IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TriOpt-Energy Aware and Secure Federated Learning in Edge-Cloud Systems

Rana Pratap [1], Uday Kumar[2], Raj Hans Singh[3], Vaishnavi Jawalkar[4], Ayushi Choudary[5]

Department of Computer Sciences[1], Vivekananda Global University, Jaipur

Department of Computer Sciences[2], Vivekananda Global University, Jaipur

Department of Hotel Management[3], Vivekananda Global University, Jaipur

Department of Forensic Sciences[4], Vivekananda Global University, Jaipur

ABSTRACT

Federated Learning allows collaborative model training without sharing raw data, making it ideal for privacy-sensitive Internet of Things (IoT) settings. However, deploying it in real-world scenarios is tough due to differences between devices, limited energy resources, and risks like model poisoning. These challenges create a difficult balance between efficiency, reliability, and model performance. This paper presents a unified orchestration framework for federated learning in a serverless edge-to-cloud environment. The framework includes two main parts. First is an energy-aware client selection mechanism called Adaptive Client Triage (ACT). It assesses devices based on energy levels, computing power, and data relevance to improve training efficiency and speed. Second is a Hybrid Security Aggregation (HSA) protocol. This combines lightweight statistical filtering at the edge with additive homomorphic encryption for safe aggregation in the cloud. This approach boosts protection against harmful updates while keeping overhead low. The system is built using a serverless architecture that spreads computation across edge and cloud resources to enhance speed and reduce costs. Experiments on the FEMNIST dataset show that the framework significantly cuts overall energy use and maintains strong model accuracy, even when facing adversarial attacks. It outperforms traditional federated learning methods.

Keywords: Federated learning, edge computing, serverless architecture, client selection, homomorphic encryption, IoT systems

INTRODUCTION

Convergence of Modern Paradigms

Modern advances in computing are being fueled by the convergence of three major paradigms. Namely, these include the internet of things (IoT), federated learning (FL), and the edge-to-cloud continuum. As a result of the emergence of these technological advancements, several innovative applications have come to fruition. For instance, applications of interest include the ability to monitor patient health information in real-time and performing predictive maintenance on industrial equipment. The idea of federated learning has become a topic of much interest, due to its potential to train machine learning models on data located at different sites, without compromising on privacy and security. Federated learning works in such

a way that there is no exchange of raw data to and from a central server, but rather the models trained on local machines only share their model weights with the central server, thus overcoming issues related to data security and ownership.In parallel, the edge-to-cloud continuum presents an opportunity to distribute work among layers of heterogeneous hardware, based on factors such as bandwidth, latency, and computing power. While this setup offers great benefits in terms of federated learning workloads, challenges arise when deploying such setups.

Core Research Problem

There are benefits associated with federated learning, yet there are also limitations when federated learning is applied in IoT and edge scenarios. First, the limitation results from the heterogeneity of the devices – from sensors to edge nodes. In federated learning, the process of choosing a client for training is usually random, and thus, it may result in energy drain on less powerful machines. Moreover, it will cause bias in the global model due to the heterogeneity of the machines that train the global model. On the other hand, even though serverless computing increases the level of scalability, its stateless feature poses issues to the iterative processes in federated learning. Second, federated learning is a technique with high levels of privacy; however, due to its decentralization, it is more susceptible to security and robustness threats. Adversarial machines might manipulate their local update process, resulting in poisoned models. Furthermore, traditional aggregation techniques like FedAvg are vulnerable to adversarial attacks.

Novel Contributions

This research offers significant advancements in the realm of federated learning, particularly within distributed and edge-cloud contexts:

Hierarchical Orchestration Architecture:

We introduce a multi-tier architecture aimed at federated learning with serverless edge-to-cloud integration. This design ensures the delegation of functions including client orchestration, intermediate aggregation, and global model management to edge and cloud tiers, leading to better scalability and reduced communication costs.

Adaptive Client Triage (ACT):

We devise a data and energy-efficient client selection algorithm that measures the fitness of clients based on an aggregated utility measure. This measure takes into account parameters like available energy, efficiency, relevancy, and past contributions, facilitating efficient client selection and faster model training.

Hybrid Security Aggregation (HSA):

We implement a two-stage approach for securing federated models from any malicious activities such as poisoned model attacks. Our hybrid approach employs statistical filtering in the edge layer with homomorphic encryption-based secure aggregation in the cloud tier.

Thorough Experimentation Evaluation:

Our framework is experimentally validated through rigorous simulations within the federated learning domain. Our experiments reveal the superiority of our framework in terms of energy consumption, convergence rate, and resistance to malicious behavior in comparison with the existing benchmark approaches.

