaluate the robustness and fairness of ML models in these different environments.

2.17 Identification of Data Poisoning Attacks

Data poisoning identification is driving new ways to respond to attacks focusing on ways to protect models without hurting accuracy. The main challenge of these attacks is that adversarial data blends in with the legitimate training data making it hard to detect. Multiple methods exist to detect anomalies and outliers that can show a poisoning attempt across data sources or models:

- Data Sanitization: One of the earliest ways to detect poisoned data is through data sanitization, cleaning the training data by filtering out anomalous inputs before they come in contact with the model. Recent work by Steinhardt, Koh, and Liang (2021) has shown the use of Influence Functions to trace model predictions back to specific training points, identify suspicious data that may have an outsized impact on the model's behavior. Then practitioners can remove or reweight the corrupted data points.
- Outlier Detection Algorithms: Advanced outlier detection techniques, like k-nearest neighbors (k-NN) and cluster-based analysis, are also effective in flagging potential attacks. We saw this in Hong and Jin (2022) who proposed a clustering-based method that identified poisoned data by observing which data points

deviated significantly from the majority class. These methods are very useful in detecting label-flipping attacks (as previously defined).

- Anomaly Detection via Model Monitoring: This is a FL centered technique, incorporating secure aggregation techniques that prevent data tampering during client-server communication and anomaly detection methods monitor model updates from each client. Using Byzantine-resilient aggregation to identify poisoned updates and keep them from being added to the global model is an example.

As soon as an attack is detected, action needs to be taken to stop its spread and harm to a model and connected systems. Multiple methods have been proposed to protect ML models from this kind of attack:

1. Adversarial Training: Adversarial training is a well known technique to strengthen models. Models can learn to detect and mitigate the effect of poisoned data by including adversarial examples, these are data points that have been tampered with, into the training process. Differentially private adversarial training which makes model updates resistant to attacks without sacrificing data privacy was explored by Nguyen et al. (2022). Since data poisoning attacks can exploit the decentralized architecture in federated learning, this is particularly important.
2. Aggregation: Given that FL is distributed, robust and secure aggregation is even more key to ensure protection. By only considering a subset of benign updates during the aggregation process, robust aggregation methods as we've seen with Krum and Trimmed Mean have been developed to reduce the impact of poisoned client updates. Building on this, Blanchard et al. (2021) introduced Multi-Krum,

which selects multiple clients' updates while excluding outliers (possibly poisoned) so it's more robust and efficient.

3. Certified Robustness: Recent work has also looked into certified defenses which provide formal guarantees a model will protect from certain types of poisoning attacks. A validated protection mechanism that limits the maximum impact of any poisoned data on the model's output was proposed by Jia, Cao, and Evans (2021). This method can help reduce the impact of malicious data while working to ensure model performance.

2.18 Research Connection

The literature review in this chapter served as the investigative foundation to launch the experimental trials and methodology that will be further explored in the next chapter that sought to answer the initial research questions posed. Through this survey it was apparent that the increasing prevalence of data poisoning attacks poses a significant challenge to both CL and FL systems across industries. This praxis strives to assess the fundamental difference between these two learning techniques, how each is influenced by data poisoning at varying levels, analyze potential transferability among data sources / nodes, explore performance tradeoffs, and reflect on architecture considerations to strengthen systems. At the same, benchmarking performance metrics around these areas arose as an area of importance in the literature review, which translated into the development of a Digital Dashboard (https://poisoning-dashboard.web.app). This application will be described in more detail in the following chapter as well as its intended future use for further research.

In reflecting across this literature review and the current state of cybersecurity, there is a noticeable gap in the ability to efficiently and effectively identify data poisoning attacks. This is an alarming realization given the proliferation of the attack types coupled with growing use of ML operations across industries and organizations, especially in mission / safety critical tasking as was identified across a number of examples. While this praxis does not focus on the initial identification of such adversarial activity, it did observe multiple levels of attack and how varying degrees of data poisoning influence model performance and degrade system outputs. As the budding field of adversarial training becomes more effective, this will help bridge the current gap in both identification and designing built-in resiliency. This is paramount as the nuanced nature of adversarial risk produces the need for more advanced, context-aware defense strategies capable of operating in diverse and evolving environments.

Chapter 3 – Methodology

3.1 Experimentation Overview

The primary aim of this research was to identify and observe the main differences between centralized and federated learning techniques, especially when data poisoning occurred across ML operations (computer vision, object detection). To accomplish this, two different environments were developed to conduct the experimentation trials: one for CL and one for FL. It is worth noting, the FL simulated environment required significant effort to create given the need to replicate local nodes working together with a universal or global model. The below high-level operational views (as previously illustrated in Chapter 1), provide an idea of what these distinct environments could look like in a Defense-related context with the first having all devices connected to one central model and the other relying on varying local models reporting to a global model that can adapt to and learn from the local models without sharing any raw data.

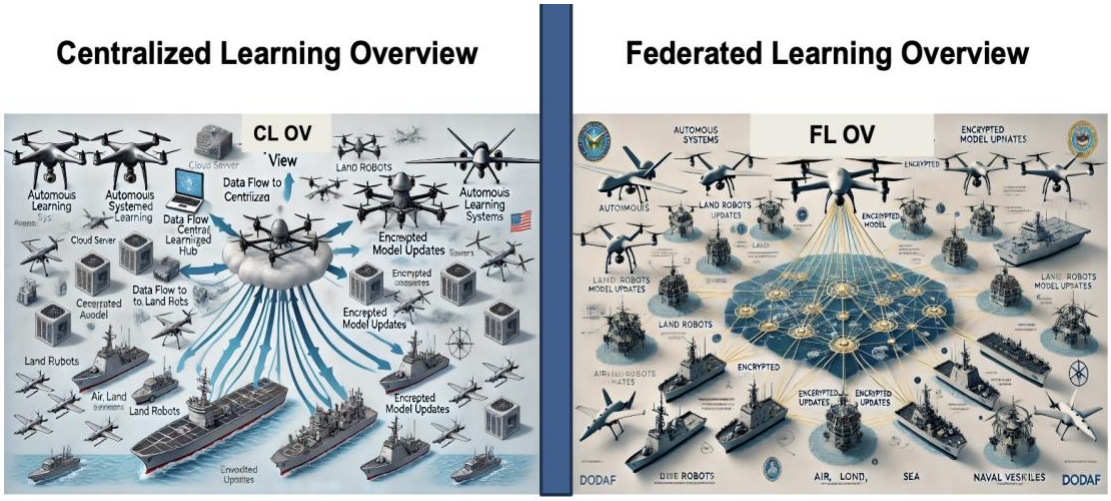


Figure 3-1: CL and FL Example Operational Views

For each of these environments (CL and FL), a series of three trials was conducted using a computer vision model, and a different object detection model for each of the datasets

(a simple convolutional neural network (CNN) for the MNIST dataset and a single layer classification head was trained on top of ResNet18 for ImageWoof). During each trial, focus was paid to evaluate general performance (or 0% poisoned data), data poisoning occurring at a 10% level, and data poisoning occurring at a 25% level to ensure robust results were achieved throughout the study for generalized poisoning (versus class specific). These trials are captured in the below graphic providing an overview of how the trials were designed.

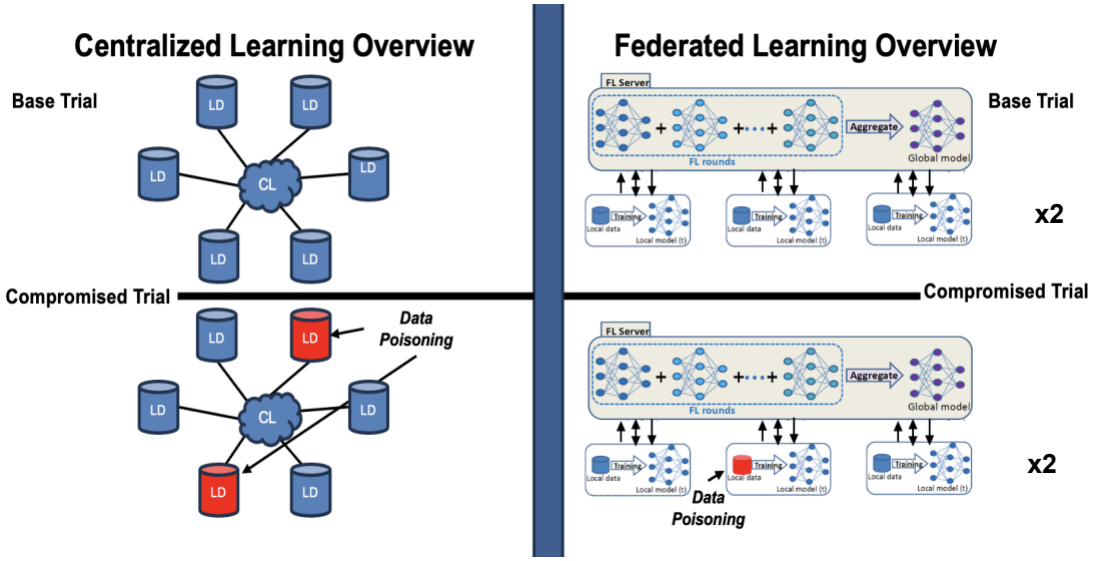
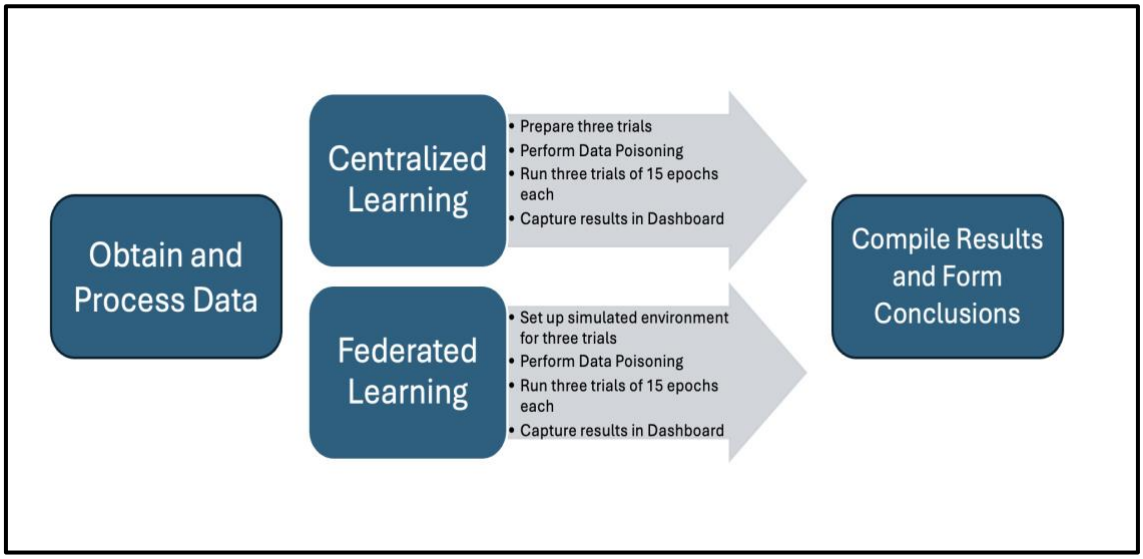


Figure 3-2: Experimental Trials Illustrative Overview

With these environments in mind, an experimentation methodology was developed to prepare, develop, and run the trials described above to test the research hypotheses. Effective datasets were identified and obtained followed by developing each of the learning environments to conduct the three trials for each of the three models tested. Each of the trials was completed with 15 epochs run for each to ensure a thorough capture of data and trends was able to be captured. At the conclusion of each trial, all results were logged in the Digital Dashboard that was developed during trial preparation. It's worth

noting that more than 15 epochs were initially run; however the results across performance metrics converged ahead of the 15th epoch, so this cutoff was selected and used to simplify reporting. From there, results were compiled with trends identified across each of the learning techniques along with data poisoning levels, and models assessing accuracy, cross entropy loss, recall, precision, and F1 score performance metrics. These metrics were selected given the need to have consistent measures across



all trials, datasets, and models, while also providing a thorough indication of model performance that is expected in typical ML Ops situations. The graphic below provides an overview of the overarching methodology that was followed to complete the research experimentation for this Praxis.

Figure 3-3: Experimentation Overview

3.2 Experimentation Approach: Dataset and Model Selection

To evaluate the impact of data poisoning on both centralized and federated learning models, two distinct datasets were chosen: MNIST and ImageWoof. These datasets were selected to represent disparate data types and challenges commonly encountered in machine learning applications as well as offering differing levels of difficulty:

- MNIST Dataset: A benchmark dataset consisting of 28x28 grayscale images of handwritten digits, widely used in computer vision and deep learning experiments.
- ImageWoof Dataset: A dataset containing 10 classes of dogs in various situations, very complex dataset, good for modeling fine-tuning capabilities of pre-trained CV models

For each dataset, a standard classifier was employed, ensuring a baseline comparison across different learning paradigms, centralized and federated. Specifically (and as previously alluded to), a simple CNN was used for the MNIST dataset, while a single layer classification head was trained on top of ResNet18 for the ImageWoof dataset.

The MNIST (Modified National Institute of Standards and Technology) dataset was a benchmark dataset in ML, especially useful in computer vision and deep learning. It had 70,000 grayscale images of handwritten digits (0–9), split into: 60,000 training images and 10,000 test images. Each image was 28x28 pixels and labeled with the digit it represented, which made it a balanced dataset with equal distribution of digit classes that also made it a great candidate for the FL trials in evenly distributing among the five clients. MNIST was great for training OCR models with handwritten digit data and was the first dataset to test deep learning architectures for neural network training. The dataset was helpful in comparing different ML algorithms initially during the design of the experimentation trials, and is recognized as a frequently used research dataset. While it is useful for introductory research, it may not generalize well to real-world handwriting datasets and its limited variability in that it lacks real-world complexities such as different fonts, backgrounds, or noise. As evidenced by these different attributes and

applications, two datasets were selected and used in this research to ensure a more robust testing scope across multiple levels of difficulty for ML functionality.

A carefully selected subset of the broader ImageNet dataset, the ImageWoof dataset was created to support studies on picture classification tasks requiring fine-grained visual distinctions through an even distribution. It included pictures of ten different dog breeds, all chosen to test classification algorithms by creating superficially similar yet hard-to-differentiate groups and contained background variables behind the dogs. The dataset consisted of more than 9,000 training images and greater than 3,000 validation images, all resized to standard resolutions such as 320px or 160px, depending on the variant. Unlike the broader ImageNet dataset, ImageWoof offers a cleaner and more targeted benchmark that emphasized the performance of ML models in discerning nuanced visual patterns, such as differences in fur texture, ear shape, and facial structure among dog breeds as well as capturing noise in the images. This made it especially useful for evaluating the robustness and generalization ability of neural networks, particularly in low-data or rapid prototyping scenarios and operational environments where background noise in imaging cannot be avoided. Furthermore, ImageWoof's moderate size and structured class distribution made it well-suited for both centralized and federated learning paradigms: in centralized environments, it enabled rapid iteration and model tuning, while in the federated environments, it offered balanced and class-consistent structure across clients supported controlled experimentation with data heterogeneity, communication efficiency, and privacy-preserving techniques. It also offered versatility across these learning settings further enhanced its value as a research benchmark.

An overview of the two datasets can be seen in the below tables (access via HuggingFace):

Table 3 – 1: MNIST Dataset

Feature	**Description**
Number of Instances	70,000 (60,000 training, 10,000 testing)
Classes	10 (digits 0-9)
Image Size	28 x 28 pixels
Image Format	Grayscale (0-255)
Data Type	Numerical (pixel values)
Target Variable	Digit (0-9) corresponding to handwritten image

Table 3 – 2: ImageWoof Dataset

Feature	**Description**
Number of Instances	12,954 (9,025 training, 3,929 testing)
Classes	10 (Dog Breeds)
Image Sizes	Full Size (original image dimensions) 320 pixels 160 pixels
Image Format	Colored Images
Data Type	Qualitative
Target Variable	Dog Breed

Exploratory Data Analysis (EDA) was performed at a limited level given that both datasets are standard, attribute-complete, and widely known to produce consistent results. The MNIST character recognition dataset is typically paired with a Convolutional

Neural Network (CNN), convolutional, pooling, and linear layers, which produce state of the art results. Hyperparameter tuning was conducted for both datasets by testing learning rates of .01, .02, .03, .04, .05, where it was found that .03 and .04 worked best for both federated and centralized learning environments. As previously mentioned, it was discovered that in both trials (datasets) and learning environments (centralized, federated) there was no observable performance increase after the 13th and 14th epochs. As a result, a total of 15 epochs were identified for all of the experimental trials. This also made it possible to complete all of the trials on a standard MacBook Pro given compute constraint, though access to the Pittsburgh Supercomputing Center (PSC) was briefly accessible for initial trials where preliminary experimentation was completed, but only for a short duration given their competing operational needs.

