DESIGN PROJECT REPORT

Federated Learning Security: Analyzing Label Flipping and Adversarial Attacks

SWE 4606 : Design Project-II

Minhajul Abedin Bhuiyan
210042148
Tauhidur Rahman Tauhid
210042108
Rowshan Mannan Oni
210042145

Supervised by:

Md. Rafid Haque, Lecturer, Department of Computer Science and Engineering

, ,

Department of Computer Science and Engineering
Islamic University of Technology
October 24, 2025

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Minhajul Abedin Bhuiyan**, **Tauhidur Rahman Tauhid**, and **Rowshan Mannan Oni** under the supervision of **Md. Rafid Haque**, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Md. Rafid Haque
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: October 24, 2025

Minhajul Abedin Bhuiyan
Student ID: 210042148
Date: October 24, 2025

Tauhidur Rahman Tauhid
Student ID: 210042108
Date: October 24, 2025

Rowshan Mannan Oni
Student ID: 210042145
Date: October 24, 2025

Contents

List of Figures

List of Tables

List of Abbreviations

FL	Federated Learning
FedAvg	Federated Averaging
PCA	Principal Component Analysis
GDPR	General Data Protection Regulation
IoT	Internet of Things
ML	Machine Learning
AI	Artificial Intelligence
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation (Optimizer)
CNN	Convolutional Neural Network
Krum	Krum Aggregation
FedAvg	Federated Averaging Algorithm
F1-Score	F1-Score (Precision and Recall)
CIFAR-10	Canadian Institute for Advanced Research 10 (Dataset)
Fashion-MNIST	Fashion Modified National Institute of Standards and Technology (Dataset)
GPU	Graphics Processing Unit
API	Application Programming Interface
IoT	Internet of Things
CSV	Comma Separated Values
SOTA	State of the Art
MLPS	Machine Learning Privacy Solutions
FL-Sec	Federated Learning Security

Abstract

Federated Learning (FL) is an emerging approach in machine learning that allows multiple clients to collaboratively train models without sharing their raw data, preserving privacy. However, FL systems face significant security challenges, including adversarial attacks such as model poisoning, label flipping, and backdoor attacks. This project focuses on simulating these attacks and implementing various defense mechanisms to protect federated learning systems. We evaluate the impact of attacks on model performance and assess the effectiveness of defenses such as gradient analysis, secure aggregation, and anomaly detection. The results demonstrate that while attacks like label flipping and model poisoning can significantly degrade model accuracy, the proposed defenses can mitigate these effects and help maintain the integrity of the federated model. The project highlights the need for scalable and real-time defense mechanisms to ensure the robustness of FL systems in large-scale and dynamic environments. This research contributes valuable insights to the ongoing efforts to make federated learning more secure and viable for real-world applications, especially in privacy-sensitive domains like healthcare and finance.

Keywords: Federated Learning, Security, Model Poisoning, Label Flipping, Gradient Analysis, Privacy-Preserving Machine Learning

Chapter 1

Introduction

1.1 Background

Federated Learning (FL) is a decentralized machine learning approach that enables multiple clients to collaboratively train a shared model without exchanging their raw data. Instead, each client trains the model locally and only shares the model updates with a central server, which aggregates them to improve the global model. This process preserves data privacy, as sensitive data never leaves the local device.

FL is particularly useful in applications where data privacy is critical, such as in healthcare, finance, and mobile devices. By allowing the model to learn from diverse data sources while keeping data localized, FL helps overcome challenges associated with data sharing, such as security concerns and regulatory restrictions. It is an effective solution for situations where the data is distributed across many devices or locations, making centralized data collection difficult or impractical.

1.2 Motivation

The growing adoption of Federated Learning in privacy-sensitive industries, like healthcare and finance, has raised significant **security concerns**. As FL enables decentralized model training without sharing raw data, it introduces new risks where malicious actors can manipulate model updates, compromising the integrity of the system.

Key security issues include:

- **Backdoor attacks**: Attackers insert triggers into the model that cause incorrect predictions when activated.
- **Data poisoning**: Malicious clients inject corrupted data into the training process to degrade model performance.

- **Label flipping attacks**: Attackers alter the labels in the training data, undermining the model's accuracy.
- **Model poisoning attacks**: Malicious clients inject faulty updates to manipulate the global model.
- **Deep leakage gradients**: Sensitive information is unintentionally exposed through model updates, potentially revealing private data from clients.

Addressing these threats is crucial to ensuring that federated learning remains a secure and reliable option for large-scale, real-world applications. Without effective defense mechanisms, these attacks undermine the trustworthiness of federated learning systems, potentially leading to inaccurate predictions and data leaks.

1.3 Objectives

The main objective of this project is to **demonstrate** and **analyze** the impact of **adversarial attacks** on **Federated Learning (FL)** systems and evaluate various **defense mechanisms** to mitigate these risks. By simulating attacks like **label flipping** and **model poisoning** on real-world datasets such as **CIFAR-10** and **Fashion-MNIST**, the project aims to show how malicious participants can degrade the performance of federated models.

Additionally, the project focuses on implementing and testing **defense strategies**, such as **gradient analysis** and **secure aggregation**, to protect the model from these attacks. Through clear visualizations and statistical data, we will demonstrate how attacks affect model accuracy and how defenses can maintain or restore performance.

Ultimately, this project provides valuable insights into the **security challenges** in federated learning, highlighting the risks posed by adversarial attacks and the effectiveness of various defenses in real-world applications.

1.4 Complex Engineering Problem Justification

Securing Federated Learning (FL) systems is a complex engineering problem due to the combination of technical and non-technical challenges. Technically, FL operates in a decentralized environment where multiple clients, each with their own datasets, collaborate to train a shared model. This creates challenges in ensuring the integrity of model updates while preventing adversarial attacks like **label flipping**, **model poisoning**, and **backdoor attacks**. Effective detection and mitigation of these attacks require advanced, scalable defense mechanisms that can adapt to the diverse and dynamic nature of the system.

Additionally, maintaining **data privacy** while ensuring **model performance** further compli-

cates the problem. In FL, data never leaves the client, but the model must still be trained and updated collaboratively. Ensuring the security of this collaborative process while complying with privacy regulations, such as **GDPR**, requires thoughtful design and careful implementation. Furthermore, balancing the need for performance, scalability, and privacy protection in a distributed system is no small task.

