UaaS-SFL: Unlearning as a Service for Safeguarding Federated Learning

Wathsara Daluwatta, Ibrahim Khalil, Shehan Edirimannage, and Mohammed Atiquzzaman, *Life Senior Member, IEEE*

***Abstract*—The rapid expansion of the Internet of Things (IoT) and network services has revolutionized technology, enabling numerous intelligent applications. However, this interconnected environment also introduces significant security challenges, particularly the susceptibility of federated learning (FL) systems to poisoning attacks. Such attacks compromise the integrity of the global model by injecting malicious data, leading to inaccurate predictions and potentially endangering system reliability and user safety. While traditional approaches, such as early detection and secure aggregation methods, aim to prevent the aggregation of malicious updates, they are ineffective in addressing threats within systems that have already been compromised and did not initially implement these safeguards. This gap highlights the urgent need for robust post-compromise mitigation strategies in FL security. To address this challenge, we introduce "Unlearning as a Service for Safeguarding Federated Learning" (UaaS-SFL), a novel service designed to seamlessly integrate with any FL management system to remove the impact of poisoning clients and restore the integrity of the global model. UaaS-SFL effectively unlearns the contributions of malicious clients, ensuring both model security and system reliability. Our empirical evaluations, conducted in a simulated IoT environment, demonstrate that our service maintains model accuracy with less than a 10% deviation from the baseline achieved through retraining from scratch, underscoring the efficacy of our methodology in safeguarding FL systems. These results highlight UaaS-SFL as a critical service for securing FL management systems, providing a robust foundation for the continued growth of secure and intelligent IoT applications.**

***Index Terms*—IoT networks, federated unlearning, federated learning, poisoning attack, unlearning service.**

I. Introduction

THE RECENT surge in the Internet of Things (IoT) and network services has created a landscape of ubiquitous sensing and computing capabilities [1]. IoT devices generate vast amounts of data, holding immense potential for intelligent applications in smart healthcare, transportation, and smart cities [2]. Developing such high-quality AI applications requires large volumes of high-quality training data and substantial computational power. Most machine learning (ML), particularly deep learning (DL), algorithms follow a centralized architecture where all data processing and model training occurs on a central server. However, storing large volumes of sensitive data on a central server introduces several security and privacy challenges, including the risk of data breaches [3]. Interacting with third-party network services further complicates these issues, potentially leading to violations of privacy regulations. Therefore, it is necessary to devise innovative strategies to achieve efficient and privacy-enhanced intelligent IoT networks and applications.

Federated Learning (FL) [4] represents a transformative approach to machine learning that addresses the growing concerns around data privacy and security. Unlike traditional centralized training paradigms, FL enables the collaborative training of models across a distributed network of devices, each maintaining local data. This decentralized process allows participants to contribute to the model's improvement without sharing sensitive information with a central server, thus mitigating privacy risks and complying with data sovereignty regulations. As depicted in Figure 1, the architecture of FL ensures that only model updates are transmitted, with no raw data leaving individual devices, enhancing both the privacy and scalability of the system. The advantages of FL have led to its adoption in various domains, particularly where data privacy is paramount. In edge computing environments, FL optimizes resource utilization and latency by performing computations closer to data sources [5]. Crowdsourced systems, benefiting from the collaborative nature of FL, leverage a diverse range of participant data without compromising individual privacy [6]. Additionally, FL has proven particularly valuable in Internet of Things (IoT) networks, where vast amounts of heterogeneous data are generated at the edge, enabling real-time insights while preserving user privacy [7]. With its potential to address the challenges of data decentralization, privacy, and large-scale deployment, FL stands as a critical solution in advancing machine learning across distributed systems.

FL models, while offering significant privacy and collaboration benefits [8], are susceptible to various types of attacks during the training process [9], [10]. These attacks aim to exploit the distributed nature of FL to compromise model integrity, effectiveness, or participant privacy. Data Poisoning Attacks represent a significant vulnerability in FL model training, where attackers introduce maliciously modified or

Received 11 March 2024; revised 29 September 2024; accepted 14 December 2024. Date of publication 19 December 2024; date of current version 22 April 2025. This work is supported by the Australian Research Council Discovery Project (DP220100215). The associate editor coordinating the review of this article and approving it for publication was H. Otrok. *(Corresponding author: Wathsara Daluwatta.)*

Wathsara Daluwatta, Ibrahim Khalil, and Shehan Edirimannage are with the School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia (e-mail: wathsara.daluwatta@student.rmit.edu.au; ibrahim.khalil@rmit.edu.au; shehan.edirimannage@student.rmit.edu.au).

Mohammed Atiquzzaman is with the School of Computer Science, The University of Oklahoma, Norman, OK 73019 USA (e-mail: atiq@ou.edu).

Digital Object Identifier 10.1109/TNSM.2024.3520109

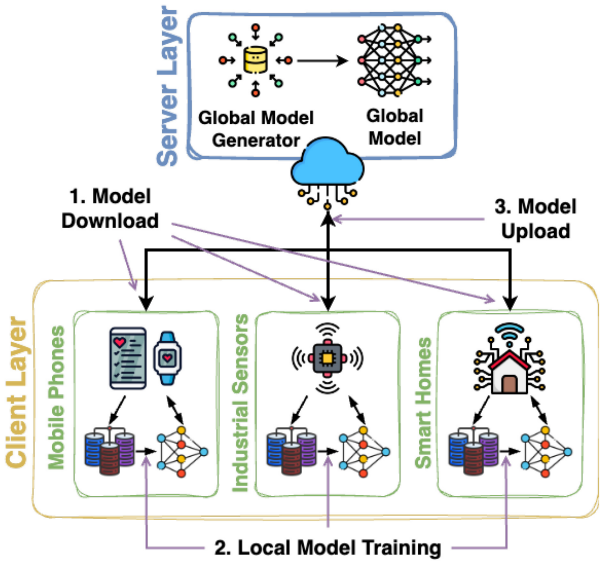

Fig. 1. High-Level Architecture and Communication Process for IoT network in Federated Learning Management System.

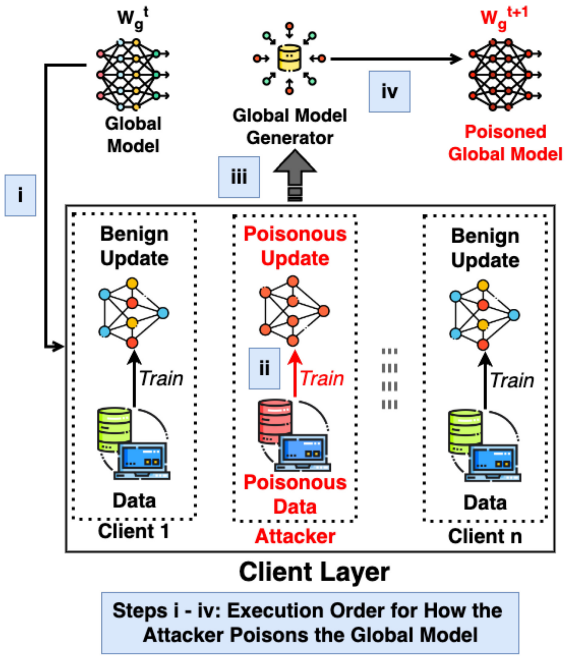

Fig. 2. Demonstrate the process of a poisoning attack in a FL management system. At iteration t, an attacker strategically injects poisoned data, such as backdoor data or label-flipped data, into the global model update w_G. The poisoned update is then sent to the global generator. As a result, the global model w_G at iteration t+1 becomes vulnerable to manipulation and may exhibit compromised performance or biased outcomes.

entirely fabricated data into the training dataset. The goal is to corrupt the learning process, leading the model to make incorrect predictions or decisions. Figure 2 demonstrates the process of a poisoning attack in an FL management system, showcasing how an adversary can tamper with data contributions. This vulnerability can critically impact the integrity, effectiveness, and security of FL models, especially in IoT applications and network services, highlighting the need for robust countermeasures.

Early detection and mitigation of poisoning attacks are crucial for maintaining the integrity of the global model in FL management systems. Traditional pre-detection methods, such as anomaly detection and outlier filtering, aim to evaluate client updates before they are aggregated into the global model [11]. These methods typically involve monitoring the gradients or updates sent by each client to identify irregularities or patterns that deviate from expected behavior, thereby filtering out potentially malicious contributions. Moreover, these techniques tend to be inadequate once the network has been compromised already, as they cannot fully address malicious updates that have already influenced the global model. This inadequacy highlights a critical gap in the security measures of FL systems, particularly in IoT networks, where compromised models pose significant risks to both system functionality and user safety.

Therefore, it becomes imperative to devise methods that can purify the global model post-poisoning attack. Such methods must efficiently identify and neutralize the impact of malicious updates, thus reinstating the model's accuracy and reliability without the necessity for complete retraining. The development of innovative approaches capable of scrutinizing the global model to reverse the effects of compromised data is essential for enhancing resilience against these attacks within federated learning systems. Ensuring this resilience is critical for safeguarding the integrity and functionality of IoT networks.

In this research paper, we introduce a novel service tailored to integrate seamlessly with existing FL management systems, to effectively mitigate the impact of malicious clients on the global model. Our method which we have coined **"Unlearning as a Service"**, employs the innovative concept of federated unlearning. This approach strategically excludes the influence of specific clients' data from the trained global model, focusing not just on the exclusion but also on maintaining the model's overall performance and reliability. The following are the key contributions of our study:

1) We propose a federated unlearning method that enables the selective erasure of specific clients' data influence from the global model in a federated learning system. Unlike existing unlearning methods that often require global retraining or centralized intervention, our approach performs unlearning locally at the client's node by leveraging gradient ascent to reverse the learning effects of their data contributions. To prevent potential degradation in model performance due to over-unlearning, we incorporate the use of Expected Calibration Error (ECE) metrics on a validation dataset. This ensures that the unlearning process does not adversely affect the model's accuracy and reliability.
2) We propose a novel framework designed to integrate seamlessly with existing federated learning systems. Our framework is capable of detecting and mitigating the adverse effects of malicious clients on the global model at any stage of the learning process, including already compromised environments. This is in contrast to traditional protective strategies that require implementation from the outset and often fail to address issues

post-compromise. UaaS-SFL provides a scalable and efficient solution for cleaning compromised global models by selectively removing the influence of poisonous clients without necessitating complete retraining.

3) We evaluate our proposed approach by conducting extensive evaluations within FL scenarios. Our benchmark comparisons involve retraining models sans the identified poisonous clients, providing a clear baseline for assessing the effectiveness of our method. These evaluations were carried out using diverse Convolutional Neural Network (CNN) architectures and three distinct datasets, demonstrating the versatility and robustness of our approach in mitigating the threats posed by poisonous clients in the FL management system.

The remainder of this article is organized as follows: Section II discusses the related work in detail, Section III introduces the preliminaries used to support the proposed framework. Section IV describes the design and specification of the proposed framework and Section V presents the main contributions. Section VII presents the experimental evaluation and Finally Section VIII concludes this article.