Two primary models were used to complete the development of the experiment trials for this Praxis, based on the two datasets used. It is important to note, that for the models used there is not a differentiation between the CL and FL trials as the same model was used; however for the FL trials, the model was used at each client. For the FL trials, an algorithm was used to aggregate the models from the clients for the global or universal model as is detailed in Section 3.4. The shallow CNN can observed in the below image:

```
self.conv1 = nn.Conv2d(in_channels=1, out_channels=32, kernel_size=3, stride=1,
padding=1)

        # Max pooling layer: reduces spatial dimensions by a factor of 2.
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

        # Fully connected layer: input features = 32 channels * 14 * 14 (after pooling
a 28x28 image),
        # output features = 10 (for 10 classes).
        self.fc1 = nn.Linear(32 * 14 * 14, 10)
```

Figure 3 – 4: MNIST Shallow CNN Model

The second model used in this research was Resnet18 used for the ImageWoof dataset for both the CL and FL trials as detailed in the below image:

```
        self.resnet = models.resnet18(weights=models.ResNet18_Weights.DEFAULT)
        num_features = self.resnet.fc.in_features
        # Replace final layer
        self.resnet.fc = nn.Linear(num_features, num_classes)
```

Figure 3 – 5: ImageWoof Model

There also was a third dataset / model originally included for the experimental trials that focused on predictive analytics using a Netflix dataset. The plan was to include this dataset as a third in addition to the MNIST and IMageWoof datasets; however after initial development in the FL trials, this third dataset was removed given the high computing cost that was necessary through Google to complete the trials as the complexity of the dataset would have required a myriad ways to be split over disparate clients. This area will be explored further in Chapter 5 as an opportunity for further research on additional datasets and would make a great follow on opportunity if minimal research funding can be obtained (approximately $1,000.00) and / or scaled FL studying is sought.

3.3 Centralized Learning with Data Poisoning

In the centralized training approach, models were trained over the course of 15 epochs, allowing sufficient learning while maintaining computational efficiency. In this context, an epoch was a complete pass through a dataset for the centralized model where every sample has been trained and the parameters of the model have been updated after each completion. The use of 15 epochs was selected for consistency across all trials as the results in thereafter epochs did not change with trends converging prior to the 15th epoch. It is also worth noting that given the size of the datasets and models used (particularly the later Federated Learning Trials), 15 epochs was the maximum number supported by a standard MacBook Pro device. The training for this experimental trial followed a

standard supervised learning paradigm, using labeled data to optimize model parameters, which was consistent across both datasets (MNIST and ImageWoof). However, the Federated Learning Trials tested different problems in ML, such as training a CNN from scratch versus fine tuning a pretrained model, which was another major difference between the two techniques / architectures.

To assess the impact of adversarial manipulation, experiments were conducted under three poisoning conditions:

- **0% Poisoning**: No modifications were made to the training or test data, serving as the control condition.
- **10% Poisoning**: 10% of the training set was changed with "adversarially injected data," designed to disrupt model training.
- **25% Poisoning**: 25% of the training set was changed to introduce adversarial noise, affecting model evaluation and training.

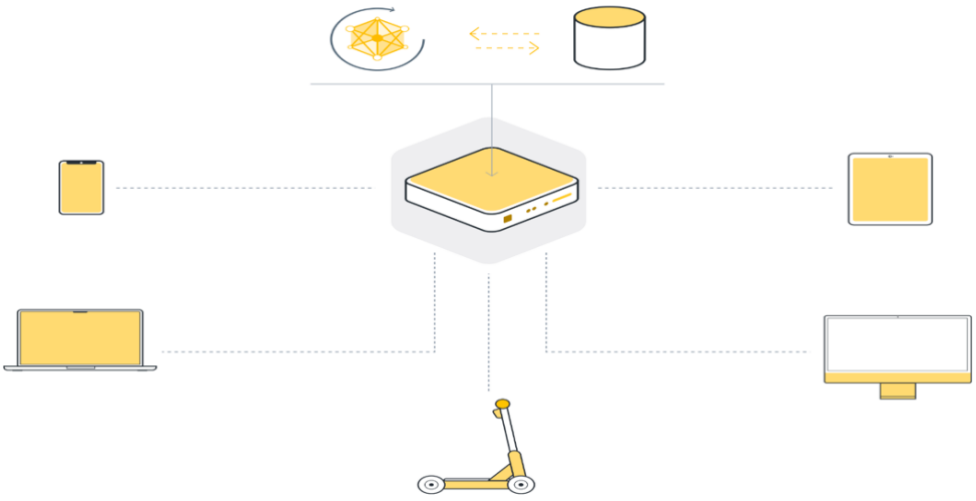
To systematically evaluate model performance under these conditions, five primary performance metrics were recorded after each training epoch across both trials (both for centralized and then for federated learning):

- **Model Accuracy**: This is the fraction of total predictions that were exactly equal to the true labels based on the training data.
- **Cross Entropy Loss**: The average negative log-likelihood of the true class under the model's predicted probability distribution. Here, lower values signified the predicted probabilities placed more mass on the correct labels with the inverse being true as well.
- **Recall:** The proportion of all actual positive instances that the classifier correctly identified while running.
- **Precision:** The proportion of predicted positives that were actually positive (not false positives) returned from the model.
- **Macro Averaged F1 Score:** Here we used the computed the F1 score (harmonic mean of precision and recall as previously defined) separately for each class, then took the unweighted arithmetic mean across classes. This treated every class equally regardless of its frequency, especially important for the FL trials.

3.4 Federated Learning Implementation

The Flower library was used across each of the FL experimental trials to mimic a distributed system with five client devices in order to investigate the impacts of data poisoning in a FL environment (this was the most accurate simulation environment given compute constraints). Flower is an "open-source framework that is available in multiple programming languages to provide a realistic simulation of FL" (Beutel et al., 2022). As previously mentioned, this required the greatest effort in development for this research given the complex nature of the environment with multiple local models and devices paired with a global or universal model. To minimize additional variables, the classes of the datasets (both MNIST and ImageWoof) were evenly distributed among the five 'devices' in this configuration to ensure balance was achieved. While real-world instances would likely have disparate data class distributions, evaluating such diversity in classes was not an objective of this praxis as the focus was on performance related to both learning technique (CL and FL) along with susceptibility to data poisoning.

Figure 3 – 6: Example Flower Configuration with 5 Clients (Bo Kan, 2023)



Python was used for the experiments across the FL experiments to develop. All the simulations were run on a Mac, so a controlled and realistic experimental setup with only

one device to simulate a wider FL system using Flower, which was done through Google Colab. The training dataset was split across 5 devices, to have a non-IID (independent and identically distributed) distribution to reflect real-world FL scenarios. Each client trained a model on its local data and updates were aggregated to improve the global model but no raw data was shared. The data was partitioned in a stratified way to have a representative distribution across clients for 15 epochs as in the Centralized Learning experiments. Here an epoch can be defined as a complete pass through a dataset for the global model where every sample and node has been trained and the parameters of the local and global models have been updated after each epoch.

The federated learning setup was configured with the following parameters:

- **Number of Clients**: 5 client devices simulated.
- **Local Epochs per Client**: Each client performed 5 local training epochs before communication.
- **Batch Size**: 32 samples per batch.
- **Optimization Algorithm**: Stochastic Gradient Descent (SGD) with a learning rate of 0.01.
- **Aggregation Method**: Federated Averaging (FedAvg) was used to combine local model updates into a global model.
- **Communication Rounds**: 50 rounds of federated training were performed to ensure convergence.

For purposes of the above parameters, a batch was defined as a subset of the training data that is passed through the model with the average of the loss calculated for each image, and the gradient calculated based on that average loss. Representative distribution referred to exhibiting the same properties (and can be represented) by the same parameters as the true distribution of the general population.

Specifically, for both datasets, the following settings were followed to accomplish the trial:

Loss Function	Cross Entropy Loss
FL Algorithm	FedAVG
Optimizer	SGD
Learning Rate	MNSIT: 0.01 ; ImageWoof: 0.003
Client Optimizer	SGD
Client Learning Rate	0.003
Epochs	15
Communication Rounds	15
Client Epochs	1
Clients	5
Batch Size	32 (for both)
Dataset Size	MNIST: 60,000 training, 10,000 testing ; ImageWoof 9,030 training, 3,930 testing

Figure 3 – 7: FL Trial Parameters

For these experiments (FL), we used the Federated Averaging (FedAVG) algorithm to train models across multiple decentralized clients without sharing raw data to simulate a distributed environment. The loss function was Cross Entropy Loss which is suitable for classification tasks as it measures the difference between predicted probability distributions and true class labels. We used Stochastic Gradient Descent (SGD) as the optimizer at both server and client level. The global model's learning rate was set to 0.01 for MNIST and 0.003 for ImageWoof as the tasks are different. All clients used SGD as the local optimizer with a uniform client learning rate of 0.003. We trained for 15 rounds, each client trained for one local epoch per round. This configuration balances the computational efficiency on the client side and the convergence of the global model. We had 5 clients in total. Each client trained with a mini-batch size of 32 which is a practical trade-off between training stability and performance.

Two datasets were used to evaluate the model: MNIST and ImageWoof. MNIST has 60,000 training samples and 10,000 testing samples, grayscale images of handwritten digits. ImageWoof is a more complex classification task with 10 dog breeds and 9,030

training images and 3,930 testing images. This setup allows us to test the three different levels of data poisoning (0%, 10%, 25%) on both a simple and a challenging dataset in a federated learning environment.

Unlike centralized training, FL involved iterative communication between clients (local models) and a central server where the universal or global model resided. After each communication round, the global model's performance was evaluated using the test set, recording the same metrics of accuracy, cross entropy loss, recall, precision, and the macro averaged F1 score as was used in the CL trials. The below flow chart captures the overall activities that were completed to develop the FL environment and related trials, specifically outlining the role of the global model and the training across local models with aggregation back to the central server.

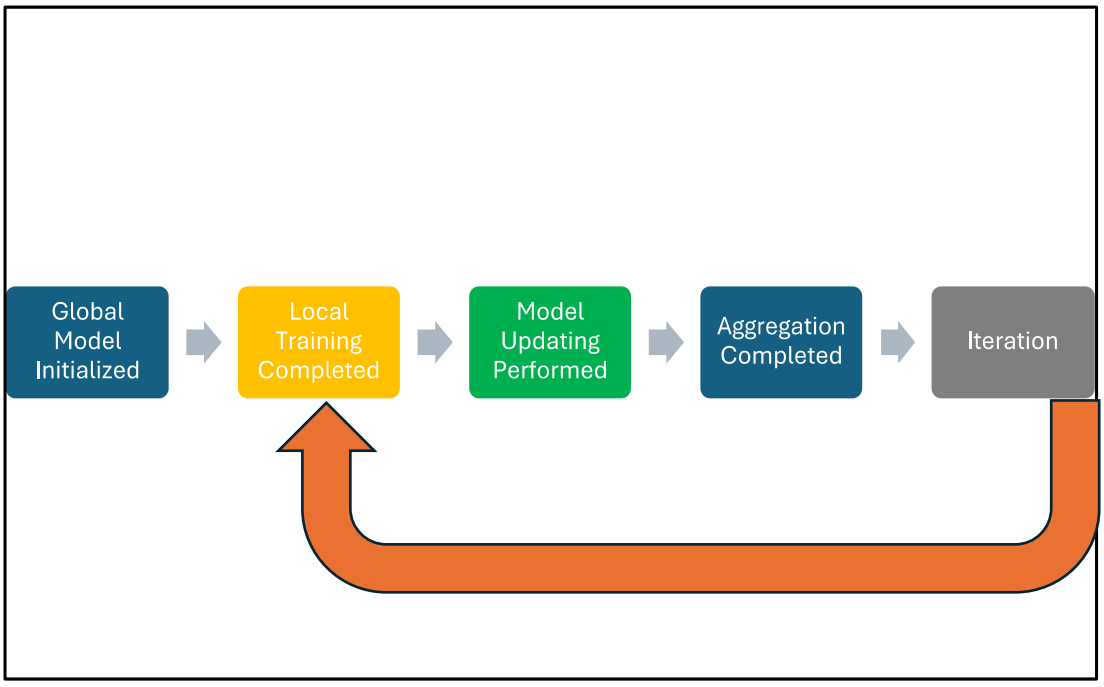


Figure 3 – 8: FL Process Flow Chart

The high-level process flow for building the federated learning environment using the Flower library is depicted in Figure 3-8. The training process followed an iterative

protocol. First, a global or universal model (this was the shallow CNN) was created on a central server and distributed to all 5 clients. Each client then trained a local copy of the global model on their own private data, keeping raw data locally and not relaying it back to the universal model. After local training occurred, clients sent their model updates (gradients or updated parameters) back to the central server. The server then aggregated these updates using the FedAVG algorithm to produce a new global model. This updated model was then sent back to the clients and the process repeated for multiple rounds across the 15 epochs. The global model improved over time while raw data was never shared outside of the client instances to simulate what would occur in a distributed AI environment where FL was training the models. For the purposes of reporting, the universal model was used to report the macro results of the FL trials, however each client trial can also be observed.

3.5 Poisoning Conditions

The main objective of this research was to understand the effect of data poisoning on ML and how it may change. During the praxis, three different poisoning levels (0%, 10%, and 25%) were used in both the centralized and federated learning testing to assess their impact under different conditions aligned with what was identified in the literature review. The poisoned data was distributed across different clients to evaluate both localized and widespread adversarial effects in the federated trials to simulate a distributed AI environment and solely the centralized model in the CL experimental trials. While poisoned data conditions were present in four of the trials (two for centralized, two for federated), it was of greatest interest to observe the differences from

the base trial of 0% data poisoning through the highest level (25% data poisoning) and how each training technique changed comparatively to the other.

These levels were selected based on recent surveys that showed their effectiveness in evaluating the robustness of systems against attacks as we better understand the attack surface of different configurations and the ability for attackers to scale their assaults on systems. One of those studies found that even a small percentage of malicious participants can hurt the model a lot, 10% of malicious data resulted in a noticeable drop in accuracy (Tolpegin, 2020). Another study found that increasing the attack level from 10% to 25% resulted in a much bigger impact of the attack, so we need to test higher poisoning levels as well (Fang, 2021). By testing these different poisoning distributions better evaluation of the resilience across the spectrum of attacks, from benign (0%) to moderate (10%) to more aggressive (25%) was able to be evaluated on ML classification tasks using different models and datasets.

3.6 Performance Evaluation Metrics

Once the testing conditions and development environment identified above were selected, it was important to identify a consistent way of evaluating model performance given the differences across models, datasets, and learning techniques. Researchers have increasingly added precision, recall, and the F1 score to measurements like accuracy and cross-entropy loss when assessing ML models in order to better solve particular problems and glean a more comprehensive picture so the same was done for this praxis. Accuracy, the proportion of correct predictions offers a broad measure of performance but is often misleading in imbalanced datasets where one class was dominant (Nguyen et al., 2022). This can be especially true when skewed datasets exist and even more so in FL

environments where they are unevenly distributed, making reconciliation by the aggregator more difficult. To measure a model's confidence, consistently cross-entropy loss is used, which penalized incorrect predictions by their certainty and was the foundation metric for training neural networks (Goodfellow et al., 2016; Huang et al., 2023). Yet in cases like medical diagnostics and fraud detection where the cost of false positives and false negatives was vastly different, precision, the ratio of true positives to all predicted positives, and recall, the ratio of true positives to all actual positives, to understand model behavior are used (Kim et al., 2024). The F1 score or harmonic mean of precision and recall, was a balanced metric when neither false positives nor false negatives could be ignored (Zhu et al., 2023). While accuracy was good for balanced classification tasks and cross-entropy loss for training the model, precision, recall and F1 score gave a more detailed view of performance, especially in high-stakes or imbalanced applications (Li et al., 2023). All of this information was considered to make the most effective decision on which metrics to include in evaluating the experimental trials and each was selected to track performance and report on results. We will now explore each of these performance metrics individually to set how calculations were reached and the figures are relevant.

Accuracy can be defined as the simplest and most common metric for classification tasks, it is the number of correct instances over the total number of instances:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

It's good for balanced datasets where class distributions are roughly equal, however accuracy is misleading when there's class imbalance (Nguyen et al., 2022) as previously mentioned. For example, in a 90:10 class ratio dataset, a naive classifier that just predicts the majority class will get 90% correct even making this a somewhat useless statistic and not necessary representative of the model. In imbalanced scenarios, we often use precision, recall and F1-score instead (Zhu et al., 2023) to gain better visibility into actual performance and better assess performance.

Cross-entropy loss (also log loss) is a common metric for training and evaluating probabilistic classification models, especially neural networks and logistic regression. It really measures the difference between the predicted probability distribution and the actual class labels. Mathematically it is expressed as:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij})$$

where N is the number of samples, C is the number of classes, yij is the true label (1 for correct class, 0 otherwise), and yij is the predicted probability for class j (Goodfellow, Bengio, & Courville, 2016). Cross-entropy loss, unlike accuracy, measures prediction confidence. An uncertain model gets a lower penalty than one that assigns high probability to wrong predictions. That's why gradient based optimization in deep learning models works well with cross-entropy loss (Huang et al., 2023). When all misclassifications are equal in balanced datasets and applications (e.g. image classification in a controlled environment), accuracy is higher. Since cross-entropy loss gives a continuous differentiable function to update the weights, it's essential for deep

learning model training especially in multiclass problems. According to recent research, accuracy alone is not enough to evaluate deep learning models especially in fraud detection and medical diagnosis where misclassification cost varies (Kim et al., 2024).

The next metric is precision which is used to measure how well a model can predict the future. The definition is the number of true positives or accurate positive predictions to all positive predictions including false positives a model may produce (both true positives and false positives):

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

When the cost of false positives is high, precision is useful. For example, important emails may be missed in spam detection if a false positive (marking a valid email as spam) occurs. But false negatives or missed positives are not considered by precision alone which may be a drawback in some cases (Kim et al., 2024).This basic example shows that the metric alone cannot be trusted as a sole measure of how well a model is performing.

Recall, sometimes called sensitivity or true positive rate, measures how well a model can find positive data points. It's defined as the number of true positives divided by all actual positives (true positives and false negatives):

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

In applications where it's more costly to miss a positive than to include false positives, recall is key. For example, in cybersecurity, a false negative like missing a hostile action or attack can be disastrous and it is better to air on the side of caution.