Related Work

This section looks at earlier research related to this study, focusing on three key areas: energy-efficient federated learning, secure and reliable ways to combine data, and management methods in serverless edge-to-cloud settings.

Energy-Aware Federated Learning

In this regard, a significant part of the literature on federated learning has concentrated on developing energy-efficient FL techniques. One of the key ways used by previous researchers to tackle energy problems has been client selection. In other words, in federated learning, a subset of participants is selected for the purpose of lowering energy costs and training latency. The existing solutions for addressing this task typically consider various device-related factors including the level of charge left in the participant's battery, computational capabilities, and network connectivity, among others. As one possible solution to this problem, adaptive FL systems such as EneA-FL may be employed to ensure that only those clients who currently have sufficient energy to participate in the training process are involved. Another possible option is to cast the client selection as a joint optimization problem aimed at minimizing training time and energy consumption. In this context, reinforcement learning techniques may also be applicable to solving the given problem. However, all these solutions are mainly device-centric and fail to account for other important factors related to data properties. For example, even if the client has adequate energy as well as higher computational capabilities, it might provide low-quality updates due to skewed data distribution at the client end. These low-quality updates might result in delayed convergence and, hence, require additional training cycles. In addition, there will be higher overall energy usage as well by the entire system. Therefore, we propose our new scheme that takes both resource and data parameters into account during client selection. Through this multi-faceted approach, better client participation is ensured through improved convergence and energy conservation.

Secure and Robust Federated Learning

FL's distributed structure, despite its advantages in ensuring privacy, presents several security issues, with model poisoning emerging as one of the most critical ones. Model poisoning refers to an attack where the clients deliberately feed the model with corrupted updates with the intention of compromising the performance of the global model or implanting a backdoor. Efforts at designing defenses against such attacks have largely followed two different directions.

The first approach deals with developing reliable aggregation methods. Unlike simple averaging in FedAvg, these approaches involve statistics to detect and ignore any malicious updates. For instance, Krum and Multi-Krum methods consider either single update or a number of updates that are close enough to their neighbors according to the distance between them in the space of parameters. There are also coordinate-wise methods that calculate aggregate parameters as trimmed mean or median for each model parameter submitted by clients. For example, Trimmed-Mean and Median. FedCC algorithm is a more advanced one, where layer-wise similarity measures, such as Centred Kernel Alignment are used to detect and isolate malicious clients and even demonstrate good results in a non-IID setup. The second approach includes cryptography. Algorithms like SMPC (Secure Multi-Party Computation) or HE (Homomorphic Encryption) enable the server to calculate the total sum of clients' updates without knowing their values. FATE is an industrial-level solution for implementing secure computation schemes based on HE and MPC. Despite their great theoretical guarantees, such approaches require significant computation and communication costs, which makes them impractical for resource-limited edge devices. The conflict between the two types of defenses becomes obvious when considering their characteristics. Although statistical robust aggregators are easy to compute and fit perfectly within the edge architecture, their major problem lies in the fact that they can only operate effectively in IID settings. Being designed to eliminate the outliers from the update population, statistical robust methods run the risk of eliminating legitimate updates made by clients who happen to be the outliers due to their non-standard distribution of data on the client. Cryptographic methods, while protecting all the updates equally, tend to be too resource-consuming for frequent communication rounds from many clients. Clearly, the best way forward is not picking one solution and sticking to it but rather designing the hybrid method that would fit the physical infrastructure of the system. As we have seen above, the hierarchical nature of edge-to-cloud communications, featuring frequent client-edge communication and less frequent edge-cloud communication, offers an excellent opportunity for hybridization. This is what the Hybrid Security Aggregation (HSA) scheme from TriOpt is based on. It utilizes a cost-effective yet efficient statistical filter (trimmed-mean) at the edge side to deal

with the large number of updates from clients and uses an encryption technique that provides security while being relatively lightweight at the same time for the edge-cloud link.

Orchestration in the Serverless Edge-to-Cloud Continuum

The serverless computing paradigm is becoming increasingly popular as a solution for deploying scalable applications because of its potential for handling infrastructural abstractions and enabling pay-per-use billing. The two traits mentioned are especially interesting in the context of applications running in edge-to-cloud distributed and heterogeneous environments. However, the adoption of the serverless paradigm to such environments comes with its own set of problems that should be addressed. Specifically, there is an increased need to effectively manage resources, preserve consistency, adapt function placement, and ensure proper security of the distributed system. Current solutions attempt to extend the serverless architecture to include edge devices as components of the system. Such systems include implementations relying on technologies like Knative and others implementing optimized resource usage by means of dedicated memory and scheduling schemes. While such solutions provide increased flexibility of deployment, they still assume a specific execution model of stateless computations triggered by events. In contrast, Federated Learning (FL) workloads require the stateful execution mode, since a globally shared model needs to be trained iteratively over many rounds. Furthermore, every training round implies synchronization between the server and participating clients, which is required in order to collect updates from all of the clients. . Using such workflows on general-purpose serverless systems would result in inefficiencies. The frequent load and save operations of model states from external storage incur latency and costs, while orchestrating synchronous activities in an asynchronously designed system becomes even more challenging. In order to overcome such challenges, there is a need for a dedicated orchestration mechanism. Our framework is built specifically considering FL processes as one of its main design principles, making it easier to handle model state and orchestrate training processes.