II. Related Work

Federated learning (FL) has emerged as a solution to the challenges posed by traditional centralized machine learning models, especially in the context of stringent data privacy regulations like the GDPR [12] and CCPA [13]. By enabling model training across multiple decentralized devices or servers without exchanging or centralizing data, FL helps maintain data privacy and security. This approach is particularly valuable in sensitive fields such as healthcare and finance, where protecting user data is paramount. The introduction of FL [14] by McMahan et al. (2017) marked a significant step forward in addressing these privacy concerns, demonstrating that it is possible to train deep learning models effectively while complying with privacy regulations and minimizing data exposure.

Machine Unlearning (MU) has emerged as a significant area of interest within the research community, driven by escalating privacy concerns and legal imperatives like the "right to be forgotten." This endeavor to reverse the impact of specific data samples on a trained model was first articulated by Cao and Yang [15]. Their pioneering methodology transformed statistical query learning into a form conducive to summation, enabling unlearning through the nuanced adjustment of certain summation components. Building upon these initial insights, Bourtoule et al. presented SISA training [16], a framework designed to accelerate the unlearning process by deliberately diminishing the influence of individual data points during training phases. Further advancements include Guo et al.'s method for certified data removal in machine learning models [17], ensuring that models behave as if certain data were never part of the training set. Neel et al. introduced the "Descent-to-Delete" approach [18], utilizing gradient-based methods to efficiently unlearn data points from models trained with convex losses. Tarun et al. proposed a machine unlearning approach employing error-maximizing noise generation and an impair-repair mechanism to modify model weights [19]. Wu et al. discussed rapid retraining methods like DeltaGrad to facilitate efficient model updates after data deletion [20]. Recent scholarly contributions, such as "Machine Unlearning: A Survey" [21], have conducted a comprehensive analysis of the machine unlearning landscape, systematically categorizing and evaluating the array of existing methodologies based on their inherent properties, and providing an insightful overview of each category's strengths and potential drawbacks. However, it is crucial to acknowledge that these centralized unlearning approaches are not directly transferable to the federated learning (FL) context due to their inherently distributed nature, where no single entity possesses access to the complete dataset. The decentralized architecture of FL introduces unique challenges for unlearning, necessitating specialized methods capable of operating within this environment.

Federated Unlearning necessitates a distinct approach due to the decentralized nature of its training mechanisms compared to typical MU. This concept is explored through three different dimensions: class-level, sample-level, and client-level unlearning [22], with our research focusing specifically on client-level federated unlearning. While a simple approach to eliminating target client influence involves retraining the recommender model from scratch after their data is removed, this method is impractical in real-world recommendation systems due to the significant time and resource constraints it imposes [23]. Liu et al. introduced FedEraser [24], a pioneering work in client-level unlearning, which reconstructs the training process by storing intermediate model updates and uses them to facilitate unlearning. However, this method requires the server to retain updates from each client for every training round, which is impractical for large-scale federated learning networks due to storage and privacy concerns. Alternatively, Wu et al. adopted a Knowledge Distillation strategy [25] aimed at selectively unlearning clients within a federated system, yet this method similarly necessitates the storage of client updates per round, posing scalability challenges. Li et al. [26] employ the Gradient Ascent with differential privacy for federated unlearning. However, the introduction of differential privacy mechanisms may add additional noise, potentially degrading the model's performance. Zhang et al. proposed Fedrecovery [27], which aims to recover federated learning models from poisoning attacks by leveraging historical model checkpoints and robust aggregation. While this method enhances model robustness, it requires the storage of historical models, increasing storage overhead. Furthermore, Wang et al. proposed a federated unlearning method utilizing class-discriminative pruning [28], which allows for the selective removal of information pertaining to specific classes or categories within a federated learning model. Instead of eliminating data associated with entire clients, this approach focuses on forgetting particular data segments based on their class labels. Diverging from these previous approaches, our approach incorporates Gradient Ascent alongside Expected Calibration Error (ECE) [29] to furnish a direct assessment of the model's reliability and performance. This strategy not only aligns with the

fundamental goals of model unlearning and refinement but also addresses the limitations of existing approaches by eliminating the need for extensive storage and offering a scalable solution for FL environments.

In the field of FL, poisoning attacks pose a critical risk to the integrity and reliability of distributed learning networks. These attacks involve malicious participants deliberately submitting harmful updates to skew the aggregated model, aiming to degrade overall performance or introduce specific vulnerabilities. Bagdasaryan et al. demonstrated the feasibility of compromising FL models through backdoor attacks [9], revealing that malicious clients could embed covert functionalities within the global model. Similarly, Bhagoji et al. analyzed FL from an adversarial perspective, highlighting vulnerabilities to poisoning attacks and their impact on model performance and efficiency [30]. To address these security concerns while maintaining the efficiency of shared models, several defense mechanisms have been proposed. Robust aggregation methods, such as Byzantine-resilient algorithms introduced by Yin et al. [31], aim to mitigate the impact of malicious updates by using statistical techniques like the median or trimmed mean during aggregation. Anomaly detection approaches, as discussed by Samy and Girdzijauskas [32], identify and exclude suspicious client updates based on deviations from expected behavior, enhancing model reliability without significant computational overhead. Differential Privacy (DP) mechanisms [33] enhance security by adding controlled noise to client updates, preserving individual data privacy while contributing to the overall efficiency of the FL system.

These protective strategies, however, typically require implementation from the outset of the FL process to ensure the integrity of the global model. In contrast, our proposed framework effectively detects and mitigates the adverse effects posed by malicious clients on the global model without the need for initial defensive measures or complete retraining. Unlike traditional protective strategies that often fail to address issues in already compromised environments, our approach excels by restoring the integrity of the global model post-compromise.

III. Preliminaries

A. Federated Learning

Federated Learning constitutes a collaborative machine learning paradigm that involves a network of *N* clients, each striving to learn a shared predictive model while keeping their data localized. This section formalizes the FL setting, adopting the notations consistent with the ones used in our analytical proofs and algorithmic descriptions. For each client *i*, let $f_i(\mathbf{w})$ denote the loss function that measures the discrepancy between the predictions generated by the model with parameters $\mathbf{w}$ and the actual targets for the *i*-th data point within its dataset. The empirical risk function associated with client *n*, where $n \in \{1, 2, \ldots, N\}$, is defined as:

$$F_n(\mathbf{w}) = \frac{1}{D_n} \sum_{i \in \mathcal{D}_n} f_i(\mathbf{w}), \tag{1}$$

where $\mathcal{D}_n$ symbolizes the local dataset of client *n* with D_n indicating the number of data points in $\mathcal{D}_n$. Here, $\mathbf{w}$ signifies the vector of model parameters across the federated network.

The principal objective in FL is to ascertain the optimal set of model parameters $\mathbf{w}^*$ that minimizes the aggregate empirical risk across all participating clients, formalized as:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{n=1}^{N} F_n(\mathbf{w}), \tag{2}$$

B. FedAvg Algorithm for Model Integration

FedAvg, or federated averaging, [14] is a cornerstone algorithm for model integration in FL, characterized by its simplicity and effectiveness. Assuming non-overlapping client datasets, i.e., $\mathcal{D}n \cap \mathcal{D}n' = \emptyset$ for any $n \neq n'$, the FedAvg procedure can be succinctly described by:

$$\mathbf{w}^{(t+1)} = \frac{1}{\sum_{k=1}^{N} D_k} \sum_{n=1}^{N} |D_n| \mathbf{w}_n^{(t)} \tag{3}$$

Here, $\mathbf{w}^{(t+1)}$ symbolizes the globally aggregated model parameters following the $(t+1)$-th federation cycle. Within each cycle, local parameters $\mathbf{w}_n^{(t)}$ are individually adjusted by clients based on their respective empirical risks $F_n(\mathbf{w})$. Subsequently, these parameters are amalgamated through the equation above, where a central server compiles and calculates the updated global parameters from all contributing clients.

These adjustments offer a more precise and conventional representation of the FL process and its key algorithms, suitable for scholarly communication.

C. Machine Unlearning

Let $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ denote the original training dataset, where x_i represents the *i*-th input sample, y_i is the corresponding target label, and *n* signifies the total number of samples in $\mathcal{D}$. The fundamental aim of machine learning is to construct a predictive model w^* characterized by a parameter set $w \in \mathcal{H}$, utilizing a designated learning algorithm. Here, $\mathcal{H}$ represents the hypothesis space, a conceptual domain wherein each hypothesis (or model configuration) corresponds to a specific set of parameters *w*.

The process is formalized through the application of a specialized unlearning algorithm $U(\cdot)$, which adjusts the model's parameters from *w*, thereby deriving a revised model *w** that more accurately aligns with the revised dataset or adheres to new data governance policies. The ultimate objective of machine unlearning is to ensure that w_U upholds or enhances certain desirable attributes of the data representation [15].

D. Federated Unlearning

In the context of FL, empowering clients with the ability to request data erasure presents a pivotal aspect of data privacy and control. Specifically, clients may petition the aggregation server to expunge certain data samples, thereby nullifying their influence within the model's training process. We introduce S_c to represent the set of deleted data samples $(\hat{x}_{i,c}, \hat{y}_{i,c})$ for a given client *c*, with $n_{s,c}$ denoting the cardinality of S_c, where

TABLE I
MAIN NOTATIONS

Notation	Definition
i, N	Client index, Client set
t, T	Federation round, Total federation rounds
$\mathcal{W}$, w	Set of Model parameters, Model parameter
w_i, w_I	Client i's model, Initial model
w_U, w_G	Unlearned model, Global model
ECE, τ	Expected Calibration Error, ECE threshold
η	Learning rate
∇L	Gradient of the loss function
$D_{\mathrm{U}}, D_{\mathrm{V}}$	Unlearning dataset, Validation dataset
E_{U}	Unlearning rounds
P	Predicted probabilities
U, C_U	Unlearning algorithm, Unlearning client

$n_s \ll n$. Assuming a subset of clients, $C_{\mathrm{U}} = \{C_j | j \in \Gamma\}$, have opted for data unlearning, the residual training dataset is represented as:

$$D_R = D \setminus S = \bigcup_{c \in C} (D_c \setminus S_c). \tag{4}$$

The objective of federated unlearning (FU) is to excise the data contributions of target clients, thereby deriving an unlearned global model w_U that approximates the model w^*, which is trained on D_R. This can be formulated as:

$$H(w_*(D_R)) \approx H_U(w_U(D)), \tag{5}$$

where $H(\cdot)$ signifies the model distribution function. Successful unlearning is achieved when the distributions of the unlearned and the original models, post-data erasure, closely align [34].

Conversely, client-level FU entails the complete removal of a target client's dataset from the global model knowledge base. This scenario is akin to the target clients opting out of the FL process entirely, described mathematically as:

$$H(w_*(D \setminus D_c)) \approx H_U(w_U(D)), \quad \text{where } c \in C_{\mathrm{U}}. \tag{6}$$

This notation and formulation articulate the processes and objectives of data erasure within the FL framework, highlighting the nuanced approach required to balance individual privacy rights with collective learning goals.

IV. SYSTEM ARCHITECTURE

A. Components

This section discusses the architecture and constituent components of the proposed FL management framework. The framework is primarily composed of clients, the FL management system, and the proposed Unlearning Service.

1) Clients: Referring to Figure 3, the framework integrates IoT networks, herein termed as clients. Each client is responsible for training a local model, denoted as $\{w_1, w_2, w_3, \ldots\}$, utilizing its local dataset. After the training phase, clients transmit their model updates to the central server to facilitate the synthesis of a global model (w_G).