Recall alone though “doesn’t account for the number of false positives which can result in too many false alarms” (Li et al., 2023) at the same time. Here too, this statistic in isolation is not enough to assess overall performance.

The F1 score is the harmonic mean of precision and recall, taking into account both false positives and false negatives, building on what we’ve covered on both of those already (although separately). It’s defined as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score is useful when dealing with imbalanced datasets where one class is more frequent than the other as we mentioned before. It gives a more complete view of a model’s performance by balancing the trade off between precision and recall, which is useful in tasks like information retrieval or fraud detection (Zhu et al., 2023) which are common in cybersecurity. For these reasons it was selected as one of the five performance metrics used to evaluate the experimental trials of this praxis. As we think about all of these metrics and their ability to provide insight into model performance together, the below table offers a good comparison.

Metric	Strengths	Weaknesses	Best Use Cases
Accuracy	- Intuitive and easy to interpret	- Misleading in imbalanced datasets	- General classification tasks with balanced class distributions
	- Useful for balanced datasets	- Treats all errors equally	
Cross-Entropy Loss	- Measures confidence of predictions	- Less interpretable for end-users	- Model training in deep learning
	- Differentiable for optimization	- Not ideal as a standalone evaluation metric	- Multi-class classification
Precision	- Highlights false positives	- Ignores false negatives	- Spam detection
	- Useful when false alarms are costly	- Can be high even with poor recall	- Medical screening (minimizing unnecessary treatments)
Recall	- Emphasizes false negatives	- Ignores false positives	- Disease detection
	- Critical when missing a positive is costly	- Can be high even with low precision	- Security applications (e.g., intrusion detection)
F1 Score	- Balances precision and recall	- Ignores true negatives	- Imbalanced classification problems
	- Useful for imbalanced datasets	- Harder to interpret than accuracy	- NLP tasks like named entity recognition

Table 3 – 3: Performance Metrics Overview

When evaluating the impact of data poisoning attacks on both CL and FL, a combination of all these metrics as shown in the table above (Table 3 – 3) was used. Each provided different insights into the performance of the model under attack on both datasets and levels of data poisoning. Accuracy was a general measure but often insufficient in the presence of class imbalances introduced by poisoning. Cross-entropy loss helped measure uncertainty and penalized large deviations in predicted probabilities which was important when analyzing the effects of subtle strategic poisoning. To further break down the impact, precision and recall were important in understanding false positive and false negative rates respectively especially when the attacker intentionally mislead the model to wrong classifications (number or dog breed). The F1 score as a harmonic mean of precision and recall gave us a balanced evaluation especially when the

attacker's goal was to either reduce the detection rate or inject erroneous samples with minimal false alarms. By combining these metrics we got a comprehensive view of how data poisoning affects both centralized and decentralized learning systems and the vulnerabilities and nuances of the model.

Early into the research phase of this praxis, it was determined that accuracy and cross entropy loss would be two key performance metrics to consistently assess the datasets, models, and learning techniques being evaluated across the varying levels of data poisoning (0%, 10%, 25%) to assess the identified research questions. However, as initial testing was conducted, a more robust and multidimensional evaluation criteria emerged to reflect the more nuanced impacts of data poisoning with the addition of the other metrics identified in this chapter. Each indicator served a unique purpose, while also worked together to convey the overall cost of model performance to data poisoning and evaluating the resiliency of the learning techniques across trials. Now that each of these metrics has been defined and compared, the following highlights why each was selected for this research in building a more comprehensive assessment of data poisoning outside of basic research.

- Accuracy was the overall correctness of predictions but not enough on its own for imbalanced or adversarial datasets.
- Precision measured how often positive predictions were correct, important when false positives can be operational risk (e.g. misidentifying threats).
- Recall measured the ability of the model to detect all true positive cases where missing a positive instance can be fatal.

- Cross Entropy Loss calculated the unpredictability and instability brought on by poisoning through measuring the difference between true labels and anticipated probability distributions.
- The F1 Score, balanced metric particularly in the presence of adversarial perturbations, was the harmonic mean of precision and recall.

These criteria made both isolated and integrated evaluations of model behavior under controlled adversarial influence while also showing how the two different datasets performed with each of their respective models across the two learning techniques (CL and FL). It was these identified performance metrics that were used to build out the Digital Dashboard (previously mentioned in this chapter), which served as the primary means for data collection, aggregation, and presentation of the overall results of the experimental trials. This dashboard was used to then conduct comparative analysis and determine conclusions from the study and will be explored in more depth through Chapter 4.

3.7 Comparison and Analysis

After the experimental trials, the performance of the models trained in both centralized and federated learning setups was compared across the different poisoning levels (0%, 10%, 25%). Each of the five different metrics was analyzed to see the differences in performance between the two learning techniques and overall vulnerability to data poisoning attacks. From there, the impact of data poisoning was assessed on both short-term degradation and long-term learning stability across the 6 trials (3 for CL, 3 for FL) to show that not just one level of difficulty was showing the observed characteristics and performance recorded, and to compare between two very different datasets and

models based on their complexity. A third was added after the end of the trials and is recorded in the Appendix of this praxis.

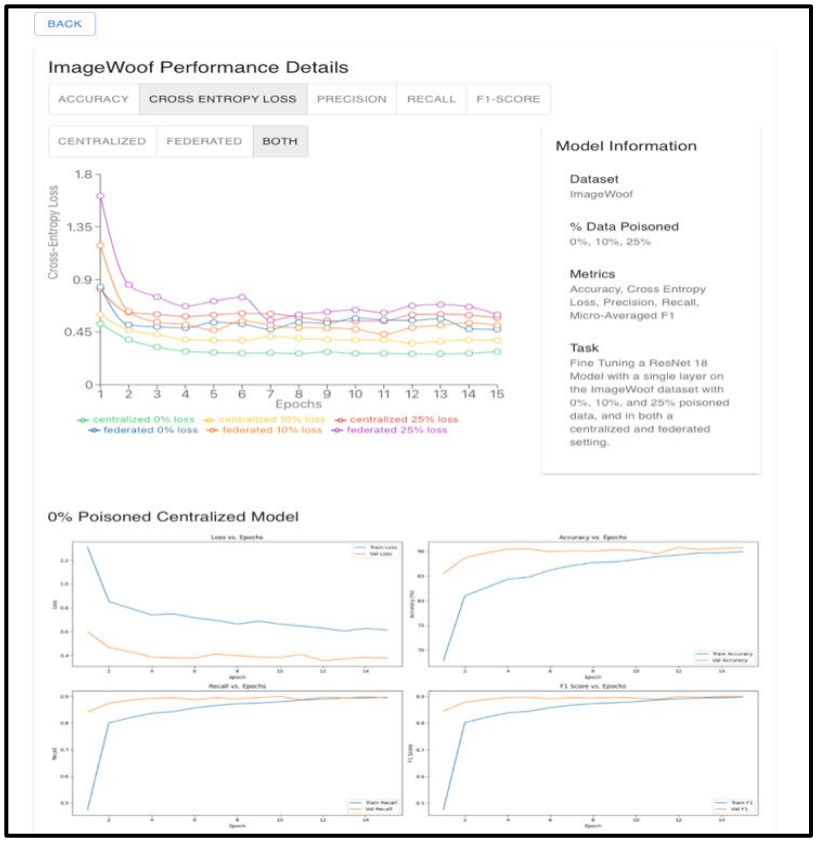
This comparison gave insights into where and how the models when exposed to adversarial data manipulation and when attacks proved most successful across 15 epochs. The results wind up telling us how resilient (or not) CL and FL are against poisoning attacks and what the implications are for securing ML systems. As trends direct toward distributed environments, this also allows for the ability to identify different ways to better harden these systems, especially in mission or safety critical applications. While the federated sim using the Flower Library was challenging in development, it did produce an effective testing environment that was helpful in assessing this newer approach.

3.8 Digital Dashboard Development

A dashboard was developed to log and compare results across the models even before the trials began given the number of variables under investigation (datasets, models, poisoning levels, and learning techniques) in this research. The dashboard (accessible at this link) was built to provide visualization into model performance and give a better comparative view in one central space. It collected, processed, and displayed critical metrics as previously identified over various experimental conditions and allowed me to efficiently rerun trials or make adjustments before too many cycles were completed. Performance patterns and trends were visualized together in CL where all data is combined into one model. A bit differently the dashboard shows some model convergence, poisoning disparities, and local training updates for FL in both client and aggregate views. Users can compare the results of the two learning approaches using the

dashboard and evaluate the trade-offs between security, efficiency and privacy while being mindful of other variables that support this decision-making (cost, infrastructure, data). The dashboard was designed with interactive elements to help support decision making when evaluating ML model safeguards taking different form based on training. While this Praxis didn't focus on development or data visualization as a main goal, it is worth sharing that this dashboard can be used for other research as well, especially those that are comparing learning approaches as inputting new data in it is a low effort. An example view of the dashboard can be seen below in Figure 3 – 9, where cross-entropy loss performance results are highlighted for the ImageWoof trials for both CL and FL. It is best to experience the dashboard through its intended dynamic nature through accessing the link included above.

Figure 3 – 9: Example View of Dashboard



3.9 Conclusion

This chapter focused on the steps taken and methodology used to carry out the praxis research objectives and address the central research problem as identified in the first chapter. After getting started on the experimental trials, it was realized that the selection of multiple datasets coupled with the assessment of two different learning techniques did make this research more complex and time-consuming in the development phase, however it was worthwhile in that it provided greater insight into the role of data difficulty and ML architecture, which proved to be important when analyzing the resulting outcomes of the experimentation. Had only one type of dataset and/or learning technique been employed (i.e. CL and MNIST), then conclusions could not have been as well informed in testing the research hypotheses of this praxis. While there were opportunities to enhance this methodology to include greater focus on preventative measures for data poisoning attacks, such as adversarial training and intrusion detection, given the scope of this Praxis and amount of development these areas were not feasible additions (though recommended for further research). The use of multiple performance metrics (accuracy, cross entropy loss, precision, recall, F1 score) served as effective mechanisms to gauge performance, versus a more limited scope of criteria (one or two of these). It is important to remember that different organizations will have different priorities over these indicators as their uses change with different priority.

Chapter 4 – Results

4.1 Establishing Desired Outcomes

This praxis set out to conduct experiments where data poisoning was applied to CL and FL models to see if/why one was more secure than the other and to analyze any correlations between learning method and attack spread, specifically as they relate to the identified performance metrics. In completing this research, all initial goals were met with results recorded and new insights into data poisoning resulting in a foundation for future research as these areas proliferate. The following four areas align with the original research objectives and were used to evaluate the results of the experimental trials.

- Quantitative Assessment of Resilience: Assess which learning technique (CL or FL) was more robust when faced with different levels (10% and 25%) of data poisoning. Overall, which did better across all reported metrics?
- Analysis of Transferability for FL: Determine how adversarial perturbations crafted in one data source can transfer (or not transfer) to another or the global model (FL), the cross-paradigm vulnerabilities. Did transferability or spread occur?
- Identification of Key Performance Deterioration Patterns: Evaluate how critical metrics (Accuracy, Precision, Recall, Cross Entropy Loss, F1 Score) changed after data poisoning attacks occurred.
- Architectural Insights for Mission / Safety-Critical Applications: Identify data driven architectural guidelines to improve system security and robustness especially for safety-critical environments or when real-time data outputs are essential.

This praxis sought to fill a gap in existing AI Engineering research by examining data poisoning not just as an isolated risk, but as a transferable attack vector across fundamentally different models and datasets, while having assessed the differences of the attack surfaces of CL and FL configurations focused on object classification. This area of research is of great importance as many organizations across industries consider adoption of more automated processes using ML and move to DAI systems where FL will play a significant role in their overall security posture, particularly for those organizations seeking enhanced data privacy, which FL offers. It is very important to understand the performance tradeoffs FL exhibits for improved privacy as infrastructure decisions are made and model training is identified.

4.2 Establishing Assessment Criteria

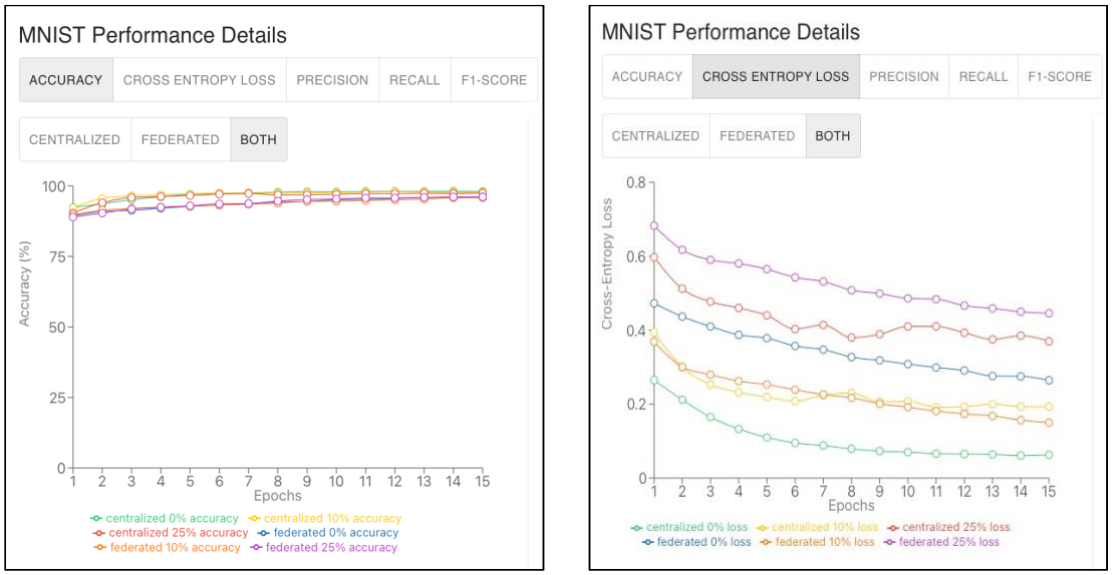
In addition to the multiple datasets and multiple models/learning techniques used in this research, it was also desired to test performance through different levels of attack, which are detailed in the next section. Across each of these variables, the same five metrics were used to evaluate trial performance: Accuracy, Precision, Recall, Cross Entropy Loss, and F1 Score as was detailed in Chapter 3. Given that different industries and applications have varying priorities, this approach was identified and followed in an effort to make the research industry-agnostic and provide a general foundation for developing effective architectural decision-making as widely as possible. Additional use cases can easily be run through the platform and dashboard developed to target specific evaluation that may be desired based on industry or operation. While the experimental trials largely focused on computer vision and classification, additional efforts were made

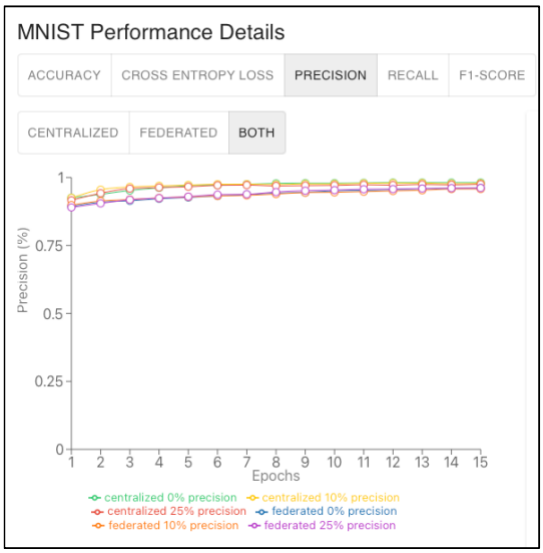
around predictive analytics as can be observed in the later trial performed with the Xray-14 dataset as described in the Appendix.