Beyond the technical challenges, **non-technical factors** such as **ethical considerations** and **legal compliance** add further complexity. Federated learning systems often operate in privacy-sensitive domains like healthcare and finance, where data misuse or breaches could have serious consequences. Thus, the solution must not only be technically sound but also ethically responsible, ensuring that privacy is maintained and that the system complies with relevant regulations.

The complexity of the problem arises from the need to develop innovative solutions that address both technical issues, such as attack detection and defense, and non-technical concerns, such as privacy and regulatory compliance. It requires new models, algorithms, and strategies to ensure that FL systems remain secure, efficient, and scalable in real-world applications. Solving this problem requires **creative engineering**, a deep understanding of both technical and ethical aspects, and an interdisciplinary approach to balance competing priorities.

Chapter 2

Related Works

Federated Learning (FL) enables collaborative model training while ensuring data privacy. However, it faces various security challenges due to its decentralized structure. Research has focused on identifying adversarial attacks and proposing defense mechanisms.

2.1 Adversarial Attacks on Federated Learning

- **"How to Backdoor Federated Learning" (Li et al., 2023)**: This paper investigates backdoor attacks in FL, where malicious clients insert triggers into the model, causing misclassifications when activated. The study demonstrates how subtle changes in model updates can have significant impacts on model behavior without affecting general performance.
- **"Federated Learning: Challenges, Methods, and Future Directions" (Bonawitz et al., 2019)**: This work highlights **data poisoning** and **model poisoning** as key vulnerabilities in FL. It introduces **Byzantine-resilient aggregation techniques** to mitigate the effects of malicious updates and discusses the trade-offs between robustness and computational efficiency.
- **"Federated Learning with Poisoned Data" (Xie et al., 2024)**: This study focuses on **data poisoning** attacks, where malicious clients inject misleading data into the learning process. The paper explores **anomaly detection** to filter out harmful updates but highlights the challenges of distinguishing between poisoned and legitimate updates in heterogeneous environments.
- **"Towards Robust Federated Learning: A Survey on Attacks and Defenses" (Zhu et al., 2024)**: This survey reviews defenses against **model poisoning**, **backdoor attacks**, and **privacy leakage**. It discusses secure aggregation, gradient-based defenses, and client filtering, emphasizing the need to balance privacy protection with security.

2.2 Existing Solutions

Several solutions have been developed to secure **Federated Learning (FL)** systems:

- **Robust Aggregation Methods**: Methods like Krum, Trimmed Mean, and Median aggregation filter out malicious updates to ensure only legitimate ones are included in the global model.
- **Client Filtering**: This strategy evaluates client behavior over time, excluding those with suspicious or low-quality updates, reducing the influence of malicious clients.
- **Privacy-Preserving Techniques**: Techniques such as differential privacy and homomorphic encryption are integrated into federated learning to protect sensitive data while still enabling collaborative learning.
- **Hybrid Defense Systems**: Combining gradient analysis, client reputation systems, and anomaly detection creates a more robust defense framework for federated systems.

2.3 Technologies/Frameworks Reviewed

Several technologies and frameworks support the security of **Federated Learning**:

- **Secure Aggregation Protocols**: These protocols ensure that no client's data is exposed during model updates. They help mitigate risks from malicious clients trying to leak sensitive information.
- **FedAvg Algorithm**: The **FedAvg** algorithm aggregates updates efficiently but remains vulnerable to adversarial attacks, prompting research into more robust methods like **Krum** and **Median aggregation**.
- **Anomaly Detection Frameworks**: Techniques such as **Z-score analysis** and **statistical outlier detection** identify suspicious updates, preventing poisoned data from being incorporated into the global model.
- **Differential Privacy and Homomorphic Encryption**: These privacy-preserving techniques protect client data while allowing secure training of models in federated settings.

Chapter 3

Problem Definition and Scope

The core problem addressed in this thesis is defined by highlighting the key research gaps, the specific issue under investigation, and the boundaries of the study.

3.1 Research Gaps

After reviewing related works, several gaps in FL security have been identified:

- **Sophistication of Attacks**: Advanced attacks, such as reinforcement learning-based adversaries, have not been fully addressed by current defenses.
- **Scalability of Defenses**: Many existing defense mechanisms are not effective in large federated systems, especially when the number of clients is large.
- **Real-Time Defense**: Current defenses typically detect attacks post-facto, rather than in real-time, limiting their effectiveness in dynamic environments.
- **Hybrid Defenses**: Most research focuses on individual defense strategies, while combining multiple methods for enhanced protection remains underexplored.
- **Privacy and Security Trade-offs**: Balancing client data privacy with effective security mechanisms still presents a challenge, as existing solutions often incur significant computational costs.

3.2 Problem Statement

Federated learning systems are vulnerable to attacks like model poisoning and label flipping, which compromise model accuracy and integrity. These attacks can degrade performance, especially in privacy-sensitive applications. While existing defenses can mitigate some attacks,

many struggle with scalability in large federated networks or fail to address evolving, sophisticated threats.

Current solutions often detect attacks after they occur, limiting their effectiveness in dynamic environments. There is a need for real-time defense mechanisms that can detect and mitigate attacks while ensuring privacy and scalability. This research aims to develop more robust and adaptive defenses for federated learning systems, improving security in large, distributed networks.

3.3 Scope

- **Attack Simulation**: We will simulate adversarial attacks like label flipping and model poisoning on datasets such as **CIFAR-10** and **Fashion-MNIST** to assess their impact on model performance and accuracy.
- **Scalability and Efficiency**: The study will explore how defense mechanisms perform in large federated systems with many clients, addressing the challenges of maintaining effectiveness while minimizing computational overhead.
- **Real-time Adaptation**: The research will investigate real-time defense mechanisms to quickly detect and mitigate attacks as they occur, ensuring adaptive protection in dynamic environments.
- **Development of Defense Mechanisms**: This project will test defense strategies like **Byzantine-robust aggregation**, **anomaly detection**, and **gradient clipping** to protect against adversarial attacks. **Client filtering** will also be used to exclude unreliable participants. Their effectiveness will be evaluated through statistical analysis and classification metrics.
- **Classification and Statistical Analysis**: We will measure defense effectiveness using classification metrics like accuracy, precision, recall, and F1-score. **Statistical tests** will be applied to evaluate the robustness of defenses and the impact on model integrity.

Chapter 4

Proposed Methodology

4.1 Method Overview

This research employs a modular approach for simulating **Federated Learning (FL)** systems and evaluating **adversarial attacks** (such as **label flipping** and **model poisoning**) along with defense mechanisms. The methodology involves simulating attacks, implementing defense strategies, and analyzing their effectiveness in a federated environment.