- *Task processor:* A critical component of our framework is the Task Processor, embedded within each client's FL application. This component serves as the runtime environment for the client-level unlearning process, autonomously managing and executing tasks dispatched by the UaaS-SFL service via the FLMS. When the FLMS detects malicious activity through the Model Detector, it invokes the unlearning service. The unlearning tasks are then communicated to the affected clients via the Task Processor. Since the Task Processor is an integral part of the client's FL application and is responsible for handling all FL-related tasks, it automatically initiates the unlearning procedure upon receiving instructions from the UaaS-SFL service, without requiring explicit consent from the client. This automated approach not only streamlines the unlearning process but also upholds the principles of FL by preserving data privacy and minimizing server-side computation. By handling unlearning on the client side, the Task Processor ensures efficient removal of malicious contributions while maintaining the integrity and robustness of the FL system. Upon completion of the client-level unlearning process, it produces an unlearned model, which is subsequently shared with the FLMS for integration and further use.

2) FL Management System: FL management system is a platform designed to orchestrate, manage, and monitor FL processes across devices or nodes participating in an IoT network. Figure 3 encapsulates the primary functions attributed to the FL management system within our framework:

- *Model Detector:* This component analyzes the updates received from clients to identify any malicious contributions, ensuring the security and integrity of the FL system. The Model Detector employs an extensible algorithm framework that can be adapted based on the specific requirements of the FL management system. In our study, we utilized anomaly detection techniques to pinpoint irregularities in client updates that may indicate malicious behavior. Additionally, this component acts as a service manager by invoking the unlearning service when necessary, sharing the relevant model parameters. It functions as middleware between the unlearning service and clients, facilitating communication and operations within the FL system.
- *Global Model Generator:* In this component, benign model updates, alongside those subjected to the unlearning process, are aggregated to construct the global model for the ensuing federation round. The aggregation leverages the Federated Averaging (FedAvg) algorithm, optimizing the global model based on contributions from participating clients.

3) Unlearning Service (UaaS-SFL): The primary functions of the unlearning service are to initiate the unlearning process by generating an initial model using the provided model parameters and then to identify the relevant parameters and tasks required for execution within the unlearning client. Subsequently, it triggers client-level unlearning through the FL management system.

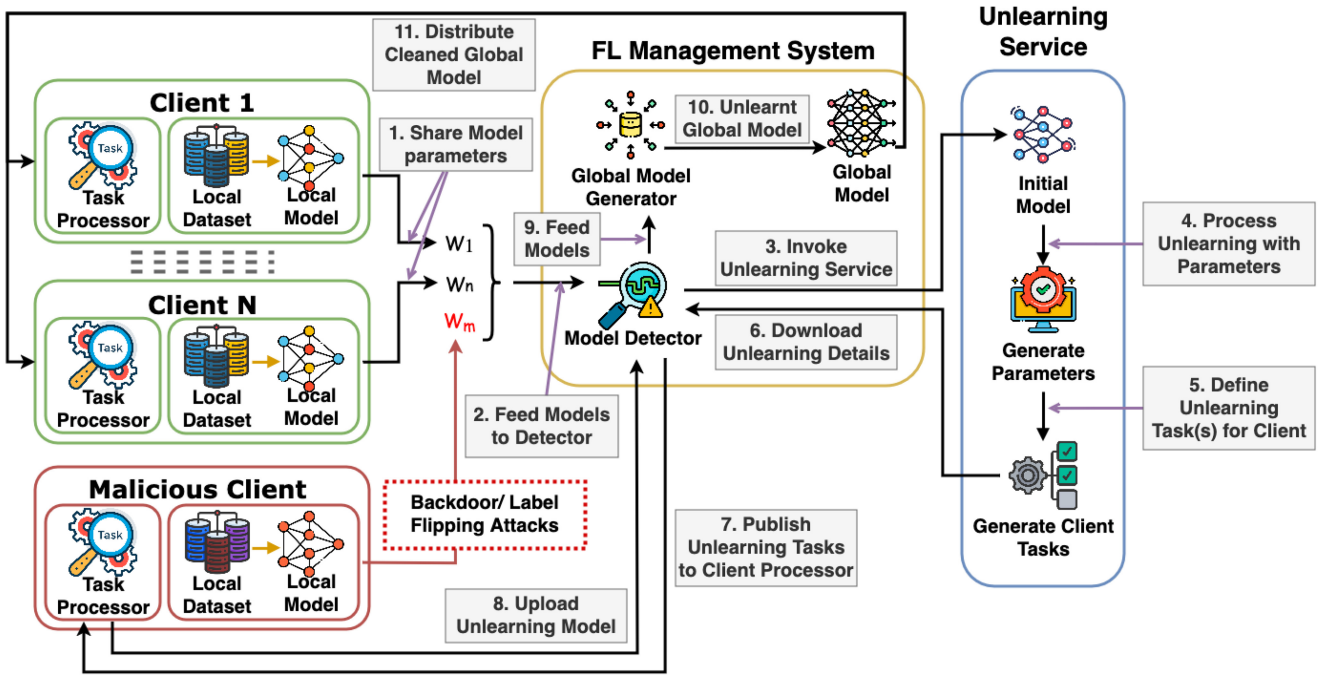

Fig. 3. The overview of the integration of Unlearning Service (UaaS-SFL) within a FLMS. UaaS-SFL manages unlearning tasks across devices, safeguards system security by mitigating poisoning attacks, and manages client contributions to ensure global model integrity for secure FL management.

B. System Overview

As depicted in Figure 3, the FL framework, integrated with the proposed UaaS-SFL service, consists of several systematic steps.

1) *Client-Side Training and Update Submission:*
   - Clients retrieve the global model from the Federated Learning Management System (FLMS).
   - They perform local training on their datasets, adapting the global model to their specific data characteristics.
   - After local training, clients send their updated models back to the central server for aggregation.
2) *Detection of Malicious Updates:*
   - The FLMS analyzes the received client updates using an extensible detection mechanism to identify malicious contributions.
   - Updates identified as detrimental are labeled as *poisonous clients*.
3) *Invocation of the Unlearning Service:*
   - Upon detecting malicious updates, the FLMS invokes the UaaS-SFL.
   - The service is provided with the necessary model parameters to initiate the unlearning process.
4) *Client-Side Unlearning Execution:*
   - The UaaS-SFL service coordinates with clients to execute unlearning tasks locally.
   - Clients perform the unlearning process by reversing the effects of their data contributions, using gradient ascent guided by parameters set by the UaaS-SFL service.
   - Upon completion, FLMS receives the unlearned models.
5) *Aggregation and Refinement of the Global Model:*
   - The FLMS aggregates the unlearned models with those from benign clients.
   - This step refines the global model, ensuring its accuracy and resilience against the influence of malicious data.
6) *Redistribution of the Purified Global Model:*
   - The updated and purified global model is redistributed to all benign clients.
   - Clients continue with the next round of training using the refined global model.

This iterative process underscores the integration of the UaaS-SFL manages robustness, accuracy, and security against potential threats within the network.

V. Proposed Framework

The proposed service introduces a novel architecture that integrates federated learning with UaaS-SFL Service, designed to enhance the security of network management systems. The algorithm, Federated Learning with Dynamic UaaS-SFL Service Invocation streamlines the global learning process by incorporating unlearning requests via Uaas-SFL service, enabling the system to dynamically adjust to evolving security requirements.

At the core of Uaas-SFL service are three pivotal algorithms: i) Initial Model Construction via Median Aggregation lays the foundation for a robust baseline model. This algorithm aggregates client models using median values to mitigate the impact of outliers, establishing a stable and reliable starting point for federated unlearning. ii) Client-level Unlearning via Gradient Ascent empowers the framework to precisely remove the influence of malicious clients iii) Early Stopping

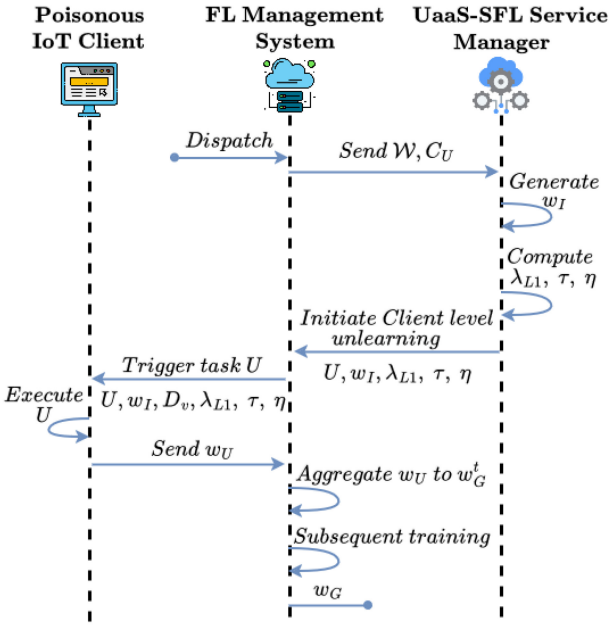

Fig. 4. This diagram illustrates the workflow by which a federated learning management system invokes the UaaS-SFL service to remove a poisonous IoT client from the global model within an IoT network. This process manages both the security and integrity of the federated learning system, effectively safeguarding it against the detrimental impacts of poisonous IoT clients.

Criterion Based on Expected Calibration Error (ECE) acts as a safeguard against over-unlearning. It maintains the model's calibration, ensuring that the unlearning and learning processes do not compromise the model's predictive performance. The following sections discuss these algorithms.

A. Federated Learning With UaaS-SFL Service Invocation

In Algorithm 1, we enhance the conventional Federated Averaging (FedAvg) methodology by integrating the UaaS-SFL service. This modified approach facilitates the targeted unlearning of data from particular clients within the FL cycle, effectively responding to events like client-based attacks. By incorporating the UaaS-SFL service, the system can detect and remove the influence of malicious clients who may attempt to poison the model or inject backdoors. Specifically, the UaaS-SFL service works in conjunction with the Model Detector component to identify anomalous client updates. When a malicious client is detected, the unlearning service is invoked to reverse the effects of the malicious updates, effectively mitigating threats in real time. This integration safeguards against potential threats by unlearning the malicious contributions from the global model, thereby preserving the model's integrity and ensuring that the learning process remains robust and reliable. By proactively addressing security risks through dynamic unlearning, our approach enhances the resilience of the FL system against adversarial attacks.

The algorithm begins with an initial global model, which is updated through standard FedAvg by aggregating client-specific model updates (line 7). The UaaS-SFL service is invoked conditionally, based on a predetermined trigger condition (lines 8–10). This conditional invocation ensures that the unlearning service is seamlessly integrated into the federated learning process, allowing for the dynamic adjustment of the global model in response to emerging requirements or challenges.

