Secure and Reversible Federated Learning for IoT-Driven Food Supply Chain Monitoring*

Mihailo Ilić[1,2,*], Nemanja Gabor[1], Mirjana Ivanović[2] and Milenko Tošić[1]

[1]VizLore Labs Foundation, Braće Ribnikar 56, Novi Sad 21000, Republic of Serbia
[2]University of Novi Sad, Faculty of Sciences, Trg Dositeja Obradovića 3, Novi Sad 21000, Republic of Serbia

Abstract

Modern food supply chains face pressing concerns regarding food safety, traceability, and quality assurance. There are many actors in global food supply chains covering various needs like transportation, storage, testing, etc. All of these numerous links in the chain carry a certain risk of compromising food quality either intentionally or unintentionally. Because of this, quality checks at each link need to be carried out for the benefit of all stakeholders involved, from the different parties in the supply chain to the end users. The combined effort of Internet of Things (IoT) and Machine Learning (ML) advancements can help minimize risks in food supply chains. In this paper we identify some of the key risks in food supply chains, propose an IoT-based approach to monitoring food quality at the edge, prepare a food quality dataset using proprietary AISpectra edge units, and validate a privacy-first distributed approach to training ML models based on Federated Learning (FL). To eliminate the significant computational overhead and software fragility associated with full retraining cycles, we optimize the process through federated unlearning. We ensure the reliability and security of the system, by applying state-of-the-art techniques (Fast-FedUL) for removing contributions of specific actors in the supply chain which choose to withdraw their data either due to security or privacy reasons, or are eliminated from the training process due to intentional or unintentional model poisoning.

Keywords

Federated Learning, Supply Chains, Federated Unlearning, Internet of Things (IoT)

1. Introduction

Supply chains are complex systems made up of various stakeholders including producers, distributors, shipping partners, and consumers. Due to large numbers of different participants in these processes, supply chains face transparency, information asymmetry, and limited collaboration challenges. These cause issues like delays, inefficiencies, and most importantly compromised product quality. Consequently, there exists a strong need for transformative solutions. To address these pressing issues and raise a shared responsibility among supply chain participants, integrating cutting-edge technologies like the Internet of Things (IoT) and machine learning (ML) should be of high priority.

IoT devices deployed strategically throughout the supply chain can be used as edge-based measurement checkpoints, enabling stakeholders to trace and monitor products throughout the supply chain. Since supply chains are often comprised of numerous participants, an implementation like this one would imply heavy data collection, which is an excellent opportunity to apply ML techniques to learn from this data. The main challenges for these kinds of setups are training of ML models in distributed environments, particularly those in which data privacy is paramount. One approach to solving this issue is the use of federated learning (FL), a privacy enhancing approach to distributed ML.

This paper focuses on a strong practical example of applying FL for risk prediction in food supply chains, where multiple stakeholders jointly train models without sharing raw operational data. To

SQAMIA2026: 13th Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, September 16–18, 2024, Niš, Serbia
*Corresponding author.
†These authors contributed equally.
mihailo.ilic@vizlore.com,milic@dmi.uns.ac.rs (M. Ilić); nemanja.gabor@vizlore.com (N. Gabor); mira@dmi.uns.ac.rs (M. Ivanović); milenko.tosic@vizlore.com (M. Tošić)
http://people.dmi.uns.ac.rs/~milic/ (M. Ilić); https://perun.pmf.uns.ac.rs/wiki/mirjana-ivanovic (M. Ivanović)
0009-0005-4492-017X (M. Ilić); 0000-0003-1946-0384 (M. Ivanović)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-19

CEUR-WS.org/Vol-4336/paper19.pdf

this end, we consider food quality analyses via IoT edge based spectrometers and cameras, collecting ultraviolet, visible, and near infrared data, called AISpectra. We collect our own dataset on extra virgin olive oil, where the ML goal is to detect food adulteration based on food samples examined by the AISpectra device.

Data collection for these kinds of scenarios would certainly be conducted in numerous points. Because of this, FL is a natural fit for training these types of models. However, due to data drifts, specific statistical distributions, device miscalibrations, and privacy concerns, certain participants in the training process may come and go at different times during training. For these reasons, some participants may wish to pull their contributions from the global model trained via FL. With clients withdrawing from the federation, it is vital to remove their contribution to the global model, and delete all training cycles records linked to them. In this paper we explore the effects that dropouts of these *dynamic* clients have on the quality of the global model, and test the viability of using certain *federated unlearning* techniques compared to full retraining of the global models.

The main contributions of this paper are:

- An IoT, privacy-based pipeline for food quality monitoring, built on AISpectra units and Federated Learning.
- A real-world food quality dataset collected with AISpectra units, targeting olive oil adulteration, from fused visual **VIS**, near infrared **NIR**, and fluoroscopic **FLUO** spectra.
- An experimental evaluation of **Fast-FedUL** against full retraining, no-unlearning and baseline scenarios for cases in which a participant must be removed.

The rest of this paper is structured as follows. Section 2 gives an overview of related work. Food supply chains and the premise of the use case described in this paper are in Section 3. The datasets and experimental setup are presented in Section 4. Results are discussed in Section 5 and concluding remarks are given in the last section.

2. Related Work

Food adulteration is a major risk in supply chains, as it is the intentional addition of certain substances into food products for numerous reasons such as hiding poor quality of the packaging material, selling decomposed foods, misbranding, and adding toxic elements to the food product [1, 2]. Because of this, adequate food testing is required to ensure the protection of subsequent actors in the food chain, and ultimately, the end consumers. Poor quality of food is associated with various health risks, including but not limited to diabetes, cancer, and cardiovascular disease [3, 2]. The main reason for food adulteration is economic benefit, to increase profit margins. Food safety can be increased in multiple ways, by securing actors in the chain that cover aspects of food processing, preparation, storage, and tracking food safety and quality indicators throughout complex supply chains [4].