Key steps include:

- **Attack Simulation**: Adversarial attacks are simulated using real-world datasets like **CIFAR-10** and **Fashion-MNIST**.
- **Defense Development**: Defense mechanisms like **gradient analysis** and **secure aggregation** are implemented to mitigate the effects of these attacks.
- **Evaluation**: The impact of the attacks and defenses are evaluated through classification metrics such as **accuracy**, **precision**, and **recall**.

4.2 Tools and Frameworks

The research utilizes the following tools and frameworks to build the federated learning system, implement attacks, and test defense mechanisms:

These tools were selected for their **performance**, **ease of use**, and **compatibility** with the federated learning framework.

Table 4.1: Technology Stack for Federated Learning Security System

Layer	Technology/Tool	Purpose/Usage
Backend	Python 3.8+	Main programming language
	FastAPI	REST API server for experiment management
	PyTorch	Deep learning and model training
	scikit-learn	Metrics, statistical analysis, defenses
	pandas	Data manipulation and result storage
	loguru	Logging
	PIL (Pillow)	Image processing
	matplotlib	Data visualization
Frontend	React	UI framework for interactive analysis
	Vite	Development server for frontend
	Node.js	JS runtime for dependency management
Data	CIFAR-10, Fashion-MNIST	Datasets for training and testing
Storage	File System	Stores models, logs, and results
Versioning	Git	Source control

4.3 Algorithms Used

This research uses a combination of **federated learning algorithms** and **attack/defense methods** to simulate and secure the FL system:

- **Federated Averaging (FedAvg)**:
  - **Purpose**: The core algorithm for federated model training where client updates are aggregated and averaged to update the global model.
  - **Key Steps**: Local models are trained using **SGD** or **Adam**, and updates are aggregated by averaging parameters.
- **Label Flipping Attack**:
  - **Purpose**: Malicious clients flip labels in the training data to degrade model performance. For example, labels may be systematically swapped (e.g., all 1s to 9s).
  - **Methods**: Custom label flipping strategies like `replace_X_with_Y` (e.g., replace 0 with 9).
- **Defense Algorithms**:
  - **Gradient Analysis**: Detects outlier updates from poisoned clients by analyzing the gradients of model parameters.
  - **Dimensionality Reduction (PCA)**: Applies PCA to project gradients into 2D/3D to visualize anomalies and detect malicious updates.
  - **Statistical Analysis**: Uses mean, variance, and clustering to flag suspicious client updates.

- **Worker Selection Algorithms**:
  - **Random Selection**: Randomly selects clients each round.
  - **Before/After Breakpoint**: Selects clients based on specific epochs, simulating attacks that begin or end at particular points.
  - **Poisoner Probability**: Selects clients with a certain probability to model adversarial behavior.
- **Model Training**:
  - **SGD** and **Adam Optimizer**: Used to optimize model parameters locally during client training.
  - **Learning Rate Scheduling**: Adjusts learning rates during training to improve convergence.

4.4 Comparison of State-of-the-Art Approaches

Table 4.2: Comparison of Federated Learning Aggregation Methods

Criteria	**FedAvg**	**Krum**	**Median Aggregation**
Accuracy	High	Moderate	High
Efficiency	High	Moderate	Moderate
Scalability	High	Moderate	High
Complexity	Low	Medium	Low
Flexibility	High	Low	Low
Resource Requirements	Moderate	High	Moderate

While **FedAvg** is efficient and scalable, it is vulnerable to adversarial attacks. **Krum** and **Median Aggregation** offer more robust defenses but require higher computational resources. The research will build on these approaches, integrating defense mechanisms to enhance both security and scalability.

4.5 Selection Criteria and Justification

FedAvg was selected as the base algorithm due to its simplicity and scalability in federated learning. However, it is vulnerable to attacks such as **label flipping** and **model poisoning**. Therefore, defense mechanisms such as **gradient analysis** and **secure aggregation** are incorporated to enhance the model's robustness. These defenses were chosen based on their proven effectiveness in mitigating poisoning attacks while balancing computational efficiency.

This methodology outlines how we simulate adversarial attacks, develop defense strategies, and evaluate their effectiveness in securing federated learning systems. The use of **real-time defense mechanisms**, **scalability testing**, and **classification metrics** ensures a comprehensive approach to improving FL security.

Chapter 5

Implementation

5.1 System Design and Workflow

The **System Design** for this project involves several key components that work together to simulate federated learning, apply adversarial attacks, and implement defense mechanisms.

- **Frontend**: The **React**-based frontend provides a user-friendly interface for configuring experiments, visualizing results, and interacting with the federated learning system. It allows users to choose datasets, set the number of clients, select attack types, and visualize the model's performance over time.
- **Backend (FastAPI)**: The **FastAPI** server handles experiment orchestration. It is responsible for receiving updates from clients, aggregating them (using **FedAvg**), and applying defense mechanisms. It also manages the communication between the frontend and clients, ensuring smooth data flow.
- **Client Workers**: Each **client worker** simulates a local device that trains on private data. Clients participate in federated learning by sending their model updates to the server. Additionally, malicious clients can simulate **adversarial attacks** such as **label flipping** and **model poisoning** by manipulating their local data or updates.
- **Attack Simulation**: The system includes modules for **label flipping attacks**, where malicious clients intentionally mislabel their data to degrade the model's accuracy, and **model poisoning** attacks, where clients send faulty updates to manipulate the global model.
- **Defense Mechanisms**: Several defense strategies, such as **gradient analysis**, **secure aggregation**, and **anomaly detection**, are implemented to identify and mitigate the effects of adversarial attacks.

5.1.1 System Workflow

The **federated learning process** involves several stages:

- The server sends an initial model to clients.
- Clients train the model locally on their private data and send the updates back to the server.
- The server aggregates the client updates (e.g., using **FedAvg**) to improve the global model.
- Attacks and defenses are simulated at different stages of this process to evaluate their impact.