Algorithm 1 Federated Averaging With UaaS-SFL Service Invocation

Require: Initial global model w_G, Number of clients N
Ensure: Updated global model w_G
1: **while** not converged **do**
2: **for** each client i in $\{1, 2, \ldots, N\}$ **do**
3: $w_i = \text{ClientUpdate}(i, w_G)$ ▷ Local update on client i
4: **end for**
5: $w_G = \frac{1}{N} \sum_{i=1}^{N} w_i$ ▷ Aggregate updates to form new global model
6: $C_{\text{malicious}} = \text{DetectMaliciousClients}(\{w_i\})$ ▷ Identify malicious clients
7: **if** $C_{\text{malicious}} \neq \emptyset$ **then**
8: Invoke UaaS-SFL service to unlearn contributions from $C_{\text{malicious}}$
9: **for** each client j in $C_{\text{malicious}}$ **do**
10: $w_j = \text{Unlearn}(w_j, w_G)$ ▷ Client j performs unlearning locally
11: **end for**
12: $w_G = \frac{1}{N} \sum_{i=1}^{N} w_i$ ▷ Re-aggregate to update global model
13: **end if**
14: **end while**
15: **return** w_G

The integration of UaaS-SFL into the federated averaging process highlights the flexibility and adaptability of the proposed service. By enabling selective unlearning within the broader context of federated learning, we address key challenges related to security and model robustness, ensuring that the federated model remains responsive to the needs and constraints of individual clients.

B. Initial Model Construction via Median Aggregation

The establishment of an Initial Model, denoted as w_{I}, plays a pivotal role in the unlearning process within a federated learning management system. This model is constructed through the median aggregation of parameters from client models, excluding those from the client requesting unlearning, denoted as w_{U}. Utilizing median aggregation ensures that w_{I} is robust and representative of the collective knowledge across the federation, effectively mitigating the influence of outliers. Consequently, w_{I} improves the effectiveness of subsequent unlearning processes.

Given a federation of client models $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$, excluding the model associated with the unlearning request w_{U}, the Initial Model w_{I} is formulated by aggregating the parameters of the remaining models in $\mathcal{W}$ using a median operation. The aggregation process for each parameter p across the model parameters is formally defined as follows:

$$w_{\text{I}} = \text{median}(\{w_i[k] | w_i \in \mathcal{W} \setminus \{w_{\text{U}}\}\}) \quad \forall k, \tag{7}$$

where k indexes over all parameters within the model, and $w_i[k]$ denotes the k-th parameter of the i-th client model within the set $\mathcal{W}$.

Algorithm 2 Construction of an Initial Model Using Median Aggregation for Federated Unlearning

Require: $\{w_i\}_{i=1}^{N}$ excluding w_{U}
Ensure: Initial Model w_I
$ModelParametersList = [\,]$
for each parameter p in w_i **do**
  for each client model w_i in $\{w_1, w_2, \ldots, w_N\}$ **do**
    $ModelParametersList \leftarrow w_i[p]$
  end for
  Sort $ModelParametersList[p]$
  $w_I[p] \leftarrow Median(ModelParametersList[p])$
end for
return w_I

The Algorithm 2 commences with the collection of model parameters from all client models in the federation $\{w_1, w_2, \ldots, w_N\}$, deliberately omitting the parameters from the unlearning client w_U. For each parameter across these models, a median value is computed to serve as a representative parameter value for the Initial Model.

C. Unlearning via Gradient Ascent

Algorithm 3 introduces an unlearning strategy within a federated learning framework, where the unlearning operation is executed locally at the client designated for unlearning first. This approach is underpinned by the deployment of an Initial Model (w_{I}). A key component of this strategy is the application of regularization, which enforces sparsity in the model parameters, thereby aiding in the unlearning process by reducing the potential for overfitting during the unlearning process [35].

Uniquely, this methodology employs gradient ascent locally at the unlearning client to maximize the loss for the data targeted for unlearning. This deliberate maximization of loss facilitates the effective removal of the client's data influence, contrasting with conventional gradient descent approaches that aim to minimize loss. The mathematical formulation for performing gradient ascent with regularization at the client level is presented as follows:

The client aims to solve the following optimization problem to perform unlearning:

$$w_U \leftarrow \max_{w \in \mathbb{R}^d} \; F_i(w) + \lambda_{L1} \cdot ||w||_1 \quad \text{subject to} \;\; \mathrm{ECE}(w) \leq \tau, \tag{8}$$

where:

- $F_i(w)$ is the loss function for the data targeted for unlearning.
- $\lambda_{L1} \cdot ||w||_1$ represents the L1 regularization term, with λ_{L1} as the regularization coefficient, promoting sparsity in the model parameters.
- ECE(w) measures the Expected Calibration Error of the model, ensuring the model's predictions remain well-calibrated.
- τ is the threshold for ECE, serving as an early stopping criterion to prevent over-unlearning by maintaining acceptable calibration levels.

Algorithm 3 Client Level Unlearning via Gradient Ascent

Require: $\{w_i\}_{i=1}^{N}$ excluding w_{U}, dataset $\mathcal{D}_{\mathrm{U}}$, learning rate η, regularization strength λ_{L1}, ECE threshold τ
Ensure: Updated model W'_{U} after unlearning
$w_{\mathrm{I}} \leftarrow \mathrm{ComputeMedianModel}(\{w_{\mathrm{i}}\}_{i \neq \mathrm{U}})$ ▷ Initial model excluding w_{U}
Initialize $w \leftarrow w_{\mathrm{I}}$
for $e = 1$ to E_{U} **do** ▷ Unlearning epochs
  for each batch $(X_{\mathrm{b}}, Y_{\mathrm{b}}) \in \mathcal{D}_{\mathrm{U}}$ **do**
    Compute loss L for unlearning data and model 2
    $L \leftarrow -L + \lambda_{L1} \cdot ||w||_1$
    $w \leftarrow w + \eta \nabla \mathrm{L}$
    $ECE \leftarrow \mathrm{CalculateECE}(w, \mathcal{D}_{\mathrm{V}})$
    if $\mathrm{EngageEarlyStoping}(w, \tau, \mathcal{D}_{\mathrm{V}})$ **then**
      break ▷ Stop if model calibration degrades
    end if
  end for
  if Early stopping condition is met **then**
    break
  end if
end for
$w_{\mathrm{U}} \leftarrow w$ ▷ Final unlearned model
return w_{U}

This optimization approach is designed to selectively forget information from the model by maximizing the specified loss, with the inclusion of L1 regularization to encourage parameter sparsity and enhance model interpretability. The constraint on ECE ensures that, throughout the unlearning process, the model's calibration is preserved, preventing the degradation of prediction reliability.

D. Early Stopping Criterion Based on Expected Calibration Error

The Expected Calibration Error (ECE) serves as a pivotal metric for assessing the calibration of probabilistic models. It quantifies the discrepancy between the model's predicted probabilities and the actual correctness of those predictions across all classes. Formally, ECE [29] is defined as:

$$\mathrm{ECE} = \sum_{m=1}^{N} \frac{|B_m|}{N} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|, \tag{9}$$

where N is the total number of samples, M is the number of bins, B_m is the set of indices of samples whose predicted confidence falls into bin m, $\mathrm{acc}(B_m)$ is the accuracy of predictions in bin m, and $\mathrm{conf}(B_m)$ is the average confidence of predictions in bin m.

The early stopping mechanism is activated when the ECE of the model on a validation set exceeds a predefined threshold τ, indicating a significant divergence between the model's confidence and its empirical accuracy. This approach ensures the model remains well-calibrated throughout the learning or unlearning process, preventing overfitting or over-unlearning that could compromise the model's reliability.

Algorithm 4 implements a vital mechanism to prevent over-unlearning by employing an early-stopping criterion based on

Algorithm 4 Early Stopping Using ECE

Require: w_{U}, $\mathcal{D}_{\mathrm{V}}$, τ

Ensure: The decision to halt unlearning early if ECE exceeds the threshold

```
Initialize ECE ← 0
for each (X_b, Y_b) ∈ D_V do
    P ← Softmax(Model(X_b, M))
    ECE ← Calculate ECE for (P, Y_b)
end for
ECE ← ECE/|D_V|
if ECE > τ then
    return True                    ▷ Engage early stopping
else
    return False
end if
```

ECE. This error measures the discrepancy between the model's predicted probabilities and the actual outcomes, serving as an indicator of the unlearning process's effect on the model's predictive reliability.

The method calculates ECE between the current unlearning model's softmax outputs and the actual outcomes across a validation dataset ($\mathcal{D}_{\mathrm{V}}$). Early stopping is activated if ECE exceeds a predefined threshold, thereby preserving the model's calibration and reliability.

After all, to enhance performance and weight balance after integrating the locally unlearned model w_U, with the global model w_{G}, it is necessary to conduct a few rounds of aggregation exclusively with benign clients.

VI. Theoretical Analysis

In this subsection, we revisit the convergence analysis of the federated learning process, with a particular emphasis on the proposed unlearning service. Given that the unlearning algorithm necessitates the removal of certain weights from the global model, this can potentially impact the model's convergence. Therefore, it is essential to analyze the mathematical implications of the proposed approach to global model convergence. In this section, we employ established theoretical frameworks to evaluate the applicability of our proposed solution.

Definition 1 (Triangle Inequality): $\|x+y\| \leq \|x\| + \|y\|$, $\forall x, y \in \mathbb{R}^d$.

Definition 2 (Linearity of Expectation): The expected value of the sum of random variables is equal to the sum of their individual expected values, regardless of whether they are independent.

Definition 3 (β-Smoothness [27]): A function $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth if it is differentiable and its gradient function $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz continuous with constant β, such that $\forall w_i, w_j \in \mathbb{R}^d$:

$$\|\nabla f(w_i) - \nabla f(w_j)\| \leq \beta \|w_i - w_j\| \tag{10}$$

Definition 4 (Smoothness of Function F [27]): If each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is β-smooth, the function $F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$ is also β-smooth, such that $\forall w_i, w_j \in \mathbb{R}^d$:

$$\|\nabla F(w_i) - \nabla F(w_j)\| \leq \beta \|w_i - w_j\| \tag{11}$$

Definition 5 (The Norm of Stochastic Gradients [27]): The expected squared norm of stochastic gradients is uniformly bounded, $\forall i, i \in [N]$ and $\forall t, t \in [T]$, there exists a $G > 0$ such that:

$$\mathbb{E}_{d^t} \|\nabla f_i(w^t)\|^2 \leq G^2 \tag{12}$$

where [N] is the client set and [T] is the total iteration set. Note that d^t represents the sampling distribution over local data on local clients.

Lemma 1: For a given federation round t, when the model update step is followed by the formula $w^t = w^{t-1} - \eta_t \Theta(w^t, d^t)$, the norm of the stochastic gradients of $F(w^t)$ satisfies the following inequality:

$$\mathbb{E}_{d^t}(F(w^t)) \leq \mathbb{E}_{d^t}\left(F(w^{t-1})\right) - \eta_t \|\nabla F\left(w^{t-1}\right)\|^2 + \frac{1}{2}\eta_t^2 \beta G^2 \tag{13}$$

Note that $\Theta(w^t, d^t)$ denotes the gradients of the function $F(w^t)$ with data sample distribution of d^t. Above Lemma 1 can be rearrange as follows according to [27]:

$$\eta_t \|\nabla F\left(w^{t-1}\right)\|^2 \leq \mathbb{E}_{d^t}\left(F(w^{t-1})\right) - \mathbb{E}_{d^t}(F(w^t)) + \frac{1}{2}\eta_t^2 \beta G^2 \tag{14}$$

Equation (14) can be updated for all the iteration rounds t, such that $t \in [T]$ where $T = \{0, 1, 2, 3, \ldots, t\}$. By summing all the inequalities over all t iterations, a summed inequality from the Equation (14) can be derived as follows:

$$\begin{aligned} &\sum_{t=0}^{t} \eta_t \|\nabla F(w^t)\|^2 \\ &\leq \mathbb{E}_d(F(w_I)) - \mathbb{E}_d(F(w^t)) + \sum_{t=0}^{t} \frac{1}{2}\eta_t^2 \beta G^2 \\ &\leq F(w_0) - F(w^*) + \frac{1}{2}\sum_{t=0}^{t} \eta_t^2 \beta G^2. \end{aligned} \tag{15}$$

where w_I is the model on the initial round and w^* represents the optimal model that satisfies the condition on the function F best after $t+1$ iterations.