According to [2], multiple analytical methods exist. **Chromatography** is a technique that separates chemical compounds and food items based on their boiling point or polarity. **Spectroscopy** is a non-destructive method that applies subsurface inspection to detect adulteration. This is a rapid analysis technique based on processing large numbers of wavelengths of electromagnetic spectra. **Electronic nose** is an electronic sensor that determines food adulteration in the physical and chemical sense, based on imitating the human sense of smell. These sensors are based on detecting certain odors at low cost. **Enzyme-linked immunosorbent assay** is an easy to perform technique for detecting adulteration of meat products. The AISpectra device is a spectroscopic sensor that utilizes a combination of visual (VIS), near infrared (NIR), and fluoroscopic (FLUO) wavelengths.

Data collection through IoT devices in the food safety domain has made it easier to apply artificial intelligence (AI) and machine learning (ML) techniques to food related issues, like adulteration. Many of these methods also rely on spectroscopic data, as it is a non-destructive way of assessing the quality of food. Examples of using spectral analysis combined with numerous AI/ML techniques like support vector machines (SVMs), neural networks (NNs), convolutional neural networks (CNNs), etc., can be

seen in literature [5, 6, 7, 8]. Techniques for ensuring safety in food supply chains now heavily rely on Industry 4.0 technologies like big data, blockchain, IoT, and federated learning (FL) [9, 10].

The inclusion of FL is particularly interesting, since it is a privacy and security enhancing technique of distributed machine learning [11, 12]. Since numerous actors are present in supply chains, with multiple roles linked to food preparation, storage, and tracking, this technique is a perfect fit to facilitate collaborative learning. First, sharing of sensitive data is not a problem in FL, and second, FL can be both horizontal and vertical [11], thus it can support collaboration between actors in the supply chain which collect different parameters of food data. In this paper, we consider the case of horizontal FL, where all participants collect the same feature columns, since they are based on the AISpectra dataset. For this purpose, we apply neural networks to tabular data derived from spectral data collected using this edge device. Food adulteration can also be posed as an anomaly detection task, where food samples are analysed and out-of-distribution (OOD) data points are declared as anomalies. Anomaly detection usually implies the use of unsupervised training, where models are only shown normal data points during training, and has been applied successfully in FL with state-of-the-art models such as memory-augmented autoencoders [13, 14].

Even though FL offers enhanced privacy and security, especially when combined with techniques like differential privacy or homomorphic encryption [11], different actors may decide to leave the training process due to numerous reasons. Conversely, some edge nodes may be flagged as clients supplying suboptimal updates to the global model either intentionally (model poisoning) [15] or unintentionally (e.g. due to edge device miscalibration). Because of this, it is important to have methods to identify these clients, and to retroactively remove their updates to the global model. A trivial approach to removing updates of compromised clients is to simply rewind model training to a point in time where the selected client was not a participant, and retraining with the rest of the clients from that point in time. However, this wastes client resources and can potentially take a significant amount of time. In the study conducted in this paper, we consider the importance of efficient federated unlearning, and focus specifically on the application of Fast-FedUL [16].

3. Domain Context and Operational Motivation

In this paper, we consider a particular case of collaborative learning based on FL, where clients may come and go at any time of the training process. Clients may abandon training due to numerous reasons, for example by being flagged as malicious nodes and being removed from the training pool, or by opting out of training due to e.g. privacy concerns. In both cases, training cannot simply continue from that point in time, since their influence is embedded into the global model. Consequently, some form of retraining must be done to "forget" the contributions of the client which left. Full retraining is suboptimal for numerous reasons. First, it is **operationally slow**. Coordinating between numerous stakeholders may be extremely difficult, since some nodes may be offline, have limited edge resources, or can be synchronized only at specific intervals. In a real supply-chain federation, participants are not always available like machines in one cloud cluster. Second, **the model may have been trained over multiple seasons**. In agriculture and food supply chains, data is seasonal. A model may have learned from harvest 2024, storage winter 2024/2025, export season 2025, and harvest 2025. Replaying all that training could take weeks or months, even if the data still exists. Third, **the clients may simply not want to rerun old training cycles**. This can happen due to numerous reasons, like system changes, evolving data schemas, or lack of storage of historical data. Fourth, **complete retraining may cause trust issues**. If every partner withdrawal forces the entire consortium to restart, the federation becomes fragile. Participants may be less willing to join if they know that one partner leaving can disrupt the whole ecosystem.

Because of this, an alternative to full retraining must be introduced. One such approach is Fast-FedUL [16]. This approach assumes that a log of model updates received from each individual edge node exists at the server side. When one or more clients are removed from the training process in training round T_{n+k}, an approximation (or *"what if"* scenario) can be calculated on what the model would

have looked like if those clients have never been included in training at all. Assuming the removed clients joined in round T_n, the k training rounds up until T_{n+k} can be approximated using Fast-FedUL completely at the server side, thus saving client resources in terms of compute time. This kind of approach makes training more robust, and ultimately increased trust in the system. Most importantly, the contributions of the removed clients are eliminated from the global model.

4. Experimental Setup

4.1. Methodology

Our experimental evaluation covers four scenarios and three different cluster configurations. The scenarios presented are:

(i) Baseline – where only base nodes train without any help of the dynamic nodes;
(ii) No Unlearning – in which the dynamic nodes' contributions were not rolled back;
(iii) Retrain – where the model was rolled backed and retrained with only the base nodes;
(iv) Fast-FedUL [16] – where federated unlearning was used in case dynamic nodes abandoned training.