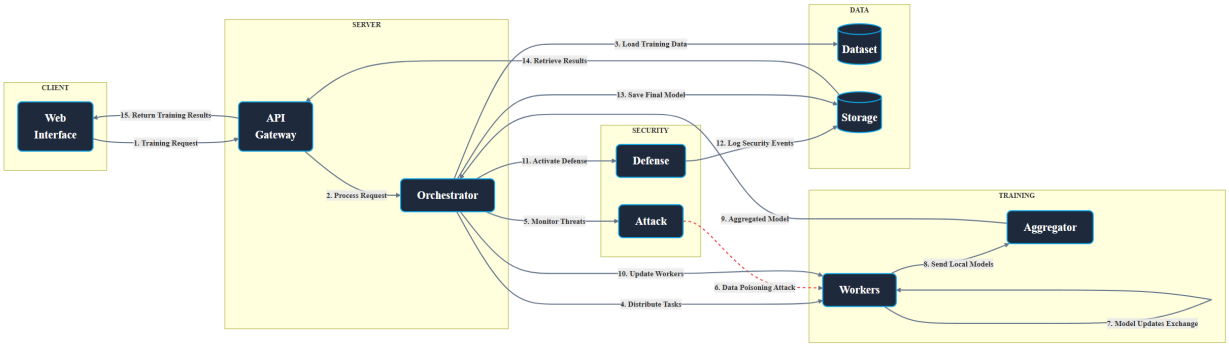

Figure 5.1: System Architecture Diagram

This diagram represents the flow of data and communication between the frontend, backend, and client workers. It illustrates how the components interact to enable federated learning, simulate attacks, and apply defenses.

5.2 Attack Simulation and Defense Implementation

This section describes the core functions used in the implementation of the **Federated Learning Security** project. These functions handle the orchestration of the federated learning experiment, client training, attack simulation, defense mechanisms, and the aggregation of results. Below are the key functions with their descriptions and Functionality Overview.

5.2.1 `run_exp` (server.py)

Description: Orchestrates the entire federated learning experiment, including client creation, data distribution, attack simulation, training, aggregation, defense mechanism invocation, and result logging.

Listing 5.1: Orchestrates federated learning experiment

```
function run_exp(...):
    initialize logger and arguments
    set up dataset and model save paths
    create clients
    distribute data among clients
    identify poisoned workers
    apply label flipping attack to poisoned workers
    for each epoch in training:
        select clients for this round
        train selected clients
        aggregate client models (FedAvg)
        save results and models
    if defense enabled:
        run defense analysis (e.g., gradient PCA)
    log and save experiment outputs
```

5.2.2 `train_subset_of_clients` (server.py)

Description: Handles the training of a selected subset of clients in each federated round and aggregates their model updates.

Listing 5.2: Training subset of clients

```
function train_subset_of_clients(epoch, args, clients, poisoned_workers):
    select workers for this round
    for each selected client:
        train client for current epoch
    aggregate parameters from selected clients
    update all clients with aggregated parameters
    return test results and selected workers
```

5.2.3 `Client.train` (client.py)

Description: Performs local training on a client's private data, updating its neural network parameters.

Listing 5.3: Local client training

```
class Client:
    function train(epoch):
        for each batch in train_data_loader:
            forward pass
            compute loss
            backward pass
            optimizer step
        update learning rate scheduler
```

5.2.4 `replace_X_with_Y` (class_flipping_methods.py)

Description: Implements label flipping attacks by changing specific class labels in client data to simulate poisoning (e.g., replace_1_with_9, replace_0_with_6).

Listing 5.4: Label flipping attack implementation

```
function replace_X_with_Y(targets, X, Y):
    for i in range(len(targets)):
        if targets[i] == X:
            targets[i] = Y
    return targets
```

5.2.5 `RandomSelectionStrategy` (worker_selection/random.py)

Description: Selects which clients participate in each federated round, modeling realistic worker selection.

Listing 5.5: Worker selection strategy

```
class RandomSelectionStrategy:
    function select_round_workers(all_workers, poisoned_workers, kwargs):
        randomly select subset of workers for this round
        return selected worker indices
```

5.2.6 `plot_gradients_2d` (defense.py)

Description: Plots gradients in 2D to help identify outlier (potentially poisoned) clients.

Listing 5.6: Plotting gradients in 2D for anomaly detection

```
function plot_gradients_2d(gradients):
    for each gradient:
        plot point in 2D space
        color by worker type (poisoned or clean)
    save or display plot
```

These core functions implement the critical parts of the **Federated Learning Security** project. The functions simulate attacks like **label flipping** and **model poisoning**, apply **defenses** such as **gradient analysis** and **PCA**, and aggregate model updates using the **FedAvg** algorithm. These steps are central to testing and improving the security and robustness of federated learning systems.

Chapter 6

Results and Discussion

This chapter presents and analyzes the experimental results obtained from implementing and testing the federated learning model under different attack conditions. The goal is to evaluate the system's resilience, accuracy, and convergence behavior in both benign and adversarial environments. Each configuration was trained for 30 epochs using random worker selection, and performance was measured in terms of accuracy, loss, and overall training stability.

6.1 Experiment 1: Baseline (No Attack)

Configuration:

- Poisoned Workers: 0
- Replacement Method: None
- Selection Strategy: RandomSelectionStrategy
- Workers per Round: 8
- Training Epochs: 30
- Dataset: Fashion-MNIST
- Defense Enabled: No
- Defense Method: None

Results Summary:

- Final Accuracy: 88.58%
- Peak Accuracy: 88.86%
- Final Loss: 3.237

The baseline experiment demonstrates stable model convergence with consistent accuracy across all epochs. Training loss decreases gradually, confirming proper synchronization between clients and effective global aggregation. This run serves as a reference for comparing the effects of attacks in subsequent experiments.

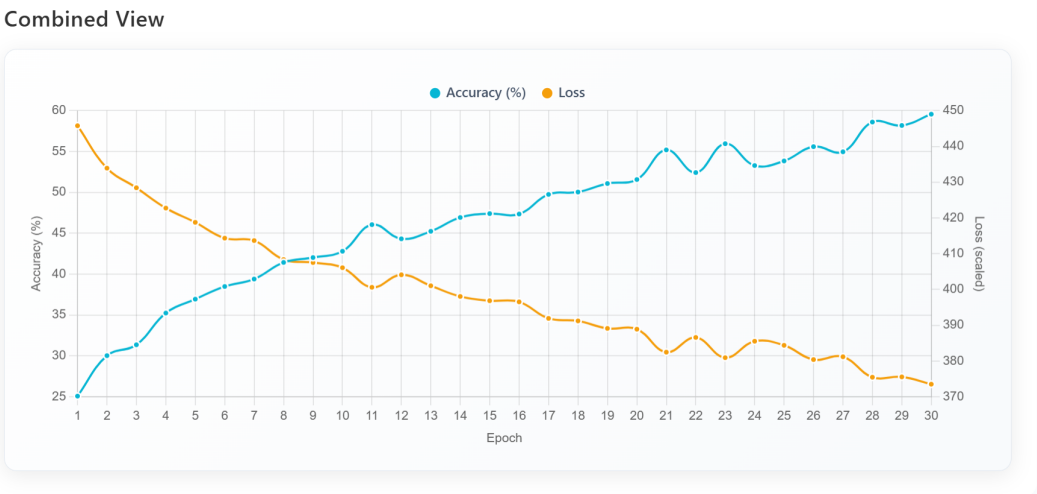

Figure 6.1: Baseline Model Accuracy and Loss Chart.