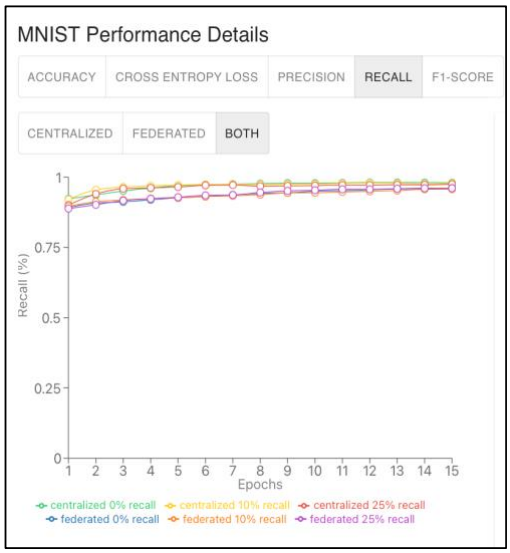
4.3 Assessing Experimental Trial Outputs

The experimental trials were divided into three main conditions per training technique: **Base Trials** (where no attack occurred or 0% poisoned data), **Compromised Trials 1** (exhibited a 10% data poisoning attack), and **Compromised Trials 2** (exhibited a 25% data poisoning attack). The following graphics showcase the performance of each dataset and respective model across CL and FL training along with the three data poisoning trials completed for each. Graphs represent the performance based by metric with each of the five having its own graph for each of the two datasets and models (a total of 10 overall graphs is thus presented):

Figure 4-1 MNIST Overall Results



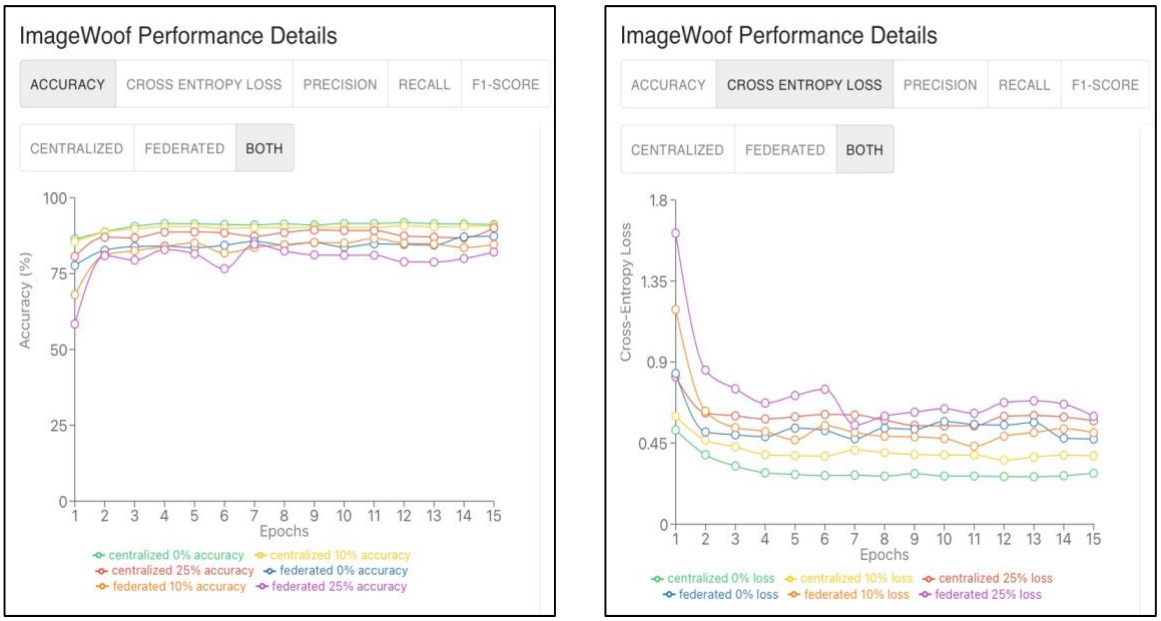


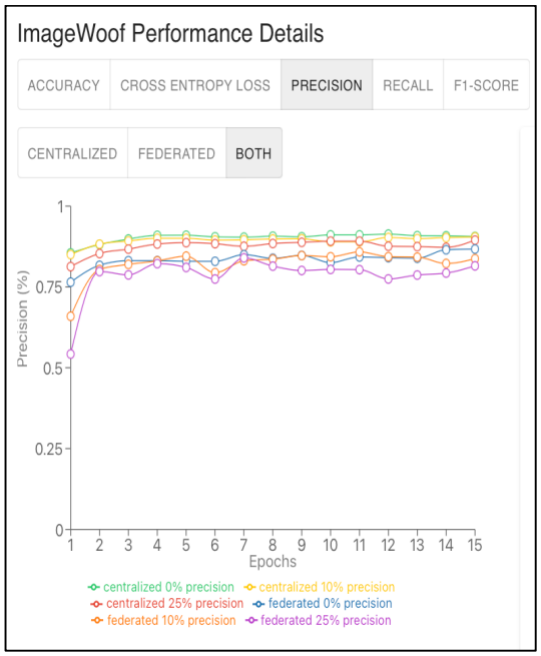


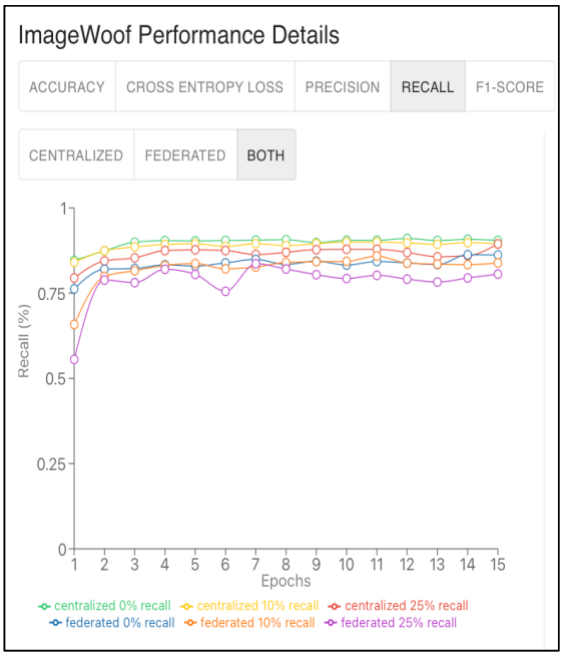
In reviewing the MNIST dataset (as can be observed in Figure 4-1) results which was paired with the shallow CNN developed, we can observe that FL consistently outperformed CL across the epochs in terms of accuracy and recall with the performance gap widening as the percentage of poisoned data increased. Note: it is best to view these graphs interactively via https://poisoning-dashboard.web.app to gain the best visual. As

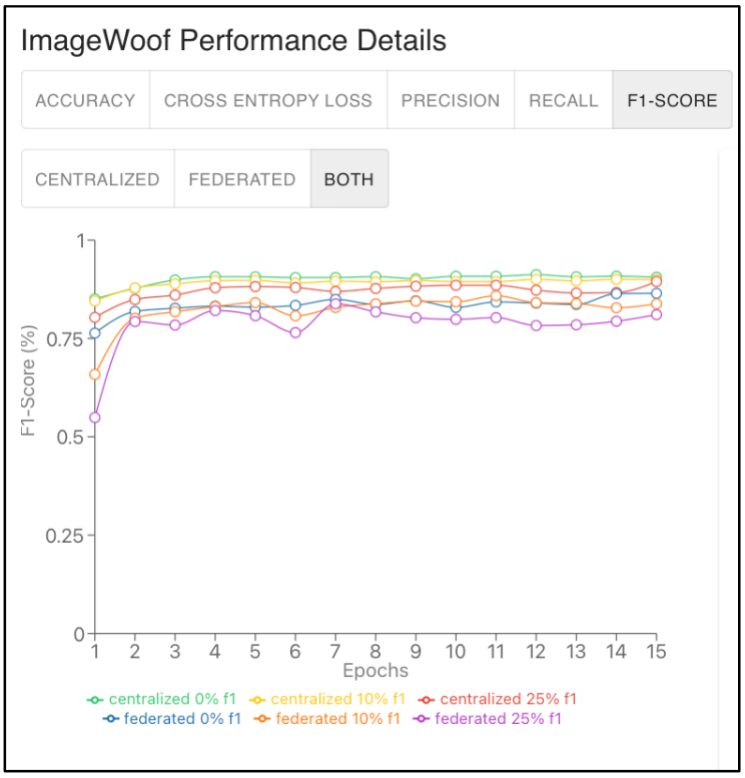
to be expected, the Base Trial without any poisoned data performed the best across both CL and FL with only minor variation between the two learning techniques for these trials. These results support the hypothesis that FL architectures can be more resilient to systemic model degradation caused by adversarial data poisoning. The MNIST dataset is the simpler of the two datasets tested in this research, so it was necessary to identify and select a second dataset to ground these trials in more realistic simulation to those datasets of an operational nature. While the MNIST dataset is noted as largely an academic dataset, it served the trial of benchmarking learning techniques and initial data poisoning. In pulling from the larger and more complex ImageNet dataset, the ImageWoof subset was selected and paired with a more robust Resnet18 model then trained a classification header on top of it. This second series of trials for the ImageWoof dataset is displayed on the next page.

Figure 4-2 ImageWoof Overall Results




ImageWoof Performance Details
ACCURACY
CROSS ENTROPY LOSS
PRECISION
RECALL
F1-SCORE
CENTRALIZED
FEDERATED
BOTH
Precision (%)
Epochs
centralized 0% precision
centralized 10% precision
centralized 25% precision
federated 0% precision
federated 10% precision
federated 25% precision


ImageWoof Performance Details
ACCURACY
CROSS ENTROPY LOSS
PRECISION
RECALL
F1-SCORE
CENTRALIZED
FEDERATED
BOTH
Recall (%)
Epochs
centralized 0% recall
centralized 10% recall
centralized 25% recall
federated 0% recall
federated 10% recall
federated 25% recall


ImageWoof Performance Details
ACCURACY
CROSS ENTROPY LOSS
PRECISION
RECALL
F1-SCORE
CENTRALIZED
FEDERATED
BOTH
F1-Score (%)
Epochs
centralized 0% f1
centralized 10% f1
centralized 25% f1
federated 0% f1
federated 10% f1
federated 25% f1

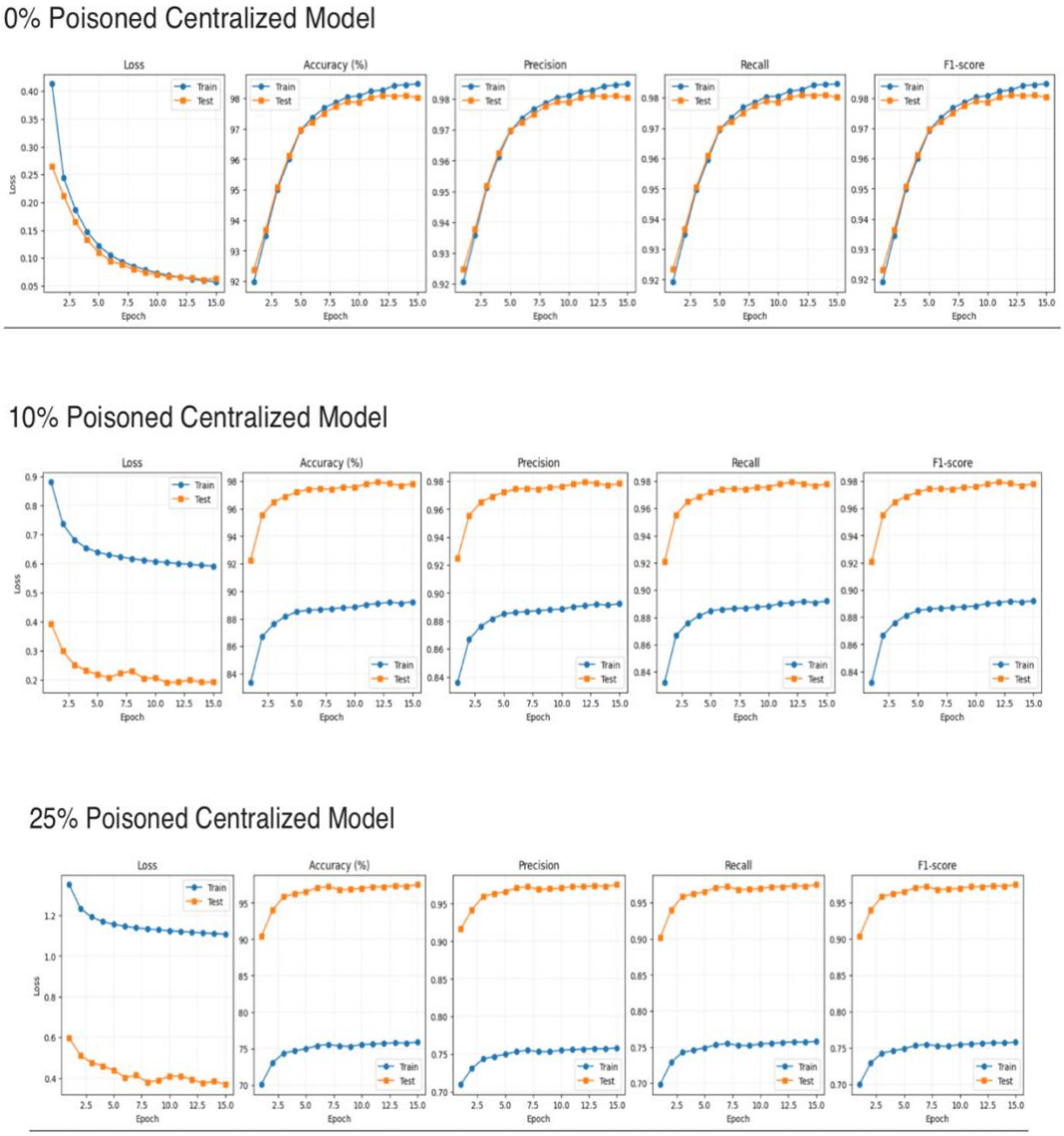
In reviewing the ImageWoof dataset results as illustrated above in Figure 4-2, we can see far greater divergence between the CL and FL trials, which is likely attributed to the MNIST dataset (more balanced) resulting in similar accuracy, precision, and recall versus the noisy ImageWoof dataset that is far more complex. A more detailed comparative analysis is explored in the next section; however it is worth identifying in these results that the tradeoff between CL and FL begins at the Base Trial with performance metrics lagging for FL compared to CL. This initial demonstration in the Base Trial where no data was poisoned showed that the data privacy offered by a distributed configuration (simulated using the Flower Library across five clients) came at a performance cost in each of the five performance metrics. That cost was approximately a 12% reduction in accuracy between the two learning techniques coupled with an approximate loss of 16%. While FL offers data privacy and segmented learning versus that CL model, it does come at a cost, which is an important consideration for organizations operating in mission/safety critical environments. It is also worth highlighting that this trend became more pronounced in the complex dataset versus the simpler one where there was not as much divergence, signaling that data complexity plays a role as well in making architectural decisions for ML operations and employment.

4.4 Comparing Results between Datasets and Learning Techniques

As the experimental trials were conducted, the MNIST dataset was tested first followed by the ImageWoof dataset across both CL and FL model training techniques. Each dataset / model was tested separately, but followed the same process to avoid unnecessary variables added to the experiments. In exploring the three trials for CL, we

can observe the following results comparing the performance of the model with training and testing data with varying degrees of data poisoning.

Figure 4-3: MNIST Centralized Learning Experimental Trials

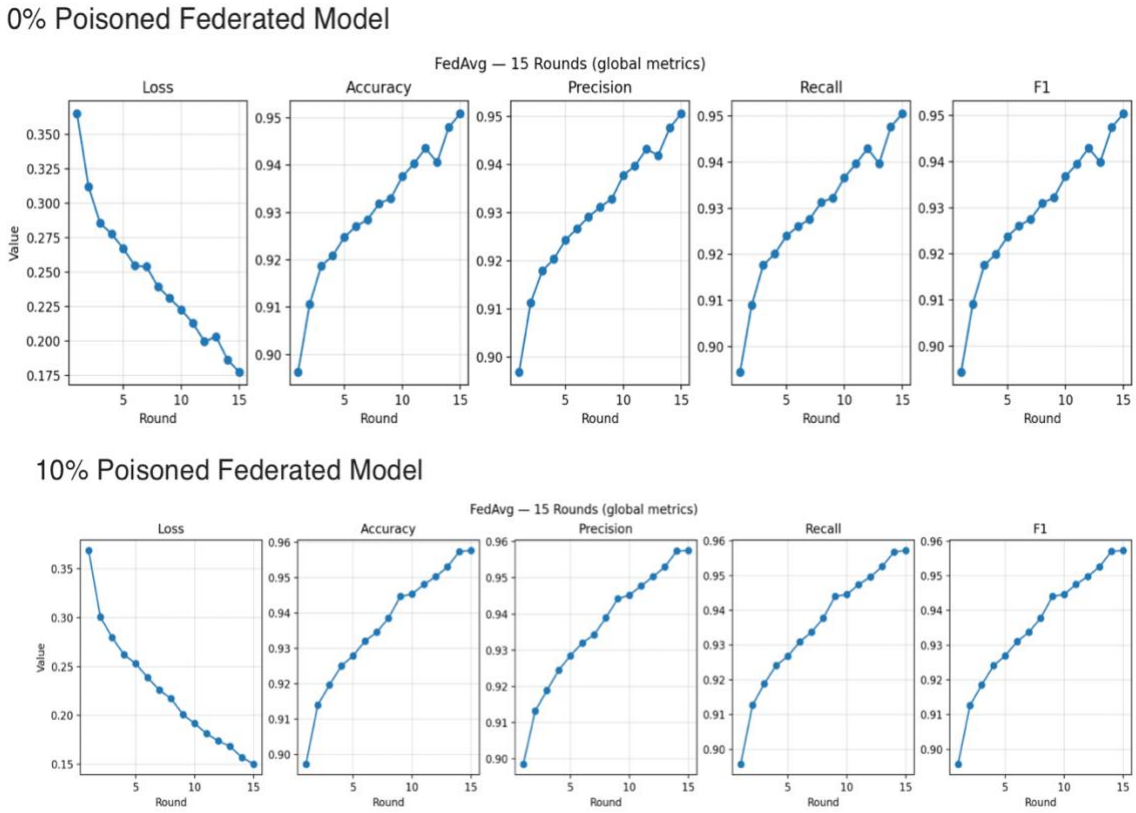


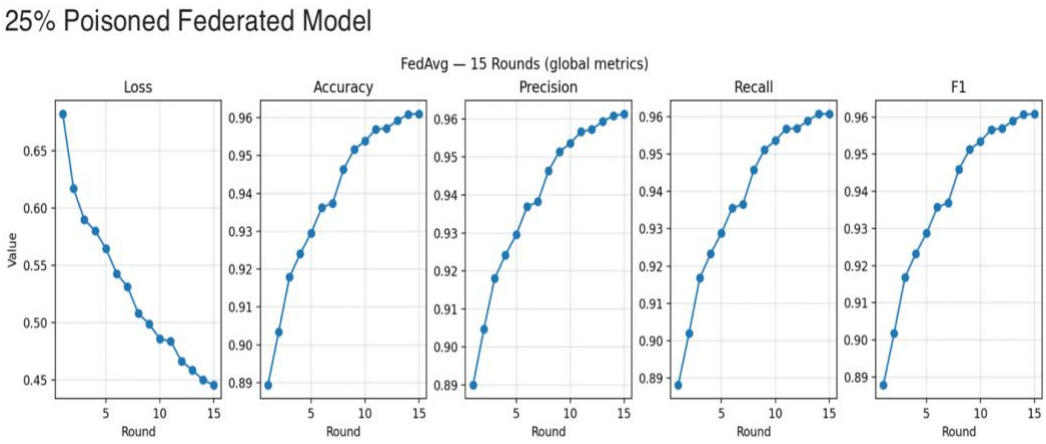
The series of graphs above offer an overview of how the multiple levels of data poisoning influenced the CL model performance for the MNIST dataset. As to be expected, model performance was degraded proportionally with each increasing level, though generally the models rebounded slightly after completion of 15 epochs. This

further demonstrates the testing environment was developed correctly with anticipated results produced for this series of trials.