In each experiment, the dynamic nodes were configured to join and leave the network simultaneously. To evaluate scalability, we utilized three distinct cluster configurations, summarized in Table 1 and visualized in Figure 1.

Table 1
Experimental Cluster Configurations

Configuration	Base Nodes	Dynamic Nodes
Small-Scale	5	1
Medium-Scale	10	1
Large-Scale	20	10

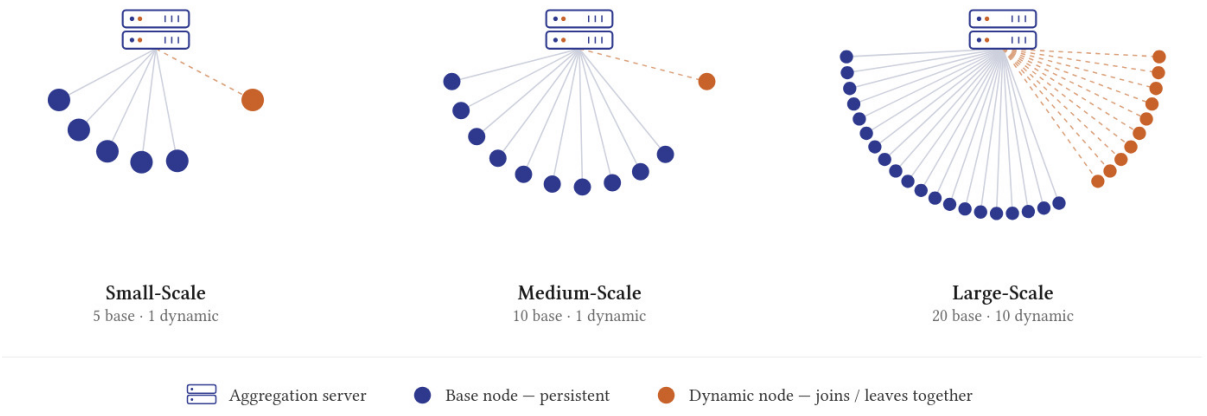


Figure 1: Evaluated cluster configurations.

In the first ten rounds only the base nodes participated in training. On the eleventh round the dynamic nodes joined and contributed to the global model as well, and left on the fifteenth round. At this point, one of the unlearning strategies was applied and then the rest of the nodes would train until the twentieth round. This process is visualized in Figure 2.

Data was distributed between nodes through Dirichlet partitioning [17], ensuring a non-IID data distribution among edge nodes, as in real-world scenarios [18]. By applying a low α value, the dataset



Figure 2: Round-by-round training timeline.

becomes more unevenly distributed. Two different aggregation strategies were used on the same setup, federated averaging (FedAvg) [19] and federated proximal optimization (FedProx) [20]. Each run used two local epochs per client. The batch size was 64 for CIFAR100 and 32 for the AISpectra dataset. Stochastic Gradient Descent was used as the optimizer and learning rate η varied by experiment, being 0.01 for CIFAR100 and 0.005 for AISpectra.

4.2. Model architecture

For the CIFAR100 dataset, a convolutional network was used. Each block applies a 3x3 convolution with padding 1, batch normalization and ReLU, and additionally the first two blocks apply a 2x2 max pooling. The channel width is doubled across the blocks, from 32 to 64 to 128, and global average pooling collapses it into a 128 dimensional descriptor. And a final linear layers maps it to 100 class logits.

The AISpectra dataset was classified with a fully connected network with two hidden layers. Each hidden layer is a linear map followed by a batch normalization, ReLU, and dropout with $p = 0.2$. From the input features we turn it into 256, and we halve it to 128 before a project to the number of class logits.

4.3. Datasets

The experiments were evaluated on two datasets, AISpectra and CIFAR100 [21]. Table 2 contains information on both datasets.

Attribute	AISpectra	CIFAR100
Datapoints	1050	60,000
Features	228 (VIS) + 288 (NIR) + 288 (FLUO) + 1 (label)	$32 \times 32 \times 3$
Modality	Tabular	Image
Number of Classes	23	100
Task	Olive oil adulteration detection	Image classification

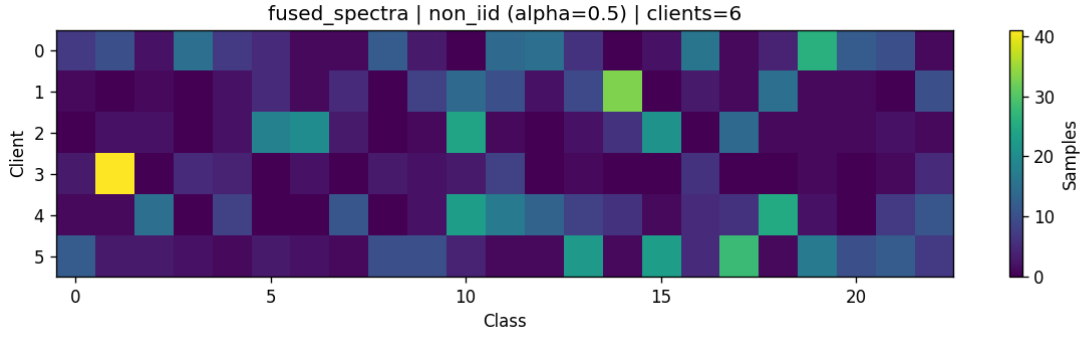
Table 2
Descriptions of the datasets used in the experiments.