Let w_G^t be the global model in the federation network before the unlearning process in the federation round t. Similarly, $\bar{w}_G^t$ represents the global model after the unlearning process. Then the global model generation for each case are:

$$w_G^t = w_G^{t-1} - \eta \cdot \frac{1}{n} \sum^{i \in N} \nabla f_i\left(w^{t-1}\right) \tag{16}$$

$$\bar{w}_G^t = \bar{w}_G^{t-1} - \eta \cdot \left[\frac{1}{n-1} \sum^{i \in N-\{u\}} \nabla f_i\left(w^{t-1}\right) + w_U\right] \tag{17}$$

To determine the convergence of the proposed approach, we considered the gradient difference of the above two global

models. To theoretically evaluate the condition on convergence, the gradient difference should have an upper bound.

Theorem 1 (Convergence of the Federated Unlearning): Let loss function F is β-smooth and $\eta < \frac{1}{t\beta}$ such that $\lim_{t\to\infty}\eta = 0$. Then the expected distance between w_G^t and $\bar{w}_G^t$ is upper-bounded by the following inequality for all t values, such that $\forall t, t \in [T]$:

$$\mathbb{E}_{\xi_t}\left\|w_G^t - \bar{w}_G^t\right\| \leq \sqrt{\sum_{t=0}^{t-1}\eta_t\left[F(w_I)) - F(w^*) + \frac{\beta G^2}{2}\sum_{t=0}^{t-1}\eta_t^2\right]} + \delta^t \quad (18)$$

Proof: Let $\mathbb{E}_d(\|w_G^t - \bar{w}_G^t\|)$ denote the norm of stochastic gradient difference of the two main global models. Applying linearity of expectation and triangle inequality together, we can obtain:

$$\begin{aligned}&\mathbb{E}_d\left(\left\|w_G^t + (-\bar{w}_G^t)\right\|\right) \to \mathbb{E}_d\left(\left\|w_G^t - w_I\right\|\right) + \mathbb{E}_d\left(\left\|\bar{w}_G^t - w_I\right\|\right)\\ &\to \left\|\sum_{t=0}^{t}\eta_t\mathbb{E}_{d^t}\left(\Theta(w^t, d^t)\right)\right\| + \delta^t\\ &\leq \sum_{t=0}^{t}\eta_t\left\|\nabla F\left(w^t\right)\right\| + \delta^t \end{aligned} \quad (19)$$

Applying squared and square root to the right-hand side of the inequality to obtain the Lemma 1.

$$\begin{aligned}&\mathbb{E}_d\left(\left\|w_G^t - \bar{w}_G^t\right\|\right)\\ &\leq \sqrt{\left(\sum_{t=0}^{t}\eta_t\right)\left(\sum_{t=0}^{t}\eta_t\|\nabla F(w^t)\|^2\right)} + \delta^t\\ &\leq \sqrt{\sum_{t=0}^{t}\eta_t\left[F(w_I) - F(w^*) + \frac{\beta G^2}{2}\sum_{t=0}^{t}\eta_t^2\right]} + \delta^t \end{aligned} \quad (20)$$

where δ^t is the global model's training trajectory constant. When $\eta \leq \frac{1}{\beta}$(fixed learning rate), the equation changes to:

$$\left\|w^t - \bar{w}^t\right\| \leq \sqrt{\eta(t+1)[F(w_I) - F(w^*)]} + \delta^t. \quad (21)$$

The biasness of the gradient estimations on the upper bound should be ignored in the fixed step size learning approach and $\sum_{t=0}^{t} = \eta(t+1)$. The bounded value indicates that after the unlearning process, the global model has a different gradient for the global model from the unlearning client model. This implies that after the unlearning process, the global model does not influence the unlearnt model. Moreover, assuming the w_G^t on the t iteration round is already converged, the $\bar{w}^t$ model is converged with upper bounded distance after the unlearning process is completed. Hence the theorem is proven. ■

VII. Results and Discussion

In this section, we provide discuss about the testing environment, experimental setup, and results. We use the MNIST [36], CIFAR-10 [37] and Fashion-MNIST [38] datasets, and perform three image classification tasks to evaluate the performance of our algorithm.

A. Datasets

- *MNIST:* This dataset comprises 60,000 training images and 10,000 testing images, each a 28x28 grayscale representation of digits from 0 to 9. Pixel values range from 0 (white) to 255 (black). For the MNIST dataset, we employ the LeNet-5 model.
- *Fashion-MNIST:* Fashion-MNIST features Zalando's article images, with a training set of 60,000 examples and a test set of 10,000 examples. Similar to MNIST, each example is a 28x28 grayscale image, categorized into one of 10 classes. The LeNet-5 model is utilized for experiments with this dataset.
- *CIFAR-10:* This dataset includes 60,000 32x32 color images in 10 different classes, with each class represented by 6,000 images. The dataset is divided into a training set of 50,000 images and a test set of 10,000 images. For CIFAR-10, we use the VGG-11 model.

B. Machine Learning Models

The experiments leverage well-established models tailored to the characteristics of each dataset:

- For both MNIST and Fashion-MNIST datasets, the **LeNet-5** model [39] is chosen due to its proficiency in handling grayscale images and its historical significance in the field of deep learning for image classification.
- The CIFAR-10 dataset experiments are conducted using the **VGG-11** model [40], selected for its ability to effectively process color images and its deeper architecture, which is suitable for recognizing more complex patterns in the data.

C. Experiment Setup

In our study, we developed a comprehensive experimental setup to evaluate the practical applications and effectiveness of FL within a simulated environment that closely mirrors real-world IoT network scenarios. To simulate the unpredictable nature of data generation in IoT networks, data was randomly distributed among clients in a manner that ensured no two clients received identical datasets, accounting for variations in node activity and availability. This approach accurately reflects the inherent data heterogeneity and client variability seen in real-world FLMS.

We employed backdoor triggers as a metric to assess the efficacy of unlearning methods within FL frameworks. Specifically, we manipulated the dataset of a designated target client by embedding a 'pixel pattern' backdoor trigger, sized 3×3 pixels, into a subset of the images. This manipulation was facilitated using the Adversarial Robustness Toolbox [41], a reputable toolkit for enhancing the security of machine learning models against adversarial threats. The insertion of these triggers rendered the global FL model vulnerable to the backdoor, thereby simulating a real-world adversarial scenario.

Our experiment employs an approach that combines anomaly detection techniques with targeted backdoor detection strategies to identify malicious client updates. This approach includes statistical analysis to monitor unexpected deviations in model performance and consistency checks to identify

anomalies in model predictions on specific inputs, such as those containing backdoor triggers. The detection mechanism can effectively identify and flag client updates introducing such vulnerabilities by analyzing the model's behavior on these manipulated inputs. This comprehensive approach enables our experiment to reliably detect and isolate harmful clients.

To further demonstrate the robustness of our unlearning approach, we integrated UaaS-SFL service at three different stages within the experimental setup: (1) at the beginning of the training process, (2) midway through the training, and (3) in a scenario where malicious data had significantly compromised the global model. This staged integration allowed us to assess our solution's effectiveness in preemptive and post-compromise scenarios, showcasing its capability to remove malicious contributions, thereby restoring the integrity of the global model across different stages of the FL lifecycle.

D. Evaluation and Comparison Metrics

An effective unlearning service is expected to compel the model to remove the information learned from the data designated for unlearning while preserving its proficiency on the remaining dataset. To quantitatively assess the performance of our proposed service, we employ the following widely accepted metrics:

Backdoor Accuracy: This metric measures the model's accuracy on backdoored data, defined as the proportion of instances that are incorrectly classified as a specific target label manipulated by an adversary. Evaluating backdoor accuracy is crucial because it indicates how effectively the model has unlearned the adversarial influence embedded within the target client's data. A lower backdoor accuracy after unlearning signifies a higher degree of successful unlearning, reflecting the model's reduced susceptibility to backdoor attacks.

Accuracy: In addition to backdoor accuracy, we assess the overall accuracy of both the global model and the model after unlearning. This metric ensures that the process of unlearning a particular client's data does not detrimentally impact the comprehensive performance of the system. Maintaining or improving overall accuracy post-unlearning indicates that the algorithm can selectively eliminate harmful data influences without compromising the model's general learning capabilities.

By employing these metrics, we provide a holistic evaluation framework for our proposed unlearning service. This framework balances the need to mitigate specific adversarial impacts with the preservation of overall system performance and knowledge retention, demonstrating the effectiveness and reliability of our approach in practical federated learning scenarios.

E. Comparison Methods

To validate the effectiveness of our proposed service, we conducted comprehensive experiments comparing it against three established methods: Federated Averaging (FedAvg), Federated Retraining (Retrain), and FedEraser.

FedAvg: FedAvg serves as the foundational method for synthesizing a global model by aggregating locally updated models from all participating clients. This method is employed as a benchmark to demonstrate the impact of data removal, particularly backdoor data, utilizing our proposed algorithm.

Retrain: This approach employs a straightforward unlearning strategy wherein the global model is retrained from scratch, excluding the data from the target client. This approach, referred to as Retrain, serves as a baseline to assess the unlearning efficiency of our proposed method relative to conventional retraining practices [24]. Although retraining the global model often demonstrates superior performance, it requires a substantial number of aggregation rounds to reach the desired performance level of a fully trained FLMS. This process consumes significant time and computational resources compared to unlearning algorithms. Despite these drawbacks, retraining provides a robust baseline for evaluating the unlearning effectiveness of our algorithm, as it represents the ideal scenario where the target client's influence is entirely removed from the global model through complete retraining.

FedEraser [24]: Recognized for its efficient unlearning capabilities, FedEraser reconstructs the model by exploiting historical parameters to exclude the influence of specific data points or clients. This method provides a comparative framework to showcase the performance of our algorithm relative to existing unlearning algorithms within the federated learning paradigm.

FedRecovery [27]: FedRecovery is an unlearning method for FL that removes the influence of specific clients or data by reversing their contributions to the global model. It negates targeted clients' updates, effectively restoring the model to a state as if their data were never included. This approach eliminates compromised data without the need to retrain the entire model, providing a benchmark to evaluate the effectiveness of our proposed unlearning service.

F. Effectiveness Assessment of the Proposed Service

We conducted an extensive evaluation of backdoor accuracy across various models and datasets, carefully monitoring this metric's progression through successive rounds of the federated learning process. This analysis is crucial as it shows the dynamics of the federated learning model's ability, particularly with the FedAvg algorithm, to integrate data compromised with backdoor triggers by the target client. The introduction of our proposed service leads to a marked decrease in backdoor accuracy across all examined datasets, bringing the results in line with those obtained through the standard practice of complete retraining.

