STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

A Low-Latency Secure Federated Learning Framework for IoT Environments Using VPN Mesh and Homomorphic Encryption over Decentralized SDS

Saad Mahmoudi [1,*], Mohamed Amnai [1], Tarik El Moudden [2]

[1] *Computer Science Research Laboratory, Faculty of Science, Ibn Tofail University, Kenitra, Morocco*
[2] *Interdisciplinary Laboratory of Sports Sciences, Institute of Sports Professions, Ibn Tofail University, Kenitra, Morocco*

Abstract
With the rapid growth of intelligent Internet of Things (IoT) applications and the increasing sophistication of cyberattacks, ensuring secure and efficient collaborative learning has become a critical challenge in Federated Learning environments. Adversarial attacks can threaten data confidentiality, disrupt the model training process, and reduce the reliability of collaboratively trained models, particularly in heterogeneous and resource-constrained IoT environments. In order to overcome these limitations, this paper proposes the Secure Federated Learning over Virtualized Hybrid Decentralized Storage (SFL-VHDS) architecture together with the Privacy-Preserving SDS Secure Federated Aggregation Algorithm (PPS-SFAA). The proposed architecture integrates Software-Defined Storage (SDS), a WireGuard VPN Mesh, and OpenFHE-based homomorphic encryption to provide authenticated communication, privacy-preserving model-update exchange, decentralized storage, and secure aggregation while maintaining low communication latency. Experimental evaluation is conducted using the NGSIM dataset under a non-IID data distribution generated through a Dirichlet partitioning strategy with a concentration parameter of $\alpha = 0.5$. The proposed framework is evaluated under representative threat scenarios, including data poisoning, model inversion, and communication-layer attacks. Experimental results show that the proposed SFL-VHDS architecture achieves a global model accuracy of 98.58% with an end-to-end latency of 31.04 s while maintaining high communication integrity and system resilience.

Keywords Federated Learning; Federated Edge Learning; Decentralized Architecture; Software Defined Storage; VPN

DOI: 10.19139/soic-2310-5070-4144

1. Introduction

The proliferation of IoT technology has resulted in the production of huge amounts of distributed data, thus making centralized machine learning challenging due to communication costs, privacy issues, and the limited computing and memory resources of edge devices. Federated Learning (FL) has emerged as a distributed learning architecture that allows various devices to train a common model without having to transfer any raw data between them. Keeping the raw data on the edge device improves privacy and minimizes the cost of communication [1]; however, securely exchanging model updates between distributed IoT devices is not an easy task.

Due to technological advancements, cloud and edge computing frameworks offer extra computing power and memory, as well as specialized hardware devices such as CPUs, GPUs, and TPUs, for FL. Nevertheless, a number of problems have emerged from the application of such frameworks. These challenges include

*Correspondence to: Saad Mahmoudi (Email: saad.mahmoudi@uit.ac.ma). Computer Science Research Laboratory, Faculty of Science, Ibn Tofail University, Kenitra, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 202x International Academic Press

high communication costs, reliance on centralized or semi-centralized frameworks, and a large volume of shared model updates that can negatively impact the training process [3]. Furthermore, FL frameworks are vulnerable to several kinds of attacks [2].

Latency issues associated with communication have been addressed in recent works utilizing fog computing and edge-assisted FL, where the computational process occurs close to the IoT devices [4, 5]. Although these methods help improve communication efficiency, a holistic approach that incorporates secure communication, privacy preservation for model updates, and decentralized storage management and aggregation has not yet been proposed . Therefore, there is a need for a secure FL framework that encompasses all these aspects.

To overcome the aforementioned limitations, the Secure FL over Virtualized Hybrid Decentralized Storage (SFL-VHDS) framework, along with the Privacy-Preserving SDS Secure Federated Aggregation Algorithm (PPS-SFAA), is proposed. The SFL-VHDS framework combines a decentralized Software-Defined Storage (SDS) system, a WireGuard VPN mesh, and homomorphic encryption using OpenFHE to provide authenticated communication, privacy-preserving model update exchanges, decentralized storage, and secure aggregation with low communication latency. Moreover, the decentralized SDS architecture provides scalable storage and resource management, making it well-suited for future FL systems involving a large number of IoT devices.