The AISpectra dataset is a private dataset containing data on extra virgin olive oil (**EVOO**), and the task was to detect adulteration, meaning impurities in the food samples. Because AISpectra is

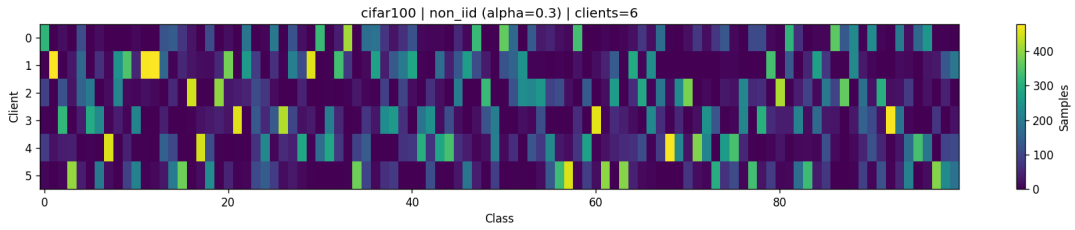
a private dataset, it is a natrual candidate for federated unlearning. Contributing parties may later want to remove their data, motivating the need for an efficient unlearning algorithm. Each data point comprises multi-modal spectroscopic measurements, specifically from the visible light region (**VIS**), Near-Infrared (**NIR**), and Fluorescence (**FLUO**) regions. The labels represent the levels of impurity in the oil samples and the following naming scheme is in place `xvoo_<adulterant><XX>`, where XX denotes how much of the adulterant is present in the extra virgin olive oil. Pure olive oilis marked as `xvoo_none`, which means the ratio is 100:0. Three adulterants are present in the dataset linseed oil (`lso`), refined olive oil (`oo`) and sunflower oil (`sfo`). For the linseed oil and refined olive oil, they are present in ten variations, done in steps of 10% starting from 10% and ending with 100%, while the sunflower oil was only present at 50% and 100%.

On the other hand CIFAR100 [21] is a public, widely used dataset for image classification, containing 100 classes grouped into 20 superclasses. It contains 60,000 images of size 32×32 pixels with three RGB channels. Each class contains 600 images which are grouped in 20 super classes.

In order to evaluate how well the unlearning algorithm performed, in this paper the two datasets were constructed through Dirichlet partitioning with $\alpha = 0.3$ for CIFAR100 and $\alpha = 0.5$ for the AISpectra dataset. Only non-IID scenarios were tested as it is a more likely scenario in federated settings. The milder skew for AISpectra reflects its smaller size, ensuring each client retains a viable number of samples per class. Data partitioning among clients remained the same for each configuration, no matter the federated aggregation function (FedAvg or FedProx) being used in that particular set of experiments. An example of partitioning in the $5 + 1$ setup is visualized in Figure 3.



(a) AISpectra ($\alpha = 0.5$)



(b) CIFAR-100 ($\alpha = 0.3$)

Figure 3: Per-client label distribution under Dirichlet partitioning, Small configuration ($5 + 1$).

5. Results & Discussion

All four scenarios were evaluated based on three conditions:

(i) **Fidelity** – How does the model recover after the dynamic clients depart? Does it recover to the same accuracy of the retraining scenario?
(ii) **Efficiency** – Is the computational cost of the algorithm smaller than the retraining scenario?
(iii) **Stability** – How disruptive are the dynamic clients and how quickly does the global model recover aftewards?

Full retraining (*Retrain scenario*) is the baseline case. In this scenario, the global model resets to where it was prior to the dynamic node joining. After this, the remaining nodes need to retrain again. So the question is how *Fast-FedUL* compares to **Retrain** and how much computational cost it saves. Table 3 shows aggregated results for FedAvg. The*Fast-FedUL* correction step size is denoted α_u to distinguish it from the Dirichlet α value.

Table 3
Final accuracy (mean over the last five rounds) under **FedAvg**. α_u is the Fast-FedUL correction step size.

					Fast-FedUL		
Dataset	**Scale**	Baseline	No-Unl.	Retrain	α_u=0.01	α_u=0.05	α_u=0.1
CIFAR100	Small (5+1)	30.53	**30.68**	30.53	30.56	30.56	30.50
	Medium (10+1)	26.79	**26.93**	26.79	26.82	26.84	26.85
	Large (20+10)	18.78	**18.93**	18.78	17.82	17.71	17.56
AISpectra	Small (5+1)	53.05	54.10	53.05	52.00	54.48	**56.29**
	Medium (10+1)	**47.14**	46.57	**47.14**	44.86	45.71	45.33
	Large (20+10)	28.57	28.57	28.57	**31.14**	29.90	27.05

Table 3 shows the final accuracy, averaged over the last 5 rounds, for every combination of dataset and cluster configuration (Small: 5 base nodes and 1 dynamic; Medium: 10 base nodes and 1 dynamic; Large: 20 base nodes and 10 dynamic). The table reports each unlearning scenario, **Baseline**, **No-Unlearning**, **Retrain**, and **Fast-FedUL** at three correction step sizes α_u=$\{0.1, 0.05, 0.01\}$, all under FedAvg. The **Baseline** and **Retrain** numbers are the same, which is intended: **Baseline** was run without the dynamic nodes, and **Retrain** was trained with the dynamic nodes but afterwards retrained on the base nodes only.

For the CIFAR100 dataset, the results on the Small and Medium configuration are within 0.2% of each other, and **Fast-FedUL** is marginally better than **Retrain**, by just $0.03\% - 0.06\%$. For the Large configuration, **Retrain** performed better than **Fast-FedUL**, but only by $0.96\% - 1.22\%$. The value of α_u made no difference, only on the Large configuration did it have a slight effect, with α_u=0.01 giving the best result.