In a similar way, the MNIST FL trials were conducted after the federated simulation environment was developed and implemented using the FLOWER library. These results can be observed below in Figure 4-4. Here we see an average line across the multiple poisoning levels that is produced using the FedAVG algorithm to accommodate reporting for the five unique clients simulated in the FL trials for each of the five performance metrics.

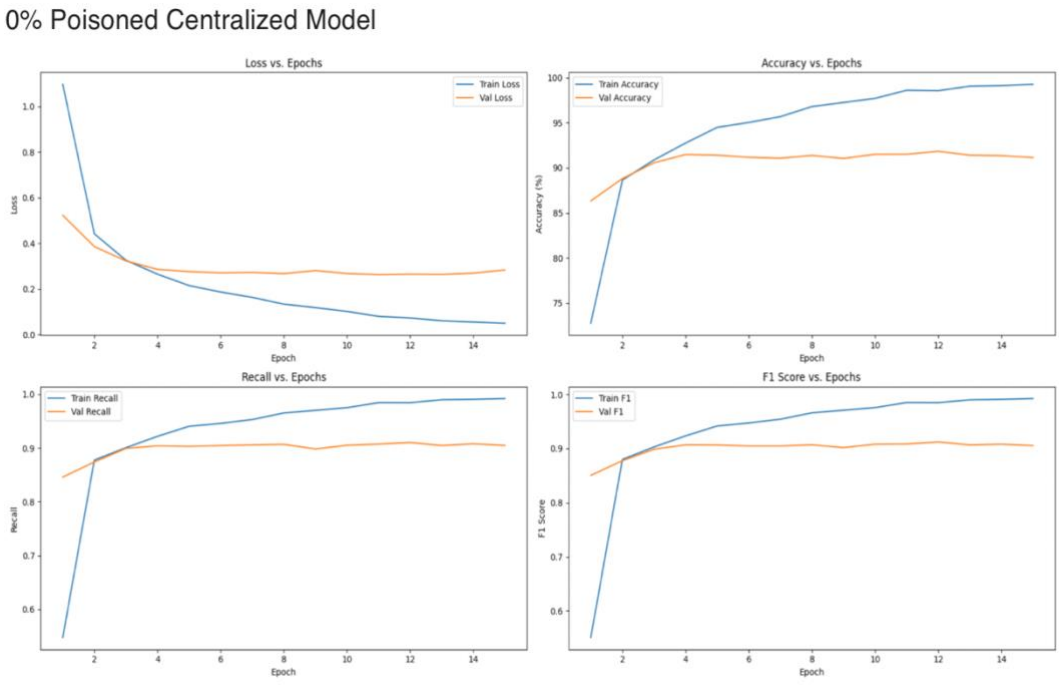
Figure 4-4 MNIST Federated Learning Experimental Trials



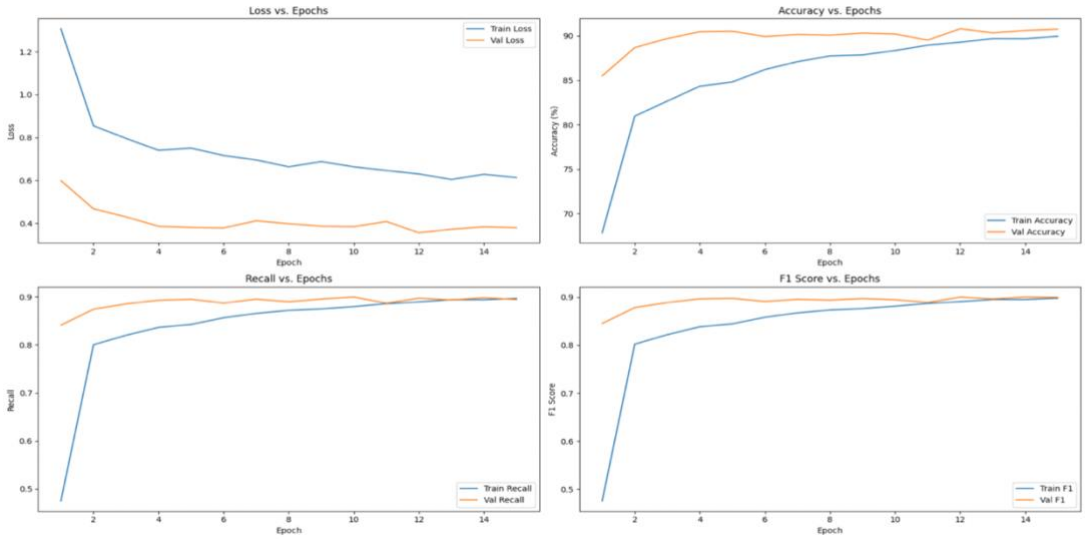


We can then explore the results of the more complex ImageWoof dataset and Resnet18 model in a similar fashion between the CL and FL trials.

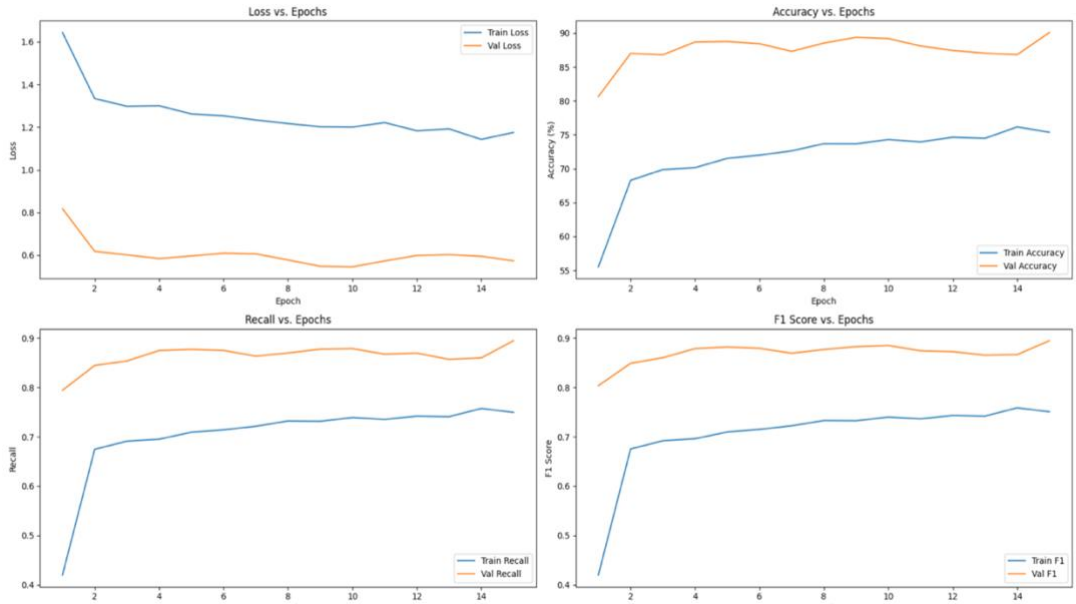
Figure 4-5: ImageWoof Centralized Learning Experimental Trials



10% Poisoned Centralized Model



25% Poisoned Centralized Model



Given the complexity of data and multiple clients for the FL trials for this dataset and model, the raw tables are being included as the FEDAVG Algorithm was employed to obtain these results as previously mentioned, which is why the graphics are not consistent with the previous trial; however the same results can be gleaned.

Figure 4-6 ImageWoof Federated Learning Experimental Trials

0% Poisoned Federated Model

```
INFO :      configure_evaluate: strategy sampled 5 clients (out of 5)
INFO :      aggregate_evaluate: received 5 results and 0 failures
INFO :
INFO :      [ROUND 15]
INFO :      configure_fit: strategy sampled 5 clients (out of 5)
E0000 00:00:1744570483.306790   44527 tcp_posix.cc:594] recvmsg encountered uncommon error: Message too
long
E0000 00:00:1744570544.174305   44527 tcp_posix.cc:594] recvmsg encountered uncommon error: Message too
long
E0000 00:00:1744570546.448986   44527 tcp_posix.cc:594] recvmsg encountered uncommon error: Message too
long
INFO :      aggregate_fit: received 5 results and 0 failures
INFO :      configure_evaluate: strategy sampled 5 clients (out of 5)
INFO :      aggregate_evaluate: received 5 results and 0 failures
INFO :
INFO :      [SUMMARY]
INFO :      Run finished 15 round(s) in 11336.20s
INFO :          History (loss, distributed):
INFO :                  round 1: 0.8371270298957825
INFO :                  round 2: 0.5119242072105408
INFO :                  round 3: 0.49602416157722473
INFO :                  round 4: 0.4857897460460663
INFO :                  round 5: 0.5334639549255371
INFO :                  round 6: 0.5199476480484009
INFO :                  round 7: 0.4741675555706024
INFO :                  round 8: 0.5330222845077515
INFO :                  round 9: 0.5269812941551208
INFO :                  round 10: 0.5695115923881531
INFO :                  round 11: 0.5538593530654907
INFO :                  round 12: 0.5506183505058289
INFO :                  round 13: 0.5640490651130676
INFO :                  round 14: 0.4774259626865387
INFO :                  round 15: 0.47310498356819153
INFO :
```

10% Poisoned Federated Model

```
INFO :      configure_fit: strategy sampled 5 clients (out of 5)
E0000 00:00:1744947678.194078  410796 tcp_posix.cc:594] recvmsg encount
long
E0000 00:00:1744947743.347740  410796 tcp_posix.cc:594] recvmsg encount
long
E0000 00:00:1744947745.559399  410796 tcp_posix.cc:594] recvmsg encount
long
E0000 00:00:1744947747.846443  410796 tcp_posix.cc:594] recvmsg encount
long
INFO :      aggregate_fit: received 5 results and 0 failures
INFO :      configure_evaluate: strategy sampled 5 clients (out of 5)
INFO :      aggregate_evaluate: received 5 results and 0 failures
INFO :
INFO :      [SUMMARY]
INFO :      Run finished 15 round(s) in 11287.98s
INFO :          History (loss, distributed):
INFO :                  round 1: 1.191460132598877
INFO :                  round 2: 0.6274880766868591
INFO :                  round 3: 0.536526083946228
INFO :                  round 4: 0.51394122838974
INFO :                  round 5: 0.46779200434684753
INFO :                  round 6: 0.546302318572998
INFO :                  round 7: 0.5099931955337524
INFO :                  round 8: 0.48790836334228516
INFO :                  round 9: 0.4844713509082794
INFO :                  round 10: 0.47486433386802673
INFO :                  round 11: 0.43264731764793396
INFO :                  round 12: 0.48855140805244446
INFO :                  round 13: 0.5087966322898865
INFO :                  round 14: 0.5293052792549133
INFO :                  round 15: 0.5078365206718445
INFO :
```

25% Poisoned Federated Model

```
(myenv) → Federated2 git:(main) ✗ python server.py
INFO :      configure_evaluate: strategy sampled 5 clients (out of 5)
INFO :      aggregate_evaluate: received 5 results and 0 failures
INFO :
INFO :      [SUMMARY]
INFO :      Run finished 15 round(s) in 11007.13s
INFO :          History (loss, distributed):
INFO :                  round 1: 1.6152173280715942
INFO :                  round 2: 0.8553057909011841
INFO :                  round 3: 0.7510891556739807
INFO :                  round 4: 0.6720835566520691
INFO :                  round 5: 0.7139086723327637
INFO :                  round 6: 0.748882532119751
INFO :                  round 7: 0.5510750412940979
INFO :                  round 8: 0.6000357866287231
INFO :                  round 9: 0.6217947602272034
INFO :                  round 10: 0.6401439905166626
INFO :                  round 11: 0.6153762340545654
INFO :                  round 12: 0.674720287322998
INFO :                  round 13: 0.6846103668212891
INFO :                  round 14: 0.6660716533660889
INFO :                  round 15: 0.5980852246284485
INFO :
```

4.5 Evaluating Overall Data Poisoning

This research offered some new insights into how varying levels of data poisoning affect ML systems differently depending on their underlying training architecture along with the level at which they are attacked. FL models were found to be especially vulnerable to even small-scale poisoning, while CL models demonstrated a higher threshold of tolerance, up to a point. This is not to say that the CL models did not suffer degradation as they clearly did across the epochs; however they fared better in the initial poisoning trials (10%) compared to FL and then generally responded more quickly at the higher poisoning level (25%). However, these results cannot be accepted at face value given the multi-dimensional nature of FL; these overarching results are true of the FEDAVG results (global aggregator). If we examine each individual node more closely, it becomes apparent that non-poisoned clients (three of the five) were nearly immune from the data poisoning of the other two clients. This difference can be traced to the

manner in which each architecture processes, aggregates, and learns from incoming data during model training and the architectural data structures. Overall, both the CL and FL trials converged by the 15th epoch across all data poisoning levels and were typically around 12% apart on average across most of the performance metrics.

Given the training process for CL systems, any poisoned data in the training pool is instantly and globally impactful, as the model parameters are updated with the entire dataset signaling the importance to build model resiliency to identify such corruption. The trials revealed that even at 10% poisoning, CL models experienced some drop in performance, especially in precision and recall. This is a non-linear degradation, where a small amount of adversarial contamination has a bigger impact. This non-linearity is due to centralized parameter convergence: because the model is updated with all the data, the malicious patterns introduced by the poisoned samples are quickly internalized and generalized across the decision boundaries. At 25% poisoning and above, the model struggled to tell the difference between real and adversarial data, and the misclassifications started to amplify with systemic risk materializing.

On the other hand, FL systems showed worse resistance to the initial poisoning compared to CL based on the average or aggregate performance; yet because of their decentralized nature those nodes / clients not suffering direct data poisoning fared with insignificant impact. In Chapter 3, it was reviewed how and why this learning techniques operates this way and the inherent data segregation that exists in federated systems. An adversarial contamination introduced at a single client has limited scope since its impact is diluted during the aggregation of updates from multiple, independent sources, which are not composed of raw data. The study found that FL

models poisoned at the same level performed worse across all metrics at 10% poisoning, while the structure protected non-poisoned clients. The decentralization created a natural isolation mechanism, effectively bundling the poisoned inputs to the couple clients though the attacks were still successful in causing global model degradation. This isolation mechanism was more effective when most clients were not compromised, thus preserving the statistical integrity of the aggregated model updates and hence the global model in the FL setup.

These results complicate a clearcut answer to which learning technique performed better in the scenarios executed as averaged results are not absolute when accounting for local versus global performance. It was observed that in scenarios where multiple clients are simultaneously compromised the threat to the FL system grows more severe, which is to be expected and is an element that cannot be as easily tested in CL systems. The experimental trials revealed that at 25% data poisoning across clients, FL performance began to erode measurably and significantly more than in the CL trials (at the same level). These findings suggest that FL defenses can be overcome not only by the volume of poisoning but also by the sophistication and alignment of attacks to specific local nodes simultaneously to further corrupt the global model.

Trust across clients and the aggregation method was obviously more important in federated systems as the Flower library simulation showed different results to some extent with individual clients performing differently. This begs the question of how to better defend these clients from themselves, while also hardening the wider system including the global model. Byzantine-resilient aggregation methods (Krum, Trimmed Mean, Bulyan), client behavior validation and coordinated anomaly detection were

considered as statistical analysis was done across these experiments to ensure FL model resilience. Moreover, FL systems can benefit from adaptive weighting of client updates based on past trust scores or behavioral baselines as poisoning attacks become more sophisticated and datasets are imbalanced (as in the Xray14 dataset in the Appendix).

This research made clear that data poisoning was a context-sensitive threat, whose effectiveness and risk profile were shaped directly by the learning technique in use as well as the distribution of the datasets comprising the models. FL systems exhibited high sensitivity to even low-level poisoning due to their concentrate client level dependence, while CL systems offered partial insulation through wider processing across the same dataset sizes. Neither of the two learning techniques was fully secure with defending against poisoning attacks requiring a combination of architectural foresight, algorithmic safeguards, and continuous operational monitoring, especially of data. The research supported claims in the literature review of multi-faceted safeguards in AI systems that must be engineered during design with architectural choices extending through training practices and deployment contexts among other considerations unique to the business / mission.