Methodology

This framework is envisaged to be developed into a system whose components include the adaptive orchestration schemes as well as the robust security measures within the realm of serverless computing paradigm. This segment outlines the architecture of the entire system and the algorithmic operations which support energy-efficient federated learning.

System Architecture and Model

The suggested framework uses a hierarchical structure with three logical layers in line with the edge-to-cloud computing paradigm, as shown in Figure 1. Such an approach facilitates effective cooperation among dispersed clients and centralized management while meeting demands for scalability, power efficiency, and security.

Client Layer:

This layer is made up of a heterogeneous collection of IoT and edge devices, which serve as federated learning clients. The capabilities of these devices range in their computational abilities, network conditions, and energy supplies. The agents of each of these clients conduct the training of models with their local data, keep track of their metrics, and send updates of the model to their respective edge nodes.

Edge Aggregator Layer:

The intermediary layer comprises edge nodes located close to client machines, for instance, those found in the immediate local data centers or in access networks. The use of serverless computing is what powers these nodes, and each one handles a cluster of clients in its proximity. The key tasks undertaken by these intermediary nodes include organizing the participation of the clients according to the commands issued by the cloud layer, disseminating the global model among selected clients, and aggregating the same.

Cloud Orchestrator Layer:

The cloud layer acts as the main orchestrator in the federated learning system. It retains the global model between multiple training cycles and performs high-level operations. These operations involve deciding the client selection strategies based on the Adaptive Client Triage (ACT) framework, conducting secure

aggregation of the global model using the HSA framework's second phase, and assessing the model performance for convergence. The communication between the different layers proceeds through an iterative training process. The orchestrator at the cloud level starts each cycle by establishing the selection method, followed by coordination at the edge layer for local participation and filtering. Meanwhile, the clients learn from their datasets.

To formalize the system objective, the framework is modelled as a multi-objective optimization problem. Let $L(w_T)$denote the loss of the final global model w_T, E_{total}represent the cumulative energy consumption of all clients, and T_{total}denote the total training duration. The objective is defined as:

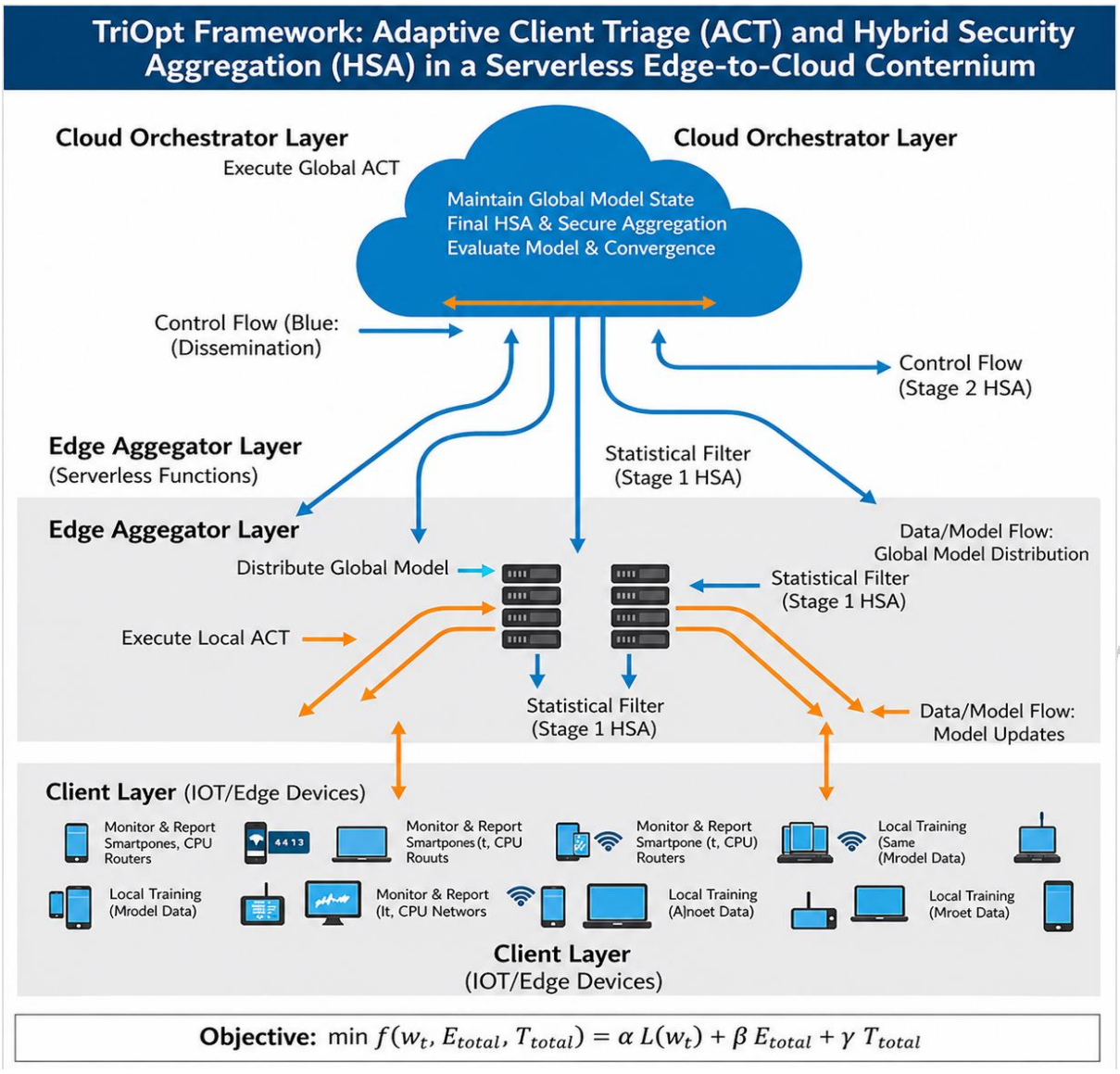

Figure 1: Architecture of the proposed TriOpt framework illustrating hierarchical orchestration across cloud, edge, and client layers. Control flow (blue) and data/model flow (orange) are depicted.

$$\boldsymbol{Objective}: \$\boldsymbol{min}_{\{S_1,S_2,\ldots,S_T\}} f(w_T, E_{\{total\}}, T_{\{total\}}) = \boldsymbol{\alpha L(w_T)} + \boldsymbol{\beta E_{\{total\}}} + \boldsymbol{\gamma T_{\{total\}}}$$

S_t: set of selected clients at round t

α, β, γ: weighting parameters

$L(w_T)$: final model loss

E_{total}: total energy consumption

T_{total}: total training time

TriOpt-E: Energy-Aware Orchestration Module

The energy efficiency of the proposed architecture lies in the TriOpt-E component, which uses the Adaptive Client Triage (ACT) algorithm. While previous strategies for choosing clients were based exclusively on hardware capabilities, the proposed method considers not only system-based parameters but also takes into account data-related information to ensure that the chosen clients can participate efficiently in model training. The overall workflow is illustrated in Figure 2.

Adaptive Client Triage (ACT) Algorithm

At the beginning of each training round t, ACT assigns a utility score $U_t(c)$to each candidate client c. This score is computed as a weighted combination of four normalized components, enabling a balanced trade-off between efficiency and learning quality:

$$(c) = we \cdot EnUtorm(c) + wt \cdot Tnorm(c) + wd \cdot Dnorm(c) + wc \cdot Cnorm(c)$$

where w_e, w_t, w_d, w_crepresent weighting coefficients that sum to 1.

The four components are defined as follows:

A. Energy Score (E_{norm}):

It relates to the amount of energy resources possessed by the client. It is calculated from the battery life left and is standardized for all machines, favoring clients who have lower chances of quitting the training process.

Time Efficiency Score (T_{norm}):

This part refers to the anticipated speed of training of the client based on its previous run-time period. The faster the client, the higher the score will be to prevent delay of slow clients.

Data Quality Score (D_{norm}):

This criterion measures the practicality of using the information collected locally by the client. It can be calculated through the comparison of the statistical resemblance between the distribution of data for the client and that of the global distribution, or the contribution towards representing minority classes.

Contribution Score (C_{norm}):

This measure determines the past effectiveness of a client's updates. This measure is obtained based on how similar the previous updates made locally are to the overall global model obtained from those updates.

The selection procedure takes place in two phases. The first phase involves the Cloud Orchestrator calculating the utilities of all possible clients and setting a threshold that depends on the training status at that moment. The threshold is then relayed to the Edge Aggregators. Every aggregator selects clients locally from among those in its cluster who fulfill the threshold condition.

This two-level selection approach allows for making decisions based on global information without losing scalability.