The AISpectra dataset painted a slightly different picture because it is small and has a significant class imbalance. The value of α_u mattered since in the Small configuration it impacted accuracy by 4%. Setting α_u=0.1 not only gave the best result for **Fast-FedUL** but it beat the **Baseline** and **Retrain** scenarios. For the Medium configuration, α_u parameter showed the same pattern, but slightly worse than the **Retrain** and **No-Unlearning** scenarios. By looking at the Large configuration, the α_u=0.01 parameter showed the best results, beating **Retrain** by 2.57%. Setting α_u=0.05 also outperformed the **Retrain** scenario, while α_u=0.1 provided the worst results by a slight margin.

It is important to acknowledge that the accuracies are low: CIFAR100 ranges from 18% to 31%, while AISpectra ranges from 27% to 56%. This is because small models were used, a short 20-round setup, and a strong non-IID split. However, the key takeaway from this set of experiments is that through the use of **Fast-FedUL**, it is possible to closely approximate the fully retrained model without the need for complete retraining.

Figure 4 visualises the α_u sweep. Odd numbered rows use the FedAvg strategy and even numbered rows use FedProx. The CIFAR100 rows are the same across step sizes, while the AISpectra rows are not. For AISpectra, the **fused_spectra_1** accuracies rise with a larger α_u, from 0.520 to 0.563. By contrast, **fused_spectra_2** tested with FedProx strategy shows no clear correlation between α_u and accuracy. Its best result was 0.504 at α_u=0.05 and worst was 0.432 at α_u=0.1.

By comparing FedAvg and FedProx in Table 4, we conclude that the FedAvg strategy provides better results compared to FedProx. On the CIFAR100 dataset the two stategies are essentially the same. On the other hand, using FedProx on the AISpectra dataset shows a large accuracy drop, in some cases by 13%. It only gives us marginally better results in the Medium cluster configuration, while the Small and Large configuration result show worse performance than FedAvg strategy. For the Medium cluster

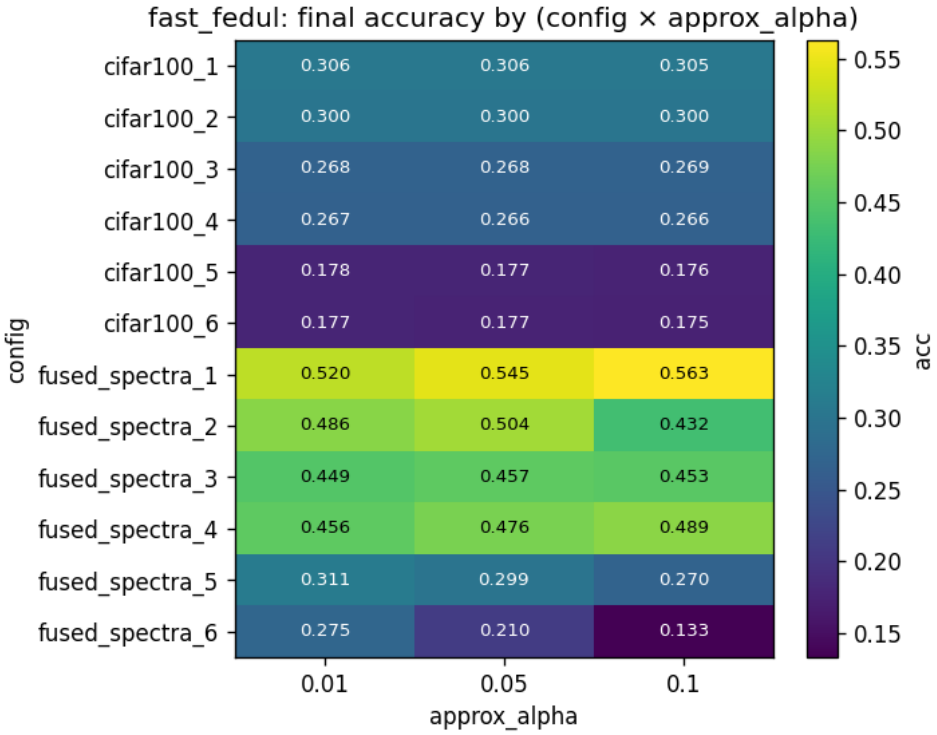


Figure 4: Fast-FedUL final accuracy across configuration (rows) and correction step α_u (columns). CIFAR100 is insensitive to α_u; AISpectra is not, and large steps are harmful under FedProx (bottom rows).

Table 4
FedAvg vs FedProx strategies on Fast-FedUL, difference in final accuracy (percentage points, + favours FedProx while − favours FedAvg) at each correction step size α_u.

Dataset	**Scale**	α_u=0.01	α_u=0.05	α_u=0.1
CIFAR-100	Small	−0.53	−0.54	−0.47
	Medium	−0.17	−0.23	−0.22
	Large	−0.07	−0.04	−0.07
AISpectra	Small	−3.43	−4.10	**−13.05**
	Medium	+0.76	+1.90	+3.52
	Large	−3.62	−8.86	**−13.71**

configuration, FedProx yields better results than FedAvg, with accuracies rising as α_u increases, whereas FedAvg performs slightly worse. Finally for the Large cluster configuration, accuracies dip with higher α_u, with $alpha_u$=0.01 giving the best result. The same holds for FedProx, where α_u=0.01 produces the best accuracies before dropping sharply to 0.133 at α_u=0.1.