4.6 Resulting Architectural Considerations

In evaluating the two different learning techniques and their responses to data poisoning, another perspective was observed, that of how architectural setup and to some extent configuration influenced susceptibility of the attacks. In the CL trials across both models and datasets, the findings revealed a definite need for more stringent data pipeline defenses as all training data and model updates converged at a central location (true of universal model in FL), where these systems were found to be particularly vulnerable to

widespread corruption stemming from even small-scale data poisoning (this was true even in minor data poisoning where 3% and 5% levels were tested early on in the experiment). To mitigate this risk, a takeaway is that FL systems should incorporate robust pre-processing filters capable of detecting anomalous or adversarial inputs before they enter the local models during the training phase, especially for data sources that may not have the most effective curation. Such filtering should continue given the dynamic nature of the models throughout training periods to help mitigate vulnerability (as no defense mechanism is certain to eliminate risk). Other real-time anomaly detection mechanisms at the data ingestion point could be beneficial for identifying and isolating suspicious data flows also given how the trials performed across the 15 epochs. Frequent model validation using trusted datasets also surfaced in the literature review, which can ensure early detection of performance degradation. Future implementations in CL configurations could explore advanced defense strategies such as differential privacy, which could obscure sensitive patterns in training data and is a built-in benefit of FL systems given the confinement of raw data locally. The role of adversarial training is still undecided given its infancy, but could also serve as another defense strategy for these same reasons of building robustness.

In continuing to explore some of the architectural lessons learned, the FL trial results offer compelling reason to better understand operational environments, the influence of clients, and distribution of data across these clients. It is clear that FL environments offer better protection given their inherent compartmentalizing of clients, yet the overall performance (averaged) did not perform as well as CL. While individual clients in FL systems could be compromised with minimal system-wide effect, the risk of

overall model poisoning via collusion (multiple malicious clients conspire to skew the global model) was recognized as a significant concern. To address this, the implementation of Byzantine robust aggregation techniques (previously described) was recommended during the literature review and adhered to during the experimental trials. These techniques helped mitigate the influence of outlier or malicious updates in order to better assess. In addition, periodic client attestation protocols and reputation-based trust scoring were identified as mechanisms for dynamically assessing the reliability of participant nodes in federated systems as mentioned in Chapter 2; however these areas were not further explored given the already complexity of the experimental trials that were conducted for the praxis.

From a broader architectural standpoint, general-purpose cybersecurity principles remained true across both CL and FL environments with these considerations offered in addition to such best practices. These included tamper evident logging systems to maintain immutable records of data provenance and other model training events with adaptive weighting schemes that dynamically adjusted the influence of data sources or clients based on observed behavior or historical trustworthiness to better protect systems. When paired comprehensively these controls work together to deliver layered defense mechanisms, improving robustness while supporting forensic analysis in the event of compromise.

4.7 Assessing Wider Cyber Vulnerabilities

This praxis focused on data poisoning alone as the source of attacks, but it is important to consider other attack types that can equally compromise ML systems. Other threats can reveal fundamental cybersecurity weaknesses in the architecture of AI

systems regardless of learning techniques. CL architectures by design consolidate training data and model updates at a single point, making them a target rich environment for attackers, especially given their popularity (and familiarity). Centralization is a single point of failure: if compromised the whole system can be taken down at scale. These architectures are vulnerable not only to data poisoning attacks but also to direct model manipulation, exfiltration and cascading performance degradation. In operational environments, especially in defense and critical infrastructure, the lack of distributed fault tolerance means you need to deploy robust cyber defenses – real time monitoring, anomaly detection across the data pipeline and zero trust at every stage of the data lifecycle.

On the other hand FL mitigates many of these risks through its own decentralization, spreading the training responsibility across multiple clients and thus limiting the global impact of a local compromise. This architecture provides fault tolerance and system redundancy so FL systems can continue to operate partially even under sustained attack. But this resilience comes with its own set of complexity. FL introduces new threat surfaces like rogue client injection, poisoning through local updates and model inversion attacks where adversaries can infer private training data from shared model gradients. These require equally advanced mitigations like secure aggregation protocols, federated anomaly detection, client attestation mechanisms and privacy preserving techniques like differential privacy and homomorphic encryption (fully and partially).

As development efforts begin, thought must be made to the operational or mission use of the CL/FL technique under consideration. The corresponding goals/requirements of the operation should play a significant role in determining which approach is appropriate. CL systems if compromised will collapse rapidly and systemically while FL architectures will degrade gracefully and preserve core functionality and support failover scenarios which is critical in mission critical or contested environments. So this research contributes not only to the understanding of ML robustness but to the broader field of cyber-resilient system engineering, especially as it relates to learning technique and the influence of data poisoning. The trials highlighted the strategic trade-offs in AI architectural decisions and the need for security aware design principles to be embedded throughout the AI development lifecycle, especially early on and how even accidental poisoning (mentioned in Chapter 2) can cause severe disruption to model performance.

4.8 Contributions to the AI Engineering Body of Knowledge

This research contributed meaningfully to the AI Engineering Body of Knowledge (AI EBoK) developed by Carnegie Mellon University's Software Engineering Institute to advance the state of practice in secure and resilient AI system design (forthcoming publication). One of the study's core contributions is aiding in the development of a comparative adversarial framework that systematically analyzes the susceptibility of CL and FL systems to data poisoning attacks at varying levels. Unlike much of the existing literature that examines these paradigms in isolation, this work offers a side-by-side empirical comparison under identical experimental conditions, thus filling a notable gap in adversarial machine learning research with datasets of varying complexity.

A significant insight emerged from this praxis, limited transferability of adversarial data poisoning transferring between data sources (clients in the FL trials) and in datasets running parallel. This finding has important implications for the broader discipline of AI engineering and cybersecurity, suggesting that architectural divergence itself can serve as a defensive mechanism by disrupting the generalizability of adversarial techniques. This area specifically should undergo additional testing to better understand the specificity and its limitations, but presents an intriguing research question for further study. This insight also supports the AIE BoK's goal of engineering for adversarial robustness and underscores the importance of design decisions in threat modeling. This praxis also introduced a structured, repeatable way to test system performance under stress. By using standard but multi-dimensional metrics (accuracy, precision, recall, F1 score, cross-entropy loss) across controlled poisoning conditions (0%, 10%, 25%) this allows for consistent benchmarking of resilience. It's a template for engineers to test their AI models in production and contributes to the AIE BoK's emphasis on evaluation protocols based on real world use cases.

In the end, the praxis showed that CL and FL both offer different benefits for a more secure and fault-tolerant base for ML operations when there is wider spread of data and varying user populations / clients connected to a network. It is worth pointing out that FL is newer than CL and therefore does not have the longevity of implementations (and resulting best practices) at this point in time. Given current trends it is likely greater study will reveal other ways to securely shape FL systems and leverage their many benefits. By bridging theory and practice this research helped to mature AI engineering as a discipline

and provides actionable insights that contributed to the AI Engineering Body of Knowledge.

4.9 Concluding Considerations

These experiments showed that overall FL was more robust to data poisoning attacks because of its decentralized and distributed architecture as across multiple controlled experiments, FL systems performed better on key metrics like accuracy, recall, precision and F1 score even under high levels of adversarial contamination. Though it is worth pointing out that its resiliency weakened as the data poisoning level increased, especially into the double digits. CL systems on the other hand showed steeper and more systemic performance decline as the level of contamination increased. These results confirmed that architectural design plays a big role in making machine learning systems resilient to adversarial stress.

One of the most interesting findings was the lack of attack transferability between the two learning paradigms. Poisoning strategies designed for CL models did not generalize well to FL systems and vice versa. This showed architectural divergence as a practical and effective way to limit the reach and impact of adversarial attacks. It means that designing systems with different structures and update mechanisms can be a form of adversarial containment, preventing attacks from spreading uniformly across different operational environments. This is very useful in scenarios where AI capabilities are deployed across heterogeneous platforms or joint networks, which is often the case in defense, intelligence and multinational infrastructure systems.

From a cybersecurity and AI engineering perspective this work has several practical implications. It shows that resilience to adversarial input must be engineered in

at the architecture not just retrofitted through reactive defenses. For CL systems this may mean more rigorous data validation pipelines, anomaly detection and adversarially hardened training procedures. For FL systems this means securing client-server interactions, improving aggregation robustness and monitoring for collusive or anomalous model updates. More generally both paradigms would benefit from layered defenses that combine statistical anomaly detection, cryptographic integrity checks and model performance monitoring to detect and isolate poisoned influences early in the training lifecycle.

This research also pointed toward the growing importance of operational context in AI system design. In mission critical domains such as military targeting, autonomous navigation, or real-time surveillance the tolerance for failure is exceptionally low. As such, the ability of FL systems to degrade gracefully under attack, rather than collapsing entirely as CL systems often did in the experiments, was a crucial advantage. Graceful degradation, in which partial system performance is preserved under stress, offers time for mitigation and recovery without total mission disruption when and where such identification can be made. Here FL is particularly more attractive for deployment in edge computing environments, disconnected operations, and contested communications scenarios as data siloing is an inherent attribute.

Chapter 5 – Research Discussion

5.1 Summary of Research and Core Findings

This chapter brings together all of the components previously described and discusses their contextual relevance and significance across wider applications. Based on the experimental trial results from Chapter 4 and observations derived throughout this praxis, two of the three hypotheses posed were supported as true and validated through empirical evidence with a partial offer of support for the third as outlined in the below table (Table 5 – 1):

Table 5 – 1: Research Hypotheses Overview

(A) Deliverable	(B) Format
Hypothesis 1	There is significant difference in the performance of models across CL and FL techniques at a foundational level.
Independent Variables	CL object detection models and FL object detection models (not poisoned).
Dependent Variable	Model performance (accuracy, cross entropy loss, precision, F1 Score, recall)
Testable	Conduct experiments deployed across CL and FL to determine which offers better performance in predictive analytics.
Hypothesis 2	Federated Learning reduces the spread and impact of data poisoning attacks given its decentralized nature.
Independent Variables	Poisoned CL object detection models and poisoned FL object detection models.
Dependent Variable	Effectiveness of FL architecture against adversarial attack, measured by the reduction in success rate of compromising models.
Testable	Experiments to evaluate the effectiveness of FL in containing an attack(s).
Hypothesis 3	When model training is completed upfront it is more effective in combating the influence of data poisoning attacks, which allow recognition of outliers, anomalies, and possible attack vectors.
Independent Variables	Model architecture and learning technique.
Dependent Variable	Use of poisoned datasets at 10% levels and 25% levels
Testable	Statistical analysis performed to identify the effectiveness of adversarial training across CL and FL.

In the figure above we can see Hypothesis 1, which claimed there would be a significant difference in model performance between CL and FL under normal (non-poisoned / 0%) and poisoned conditions which proved to be true. The experimental trials showed consistent differences across all metrics (accuracy, precision, recall, F1 score,

cross-entropy loss) for the poisoned trials with well over 15% greater performance in CL coupled with less difference between the CL and FL base trials (or when data poisoning occurred), which were closer to 5% worse for FL. The CL models did perform as well as or better than FL models in terms of generalization and stability across epochs overall even during the base trials. The takeaway here is that learning technique has an impact on model performance including when data poisoning occurs, but not as significantly without it (at least in the models and datasets tested in this praxis).

Hypothesis 2 claimed FL would reduce the spread and impact of data poisoning attacks and the data sort of showed that to a degree. When we introduced poisoned inputs at 10% and 25% levels, FL systems performed worse than CL, however as previously detailed this was at the aggregate level (with non-poisoned local clients nearly unaffected). The decentralized nature of FL seemed to compartmentalize the poisoned data and limit its effect on connected clients yet did negatively impact the global model parameters. CL models absorbed the poisoned data into one point and amplified the attack and the performance degradation, but given the concentrated data spread seemed to limit the effect of the data poisoning especially at the 10% level (this wasn't as significant in the 25% trials). Recall and F1 score dropped much faster in FL than in CL which proves that CL architecture provides a defensive advantage through structural fault isolation, yet the compartmentalization of the local nodes in the FL were better protected experiencing nearly no impact.

Hypothesis 3 was supported though it could be argued with some caveats. Neither architecture was completely immune to poisoning but models that had redundant initial training and robust aggregation did perform better. In FL the distributed and iterative

nature of training helped to mitigate poisoned gradients especially when only a subset of clients were poisoned (two clients per trial). In CL the adversarial influence propagated faster, so additional layers of training or anomaly detection helped to promote resilience. It would have been interesting to include adversarial training across these models as well and is an opportunity for further investigation (it was not here due to compute constraints as well as scope refinement). Across both learning techniques the results show that the effectiveness of model training in helping to resist data poisoning depends on both the training environment and coordination of poisoned inputs. A recap of the results of each of these hypotheses can be seen in Table 5 – 2 below:

Table 5 – 2: Results of Research Hypotheses

Hypotheses	Description	Conclusion
Hypothesis 1	There is significant difference in the performance of models across CL and FL techniques at a foundational level.	Proven Correct
Hypothesis 2	Federated Learning reduces the spread and impact of data poisoning attacks given its decentralized nature.	Partially Proven Correct And Incorrect
Hypothesis 3	When model training is completed upfront it is more effective in combating the influence of data poisoning attacks, which allow recognition of outliers, anomalies, and possible attack vectors.	Proven Correct

These hypotheses are supported through statistical analysis as demonstrated in the below paired t-tests across the experimental trials that were conducted in Minitab. The first hypothesis is validated through the paired t-test where H_0 represents that CL and FL techniques performed the same and H_1 represents that the techniques performed differently. Using an alpha of 0.05, it can be observed that the p-value is 0.04 which is less than the alpha of 0.05, therefore we reject the null hypothesis (H_0). This supports the hypothesis that CL and FL performed differently. The average of trial values was used to complete this paired t-test across all five performance metrics (as previously identified

throughout the praxis). This can be observed in Table 5 – 3 below, which is the report output from Minitab for the first hypothesis.

Table 5 – 3: Hypothesis 1 Paired t-test

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
CL Base	15	81.38	26.68	6.89
FL Base	15	73.12	14.09	3.64

Estimation for Paired Difference

Mean	StDev	SE Mean	95% CI for μ_difference
8.26	14.09	3.64	(0.46, 16.06)

μ_difference: population mean of (CL Base - FL Base)

Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
2.27	0.040

In a similar way, a paired t-test was completed for Hypothesis 2, which stated that FL would do a better job than CL in limiting the influence and spread of the data poisoning attacks. Using the performance metric averages across all five metrics, the following test was conducted in Minitab, where the H_0 was that FL would do a better job than CL and the H_1 was that FL would not do a better job than CL. Here the p-value is just above the alpha at 0.063, which signifies that we fail to reject the null hypothesis. Furthermore, we observed in the experimental trials that the individual clients in the FL trials that were not targeted did not experience degraded performance, further complicated this hypothesis. For this reason, we cannot support the second hypothesis as shown in Figure 5 – 4 below.

Figure 5 – 4: Hypothesis 2 Paired t-test

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
CL Poison	10	82.59	23.96	7.58
FL Poison	10	73.14	12.08	3.82

Estimation for Paired Difference

Mean	StDev	SE Mean	95% CI for μ_difference
9.46	14.12	4.47	(-0.65, 19.56)

μ_difference: population mean of (CL Poison - FL Poison)

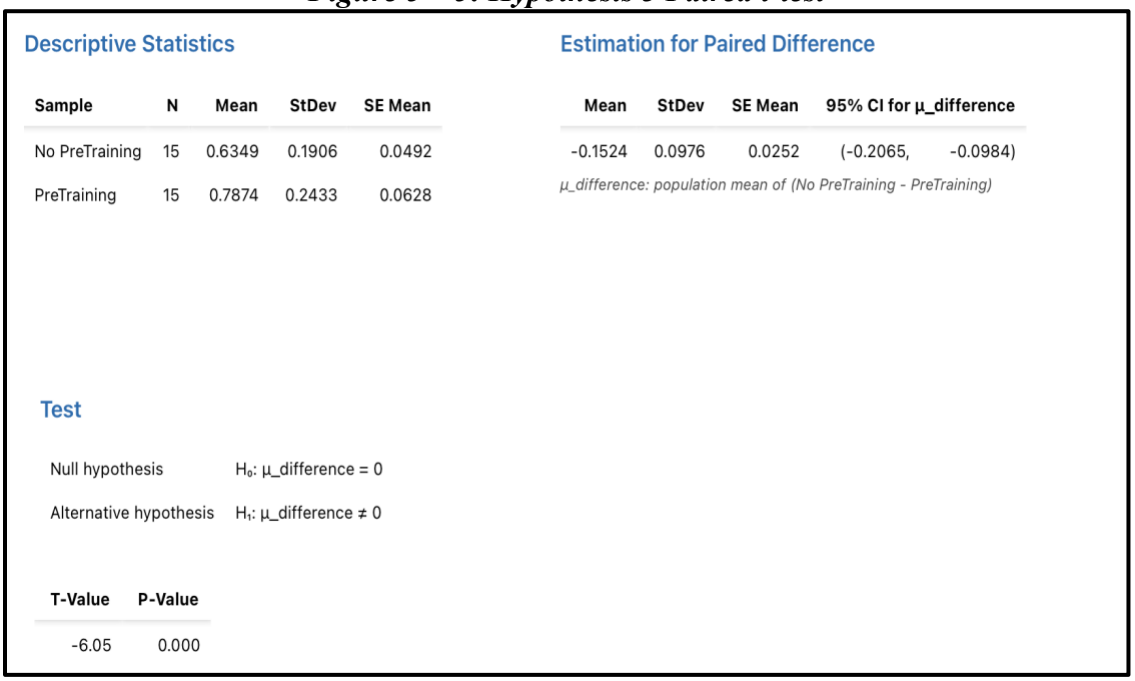
Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
2.12	0.063

Finally, the third hypothesis which stated that pretraining the models would aid in deterring the influence and spread of data poisoning attacks was tested using the same statistical analysis. Here though, the five performance metrics were examined across both learning techniques and models with pretraining taking place and without the pretraining taking place. The H_0 was there would be no change between pretraining in deterring data poisoning attacks and the H_1 was that there would be a difference in the use of pretraining the models. In reviewing Table 5 – 5, we can observe that the p-value is 0, meaning there is overwhelming evidence against the null hypothesis (we reject it). This supports that Hypothesis 3 is proven correct, there is in fact a difference in performance when pre-training of the models was conducted as can be seen in Figure 5 – 5 below:

Figure 5 – 5: Hypothesis 3 Paired t-test



Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
No PreTraining	15	0.6349	0.1906	0.0492
PreTraining	15	0.7874	0.2433	0.0628

Estimation for Paired Difference

Mean	StDev	SE Mean	95% CI for μ_difference
-0.1524	0.0976	0.0252	(-0.2065, -0.0984)

μ_difference: population mean of (No PreTraining - PreTraining)

Test

Null hypothesis	H_0: μ_difference = 0
Alternative hypothesis	H_1: μ_difference ≠ 0

T-Value	P-Value
-6.05	0.000

5.2 Research Implications

This praxis revealed implications for not just the field of adversarial machine learning, but also how to better build secure AI systems where model training is so

important. It advanced both theory and practice, especially for data poisoning attacks that are poised to significantly grow. This with setting boundaries for attack transferability, showed that ML is deeply contextual and tightly connected to architectural design decisions made upfront. This is not just important in the classroom but valuable in various use cases where AI must operate under partial trust or in contested networks while balancing varying tradeoffs. The primary outcomes from the praxis can be viewed through both theoretical and practical benefits with some overlap between the two.