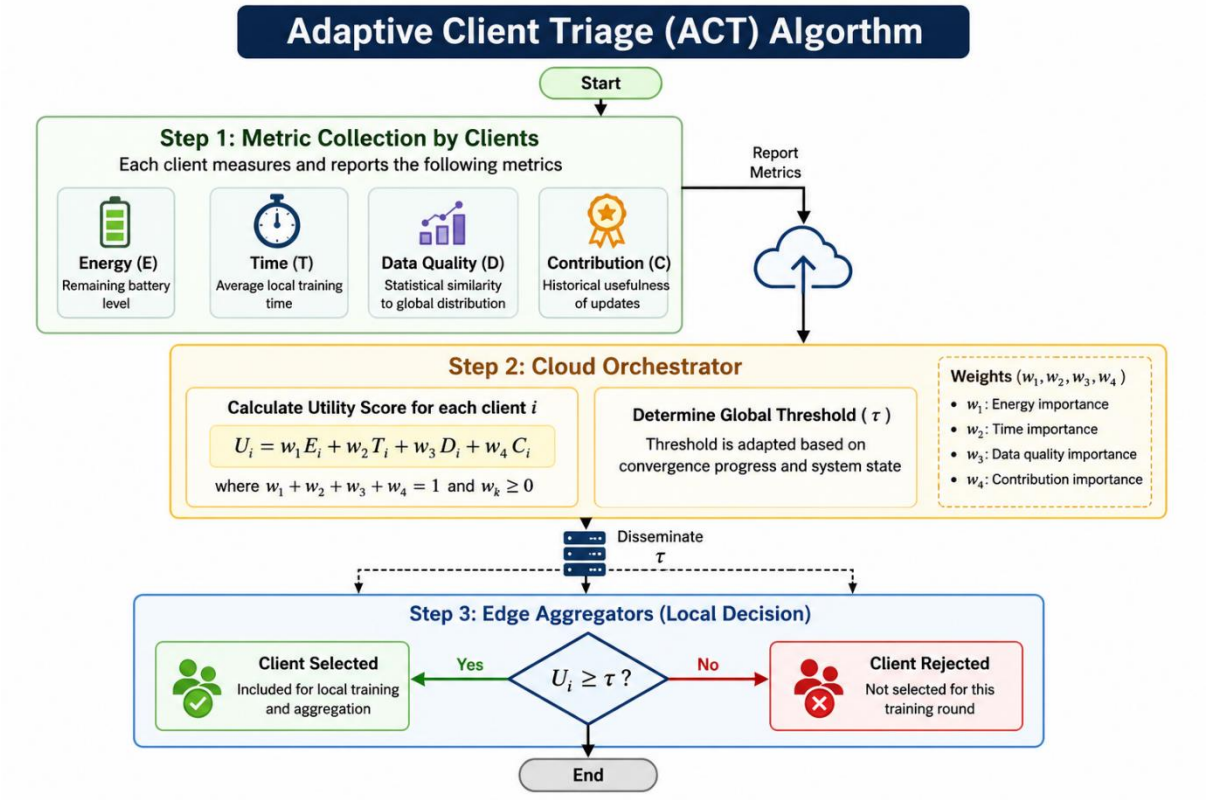

Figure 2: *Flowchart of the Adaptive Client Triage (ACT) algorithm showing metric collection, utility score computation, and threshold-based client selection.*

TriOpt-S: Secure Orchestration Module

For increasing resilience against model poisoning attacks, the TriOpt-S component adopts the Hybrid Security Aggregation (HSA) scheme. The HSA scheme is a dual-layer technique that can be used in harmony with the hierarchical edge-cloud setting, incorporating statistical filtering and cryptographic shielding without imposing any extra computational burden. The workflow of the protocol is illustrated in Figure 3.

Hybrid Security Aggregation (HSA) Protocol

Stage 1: Robust Statistical Filtering at the Edge

Once training is done locally, the selected clients will send their model updates (e.g., weight or gradient) to the respective edge aggregators. Rather than conducting regular averaging, the edge aggregator uses Coordinate-Wise Trimmed-Mean for the aggregation process. With regard to each model parameter, the received client values are sorted, and the top β and bottom β fractions of the received values are removed. Afterward, averaging is conducted on the remaining values, thus generating a filtered update value. The method is computationally efficient and useful in minimizing the effect of malicious updates.

Stage 2: Secure Aggregation via Homomorphic Encryption

The filtered updates from each of the nodes in the edge layer are then safely transmitted to the cloud layer for further processing. In order to maintain secrecy, each of the edge aggregators encrypts the updates based on the public key available from the cloud orchestrator. The cloud can perform an addition of encrypted updates due to the additive nature of homomorphic encryption and does not require decryption in this process. The final output is subsequently decrypted in the cloud using the corresponding private key.