Efficiency is important: if **Fast-FedUL** performed similarly to **Retrain**, there would be little reason to choose it. The **Retrain** scenario runs 5 extra rounds because it needs to train from the point when the nodes have joined up until they leave. **Fast-FedUL** removes the departed nodes' influence using the models' history, scaled by step size α_u, and completes without any extra rounds.

As shown in Table 5, **Fast-FedUL** is consistently faster than **Retrain** across all cluster configurations and datasets. The advantage of **Fast-FedUL** over **Retrain** becomes more apparent in situations with a large number of clients, and where the clients which need to be removed have been active for more training rounds. The longer training went on, the more global training rounds need to be reran, which raises the computational cost for the clients. The longer the retraining run, the greater the incentive for using **Fast-FedUL**.

Figure 5 answers the question of how the global model recovers after the fifteenth round where the

Table 5

Total wall-clock cost under FedAvg (mean over three seeds). Retrain executes 25 effective rounds (original training plus replay from the join round); Fast-FedUL completes in 20 rounds via a single correction.

Dataset	Scale	Retrain (s)	Fast-FedUL (s)	Speed-up
CIFAR100	Small	605.9	489.2	1.24×
	Medium	658.1	529.6	1.24×
	Large	614.3	502.9	1.22×
AISpectra	Small	2.84	2.21	1.28×
	Medium	3.44	2.75	1.25×
	Large	4.39	3.59	1.23×

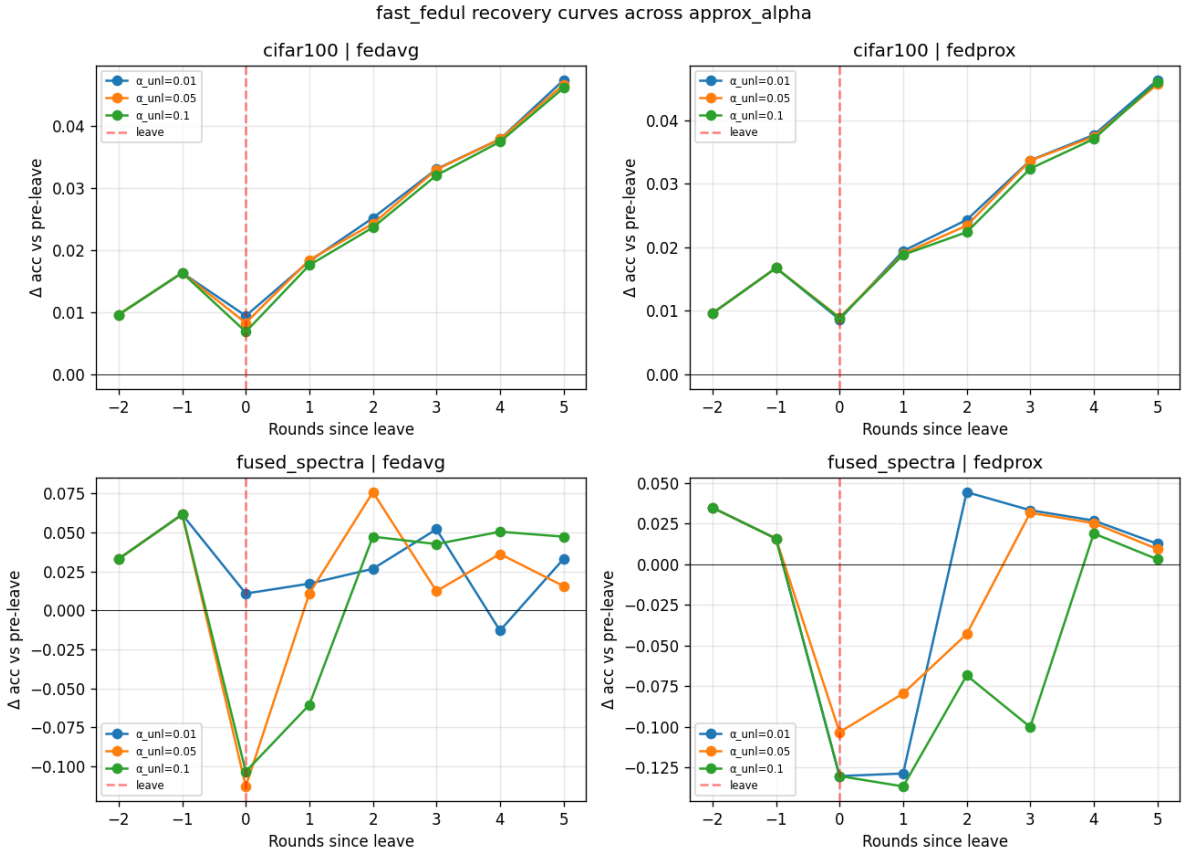


Figure 5: Fast-FedUL recovery curves: change in global accuracy relative to the pre-departure mean as a function of rounds since the dynamic client left (red dashed line), for each correction step α_u and the baseline. Top: CIFAR100; bottom: AISpectra; left: FedAvg, right: FedProx.

dynamic clients leave. On CIFAR100, the curves are all tightly clustered and rise smoothly from the shallow dip. Under both FedAvg and FedProx, the model effectively recovers within a single round and is insensitive to changes in α_u. AISpectra paints a very different picture, the curves are jagged and the differences between α_u values are visible. This indicates that removing influential clients takes a big hit on the small, imbalanced model. Recovery slows as α_u grows under FedAvg, and the slowdown is more pronounced under FedProx with the largest step dropping by ≈ 18 points and taking about 2 rounds to recover. This confirms that a conservative correction step and FedAvg aggregation are the right choice for the AISpectra dataset