The proposed framework is validated using the NGSIM dataset under a realistic non-IID data distribution generated through a Dirichlet partitioning strategy with a concentration parameter of $\alpha = 0.5$. The experimental evaluation considers representative threat scenarios, including data poisoning, model inversion, and communication-layer attacks. Performance is assessed in terms of communication latency, global model accuracy, communication integrity, and system resilience. In particular, communication integrity measures the reliability of encrypted model-update transmission over the WireGuard VPN Mesh, whereas resilience evaluates the ability of the framework to sustain secure FL operations under adverse conditions. Comparative experiments against representative secure FL architectures further demonstrate the effectiveness of the proposed approach under identical experimental settings.

The main contributions of this paper are summarized as follows:

- We propose the Secure FL over Virtualized Hybrid Decentralized Storage (SFL-VHDS) framework, which integrates decentralized Software-Defined Storage (SDS), a WireGuard VPN Mesh, OpenFHE homomorphic encryption, and the proposed PPS-SFAA secure aggregation workflow to provide authenticated communication, decentralized storage, privacy-preserving model-update exchange, and secure collaborative learning.
- We present the PPS-SFAA secure federated aggregation algorithm, which coordinates client authentication, secure communication establishment, encrypted model-update exchange, integrity verification, and secure aggregation within the proposed SFL-VHDS architecture.
- We conduct a comprehensive experimental evaluation using the NGSIM dataset under a realistic non-IID setting generated with a Dirichlet concentration parameter of $\alpha = 0.5$. The proposed framework is evaluated under representative threat scenarios, including data poisoning, model inversion, and communication-layer attacks, and compared with representative end-to-end secure FL architectures. The experimental results demonstrate high global model accuracy, low communication latency, reliable encrypted model-update transmission, and strong operational resilience.

2. Related work

Recent studies highlight the growing importance of secure, intelligent infrastructures for distributed Internet of Things (IoT) systems. Tawfik et al. in [6] provide a systematic review of the application of LLMs to IoT security and discuss the issues associated with intelligent and trustworthy IoT systems. On the other hand, War et al. [7] conduct a review of FL models for cyber-physical system security and

present issues concerning privacy preservation, communication efficiency, scalability, and collaborative learning. Moreover, Sharma et al. [8] propose a blockchain-based secure communication infrastructure for cloud environments and show that encrypted communication and distributed data management may improve data confidentiality and integrity. In this regard, based on these recent studies, this section presents a review of the FL architectures most relevant to the SFL-VHDS framework.

2.1. Deployment of FL in a Centralized VPN-Based Architecture (FL-CVPN)

In this architecture, FL is deployed and centrally controlled, where updated model parameters are exchanged between IoT devices and the aggregation server. Each IoT device performs local training of the model using its own data, thereby preserving data privacy at the device level [9, 10]. Once local training is completed, the updated model parameters are transmitted to the aggregation server through a secure tunnel created by a Virtual Private Network (VPN) server. This communication process relies on a virtual private network that ensures optimized and secure decentralized learning among the participating IoT devices while maintaining data integrity and reliability during the training phase of the federated model [11].

In addition, the VPN server is designed to verify the identity of each IoT device to ensure authenticity and proper access rights to the training process, whether over a local network or a wide-area network (WAN) across a smart city. This verification mechanism, while essential for security, introduces additional overhead, leading to increased network load during the training phase and higher latency between IoT devices and the aggregation server [12]. For model aggregation, the aggregation server employs the ProxAvg algorithm to combine the updated model parameters received from all participating IoT devices. At each communication round, corresponding to the defined number of training epochs initialized by the global model, the aggregation server redistributes the updated global model to the IoT devices iteratively until the training process is completed [12].