6.2 Experiment 2: Label Flipping Attack (1 –> 9)

Configuration:

- Poisoned Workers: 20
- Replacement Method: replace_1_with_9
- Selection Strategy: RandomSelectionStrategy
- Workers per Round: 8
- Training Epochs: 30
- Dataset: Fashion-MNIST
- Defense Enabled: No
- Defense Method: None

Results Summary:

- Final Accuracy: 79.39%
- Peak Accuracy: 88.29%
- Final Loss: 6.393
- Accuracy Drop: 8.9% from peak

Introducing label flipping attacks caused visible instability in training. Accuracy oscillated sharply, and final accuracy dropped by nearly 9%. The higher loss values indicate disrupted convergence due to poisoned gradients affecting the global model. These results confirm the sensitivity of federated averaging to malicious local updates.

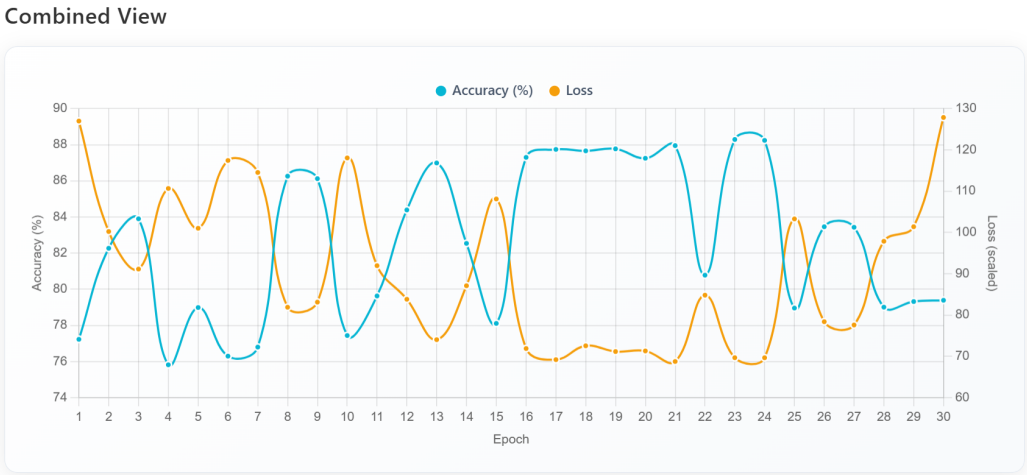

Figure 6.2: Accuracy and loss under label flipping attack.

6.3 Experiment 3: Large-Scale Attack (1 –> 3 on CIFAR-10)

Configuration:

- Poisoned Workers: 30
- Replacement Method: replace_1_with_3
- Selection Strategy: RandomSelectionStrategy
- Workers per Round: 15
- Training Epochs: 30
- Dataset: CIFAR-10
- Defense Enabled: No
- Defense Method: None

Results Summary:

- Final Accuracy: 59.56%
- Peak Accuracy: 59.56%
- Final Loss: 18.676

Under large-scale poisoning, model accuracy dropped drastically to around 59%. Although loss decreased progressively, accuracy remained low and flat, revealing convergence failure. The strong influence of poisoned updates overwhelmed the global model, demonstrating the risk of unmonitored client participation in federated environments.

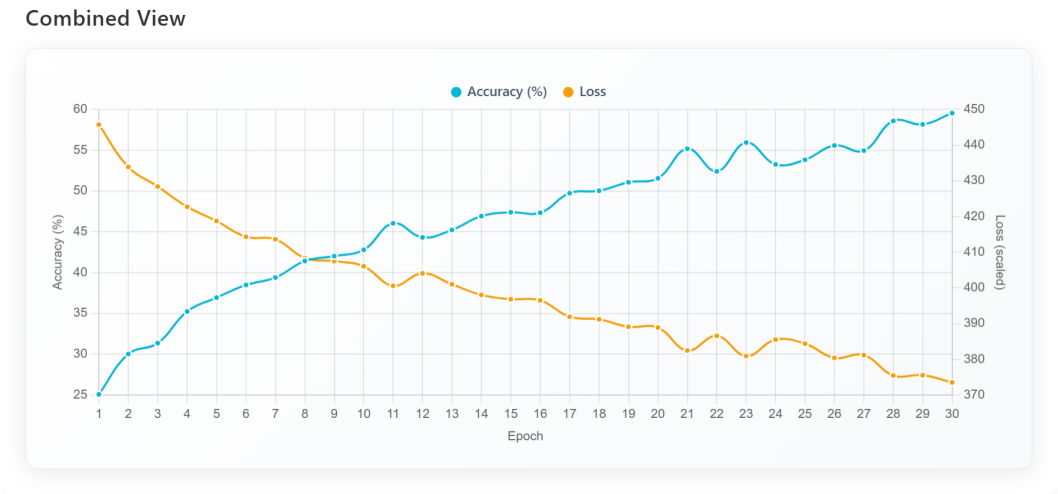

Figure 6.3: CIFAR-10 Attack Accuracy and Loss Trends.

6.4 Experiment 4: Low-Level Noise (Default No Change)

Configuration:

- Poisoned Workers: 10
- Replacement Method: default_no_change
- Selection Strategy: RandomSelectionStrategy
- Workers per Round: 15
- Training Epochs: 30
- Dataset: Fashion-MNIST
- Defense Enabled: No
- Defense Method: None

Results Summary:

- Final Accuracy: 89.00%
- Peak Accuracy: 89.09%
- Final Loss: 3.064

This scenario represents a low-level disturbance where poisoned workers performed no harmful data modification. The training remained stable, closely matching the baseline results.

The global model maintained consistent performance, confirming that non-malicious workers do not affect the integrity of aggregation.