Figure 6 summarizes how the correction step α_u affects **Fast-FedUL**. The top left panel shows that CIFAR100 accuracy stays the same across α_u values, and between FedAvg and FedProx aggregation strategies, FedAvg produces slightly better results than FedProx. For the AISpectra dataset, FedAvg is

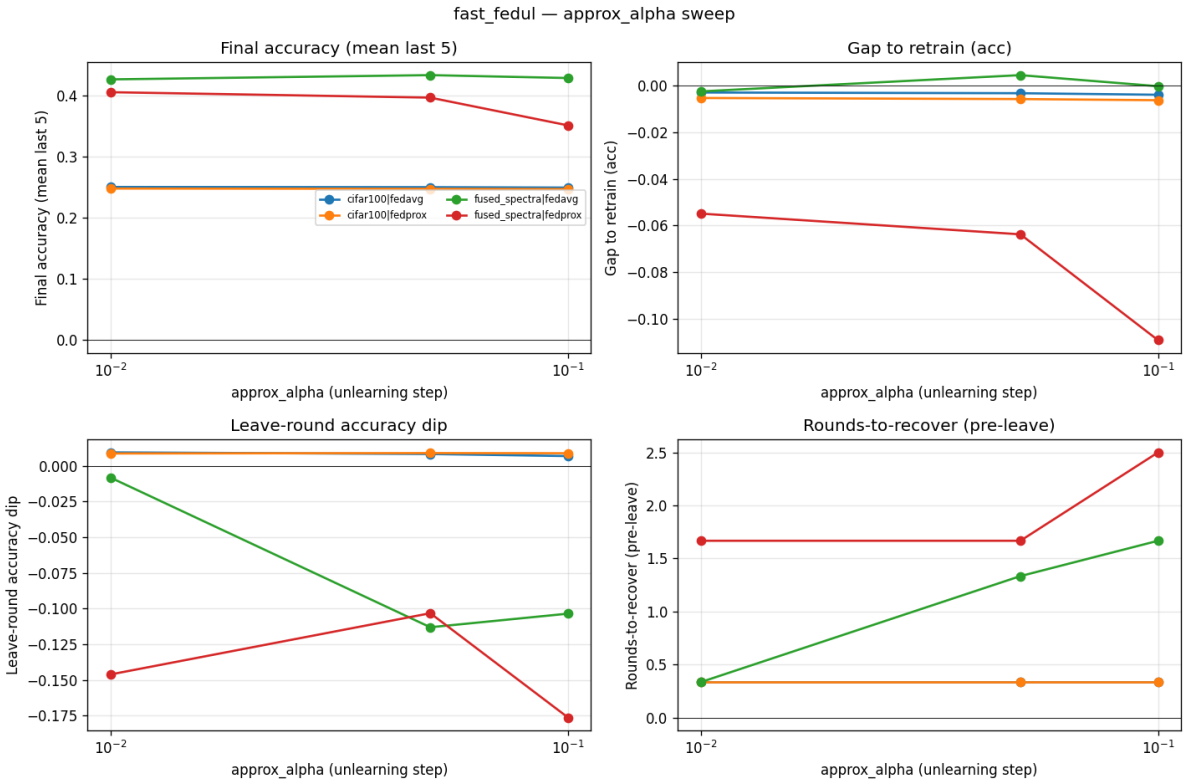


Figure 6: Effect of the Fast-FedUL correction step α_u, averaged over the three cluster scales, for each dataset and aggregation strategy. Panels: final accuracy (mean over the last five rounds), gap to Retrain, accuracy dip at the leave round, and rounds-to-recover.

ahead of FedProx by ≈ 3 points, and FedProx shows a deeper dip. The *gap to Retrain* stays the same for CIFAR100 across both aggregation strategies, but for AISpectra there is a large gap indicating that FedAvg strategy is the better for this dataset. The leave round accuracy for CIFAR100 is a flat line, with no visible dip, but for AISpectra the FedAvg again outperforms FedProx strategy which shows a dip at the end. Finally, the rounds to recover stay the same for both strategies on the CIFAR100 dataset, while for the AISpectra FedProx takes one more round to recover than FedAvg.

Taken together, the fidelity, efficiency, and stability results give a clear picture. On the *fidelity* side, **Fast-FedUL** stays faithful to the Retrain baseline, and in some cases on the AISpectra dataset it even surpasses **Retrain**. For *efficiency*, it is always faster than **Retrain** across all cluster configurations and its advantage grows as the interval between the join and leave rounds increases. This happens because it removes the departed client's influence with a closed form correction over the stored update history, rather than replaying the post join rounds. Finally, for *stability*, it recovers immediately on CIFAR100 and within two rounds on the AISpectra dataset.

6. Conclusion

This paper presents an IoT privacy-based pipeline for food quality monitoring, built on AISpectra spectroscopic units and Federated Learning. Collecting an olive oil adulterated dataset from fused VIS, NIR, and FLUO spectra. And evaluated with **Fast-FedUL** against full **retraining**, **no-unlearning** and **baseline** scenarios for cases in which a node must be removed.

From this evaluation two conclusions are drawn. First, the FedAvg aggregation strategy is safer for unlearning, it matches or beats FedProx in every configuration. Second, a small correction step α_u=0.01 is the most robust, since larger steps only help occasionally on CIFAR100, but disrupts the AISpectra dataset. Overall the Fast-FedUL algorithm is a very good replacement for full retraining when

a client influence must be removed, offering comparable accuracy at a lower cost and is best paired with FedAvg and a small correction step.