This two-stage process ensures that HSA is efficient and secure. The edge layer filters out malicious updates, whereas the encryption ensures that the transmission between layers remains confidential. Thus, the protocol effectively counteracts poisoning attacks without causing heavy computations for clients.

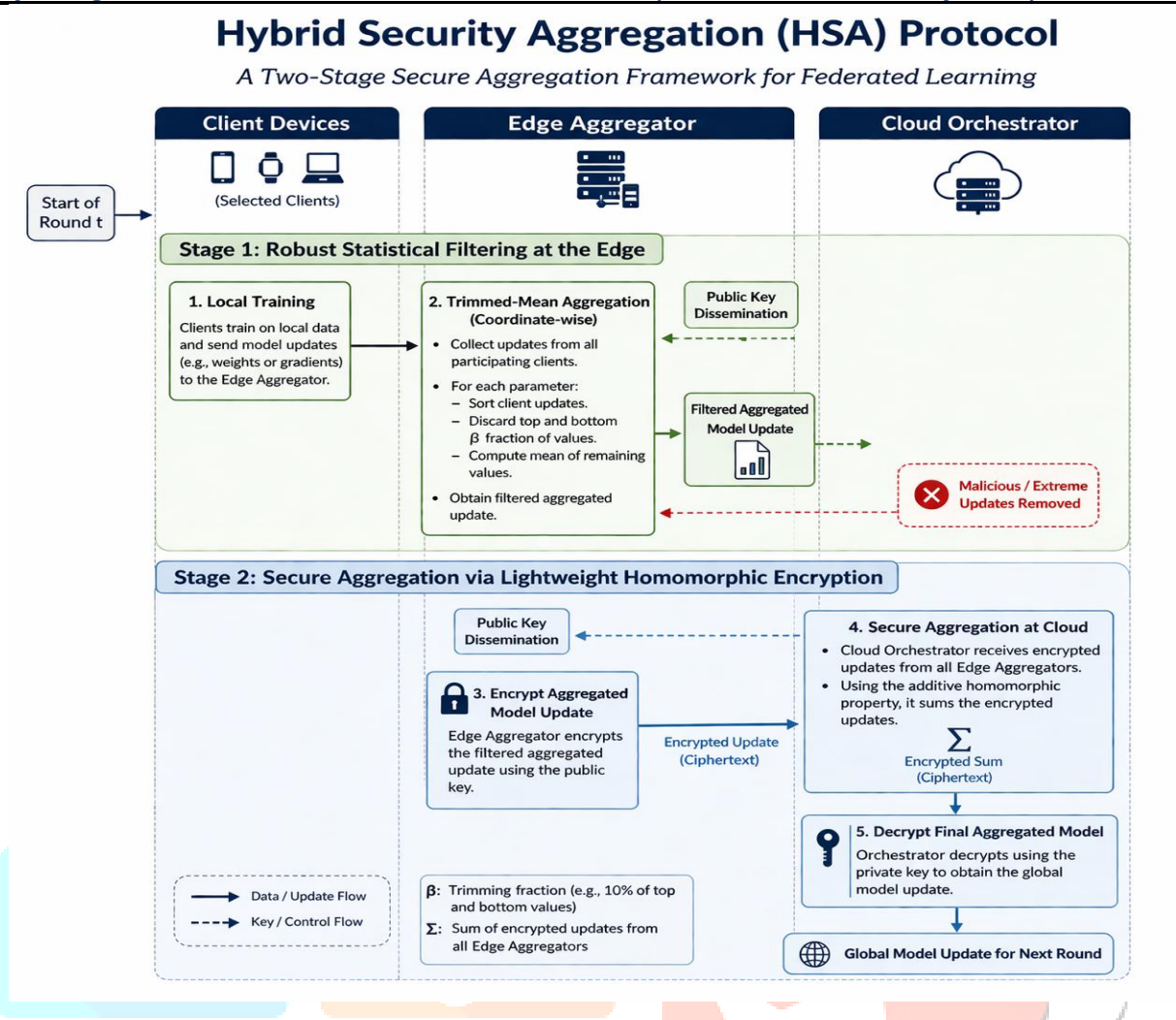

Figure 3: Sequence diagram of the HSA protocol showing edge-level filtering and secure aggregation via homomorphic encryption at the cloud.

Results

This section presents the experimental evaluation of the proposed framework, comparing its performance with established baselines under both normal and adversarial conditions. The analysis also examines the trade-off between energy efficiency, security, and model accuracy.

Performance Under Normal Conditions (No Attack)

In the benign setting without adversarial clients, the evaluation focuses on the efficiency and effectiveness of different client selection strategies.