2.2. Implementation of FL in a Software-Defined Networking-Based Architecture (FL-SDN)

In this architecture, the FL model is optimized in terms of latency using software that centrally controls and manages network traffic flow. This centralized management governs how updated model parameters are transmitted to the ag-gregation server. The process occurs directly between the aggregation server and the end devices participating in the AI model training phase, according to a pre-defined number of epochs set by the global model [13].

The global model is hosted within the SDN controller, specifically in the control plane, which serves as the core component of the SDN architecture. This control plane is responsible for making intelligent decisions about how network traffic flows across the system, with the objective of optimizing performance, minimizing latency, and avoiding network congestion or excessive bandwidth consumption [14, 15].

The learning phase starts by distributing the global model through the SDN controller to the IoT devices. After that, the IoT devices train the model on their own datasets according to the FL approach. This process enhances data security by keeping sensitive information on each device, reducing the risk of data leakage and inference attacks [16]. The SDN architecture performs intelligent route selection to optimize the transfer of updates from the IoT devices to the aggregation server. The latter employs the FedProx algorithm to aggregate the learning parameters generated by the trained models [17].

2.3. SSL-FL Authentication Framework (SSL-FL-AF)

This framework provides secure communication between participating IoT devices and the aggregation server while reducing communication costs and computational resource consumption during the authentication phase. In traditional FL, each participant trains the model locally on private data and shares only the updated model parameters. However, this training process is constrained by the limited

resources of IoT devices. For this reason, the lightweight authentication mechanism proposed in SSL-FL enables the training process to operate efficiently in constrained environments [18].

The Framework is based on an authentication mechanism using PUF (Physical-ly Unclonable Function), which relies on unique device characteristics generated during chip manufacturing. This mechanism enables device identification with-out storing encryption keys inside the device, thereby reducing the risks of key theft and impersonation attacks. In this Framework, each device must authenticate with the authentication server using lightweight cryptographic operations before being authorized to participate in the FL training phase [19]. The Framework also improves AI model training in a decentralized architecture while reducing latency, while considering the limited computational resources of IoT devices and network bandwidth constraints [19].

2.4. Privacy Preserving Verifiable FL (PP-VFL)

The PPVFL Framework is designed to ensure an efficient security mechanism through a decentralized architecture combining blockchain technology, verifia-ble aggregation, and homomorphic encryption. This approach enables participating devices to encrypt their machine learning updates before transmitting them to the aggregation server using homomorphic encryption, thereby protecting them against Data Poisoning, Model Inversion, and Integrity Attacks, and other malicious threats while preserving data privacy [20].

To further enhance security and integrity during the training phase, the Framework incorporates a verifiable aggregation mechanism that certifies the updates shared by IoT devices before the aggregation process. This helps protect the global model from malicious devices that may attempt to alter updates transmitted over the network during training. To ensure transparency, traceability, and confidentiality in the exchange of learning updates and aggregation operations, the Framework relies on a blockchain-based decentralized architecture [20, 21]. All training processes are recorded on the blockchain ledger, ensuring that both updates and the aggregated model remain immutable throughout the learning process. The combination of these technologies and security mechanisms improves the efficiency, correctness, and trustworthiness of the resulting system [21].

Even though these architectures provide improvements in secure communication, authentication, privacy protection, and decentralized communication systems, none of them integrates Software-Defined Storage (SDS), a WireGuard VPN Mesh, OpenFHE-based homomorphic encryption, and secure aggregation to provide a decentralized architecture that ensures both data privacy and low communication latency. This limitation motivated the development of the proposed framework and algorithm presented in this paper.

3. Method and Materials

In FL architectures deployed over cloud, fog, or edge computing environments, IoT devices still face two major challenges: the security of data and updated learning parameters, and the latency generated during the exchange of these parameters within the ecosystem managing AI models of participating IoT devices [22, 23]. These challenges mainly arise because the sharing of model parameters relies on communication over wide-area networks (WANs), which may introduce network congestion, delays, and potential security vulnerabilities.