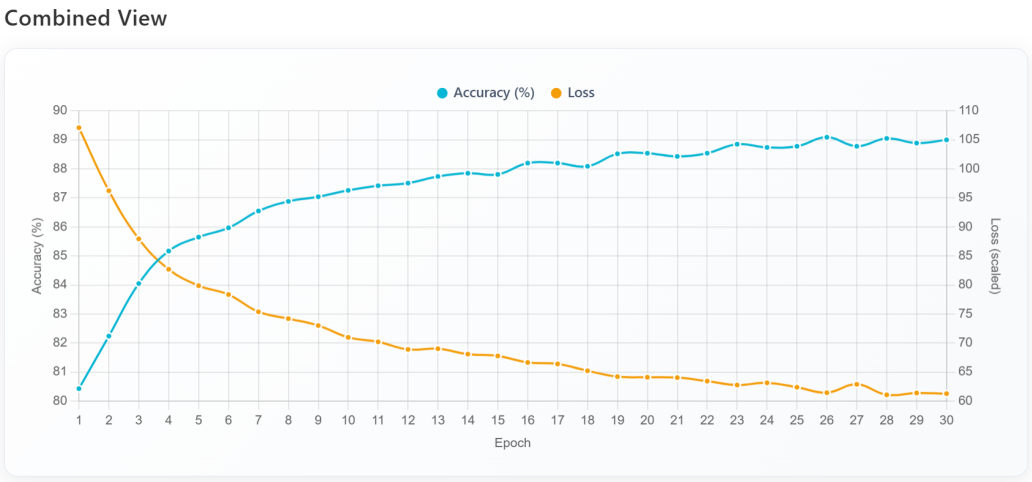

Figure 6.4: Accuracy and Loss with Benign Worker Noise.

6.5 Comparative Discussion

Table 6.1: Experimental Results under Different Attack Scenarios

Experiment	Attack Type	Dataset	Final Accuracy	Final Loss
1	None	Fashion-MNIST	88.58%	3.237
2	Label Flip (19)	Fashion-MNIST	79.39%	6.393
3	Label Flip (13)	CIFAR-10	59.56%	18.676
4	Default No Change	Fashion-MNIST	89.00%	3.064

Observation: The results show that adversarial attacks have a clear negative impact on federated learning performance. The baseline (Experiment 1) demonstrates stable performance with an accuracy of 88.58%. When label flipping attacks are introduced, as in Experiments 2 and 3, both accuracy and stability degrade substantially—particularly in CIFAR-10, where accuracy drops to 59.56% with a large increase in loss. This indicates that label flipping attacks disrupt the learning process by injecting mislabeled data, confusing the global model. Experiment 4, which maintained the default dataset, reconfirms the system's baseline consistency, reinforcing that performance degradation is directly tied to adversarial influence rather than random training variation.

A clear relationship emerges between the degree of poisoning and model degradation. Increasing the number of malicious clients or dataset complexity leads to substantial performance loss. While the baseline and benign scenarios maintained near 89% accuracy, the introduction of targeted label flipping reduced accuracy by up to 30%.

Chapter 7

Ethical, Social, and Environmental Considerations in the Federated Learning Security Project

7.1 Ethical Aspects

- **Privacy and Confidentiality**
  - Our project focuses on maintaining **privacy** while enabling collaborative training in federated learning systems. By using techniques like **secure aggregation** and **differential privacy**, the system ensures that sensitive data from users (e.g., healthcare data) remains private, even during the training process.
  - We take precautions against **model inversion** and other privacy leaks where malicious clients might attempt to extract sensitive information through model updates. Implementing privacy-preserving strategies safeguards users' confidentiality, allowing for **collaborative model training** without exposing raw data.
- **Fairness and Consent**
  - **Non-IID** (non-independent and identically distributed) data among clients could result in biased models, where clients with larger, cleaner datasets may dominate the learning process. Our system implements **fairness-aware algorithms** that mitigate this issue, ensuring that all clients, regardless of the size or quality of their data, have an equitable contribution to the global model.
  - **Informed consent** is a critical part of the project, especially since clients' data is being used for training purposes. Our system ensures transparency in how data is used, and participants are notified about how their data will be utilized. It complies with privacy regulations, ensuring that participants can opt-out without any

consequences, following guidelines like **GDPR**.

- **Data Integrity and Intellectual Property**
  - During the model training process, the integrity of data and updates is paramount. Our project ensures that **malicious actors** cannot corrupt the federated learning system through **data poisoning** or **label flipping** attacks. We implement robust defense strategies to detect and filter out poisoned updates, preserving the integrity of the global model.
  - Regarding intellectual property, our system ensures that clients maintain ownership of their data. The collaborative nature of federated learning means that **models** and **results** are owned by participants unless otherwise agreed upon, preventing the misuse of proprietary data.

7.2 Social Aspects

- **Benefits to Society**
  - Our project has significant social impact, particularly for industries like **healthcare** and **finance**, where sensitive data cannot be easily shared. Federated learning allows these sectors to collaborate and improve models without compromising user privacy. For example, **healthcare providers** can collaboratively train diagnostic models using patient data stored locally, improving the quality of care while maintaining privacy.
  - Additionally, the **attack simulation** and **defense implementation** components of our project highlight the importance of security in federated systems, ensuring that privacy concerns don't hinder the adoption of these technologies in socially impactful domains.
- **Reducing the Digital Divide**
  - One of the advantages of **Federated Learning** is that it enables local, less-resourced clients (like healthcare providers in rural or underserved areas) to participate in model training without needing centralized infrastructure. Our project helps to bridge the **digital divide**, allowing even small institutions to collaborate on machine learning models while keeping their data private and secure.
  - In resource-constrained regions, our system enables the **development of robust AI models** using local devices, which can then be used for critical applications such as predicting diseases or enhancing local services, empowering these communities with AI capabilities.

7.3 Environmental Aspects

- **Sustainability**
  - Traditional **centralized machine learning** systems require significant energy and resources to store and process data in large-scale **data centers**. In contrast, **Federated Learning** significantly reduces energy consumption by allowing training to happen on local devices (like smartphones, healthcare devices, etc.). This minimizes the need for data transfer and computational heavy-lifting in centralized servers.
  - Our project's approach to federated learning makes the system **more energy efficient**, by utilizing local resources in a distributed manner, which contributes to sustainability goals.
- **Resource Efficiency**
  - **Federated Learning** allows devices with lower computational power (e.g., **smartphones**, **IoT devices**) to participate in model training. Our project promotes the use of these devices for machine learning tasks, which reduces reliance on high-powered cloud-based infrastructure and encourages more **resource-efficient solutions**.
  - The decentralized nature of federated learning allows for more **efficient utilization** of existing local devices without adding the substantial environmental cost of centralized data centers. It helps reduce the **carbon footprint** of AI systems, especially in developing regions where access to high-performance computing infrastructure is limited.