From a theoretical perspective, this work contributed to the growing body of work on adversarial robustness in ML (not to mention the AIEBoK). It delivered a structured way to evaluate the effect and transferability of data poisoning attacks across different ML training techniques and filled a gap in the existing literature which often treats FL and CL in isolation and ignores the cross-architecture dynamics where cybersecurity is concerned. Going deeper, it challenged the conventional wisdom that poisoning attacks have uniform effect across different systems. Instead, it showed empirically that vulnerability was heavily dependent on the learning style and the inherent characteristics that come with each. Specifically, decentralization in FL acted as a distributed defense and prevented local propagation of poisoned data to other clients, though suffering performance by the global aggregator. Given the hands on nature of a DEng praxis, this research focused on pragmatic experimentation that directly translated current ML study as is detailed in Section 5.4 where we discuss in greater detail how the research novelty is already being applied. Similarly, the trials also highlighted that metric selection must be mission aligned based on the context of the operational goals and desired outcomes organizations set. Some systems that detect threats or anomalies may prioritize Recall so

true positives are not missed even if false positives increase a little bit, while Accuracy may be a higher priority in other settings. Regardless of industry or use case, this gives practical architectural guidance and evaluation strategies based on their operational risk tied to the models and datasets tested.

5.3 Opportunities for Future Research

The observations from this research were on identifying ways to facilitate architectural resilience against data poisoning with the ability to open additional ways for future work. The use of the MNIST dataset while good for control and replication was simple and more complex sets should be used to evaluate model robustness on higher dimensional real world datasets like CIFAR10, ImageNet or COCO. These datasets better reflect operational data and while ImageWoof was a subset of ImageNet, more diverse complex sets will strengthen the hypotheses tested. Additionally with the growing number of multimodal systems we need to explore poisoning vulnerabilities across combined data types like image-text pairs or sensor fusion environments given the proliferation of data poisoning attacks and expanded use of ML.

This research focused on proportion based (even distributions) and randomized poisoning attacks across multiple levels (0%, 10%, 25%) with some varied experimentation with ad-hoc levels in early experimentation to assess and fine tune the dev environment. Further research is needed to explore more sophisticated adversarial attacks like targeted attacks on specific output classes, backdoor attacks that embed hidden triggers and adaptive poisoning that evolves in response to defensive countermeasures. These attack vectors are more likely of what is to come as adversaries enhance their approaches and will give us a better understanding of the full threat

landscape, especially when these attacks are paired with others to further compromise or disrupt a system.

Overall broader system level and human centric considerations could also be included in follow on work like how alerting systems, audit trails and explainable AI tools can support human operators in detecting and responding to poisoning attacks (human-machine teaming). Similarly evaluating the trade-offs between communication overhead and robustness in FL under contested or bandwidth constrained conditions will help refine deployment strategies and balance tradeoffs for training / architectural selection.

5.4 Applying Research Novelty

There were multiple dimensions to the research explored in this praxis with support for some existing cybersecurity principles as they relate to ML and data poisoning, but more importantly giving new insight into the comparison of model training and how performance cannot be viewed singularly between CL and FL. As observed, there were two different dimensions to the results of the FL trials, the first as it related in aggregate state (FEDAVG) and the second at the client or local state. This offers a new view to assess and evaluate the different learning techniques, while foot stomping operational need to play a larger role in deciding how best to approach architectural considerations. FL, while not the overall top performer between the two in summary, did reveal benefits that did not exist in CL (and are missing from most systems today) in that data segregation, operational redundancy, and faster model output. These assets are of huge desire across many applications and industries when considering the increasing role ML is playing in daily life and should not be discounted.

Already the Auton Lab at Carnegie Mellon University is advancing this research by now evaluating the savings in network latency for model training in FL use cases and building upon the operational redundancy that was observed here to better provide consistent uptime in contested environments (Auton Lab, 2025). Both aspects are of critical importance when considering the dependency upon ML outputs in national security or related contexts like healthcare, in which model output latency and/or unavailability of results is unacceptable or even life threatening. The research of this praxis and this further investigation are changing the paradigm of AI for the future as AI is brought to the data through FL versus bringing the data to the AI (in CL and current applications). This DAI approach is poised to takeoff and significantly change not just the operational benefits of ML, but also require new thought in securing it.

5.5 Conclusion

AI is increasingly part of national security, public infrastructure and healthcare, resulting in greater need to secure and defend it, which too often is overlooked in the race to improve capabilities. This research showed how FL and CL stand up to data poisoning attacks in different ways with different costs (both operational and infrastructurally) and that consistent measurements alone are not enough when fully evaluating the dynamic nature of ML. This pushes another urgent need across technology now in adjusting the way we conduct testing and evaluation (T&E) of systems, especially in mission or safety critical applications. We still approach T&E from the antiquated hardware lens where it is merely a step on a checklist that is completed prior to deployment and done. This is no longer effective when ML models are dynamic with parameters and the very nature of models (and data) can significantly change days after such fielding, maybe even sooner.

No system or technology will be without vulnerability, especially with adversarial threat growing along with the enhancement of capabilities made for good. As adversarial techniques get more sophisticated and subtle, AI defenses must too with the topic of securing ML and adversarial ML being an area that seems often neglected today. Models must be more adaptive, more integrated into the training pipeline and more informed by the operating environment in which they serve. Future research must look at how models learn under attack and how the system as a whole (architecture, interface, user oversight) functions in adversarial conditions in order to promote greater resiliency and ensure proper T&E is conducted. This includes new defense strategies, addressing limits of current mitigation techniques and the complex relationship between model vulnerability and operational resilience.

Overall this praxis contributed to the field of secure AI engineering and delivered a scalable testing framework that has already garnered further research to better understand future directions for DAI and FL. The initial investigation sought specific objectives that were not only met, but also led into new dimensions that provided new insight into how to address the shift to FL and the change of attack surfaces. It supports the effort to build AI systems that are intelligent, trustworthy, resilient and designed with adversarial risk as a core engineering priority, an important reminder in the current rush to use AI.

Appendix

After the conclusion of the experimental trials and praxis writeup, a third dataset was identified focused on predicting disease based on x-ray images. Since there was time before final delivery of this research, the author and his advisor agreed a different dataset focused on a new use case would make a valuable addition. Given the differences in application and dataset attributes, the Chest X-ray14 dataset was selected to run the same series of trials as was conducted for the MNIST and ImageWoof ones during the earlier stages of the praxis. This dataset is a large-scale dataset provided by the NIH Clinical Center containing 112,120 frontal-view chest X-ray images from 30,805 unique patients. The dataset was originally created to facilitate research in the automated analysis of chest radiographs using deep learning and computer vision techniques with many citations across current efforts. It included 14 common thoracic pathologies annotated via automated text mining of radiology reports. This dataset was particularly well-suited for multi-label image classification and disease localization. Images were in grayscale (similar to MNIST) and the annotations were provided in a CSV file format, with each image labeled for the presence or absence of disease.

Given the size of this dataset, expenses were required to complete the work via Google Colab (not as significant as the earlier mentioned Netflix dataset, but approximately $100). Cross Entropy Loss and Accuracy were both used in these trials (as in the previous ones); however ROC-AUC was added given its better suitability for skew of this dataset (most patients were healthy and free of disease largely skewing this dataset). Since this is a new metric it is worth defining it. The area under the Receiver Operating Characteristic curve (AUC) represents the probability the model, if given a

randomly chosen positive and negative example, will rank the positive over the negative (Ozdayi, 2021). Most of the entities in this dataset represented healthy patients (not positive for disease), therefore it was important to properly assess. A higher AUC score means better model performance with a AUC of 1 signaling a perfect classifier. This trial also required so much cache that the MacBook Pro used had to be restarted at every epoch given how much data that had to load, requiring a significant amount of time to complete all trials.

A couple major takeaways from this complex dataset included when data is very skewed, data poisoning affects model performance significantly more. In the FL trials, it demonstrated how a complicated dataset results in significantly worse performance even without the influence of data poisoning. The same was true for the base trial of CL, which experienced tremendous loss, but rebounded in Accuracy. This can be observed in the following graphs from the Xray14 experimental trials.