The increasing number of IoT devices connected over a network may further complicate the training phase due to several issues, including communication overload, cyberattacks aimed at altering learning parameters or making the aggregation server unavailable, and the possibility of tampering with model parameters, thereby compromising the integrity and reliability of the AI model [24, 25].

To address these challenges, an approach based on a decentralized SDS architecture is proposed, as depicted in Fig. 1. This architecture centralizes the updated model parameters within a virtual storage space provided by SDS before they become accessible to the aggregation server through the SDS controller

API. SDS refers to a storage architecture in which the storage management layer is decoupled from the underlying physical storage hardware and is controlled through software-based policies [26]. The proposed architecture supports collaborative learning through the secure sharing of parameters in a controlled environment, where all components are interconnected through VPN tunnels based on a mesh topology. This connection mechanism links IoT devices and the aggregation server through the SDS architecture.

The Framework SFL-VHDS deploys a proposed algorithm that handles learning updates through a virtual storage space created by SDS, which is intended to softwarize storage resources, including physical disks, cloud storage, and solid-state drives (SSDs). SDS separates the software layer responsible for managing read and write operations from the underlying storage hardware. In other words, a centralized software system manages and controls multiple storage devices, which may belong to different IoT devices, and presents them as a single virtual storage space shared among participating devices. This storage process enables efficient collaboration during the training phase of models while reducing latency.

After securely sharing the updates generated by IoT devices or the aggregation server, the aggregation server accesses the shared storage space to retrieve the updated parameters generated during the training phase by IoT devices. The parameters are then aggregated using FL algorithms such as FedProx [27]. Therefore, the proposed approach reduces network overhead and avoids frequent parameter exchanges between IoT devices and the aggregation server, while enforcing strict access control to the shared storage space, guaranteeing that only authorized IoT devices have access to specific data or learning parameters by verifying the identity of each device attempting to read or upload model updates.

In the proposed system, IoT devices may send their updated parameters either directly to the SDS infrastructure or through an intermediate gateway connected to the SDS platform. To ensure secure communication, we implement two architectural variants, both designed to enhance and secure the training phase of AI models. These architectures rely on a VPN to establish encrypted tunnels between IoT devices and the SDS platform. This secure communication mechanism restricts participation to authorized devices and protects the training process against intrusion attempts and man-in-the-middle attacks that could intercept or manipulate the exchanged parameters. Such attacks could otherwise exploit device identities to infer sensitive information or bias the AI model during the training phase.

3.1. Detailed Description and Composition of the SFL-VHDS Framework

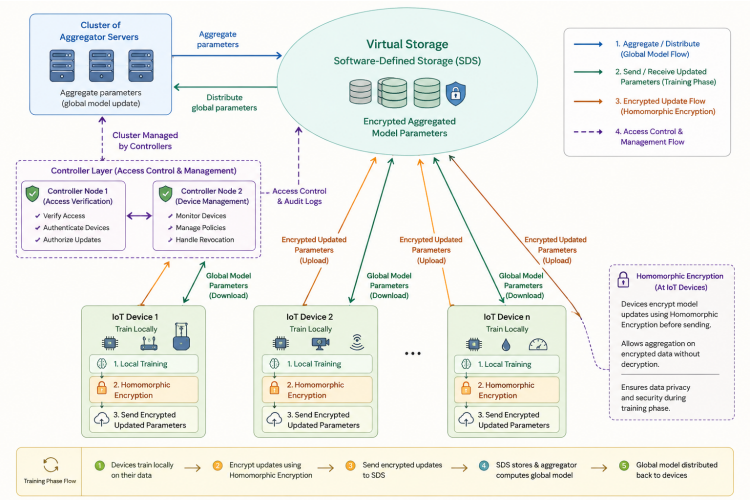


Figure 1. Architecture of the proposed Framework for securing and controlling AI model updates using a decentralized SDS architecture connected through a VPN mesh topology

In this architecture, we use an SDS controller that hosts the software responsible for managing and creating a virtual storage space from the storage resources available on IoT devices, regardless of the hardware they use. This process also addresses another challenge that can degrade the learning performance of AI models, namely hardware heterogeneity. For this reason, the SDS approach addresses this issue by abstracting the physical storage layer, independent of the storage hardware type, and creating a virtualized storage space in which all participating IoT devices can upload their updated AI model parameters and make them accessible to other authorized devices.