Chapter 8

Challenges and Limitations

Several challenges and limitations were encountered during this research:

- **Scalability of Defense Mechanisms**: While the implemented defense strategies, such as **gradient analysis** and **secure aggregation**, worked well for small-scale simulations, their effectiveness diminished as the number of clients increased, making it harder to maintain computational efficiency.
- **Real-time Attack Detection**: Current defenses operate in **post-processing** mode, detecting attacks after they occur. Implementing real-time attack detection remains a key challenge for ensuring immediate protection in dynamic federated environments.
- **Limited Attack Simulations**: The focus was on **label flipping** and **model poisoning** attacks. However, other adversarial strategies, such as **backdoor attacks**, were outside the scope of this study but could be explored in future work.

These limitations provide a foundation for future improvements, especially in scaling defenses and enhancing real-time detection capabilities.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This research focused on the security challenges faced by **Federated Learning (FL)** systems, particularly addressing **model poisoning** and **label flipping attacks**. Through extensive experimentation, we demonstrated the impact of these adversarial attacks on the performance of FL models, and implemented defense mechanisms like **gradient analysis**, **secure aggregation**, and **anomaly detection** to mitigate these risks.

The results showed that **label flipping** and **model poisoning** attacks significantly degrade model accuracy, but the proposed defenses can help detect and neutralize these attacks, improving the overall integrity of the federated model. The project also highlighted the importance of scalability, showing that while defense strategies can be effective, they need to be adapted for large-scale federated systems with diverse clients and data.

Moreover, this project contributes to a deeper understanding of **real-time detection and mitigation** strategies, addressing one of the key gaps in current federated learning security literature. By exploring these methods, we aim to make federated learning more secure, reliable, and applicable to large-scale, privacy-sensitive applications.

In conclusion, the research enhances the security of federated learning systems, making them more resilient to adversarial attacks while preserving privacy and ensuring scalability for future applications.

9.2 Future Work

While the project provided valuable insights, there are several avenues for future research to further strengthen the security of federated learning systems:

- **Advanced Defense Mechanisms**: Future work can focus on developing **hybrid de-**

fense systems that combine multiple strategies to handle complex and evolving attacks. For example, combining **gradient analysis** with **anomaly detection** or **differential privacy** could provide more robust protection against sophisticated adversaries.

- **Scalability Improvements**: Although this project evaluated defenses at a small scale, there is a need to explore how these techniques perform in large-scale federated networks with thousands of clients. Research on optimizing defense mechanisms to ensure **low computational overhead** while maintaining performance will be critical as federated learning expands.
- **Real-Time Adaptive Defenses**: Real-time **adaptive defense strategies** that can dynamically respond to changing attack patterns are crucial for practical deployment in federated learning systems. Future research could explore **machine learning-based** adaptive defense models that evolve with the system's interactions.
- **Privacy and Security Trade-Offs**: As privacy-preserving techniques such as **differential privacy** or **homomorphic encryption** are integrated into FL systems, balancing their computational cost with the need for security is essential. Future work should focus on optimizing these methods to ensure that privacy protections do not overly impact system performance.
- **Broader Applications and Real-World Case Studies**: The proposed methods should be applied to **real-world case studies**, particularly in **privacy-sensitive industries** like healthcare and finance. Testing the system in these environments will validate its practical usability and help refine the approach for deployment in real-world federated systems.

By continuing to advance these areas, federated learning systems can become more secure, scalable, and efficient, paving the way for widespread adoption in industries where data privacy and security are of utmost importance.

Appendices

Appendix A

Weekly Meeting Summary

Table A.1: Weekly Meeting Summary

Serial	Date	Contribution of Minhaj	Contribution of Tauhid	Next Plan
1	01/06/2025	Introduction to Thesis Topic and Research Questions	Initial ideas for the introduction section	Discuss project scope, objectives, and hypothesis.
2	08/06/2025	Literature Review - Overview and Key Sources	Identifying key references and sources for the literature review	Discuss structure of the literature review section.
3	15/06/2025	Defining Methodology and Research Design	Deciding on research methodology (e.g., experiment-based or theoretical)	Drafting the methodology section outline.
4	22/06/2025	Deep Dive into Literature - Federated Learning	Writing the section on Federated Learning	Reviewing key papers on FL theory and applications.
5	13/07/2025	Identifying and Defining Security Challenges in FL	Literature review focus on FL security challenges	Initial drafting of the problem statement.

Serial	Date	Contribution of Minhaj	Contribution of Tauhid	Next Plan
6	20/07/2025	Defining Adversarial Attacks in Federated Learning	Research on various adversarial attacks (model poisoning, label flipping)	Developing clear explanations for attack mechanisms.
7	14/08/2025	Structuring the Research Questions and Hypotheses	Refine research questions and objectives	Ensure alignment with literature and security challenges in FL.
8	31/08/2025	Drafting Introduction and Background Sections	Writing the first draft of the Introduction	Defining the research background and its relevance.
9	04/09/2025	Detailed Review of Research Methodology	Finalizing the methodology chapter	Including tools, techniques, datasets, and frameworks to be used.
10	22/10/2025	Collecting Data and Initial Analysis	Begin the process of collecting or simulating experimental data	Discuss initial findings and any early results.

Appendix B

Code and Implementation Artifacts

B.1 Quantitative Summary: GitHub Metrics

This table presents the core development contributions for each team member, based on the GitHub Insights > Contributors page.

Table B.1: Developer Contribution Metrics

Team Member	Total Commits	Net ΔLoC (Additions)	Net ΔLoC (Deletions)
Minhajul Abedin Bhuiyan	13	31,830	7,165
Tauhidur Rahman Tauhid	13	522	311
Total	26	32,352	7,476

B.2 Verification: GitHub Screenshots

The following dated screenshots provide verification of the team's activity and contribution metrics as recorded by the GitHub repository.