For future work, optimizing the unlearning process is crucial. For instance, designing a runtime optimization loop that dynamically scales α_u. Automatically calculating the mathematically optimal step size per round to minimize recovery time while maximizing model performance will go a long way in these kind of setups.

Declaration on Generative AI

The authors have employed Generative AI tools to create Figures 1 and 2.

References

[1] S. Bansal, A. Singh, M. Mangal, A. K. Mangal, S. Kumar, Food adulteration: Sources, health risks, and detection methods, Critical reviews in food science and nutrition 57 (2017) 1174–1189.

[2] Y. K. Anagaw, W. Ayenew, L. W. Limenh, D. T. Geremew, M. C. Worku, T. A. Tessema, W. Simegn, M. L. Mitku, Food adulteration: Causes, risks, and detection techniques, SAGE Open Medicine 12 (2024) 20503121241250184.

[3] B. M. Appelhans, S. A. French, C. C. Tangney, L. M. Powell, Y. Wang, To what extent do food purchases reflect shoppers' diet quality and nutrient intake?, International Journal of Behavioral Nutrition and Physical Activity 14 (2017) 46.

[4] M. Lv, Y. Liu, J. Geng, X. Kou, Z. Xin, D. Yang, Engineering nanomaterials-based biosensors for food safety detection, Biosensors and Bioelectronics 106 (2018) 122–128.

[5] P. Balakrishnan, A. Anny Leema, N. Jothiaruna, P. J. Assudani, K. Sankar, M. B. Kulkarni, M. Bhaiyya, Artificial intelligence for food safety: From predictive models to real-world safeguards, Trends in Food Science & Technology 163 (2025) 105153. URL: https://www.sciencedirect.com/science/article/pii/S0924224425002894. doi:https://doi.org/10.1016/j.tifs.2025.105153.

[6] J. Li, A. Wu, L. Liu, A. Qu, C. Xu, H. Kuang, L. Xu, Analysis of food safety based on machine learning: A comprehensive review and future prospects, Food Chemistry 490 (2025) 145170. URL: https://www.sciencedirect.com/science/article/pii/S0308814625024215. doi:https://doi.org/10.1016/j.foodchem.2025.145170.

[7] X. Wang, Y. Bouzembrak, A. O. Lansink, H. J. van der Fels-Klerx, Application of machine learning to the monitoring and prediction of food safety: A review, Comprehensive Reviews in Food Science and Food Safety 21 (2022) 416–434. URL: https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1541-4337.12868. doi:https://doi.org/10.1111/1541-4337.12868. arXiv:https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1541-4337.12868.

[8] R. Ben Ayed, M. Hanana, Artificial intelligence to improve the food and agriculture sector, Journal of Food Quality 2021 (2021) 5584754. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/5584754. doi:https://doi.org/10.1155/2021/5584754. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/5584754.

[9] S. Gbashi, P. B. Njobeh, Enhancing food integrity through artificial intelligence and machine learning: A comprehensive review, Applied Sciences 14 (2024). URL: https://www.mdpi.com/2076-3417/14/8/3421. doi:10.3390/app14083421.

[10] J. Yu, Y. Chen, Z. Wang, J. Liu, B. Huang, Food risk entropy model based on federated learning, Applied Sciences 12 (2022). URL: https://www.mdpi.com/2076-3417/12/10/5174. doi:10.3390/app12105174.

[11] P. Kairouz, H. B. McMahan, Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (2021) 1–210. URL: https://doi.org/10.1561/2200000083. doi:10.1561/2200000083.

[12] M. Ilić, M. Ivanović, Federated learning - opportunities and application challenges, in: N. T. Nguyen, J. Botzheim, L. Gulyás, M. Nunez, J. Treur, G. Vossen, A. Kozierkiewicz (Eds.), Advances in Computational Collective Intelligence, Springer Nature Switzerland, Cham, 2023, pp. 481–492.

[13] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1705–1714.

[14] M. Ilić, M. Savić, V. Kurbalija, M. Ivanović, G. Fortino, D. Jakovetić, Federated attention autoencoders with a stochastic aggregation scheme for anomaly detection, in: 2025 3rd International Conference on Federated Learning Technologies and Applications (FLTA), 2025, pp. 270–277. doi:10.1109/FLTA67013.2025.11336743.

[15] R. Gosselin, L. Vieu, F. Loukil, A. Benoit, Privacy and security in federated learning: A survey, Applied Sciences 12 (2022). URL: https://www.mdpi.com/2076-3417/12/19/9901. doi:10.3390/app12199901.

[16] T. T. Huynh, T. B. Nguyen, P. L. Nguyen, T. T. Nguyen, M. Weidlich, Q. V. H. Nguyen, K. Aberer, Fast-fedul: A training-free federated unlearning with provable skew resilience, in: A. Bifet, J. Davis, T. Krilavičius, M. Kull, E. Ntoutsi, I. Žliobaitė (Eds.), Machine Learning and Knowledge Discovery in Databases. Research Track, Springer Nature Switzerland, Cham, 2024, pp. 55–72.

[17] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, 2019. URL: https://arxiv.org/abs/1909.06335. arXiv:1909.06335.

[18] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, A. Talwalkar, LEAF: A benchmark for federated settings, CoRR abs/1812.01097 (2018). URL: http://arxiv.org/abs/1812.01097. arXiv:1812.01097.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: A. Singh, J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 1273–1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[20] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: I. Dhillon, D. Papailiopoulos, V. Sze (Eds.), Proceedings of Machine Learning and Systems, volume 2, 2020, pp. 429–450. URL: https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

[21] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).