The accessibility to this shared storage environment is strictly controlled by the SDS system through access control policies and security rules used to verify the identity of each device wishing to contribute to the training phase of the AI model. These rules rely on several parameters, including IP addresses, communication ports, and the hardware used by the devices. The architecture of depolyment of the proposed Framework, illustrated in Fig. 1, is organized into four main layers, which are described below.

- **Access Layer:** In this layer, IoT devices are connected to the network through the IoT gateway to access the virtual storage space provided by SDS. The devices then start the local training of AI models using their private datasets and encrypt the generated updates using homomorphic encryption while sharing only the encrypted updated parameters.
- **Connection Layer:** This layer represents the equipment that enables IoT devices to connect to the network and ensures secure traffic communication between all components. Secure communication is established through VPN tunnels based on a mesh topology. This VPN provides effective and secure communication even when IoT devices are deployed in different smart city environments.
- **SDS Layer:** This layer is responsible for managing the storage of updated model parameters while reducing communication costs and network traffic overload, and ensuring the availability of parameters to support a reliable training phase of AI models. This layer includes an SDS controller acting as an intermediate control layer between IoT devices and the aggregation server, enforcing an additional verification phase to authenticate participating IoT devices and enabling efficient communication between the virtual storage environment and participating devices during the training process.
- **Aggregation Layer:** In this layer, the aggregation process of all updates stored in the virtual SDS space is performed using the FedProx algorithm, leveraging the high computational resources of the edge cloud while ensuring scalability when the volume of updates increases.

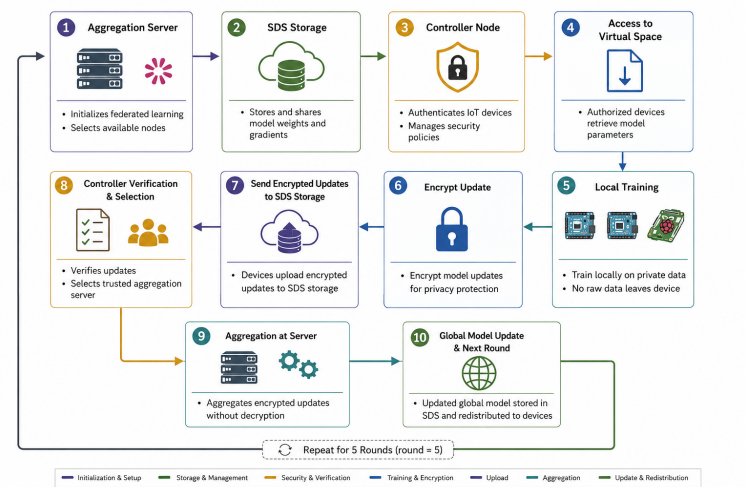


Figure 2. Workflow of SFL-VHDS Framework for privacy-preserving IoT traffic classification.

3.2. Workflow of SFL-VHDS Framework for privacy-preserving IoT traffic classification.

The workflow of the SFL-VHDS Framework implementing the PPS-SFAA algorithm, as depicted in Fig. 2, illustrates the communication flow among all components, ensuring the security and integrity of the updated model in a FL environment while accelerating model training.