In Figure 5, we present the backdoor accuracy versus aggregation rounds for the MNIST dataset, demonstrating the effectiveness of our proposed service at various stages of the FLMS. Before invoking the unlearning service at Rounds 6, 16, and 25, clients with backdoor images had been aggregated into the global model, leading to elevated backdoor accuracy and indicating the presence of backdoor vulnerabilities. At these critical points, we connected the unlearning service, which detects and triggers the removal of harmful clients'

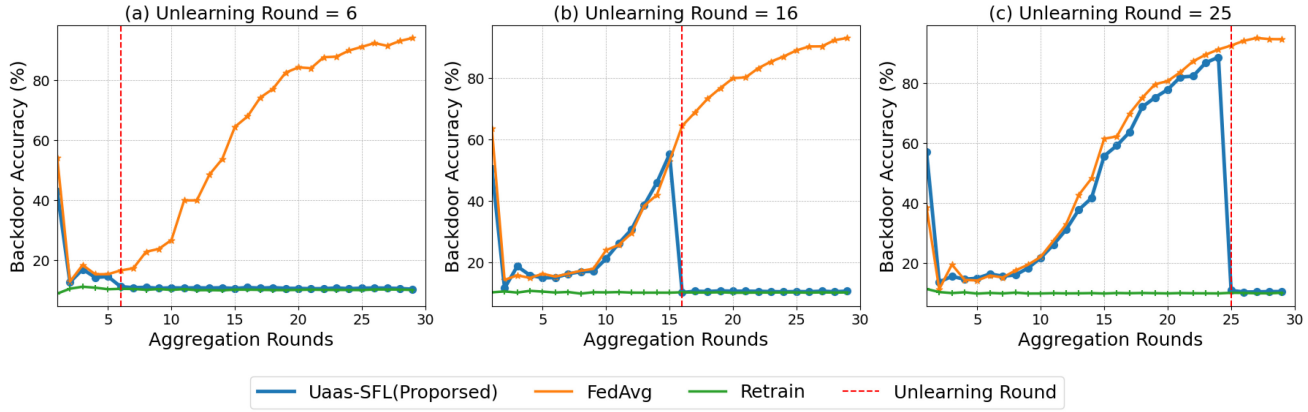

Fig. 5. MNIST Dataset: Backdoor Accuracy vs. Aggregation Rounds, demonstrating the effectiveness of the proposed service at various stages of the Federated Learning Management System (FLMS). Specifically, (a) Round 6 illustrates the service's effectiveness assessment in the early stage of FLMS, (b) Round 16 showcases its performance during the mid-training phase, and (c) Round 25 highlights its effectiveness in a largely trained model. The results indicate the service's capability to restore backdoor accuracy to baseline levels across different training stages.

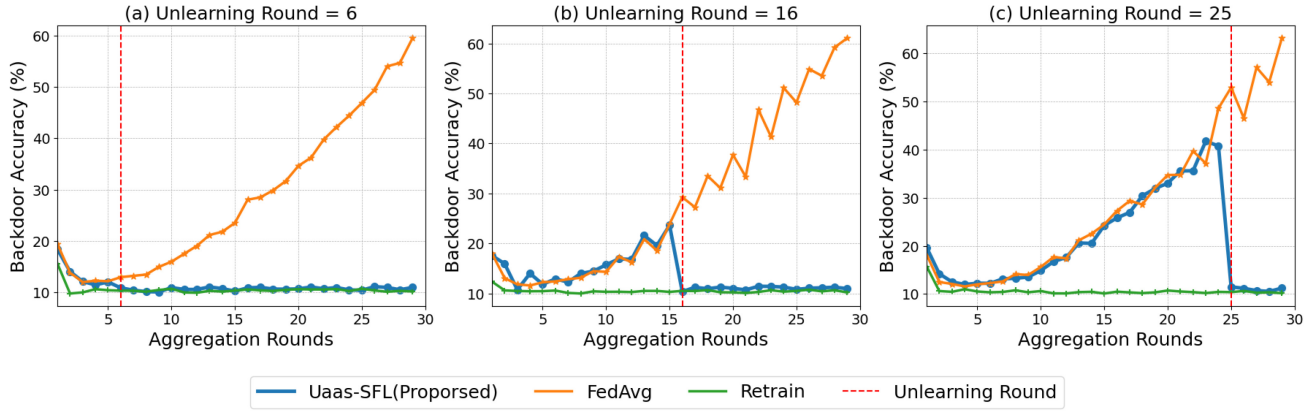

Fig. 6. Fashion-MNIST Dataset: Backdoor Accuracy vs. Aggregation Rounds, demonstrating the effectiveness of the proposed service at various stages of the Federated Learning Management System (FLMS). Specifically, (a) Round 6 illustrates the service's effectiveness assessment in the early stage of FLMS, (b) Round 16 showcases its performance during the mid-training phase, and (c) Round 25 highlights its effectiveness in a largely trained model. The results indicate the service's capability to restore backdoor accuracy to baseline levels across different training stages.

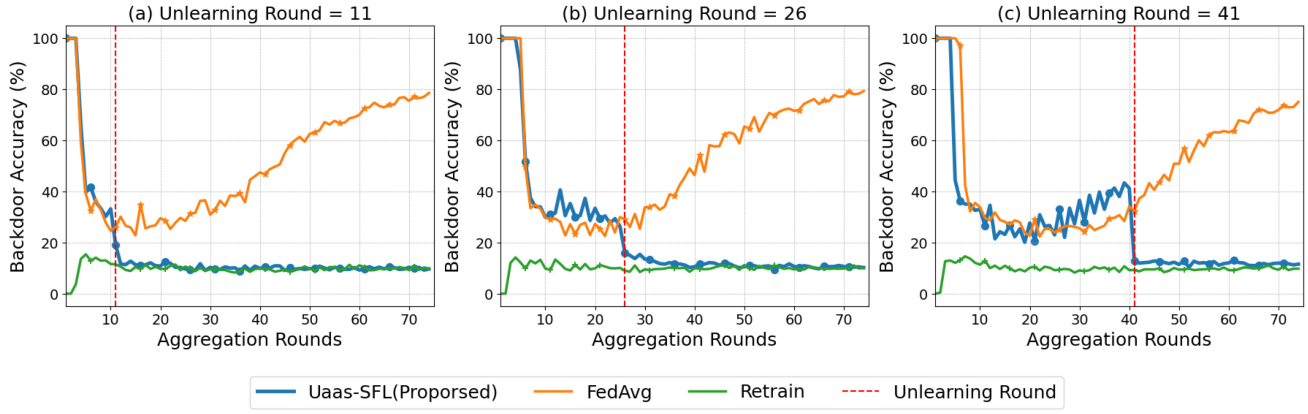

Fig. 7. CIFAR10 Dataset: Backdoor Accuracy vs. Aggregation Rounds, demonstrating the effectiveness of the proposed service at various stages of the Federated Learning Management System (FLMS). Specifically, (a) Round 11 illustrates the service's effectiveness assessment in the early stage of FLMS, (b) Round 26 showcases its performance during the mid-training phase, and (c) Round 41 highlights its effectiveness in a largely trained model. The results indicate the service's capability to restore backdoor accuracy to baseline levels across different training stages.

contributions. Specifically, at Round 6, representing the early stage of FLMS, the service swiftly reduces the backdoor accuracy to baseline levels comparable to those achieved through retraining, showcasing its ability to counteract backdoor effects before they become deeply embedded in the global model. At Round 16, during the mid-training phase, the service continues to effectively mitigate the backdoor, even as the model begins to solidify learning patterns from client updates. At Round

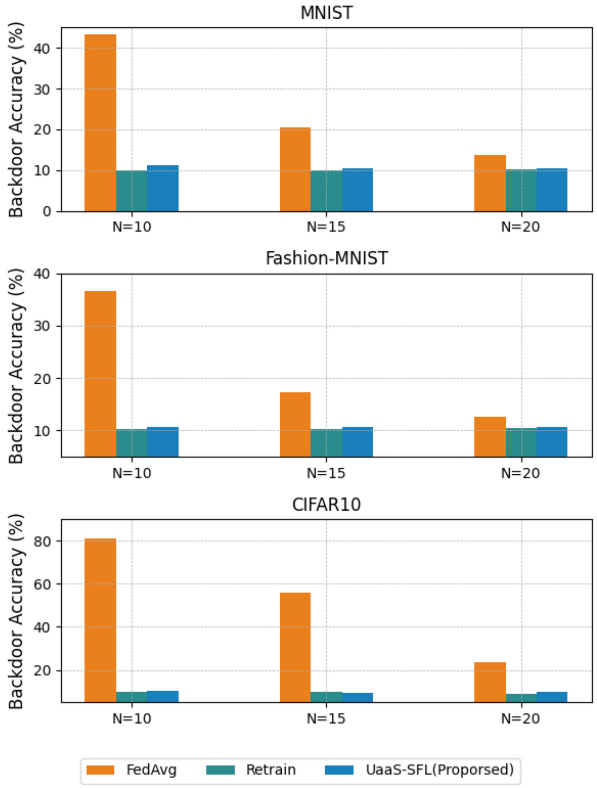

Fig. 8. A Comparison of UaaS-SFL service's Effectiveness Across Varied Client Populations (N) in MNIST, Fashion-MNIST and CIFAR10 Datasets.

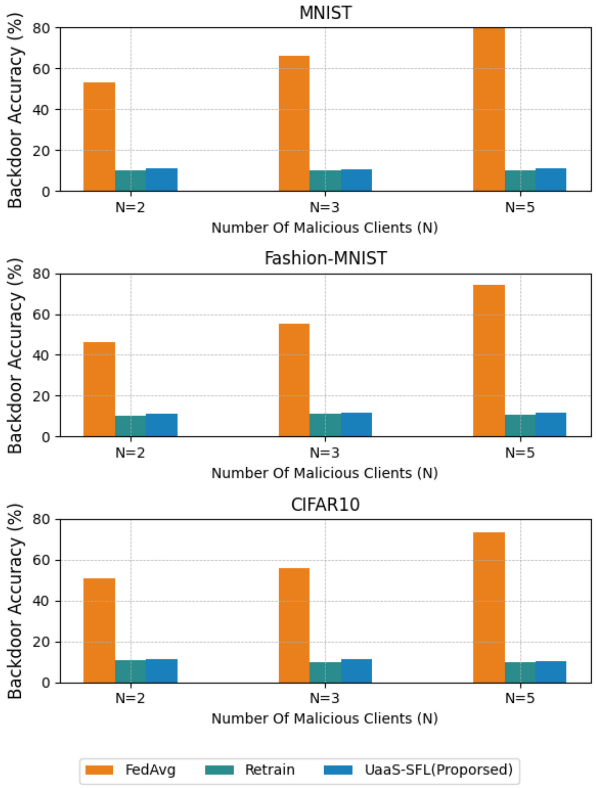

Fig. 9. Comparative Analysis of UaaS-SFL Service Effectiveness with Varying Numbers of Malicious Clients (N) across MNIST, Fashion-MNIST, and CIFAR-10 Datasets.

25, corresponding to a largely trained model, the service can unlearn backdoor effects even when the model has extensively integrated malicious client updates, restoring the backdoor accuracy to baseline levels. Similarly, Figures 6 and 7 exhibit comparable outcomes for the Fashion-MNIST and CIFAR-10 datasets, respectively, reaffirming the service's generalizability and adaptability to various data distributions and complexities. Overall, our proposed service proves efficient in counteracting the effects of manipulated data introduced by malicious clients within the federated learning framework, irrespective of the stage at which it is invoked. This underscores the service's flexibility and its capability to be integrated at any point within the federated learning system, showcasing its essential role in enhancing the system's resilience by effectively detecting and removing the poisoned clients' data.