A. Model Convergence and Accuracy:

Table I: Highlights the test accuracy growth for 50 communication rounds. Even though all techniques manage to reach a similar level of accuracy eventually, the framework converges much faster. Specifically, it manages to get to the 85% accuracy point faster compared to the other techniques. This is due to the use of the ACT technique, where clients are selected not only based on their available resources but also based on the usefulness of the data they contribute**.**

Communication Round	Standard FedAvg Accuracy (%)	EneA-FL Accuracy (%)	TriOpt Accuracy (%)
0	10	10	10
10	55	58	72
20	75	78	86

30	84	86	90
40	90	90	92
50	92.1	91.5	92.5

Table I: Test accuracy across communication rounds under normal conditions, showing faster convergence of the proposed method.

B. Energy and Time Efficiency:

Table II: Emphasizes the energy efficiency and time efficiency of the proposed models. In terms of energy efficiency, the proposed model is seen to have the minimum cumulative energy cost for 50 training iterations, which shows higher energy efficiency. Besides, it takes far less time than the other methods to achieve the required accuracy level of 85%. Though the energy-aware method is seen to consume less energy because it selects only high-energy clients, the proposed method saves more energy due to faster convergence owing to smart client selection.

Framework	Total Energy Consumption (kJ)	Time to 85% Accuracy (s)
Standard FedAvg	1250	1800
EneA-FL (inspired)	980	1650
Fed-Krum	1350	2100
TriOpt (Proposed)	750	1300

Table II: Total energy consumption (left) and time to reach 85% accuracy (right) under normal conditions. TriOpt demonstrates superior efficiency in both metrics.

C. Robustness Against Model Poisoning Attacks:

In the second set of experiments, a label-flipping attack was created by marking 20% of the clients as malicious to test the robustness of the framework in terms of security.

Performance Under Attack of Models: Table III. Demonstrates the effect of the attack on model accuracy. The approaches lacking sufficient defense mechanisms, such as FedAvg and EneA-FL, show a drastic decrease in their performance, being unable to converge and reaching only 55% accuracy because of the effect of poisoned data samples.

Fed-Krum increases robustness by eliminating the influence of malicious clients but still shows worse results than the proposed framework and performs slightly worse than in the benign case because of eliminating valid data samples in non-IID settings.

On the other hand, the proposed framework is highly resilient to the attack because of employing the HSA protocol and minimizing the effect of poisoned data samples, thus converging steadily and preserving almost full accuracy.

Communication Round	Standard FedAvg Accuracy (%)	EneA-FL Accuracy (%)	Fed-Krum Accuracy (%)	TriOpt Accuracy (%)
0	10.0	10.0	10.0	10.0
10	45.1	44.8	55.2	60.5
20	50.3	49.7	75.8	80.1
30	54.9	54.2	82.3	88.4
40	55.1	55.0	85.1	91.2
50	55.3	54.8	86.5	91.8

Table III: Test accuracy versus communication rounds in the presence of a 20% model poisoning attack. TriOpt's HSA protocol allows it to maintain high accuracy, while unprotected baselines fail.

Attack Success Rate: Table IV: Presents the Attack Success Rate (ASR) as a measure of security robustness. FedAvg and EneA-FL exhibit a rapid increase in ASR, approaching 90%, indicating a high rate of misclassification due to poisoned updates. In contrast, Fed-Krum reduces the attack impact by filtering malicious contributions, resulting in a significantly lower ASR. The proposed framework achieves the lowest ASR, demonstrating the effectiveness of the HSA protocol in mitigating adversarial updates through edge-level filtering and secure aggregation.

Communication Round	FedAvg ASR (%)	EneA-FL ASR (%)	Fed-Krum ASR (%)	TriOpt ASR (%)
0	0.0	0.0	0.0	0.0
10	80.5	81.2	10.1	2.0
20	85.3	86.0	11.2	2.1
30	87.1	88.3	12.0	2.2
40	88.0	88.9	12.3	2.2
50	88.2	89.1	12.5	2.3

Table IV: Attack Success Rate (ASR) versus communication rounds. TriOpt achieves the lowest ASR, indicating the most effective defense against the poisoning attack.

D. Analysis of the Energy-Security-Accuracy Trade-off.

From the findings, the hybrid approach appears to have successfully achieved balanced optimization of multiple objectives instead of optimizing one after another. For instance, the implementation of ACT has made it possible for the convergence process to be faster owing to its focus on ensuring quality of clients participating in the process; hence fewer communication rounds will be needed. As a result, less energy and time will be used during the training process despite the extra costs brought about by security mechanisms. The overhead involved in the application of the secure module is bearable.