Figure A – 1: Overall Cross-Entropy Loss of Xray14

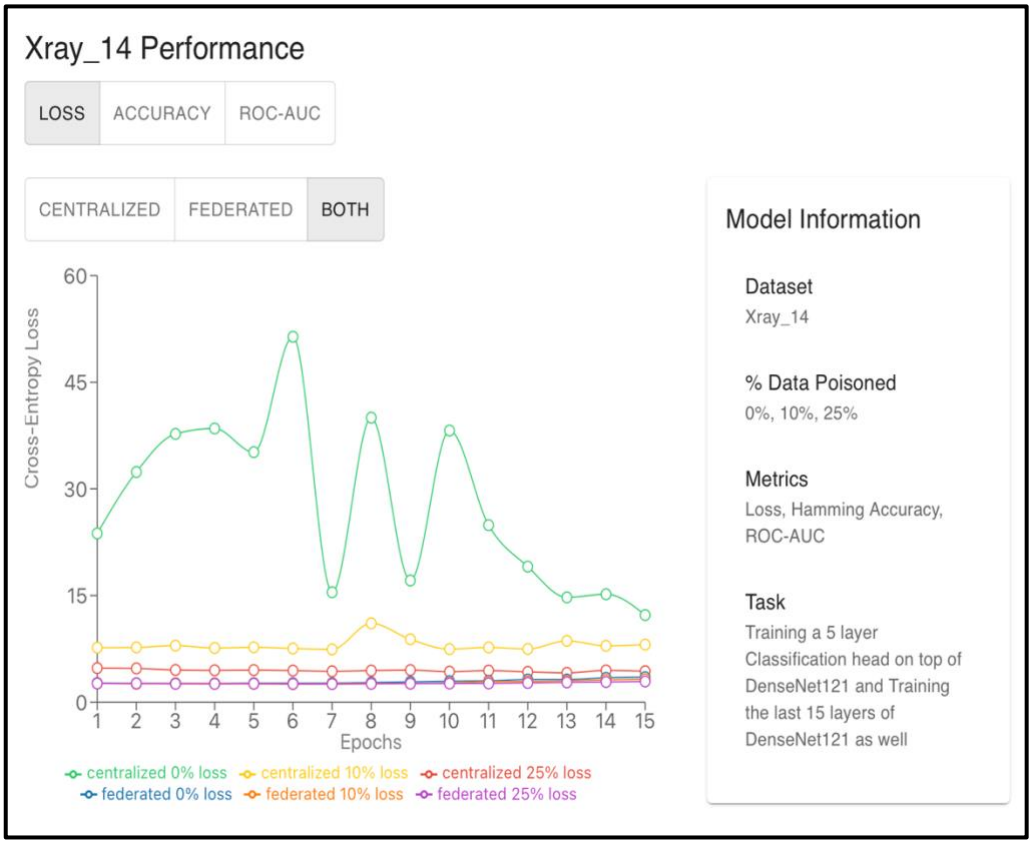


Figure A – 2: Overall Accuracy of Xray14

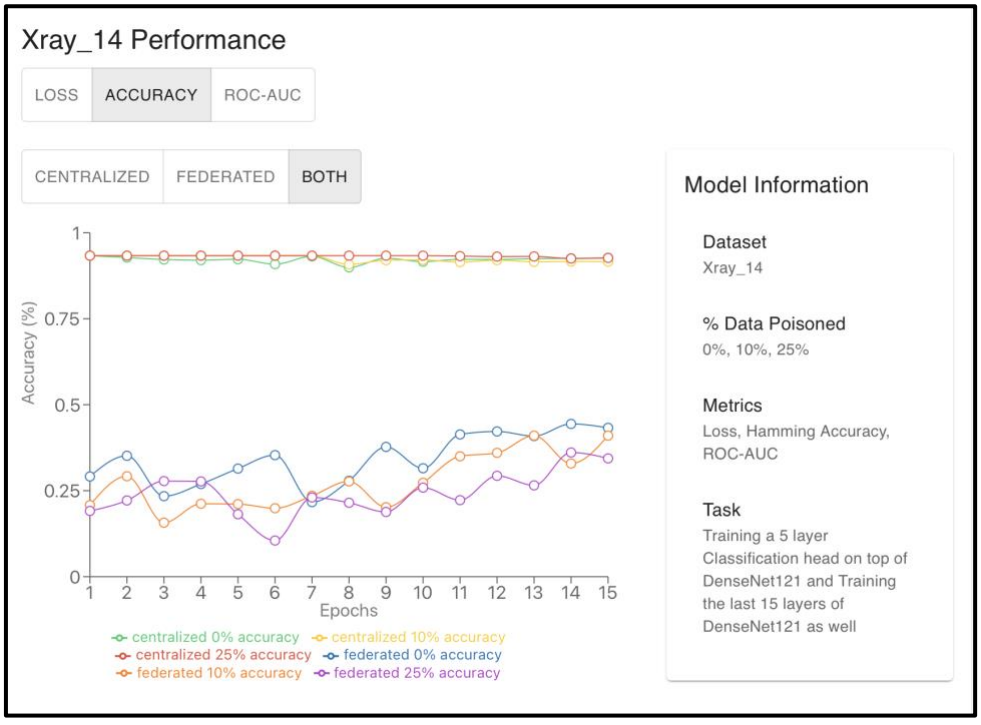


Figure A – 3: Overall ROC-AUC of Xray14

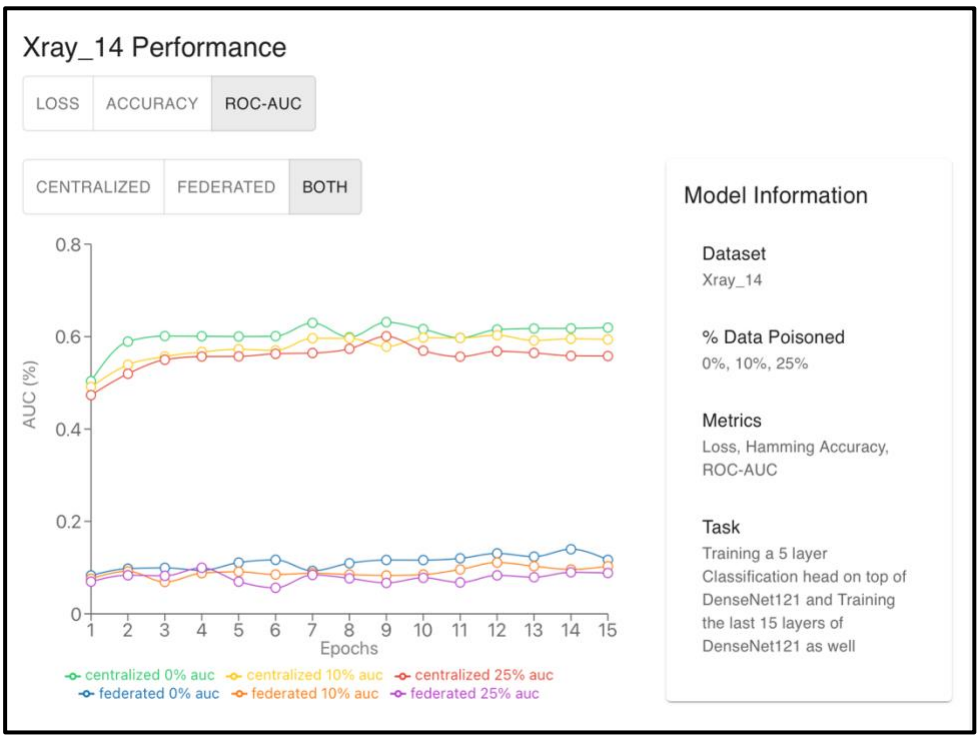


Figure A – 4: Xray14 Poisoned Levels for CL Trials

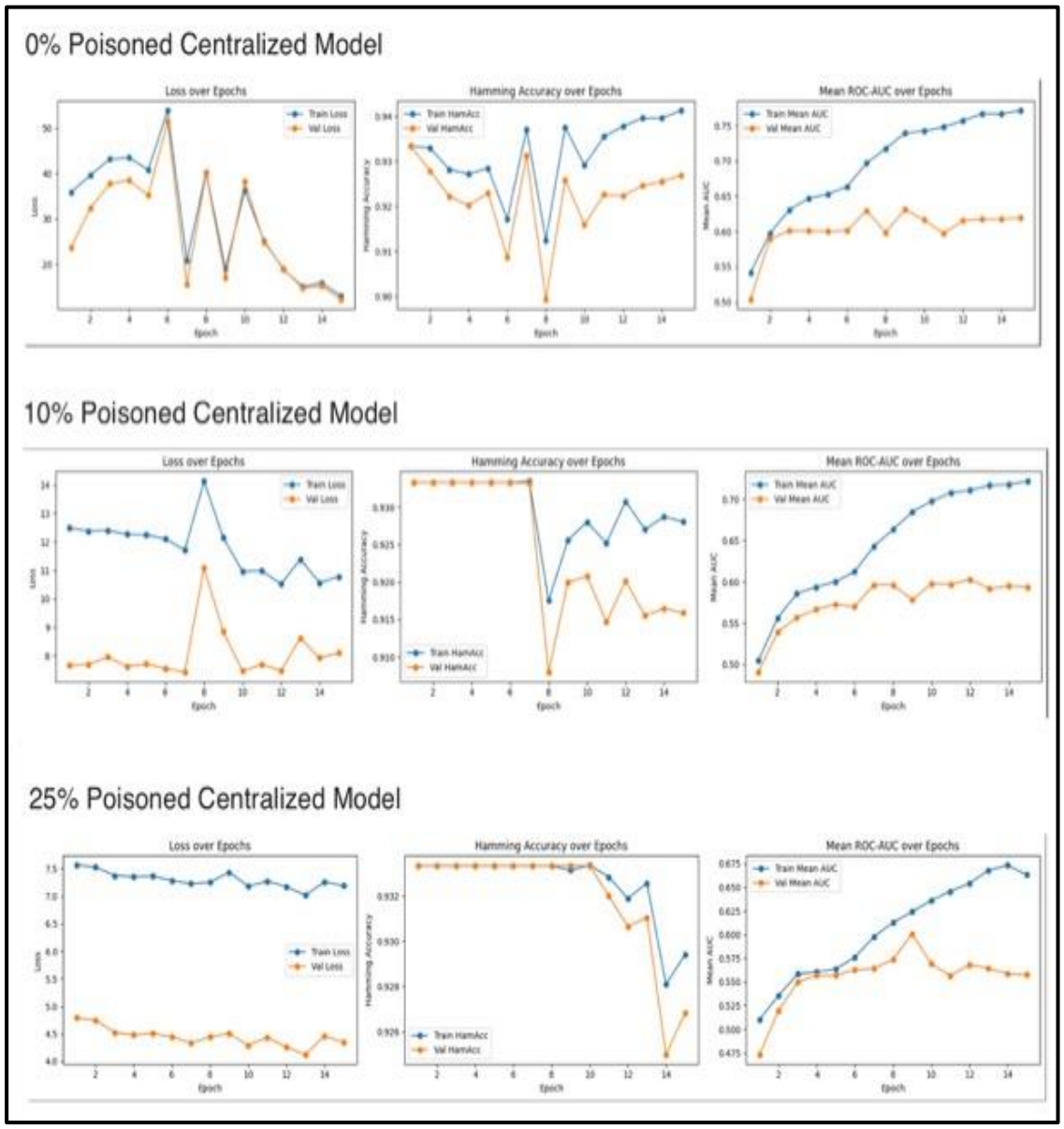
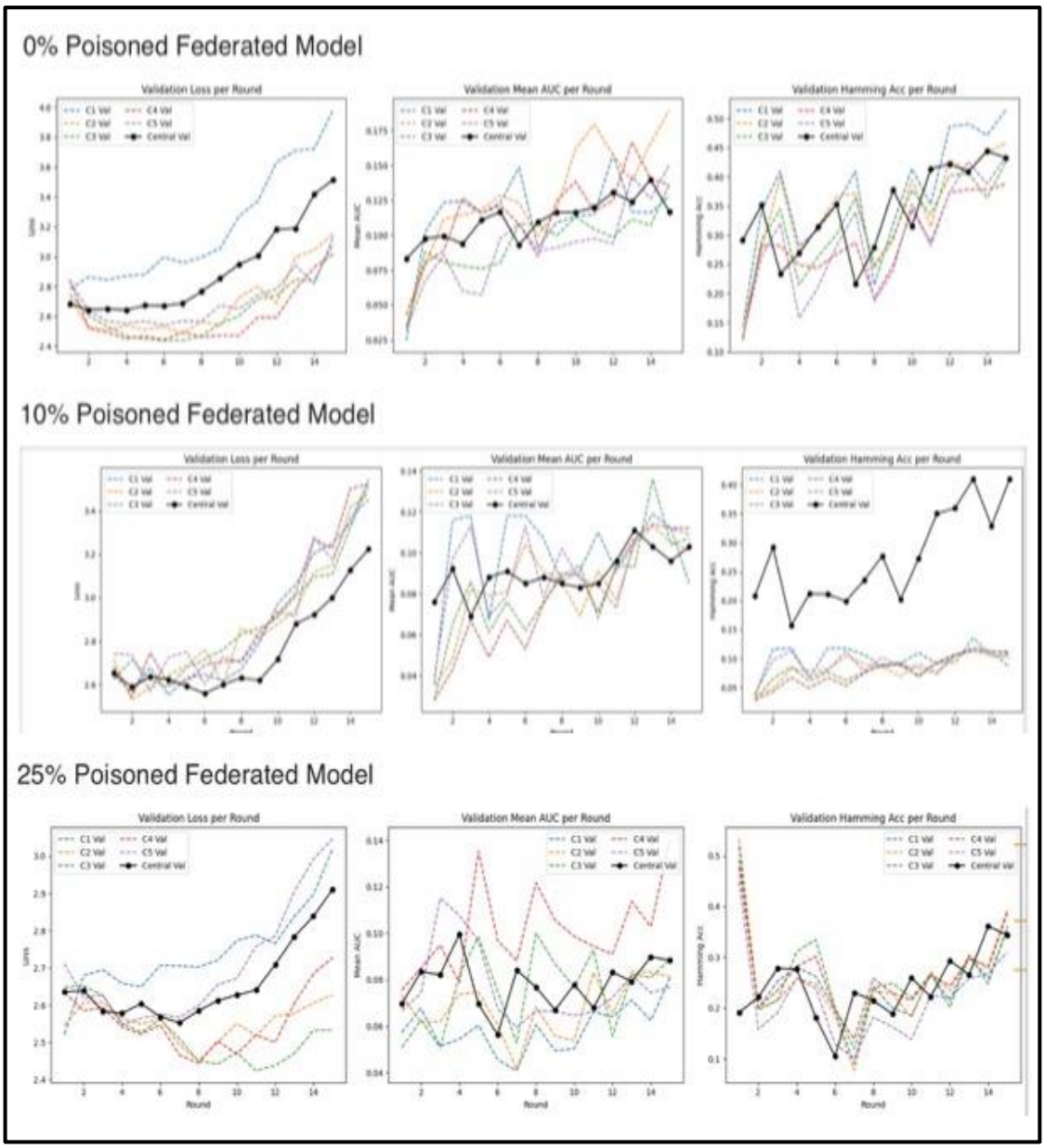


Figure A – 5: Xray14 Poisoned Levels for FL Trials



References

Aljanabi, M., Omran, A. H., Mijwil, M. M., Abotaleb, M., El-kenawy, E. M., Mohammed, S. Y., & Ibrahim, A. (2024). *Data poisoning: Issues, challenges, and needs*. In *7th IET Smart Cities Symposium (SCS 2023)* (pp. 359–363). IET. https://doi.org/10.1049/icp.2024.0951

Anwar, A., Baracaldo, N., Chen, P.-Y., Tian, Y., Yan, F., Zawad, S., & Zhou, Y. (2021, February 1). *Curse or redemption? How data heterogeneity affects the robustness of federated learning*. https://ojs.aaai.org/index.php/AAAI/article/view/17291

Auton Lab. (n.d.). *Research*. Auton Lab – Carnegie Mellon University. Retrieved May 24, 2025, from https://autonlab.org/research_areas.html

Caldas, S., Wu, P., Li, T., Konecny, J., McMahan, H. B., & Smith, V. (2020). *LEAF: A benchmark for federated settings*. *arXiv preprint arXiv:1812.01097*. https://arxiv.org/abs/1812.01097

Cao, X., Wu, Z., Chen, Y., B. (2023). *Provably Robust Learning Against Backdoor Attacks*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1243–1252.

Cao, Y., Li, Y., Zhang, H., & Li, B. (2023, May 3). *LearnDefend: Learning to defend against targeted model-poisoning attacks on federated learning*. https://prefer-nsf.org/pdf/DeFL_AAAI2023.pdf

Chen, X., Zhang, Y., Liu, X., & Ma, S. (2024, May 21). *A GAN-based data poisoning attack against federated learning systems and its countermeasure*. https://www.researchgate.net/publication/380730558_A_GAN-Based_Data_Poisoning_Attack_Against_Federated_Learning_Systems_and_Its_Countermeasure

Croce, F., & Hein, M. (2020). *Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks*. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 22063–22075). Curran Associates, https://proceedings.neurips.cc/paper/2020/file/8d30aa96e72440759f74bd2306a6b7f4-Paper.pdf

Fang, M., Cao, X., Yang, J., & Liu, N. (2024, July 9). *Tracing back the malicious clients in poisoning attacks to federated learning*. https://link.springer.com/article/10.1007/s44443-025-00115-1

Ghosh, A., Li, X., & Zhang, T. (2024, May 26). *Visualizing the shadows: Unveiling data poisoning behaviors in federated learning*. arXiv. https://arxiv.org/abs/2405.16707

Glickman, Y., Kfir, Y., & Avraham, T. (2021, December 6). *ML attack models: Adversarial attacks and data poisoning attacks*. arXiv. https://arxiv.org/abs/2112.02797

Goldblum, M., Fowl, L., & Goldstein, T. (2022). *Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses*. https://ieeexplore.ieee.org/ielaam/34/10008914/9743317-aam.pdf?tag=1

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Guo, Y., Chen, K., Zhang, L., & Wang, X. (2022, February 13). *Defense strategies toward model poisoning attacks in federated learning: A survey*. https://scholarworks.indianapolis.iu.edu/server/api/core/bitstreams/efc33e96-6e4e-4f8d-b34d-13827e82be0b/content

Guo, Y., Jiang, Y., Xu, H., & Li, J. (2021, September 21). *DeSMP: Differential privacy-exploited stealthy model poisoning attacks in federated learning*. https://gsconlinepress.com/journals/gscarr/sites/default/files/GSCARR-2024-0451.pdf

He, M., Dong, Y., Pang, T., & Zhu, J. (2022, February 17). *What doesn't kill you makes you robust(er): How to adversarially train against data poisoning*. https://openreview.net/pdf/8d8bca700348725f3f880cfc992979f63c4e629d.pdf

Huang, Q., Xie, C., Li, Y., & Zhao, X. (2023, April 30). *Assessing vulnerabilities of adversarial learning algorithm through poisoning attacks*. https://www.researchgate.net/publication/370443321_Assessing_Vulnerabilities_of_Adversarial_Learning_Algorithm_through_Poisoning_Attacks

Idika, T., & Akturk, I. (2021, December 3). *Attack-centric approach for evaluating transferability of adversarial samples in machine learning models*. arXiv. https://arxiv.org/abs/2112.01777

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). *Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning*. IEEE Symposium on Security and Privacy (SP), 19–35.

Jiang, Y., Liu, S., Li, M., & Wu, F. (2023, May 31). *How to sift out a clean data subset in the presence of data poisoning?* https://www.usenix.org/system/files/sec23fall-prepub-99-zeng-yi.pdf

Kim, J., Kim, J., Lee, A., Kim, J., & Diesner, J. (2024). LERCause: Deep learning approaches for causal sentence identification from nuclear safety reports. *PLOS ONE, 19*(8), e0308155. https://doi.org/10.1371/journal.pone.0308155

Kumar, A., Mehta, S., & Vijaykeerthy, D. (2017). An introduction to adversarial machine learning. In N. Chauhan, S. Agarwal, & D. Gupta (Eds.), *Big Data Analytics: 5th*

International Conference, BDA 2017, Hyderabad, India, December 12–15, 2017, Proceedings (pp. 293–299). Springer. https://doi.org/10.1007/978-3-319-72347-1_25

Li, X., Guo, Y., Jiang, Y., & Xu, H. (2023, November 30). *Defending against adversarial denial-of-service data poisoning attacks*. https://www.researchgate.net/publication/357300794_Defending_Against_Data_Poisoning_Attacks_From_Distributed_Learning_to_Federated_Learning

Liu, Y., Gao, S., Song, Y., & Chen, K. (2024, March 8). *Machine learning security against data poisoning: Are we there yet?* https://www.researchgate.net/publication/359917813_Machine_Learning_Security_against_Data_Poisoning_Are_We_There_Yet

Lyu, L., Yu, H., & Yang, Q. (2022, February 22). *Poisoning attacks and defenses on artificial intelligence: A survey*. https://www.academia.edu/83265573/Poisoning_Attacks_and_Defenses_on_Artificial_Intelligence_A_Survey

Ma, X., Cao, X., Li, B., Bailey, J., & Lu, F. (2022). *Backdoor Attacks and Countermeasures in Deep Learning: A Survey*. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2022.3152527

Ma, Y., Liu, S., & Gao, Y. (2023, December 19). *Mithridates: Auditing and boosting backdoor resistance of machine learning pipelines*. https://dl.acm.org/doi/10.1145/3658644.3690337

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (Vol. 54, pp. 1273–1282). PMLR.

Mohri, M., Sivek, G., & Suresh, A. T. (2020). *Agnostic federated learning*. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 24646–24657). Curran Associates https://proceedings.neurips.cc/paper/2020/file/f4ef20325c66d1d430b7b8ce2a2d8dca-Paper.pdf

Nguyen, H., Nguyen, M., & Nguyen, P. (2023, December 6). *Temporal robustness against data poisoning*. arXiv. https://arxiv.org/abs/2302.03684

Nowroozi, E., Haider, I., Taheri, R., & Conti, M. (2024, March 5). *Federated learning under attack: Exposing vulnerabilities through data poisoning attacks in computer networks*. https://dl.acm.org/doi/10.1109/TNSM.2025.3525554

Ozdayi, M. A., Demir, S., & Ferhatosmanoglu, H. (2021, June 17). *Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks*. https://proceedings.mlr.press/v139/schwarzschild21a/schwarzschild21a.pdf

Paudice, A., Munoz-Gonzalez, L., Gyorgy, A., & Lupu, E. (2018). *Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection*. https://www.studocu.com/en-au/document/macquarie-university/data-science/detection-of-adversarial-training-examples-in-poisoning-attacks/130324289

Qin, Z., Zhang, Y., & Yang, R. (2023, October 24). *Poison is not traceless: Fully-agnostic detection of poisoning attacks*. https://dblp.org/rec/journals/corr/abs-2310-16224.html

Shejwalkar, V., & Houmansadr, A. (2021, December 13). *Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning*. https://www.semanticscholar.org/paper/Back-to-the-Drawing-Board%3A-A-Critical-Evaluation-of-Shejwalkar-Houmansadr/fcf391eeded299fc8761f114302b558a7a8e4df7

Shen, S., Jin, C., Zhang, W., & Wang, B. (2021). *Backdoor Scanning for Deep Neural Networks through K-Arm Optimization*. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 10832–10840.

Software Engineering Institute, Carnegie Mellon University. (2025, March). *Improving disaster response with the xView2 Challenge*. Retrieved January 18, 2025, from https://insights.sei.cmu.edu/projects/xview-2-challenge/

Sun, X., Jiang, Y., & Liu, X. (2023, July 28). *Adversarial training for tabular data with attack propagation*. https://www.researchgate.net/publication/372766979_Adversarial_training_for_tabular_data_with_attack_propagation

Szyller, S., Pawlicki, M., & Szczepaniak, P. (2023, March 9). *Wild patterns reloaded: A survey of machine learning security against training data poisoning*. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934932

Tramèr, F., Carlini, N., Brendel, W., & Madry, A. (2020). *On adaptive attacks to adversarial example defenses*. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)* (pp. 1539–1557). Association for Computing Machinery. https://doi.org/10.1145/3372297.3423370

Wang, H., Liu, Y., Zhang, J., & Wang, X. (2024, April 2). *Vulnerabilities of foundation model integrated federated learning under adversarial threats*. arXiv. https://arxiv.org/abs/2401.10375

Xie, Y., Fang, M., & Gong, N. Z. (2024, June 6). *Model poisoning attacks to federated learning via multi-round consistency*. arXiv. https://arxiv.org/abs/2404.15611

Yang, Z., Wang, J., Zheng, H., & Zhao, B. Y. (2023, September 8). *Avoid adversarial adaption in federated learning by multi-metric investigations*. https://link.springer.com/article/10.1007/s40747-024-01664-0

Yerlikaya, F. A., & Bahtiyar, Ş. (2022). *Data poisoning attacks against machine learning algorithms*. Expert Systems with Applications, 208**,** Article 118101. https://doi.org/10.1016/j.eswa.2022.118101

Yuan, B., & Wu, Y. (2021, January 26). *Property inference from poisoning*. https://eprint.iacr.org/2021/099.pdf

Zhang, H., Li, Y., Xie, Y., & Fang, M. (2021, May 12). *Subpopulation data poisoning attacks*. arXiv. https://arxiv.org/abs/2006.14026

Zhang, Y., & Wang, H. (2021, March 31). *Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses*. https://experts.illinois.edu/en/publications/dataset-security-for-machine-learning-data-poisoning-backdoor-att