In Figure 8, we demonstrate the independence of our proposed unlearning service from the number of clients within the federated learning ecosystem. For this analysis, we trained the global model for 20 rounds, with varying numbers of clients: N = 10, N = 15, and N = 20, for both the MNIST and Fashion-MNIST datasets and we trained CIFAR10 for 100 rounds with same client combinations. We observed that the backdoor accuracy of the global model, using the FedAvg strategy, decreases as the number of clients increases, given that only one client introduces a backdoor for our experimental purposes. The diagram clearly shows that our proposed service effectively reduces the backdoor accuracy, aligning it closely with that of the retraining strategy, which serves as our baseline. Systematically, our solution maintains robust performance, illustrating its effectiveness irrespective of the client count in the federated learning network.

In Figure 9, we illustrate that our proposed unlearning service effectively reduces backdoor accuracies to baseline levels regardless of the number of malicious clients within the federated learning ecosystem. For this analysis, we trained the global model for 20 rounds with 15 clients, varying the number of malicious clients (N = 2, N = 3, and N = 5) across the MNIST and Fashion-MNIST datasets, and trained CIFAR-10 for 100 rounds with the same client configurations. Our results demonstrate that while the backdoor accuracy of the global model using the FedAvg strategy decreases as the number of clients increases, our framework consistently restores backdoor accuracies to baseline levels irrespective of the number of malicious clients. This showcases the capability of our framework to effectively mitigate backdoor threats while maintaining robust model integrity across varied client populations.

G. *Efficiency Assessment Post-Execution*

We evaluated the overall model accuracy to demonstrate that our proposed unlearning service does not compromise performance post-unlearning and that the model converges in subsequent training rounds. During the unlearning round, we observe an initial accuracy drop of approximately 5%, as shown in Figures 10, 11, and 12 for the MNIST, Fashion-MNIST, and CIFAR-10 datasets, respectively. This temporary decrease results from the unlearning process increasing the empirical loss associated with malicious clients' data to

TABLE II
COMPARATIVE ANALYSIS OF THE PROPOSED FRAMEWORK (UAAS-SFL) AGAINST FEDRECOVERY AND FEDERASER ON MNIST AND FASHION-MNIST DATASETS. RETRAIN IS INCLUDED AS A BASELINE REFERENCE. BOLD VALUES INDICATE THE BEST PERFORMANCE AMONG PRACTICAL METHODS

Method	MNIST		Fashion-MNIST	
	Clean Accuracy (%)	Backdoor Accuracy (%)	Clean Accuracy (%)	Backdoor Accuracy (%)
FedRecovery [27]	94.78	11.20	77.43	10.93
FedEraser [24]	96.32	11.23	80.61	11.17
UaaS-SFL	**97.01**	**10.97**	**82.50**	**10.89**
Retrain (Baseline)	97.25	10.13	82.11	10.37

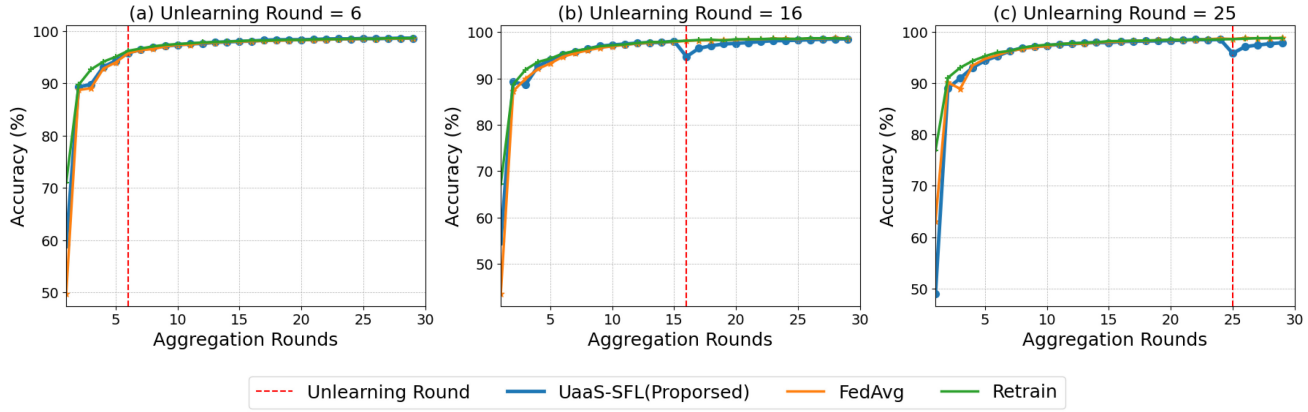

Fig. 10. MNIST Dataset: Accuracy vs. Aggregation Rounds. Illustrating the Efficiency of the Proposed Service at (a) Round 6, (b) Round 15, and (c) Round 24 in Converging the Global Model in Subsequent Training Rounds.

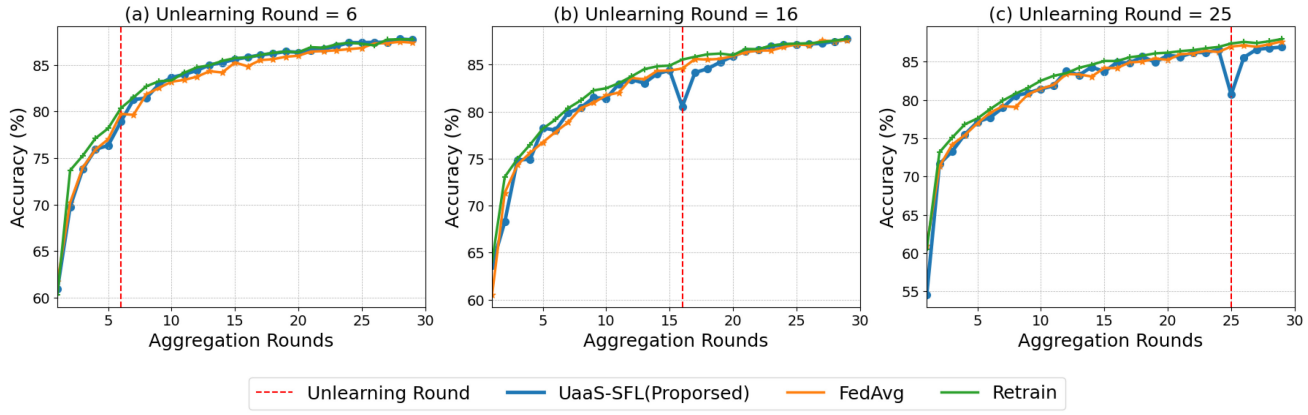

Fig. 11. Fashion-MNIST Dataset: Backdoor Accuracy vs. Aggregation Rounds. Illustrating the Efficiency of the Proposed Service at (a) Round 6, (b) Round 15, and (c) Round 24 in Converging the Global Model in Subsequent Training Rounds.

remove their influence from the global model. However, after aggregating the unlearned models with those from benign clients in subsequent rounds, the overall accuracy quickly recovers to baseline levels. Collaborative learning adjusts the model parameters back to optimal values, compensating for the initial drop. Importantly, the model consistently converges in subsequent rounds regardless of when the unlearning service is invoked, underscoring the robustness and effectiveness of our approach. This demonstrates that our service can be deployed at any stage to eliminate malicious client influence, preserving the model's overall performance and reliability despite temporary fluctuations in accuracy.

In Figure 13, we present a comparative analysis of the evaluation accuracy for local clients after the implementation of our proposed unlearning service. This evaluation encompasses a selection of random clients from both the MNIST and CIFAR-10 datasets, with our federated learning configuration comprising 10 clients for MNIST and 25 clients for CIFAR-10. As illustrated in Figure 13, across all scenarios, the post-unlearning performance of local clients consistently matches the benchmarks set by both Retrain strategies. This alignment underscores the effectiveness of our unlearning approach in maintaining the integrity of local client performance within the federated learning system.

H. Comparison With Existing Methods

We conducted a comparative analysis of our proposed service against FedRecovery, Retrain, and FedEraser [24], evaluating their impact on clean and backdoor accuracy

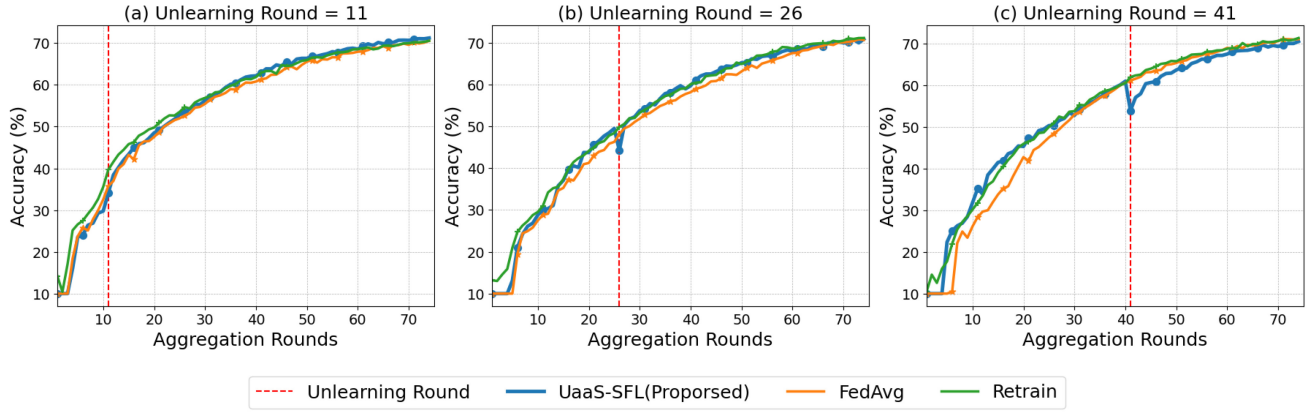

Fig. 12. CIFAR10 Dataset: Backdoor Accuracy vs. Aggregation Rounds. Illustrating the Efficiency of the Proposed Service at (a) Round 10, (b) Round 26, and (c) Round 41 in Converging the Global Model in Subsequent Training Rounds.

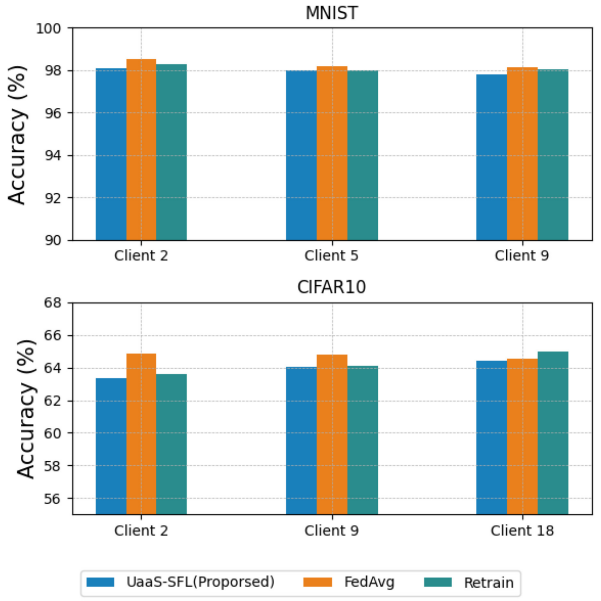

Fig. 13. Comparative evaluation of local clients' accuracy before and after our unlearning methodology, spanning MNIST and CIFAR-10 datasets with configurations of 10 and 25 clients respectively.

using the MNIST and Fashion-MNIST datasets, as shown in Table II.