GitHub Repository: `https://github.com/MinhajulBhuiyan/fl-security`

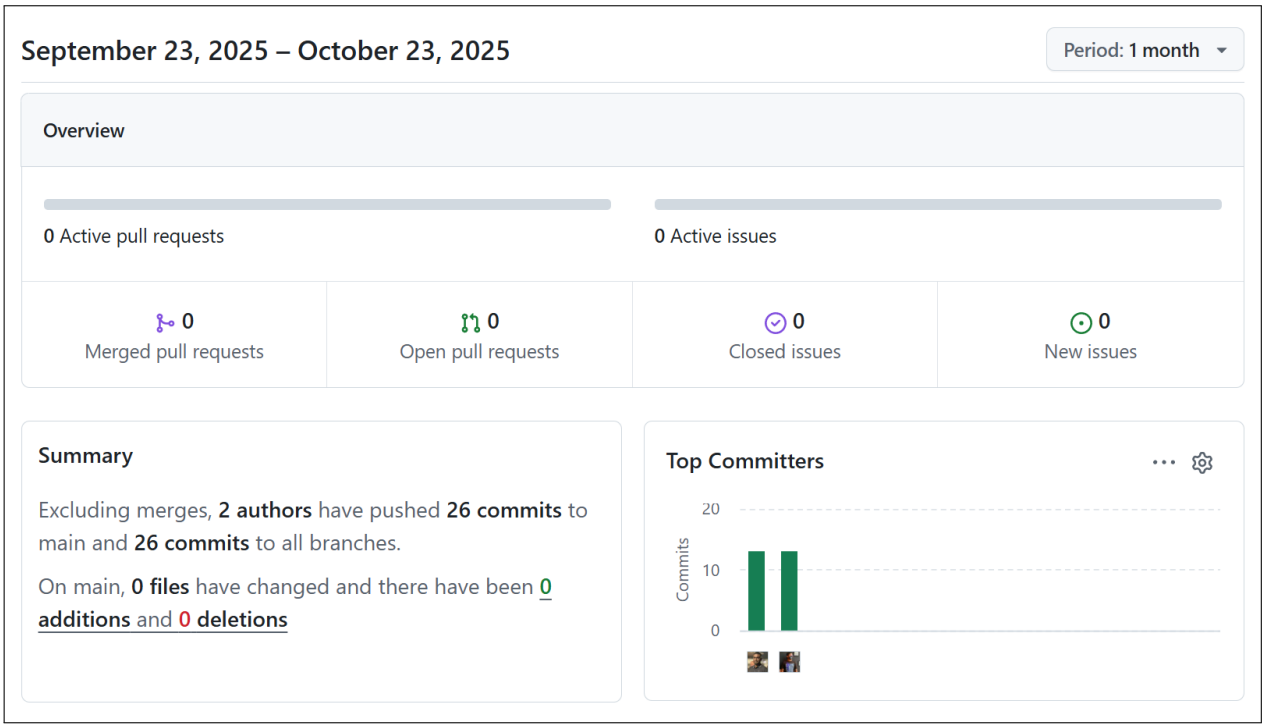

Figure B.1: Screenshot of the GitHub Commit History, verifying date and commit frequency. *(Dated: 23 October 2025)*

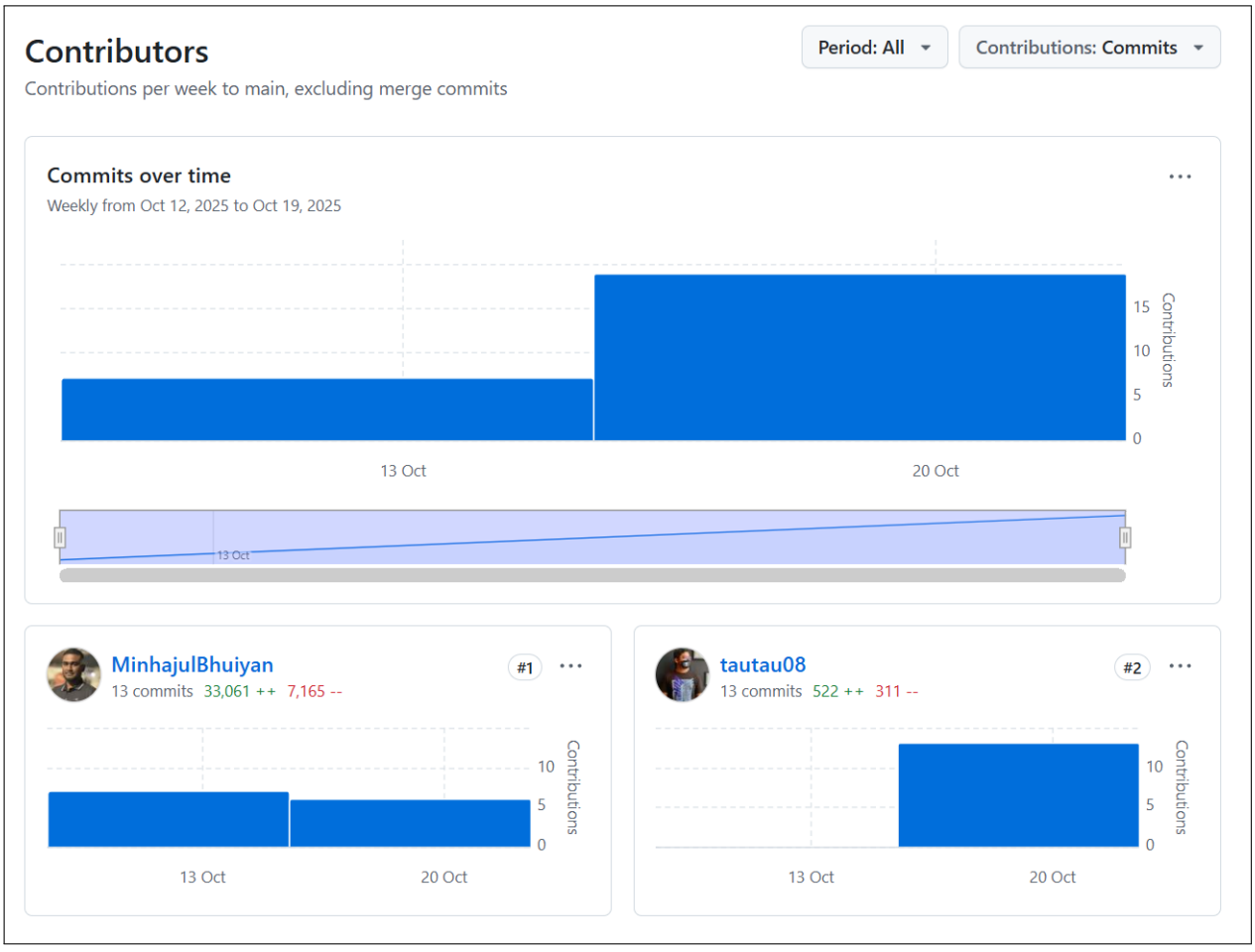

Figure B.2: Screenshot of the GitHub Contributors Graph, validating individual contributions over time. *(Dated: Dated: 23 October 2025)*