For instance, the use of the trimmed-mean operation in the edge does not involve high costs in terms of computations. Also, homomorphic encryption is used only during the aggregation process of the edge. In contrast, Fed-Krum uses more energy during computations as it uses pairwise distances

Conclusion and Future Work

Summary of Findings

The current research solves a number of problems associated with implementing federated learning in serverless edge-to-cloud architectures, which include heterogeneity in energy usage and potential threats posed by security issues. Specifically, an integrated architecture is developed that ensures the optimization of energy savings, model robustness, and high performance using the multi-layer design approach, including client selection and security measures during model aggregation. Our experimental analysis performed based on the FEMNIST data indicates that the developed system not only saves energy but also decreases the duration of training, ensuring that the final performance level is comparable to the existing benchmarks. Furthermore, our model has been proven resilient to the poison model attacks, significantly outperforming competing approaches

Broader Impact

There are numerous implications for the future of distributed machine learning associated with the outcomes of this study. In fact, by offering a realistic and efficient way to implement federated learning in practice, TriOpt provides a solid example of how theoretical advancements can be used in practical settings. Moreover, the possibility to perform model training securely and efficiently using large fleets of heterogeneous and limited devices will enable faster progress in sectors that require advanced technologies, such as decentralized healthcare systems, smart transportation networks, and industrial IoT.

Future Research Directions

This paper explored some of the main issues faced in implementing federated learning in serverless edge-to-cloud settings, including energy variability and susceptibility to model poisoning attacks. Our approach involved formulating an orchestration framework that combines adaptive client selection and secure aggregation in a hierarchical structure. By incorporating the ACT and HSA schemes, our method ensures efficient, resilient, and scalable federated learning in distributed networks. The results from experiments conducted using the FEMNIST dataset show that the framework is able to save energy and reduce training times without compromising on model accuracy. It also exhibits resilience to attack attempts, surpassing both conventional and energy-efficient baselines, as well as traditional robust aggregation techniques. These findings underscore the benefits of optimizing energy efficiency and security together. Such methods could enable massive-scale privacy-preserving applications in fields like healthcare, transportation, and industry IoT. Future research would involve improving the architecture in several areas. The integration of more advanced privacy techniques like differential privacy could better protect the application from inference attacks. Flexible federated learning topologies that include decentralized and hierarchical architectures could be explored for better scalability. Dynamic allocation of resources to serverless functions could also prove beneficial for efficiency. Moreover, evaluating the architecture on actual hardware platforms could lead to additional insight.

References:

[1] B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Proc. AISTATS, 2017.

[2] S. Li, A. Kadav, I. Durme, and C. Ré, “Federated Optimization in Heterogeneous Networks,” Proc. MLSys, 2020.

[3] “Federated Learning,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Federated_learning

[4] K. Bonawitz et al., “Practical Secure Aggregation for Privacy-Preserving Machine Learning,” Proc. ACM CCS, 2017.

[5] P. Blanchard et al., “Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent,” Proc. NeurIPS, 2017.

[6] Y. Zhao et al., “Federated Learning with Non-IID Data: A Survey,” ACM Comput. Surveys, 2023.

[7] “Flower: A Friendly Federated AI Framework.” [Online]. Available: https://flower.ai/

[8] “FEMNIST Dataset,” Hugging Face. [Online]. Available: https://huggingface.co/datasets/flwrlabs/femnist

[9] F. Sattler et al., “Robust and Communication-Efficient Federated Learning,” IEEE Trans. Neural Netw. Learn. Syst., 2020.

[10] H. Xu, J. Li, and Y. Hu, “Resource-Aware Client Selection for Efficient Federated Learning,” IEEE Trans. Mobile Comput., 2022.

[11] L. Zeng et al., “Energy-Efficient Federated Learning for Mobile Edge Devices,” IEEE Internet Things J., 2021.

[12] A. Fang et al., “Local Model Poisoning Attacks to Byzantine-Robust Federated Learning,” Proc. USENIX Security, 2020.

[13] C. Zhang et al., “BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning,” Proc. USENIX ATC, 2020.

[14] “Homomorphic Encryption,” MPCVault Docs. [Online]. Available: https://docs.mpcvault.com/blog/homomorphic-encryption/

[15] “Serverless Computing in the Edge–Cloud Continuum,” ResearchGate. [Online].

[16] “EneA-FL: Energy-Aware Orchestration for Serverless Federated Learning,” IRIS/Unibo.

[17] “A Survey on Energy-Efficient Design for Federated Learning,” MDPI Energies, 2025.

[18] “A Survey on IoT-Edge-Cloud Continuum Systems,” MDPI, 2023.

[19] “Serverless Workflow Management on the Computing Continuum,” SPEC, 2024.

[20] “Federated Learning for Edge Computing: A Survey,” MDPI Applied Sciences, 2022.