For the MNIST dataset, the FedRecovery method achieved a clean accuracy of 94.78% and a backdoor accuracy of 11.20%. While this method shows acceptable performance, there is a notable decrease in clean accuracy compared to other approaches. FedEraser yielded a clean accuracy of 96.32% and a backdoor accuracy of 11.23%. Retraining without the compromised client improved clean accuracy to 97.25% and reduced backdoor accuracy to 10.13%, serving as the baseline. Our proposed service closely approximated the baseline, attaining a clean accuracy of 97.01% and a backdoor accuracy of 10.97%, demonstrating improved performance over FedRecovery and effectively balancing high clean accuracy with diminished backdoor threats. In the case of the Fashion-MNIST dataset, as presented in Table II, FedRecovery demonstrated a clean accuracy of 77.43% and a backdoor accuracy of 10.93%. Retraining enhanced clean accuracy to 82.11% and lowered backdoor accuracy to 10.37%, mirroring the patterns observed with the MNIST dataset. The FedEraser strategy achieved a clean accuracy of 80.61% and a backdoor accuracy of 11.17%, indicating a successful reduction in the influence of compromised client data while maintaining commendable performance. Conversely, our service attained a high clean accuracy of 82.50% and a backdoor accuracy of 10.89%, highlighting its efficiency in improving clean accuracy and fortifying defenses against backdoor incursions. These findings underscore the effectiveness of our proposed strategy in striking an optimal balance between achieving high clean accuracy and effectively removing compromised client data from the global model.

Overall, our proposed service consistently demonstrates superior performance across both datasets, closely matching the baseline results obtained through retraining but without the associated computational costs. This affirms the efficacy of our approach in maintaining model integrity and mitigating backdoor vulnerabilities while being more practical for real-world applications.

VIII. Conclusion & Future Works

This study introduces UaaS-SFL, a novel service designed to secure the management of FL systems in IoT networks against data poisoning attacks. Our objective is to maintain the global model's integrity by negating the harmful influence of attackers, thus ensuring the system's reliability and optimal performance. Our evaluations revealed that our service adeptly ensures the accuracy in removing poison clients deviates by less than 10% from the baseline achieved through retraining from scratch. This finding highlights the efficacy of our methodology in safeguarding FL management systems.

At the heart of our approach is a systematic unlearning service that seamlessly integrates into existing FL frameworks. This technique employs specific criteria to identify and neutralize the influence of malicious clients, effectively cleansing the global model. Empirical evidence from real-world experiments supports the efficacy of our method in warding off data poisoning attacks, underscoring the significance of our service in enhancing the safety and efficiency of IoT networks.

While our framework effectively ensures that malicious clients participate in the unlearning process, we recognize that determined adversaries may still attempt to interfere with or bypass these mechanisms. In future work, we plan to focus on enhancing the security and resilience of the system to prevent such interference. By strengthening the system's defenses against potential manipulations, we aim to secure the FL environment further and uphold the integrity and reliability of the collaborative learning process.

ACKNOWLEDGMENT

The authors would like to extend their acknowledgment to Robert Shen from RACE (RMIT AWS Cloud Supercomputing Hub) for their invaluable provision of computing resources.

REFERENCES

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 94–101, Feb. 2018.

[3] Y. Gahi, M. Guennoun, and H. T. Mouftah, "Big data analytics: Security and privacy challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2016, pp. 952–957.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[5] L. U. Khan, Z. Han, D. Niyato, and C. S. Hong, "Socially-aware-clustering-enabled federated learning for edge networks," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 2641–2658, Sep. 2021.

[6] B. Zhao, X. Liu, W.-N. Chen, and R. Deng, "CrowdFL: Privacy-preserving mobile crowdsensing system via federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4607–4619, Aug. 2023.

[7] X. Jiang, J. Zhang, and L. Zhang, "FedRadar: Federated multi-task transfer learning for radar-based Internet of Medical Things," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1459–1469, Jun. 2023.

[8] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2nd Quart., 2021.

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, Aug. 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[10] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.

[11] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022.

[12] B. Custers, A. M. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. van der Hof, *EU Personal Data Protection in Policy and Practice* (Information Technology and Law Series), 1st ed. Hague, The Netherlands: T.M.C. Asser Press, 2019. [Online]. Available: https://www.springer.com/gp/book/9789462652811

[13] P. Bukaty, *The California Consumer Privacy Act (CCPA): An Implementation Guide*. Cambridgeshire, U.K.: IT Gov. Publ., 2019. [Online]. Available: https://books.google.com.au/books?id=vGWfDwAAQBAJ

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Apr. 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[15] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *Proc. IEEE Symp. Security Privacy*, 2015, pp. 463–480.

[16] L. Bourtoule et al., "Machine unlearning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2021, pp. 141–159.

[17] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten, "Certified data removal from machine learning models," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 3832–3842.

[18] S. Neel, A. Roth, and S. Sharifi-Malvajerdi, "Descent-to-delete: Gradient-based methods for machine unlearning," in *Proc. 32nd Int. Conf. Algorithmic Learn. Theory*, Mar. 2021, pp. 931–962. [Online]. Available: https://proceedings.mlr.press/v132/neel21a.html

[19] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 13046–13055, Sep. 2024.

[20] Y. Wu, E. Dobriban, and S. Davidson, "DeltaGrad: Rapid retraining of machine learning models," in *Proc. 37th Int. Conf. Mach. Learn.*, Jul. 2020, pp. 10355–10366. [Online]. Available: https://proceedings.mlr.press/v119/wu20b.html

[21] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu, "Machine unlearning: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1–36, 2023. [Online]. Available: https://doi.org/10.1145/3603620

[22] L. Wu, S. Guo, J. Wang, Z. Hong, J. Zhang, and Y. Ding, "Federated unlearning: Guarantee the right of clients to forget," *IEEE Netw.*, vol. 36, no. 5, pp. 129–135, Sep./Oct. 2022.

[23] W. Yuan, H. Yin, F. Wu, S. Zhang, T. He, and H. Wang, "Federated unlearning for on-device recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Min.*, 2023, pp. 393–401. [Online]. Available: https://doi.org/10.1145/3539597.3570463

[24] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, "FedEraser: Enabling efficient client-level data removal from federated learning models," in *Proc. IEEE/ACM 29th Int. Symp. Qual. Service (IWQOS)*, 2021, pp. 1–10.

[25] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," 2022, *arXiv:2201.09441*.

[26] G. Li, L. Shen, Y. Sun, Y. Hu, H. Hu, and D. Tao, "Subspace based federated unlearning," 2023, *arXiv:2302.12448*.

[27] L. Zhang, T. Zhu, H. Zhang, P. Xiong, and W. Zhou, "FedRecovery: Differentially private machine unlearning for federated learning frameworks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4732–4746, 2023.

[28] J. Wang, S. Guo, X. Xie, and H. Qi, "Federated unlearning via class-discriminative pruning," in *Proc. ACM Web Conf.*, 2022, pp. 622–632. [Online]. Available: https://doi.org/10.1145/3485447.3512222

[29] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[30] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 634–643. [Online]. Available: https://proceedings.mlr.press/v97/bhagoji19a.html

[31] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a.html

[32] A. E. Samy and Š. Girdzijauskas, "Mitigating sybil attacks in federated learning," in *Proc. Int. Conf. Inf. Security Pract. Exp.*, 2023, pp. 36–51. [Online]. Available: https://doi.org/10.1007/978-981-99-7032-2_3

[33] M. T. Hossain, S. Islam, S. Badsha, and H. Shen, "DeSMP: Differential privacy-exploited stealthy model poisoning attacks in federated learning," in *Proc. 17th Int. Conf. Mobility, Sens. Netw. (MSN)*, 2021, pp. 167–174.

[34] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1749–1758.

[35] F. Kamalov and H. H. Leung, "Deep learning regularization in imbalanced data," in *Proc. Int. Conf. Commun., Comput., Cybersecurity, Inform. (CCCI)*, 2020, pp. 1–5.

[36] Y. LeCun, C. Cortes, and C. Burges. "MNIST handwritten digit database." 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist

[37] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

[38] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[39] S. Tan and Z. Tan, "Improved LeNet-5 model based on handwritten numeral recognition," in *Proc. Chin. Control Decision Conf. (CCDC)*, 2019, pp. 6396–6399.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[41] M.-I. Nicolae et al., "Adversarial robustness toolbox v1.2.0," 2018, *arXiv:1807.01069*.

Wathsara Daluwatta recieved the Bachelor of Science (Hons.) degree in software engineering from the University of Colombo School of Computing, Sri Lanka. He is currently pursuing the Ph.D. degree with RMIT University, Melbourne, VIC, Australia. His research interests lie in the areas of machine learning, privacy, cybersecurity, and distributed systems.

Ibrahim Khalil received the Ph.D. degree in computer science from the University of Bern, Switzerland, in 2003, marking the beginning of a career that would span continents and sectors. He is a Professor with the School of Computing Technologies, RMIT University, Melbourne, Australia. Before his tenure with RMIT University, he amassed significant experience in the tech hubs of Silicon Valley, where he worked as a software engineer focused on secure network protocols and smart network provisioning. His academic journey also includes valuable stints with EPFL, Switzerland, the University of Bern, and Osaka University, Japan. A prolific contributor to the fields of blockchain and privacy, he has led several high-profile ARC discovery and linkage grants in Australia from 2017 to 2021, alongside securing international European grants, industry grants, and a QNRF Grant from Qatar. His research portfolio is diverse, encompassing privacy, blockchain, secure AI data analytics, distributed systems, e-health, wireless and body sensor networks, biomedical signal processing, and network security, reflecting a broad and impactful engagement with the frontiers of computing and technology.

Shehan Edirimannage received the Bachelor of Science (Hons.) degree in computer science from the University of Colombo School of Computing, Colombo, Sri Lanka, in 2022. He is currently pursuing the Ph.D. degree in computer science with the School of Computing Technologies, RMIT University, Melbourne, VIC, Australia. His research interests include machine learning, distributed systems and computing, blockchain, privacy, and secure computing.

Mohammed Atiquzzaman (Life Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering and electronics from the University of Manchester, U.K., in 1987. He is the Edith Kinney Gaylord Presidential Professor and a Hitachi Chair with the School of Computer Science, University of Oklahoma, Norman, OK, USA. His research interests span a wide array of topics within communications, machine learning, computer network protocols, wireless and mobile networks, satellite networks, and optical communications. In addition to his research endeavors, he holds the prestigious position of Editor-in-Chief of the *Journal of Networks and Computer Applications*, and is the Founding Editor-in-Chief for *Vehicular Communications*. His editorial acumen also extends to various IEEE journals, and his organizational skills have been instrumental in co-chairing numerous IEEE international conferences, including IEEE Globecom, further highlighting his profound impact on the global communications and networking community.