3.3. PPS-SFAA: Privacy-Preserving SDS Secure Federated Aggregation Algorithm

Algorithm 1 PPS-SFAA: Privacy-Preserving SDS Secure Federated Aggregation Algorithm

Require: Number of communication rounds R, IoT clients N, NGSIM dataset D (10,000 samples), Dirichlet concentration parameter $\alpha = 0.5$, CNN model implemented in PyTorch, batch size B, Flower FL framework, OpenFHE homomorphic encryption, SDS storage layer, WireGuard VPN Mesh, SDS controller

Ensure: Trained global CNN model M_G

Initialize the Flower aggregation server
Initialize the global CNN model M_G
Partition the NGSIM dataset D among the N IoT clients using a Dirichlet distribution with concentration parameter $\alpha = 0.5$ to generate heterogeneous non-IID local datasets $\{D_1, D_2, \ldots, D_N\}$
Deploy the initial global CNN model into the SDS virtual storage
Initialize the SDS controller for authentication, authorization, and secure session management
for $r = 1$ to R **do**
    Discover newly connected IoT clients
    Authenticate each client through the SDS controller
    Establish WireGuard VPN Mesh tunnels with authenticated clients
    Distribute the current global CNN model M_G to all authenticated clients
    for each authenticated IoT client C_i **do**
        Load the local non-IID dataset partition D_i
        Train the local CNN model using PyTorch
        Compute the local CNN model update ΔM_i
        Encode ΔM_i using OpenFHE
        Encrypt ΔM_i
        Upload the encrypted model update to the SDS virtual storage through the WireGuard VPN Mesh
    end for
    SDS controller verifies:
        • Client authentication
        • Access permissions
        • Integrity of encrypted model updates
    Retrieve validated encrypted model updates from the SDS layer
    Perform secure homomorphic aggregation on the encrypted CNN model updates
    Decrypt the aggregated global model update
    Update the global CNN model M_G
    Store the updated global CNN model in the SDS virtual storage
end for
return Trained global CNN model M_G

4. Experimental Methodology

4.1. Experimental Methodology and System Configuration

To implement the proposed SFL-VHDS architecture, we create a distributed IoT environment using a decentralized storage system for simulation. The WireGuard VPN Mesh establishes authenticated peer-to-peer encrypted connections among all participants in the FL process, including the SDS controller, IoT clients, and the edge aggregation server. This design enables the confidential transmission of model updates while minimizing communication latency. The Flower FL framework is utilized to manage distributed training across multiple clients. Each client performs local training on its assigned portion of the NGSIM dataset using a CNN implemented in the PyTorch framework. At the end of each FL round, the CNN model updates are serialized and encrypted using the OpenFHE-Python library based on the CKKS homomorphic encryption scheme before being transmitted through the WireGuard VPN Mesh to the aggregation server for global model aggregation. Thus, the training data remain stored on each client and are never transferred during the FL process.

To implement the decentralized storage scheme in the proposed framework, Ceph is used as the SDS, and a secure Virtual Private Network (VPN) mesh is established using WireGuard among all participating nodes. This facilitates secure peer-to-peer communication between clients and the aggregation server, enabling model updates to be exchanged in encrypted form. Furthermore, model parameters are serialized and homomorphically encrypted before transmission.

The experimental setup is implemented on a virtualized platform using Proxmox VE. Two virtual machines are deployed: one for the Ceph SDS services and the other for the FL setup, with a virtual switch enabling networking among all involved components. This configuration eliminates network variability and enables an accurate evaluation of the latency introduced by the FL setup.

The hardware platform comprises a laptop running Proxmox VE virtualization technology and a desktop machine hosting the decentralized storage environment. While the laptop is equipped with an Intel Core i5-8250U processor (1.60 GHz, 8 cores) and an NVIDIA GeForce GTX 1650 Ti GPU with 16 GB of VRAM, the storage server is equipped with an Intel Core i7-8250U processor (1.60 GHz, 8 cores) and an NVIDIA GeForce GTX 1650 Ti GPU with 16 GB of VRAM. The decentralized storage environment includes two internal 250 GB SSDs and two 1 TB HDDs to deploy the Ceph SDS cluster for hosting and securely storing encrypted FL model updates. The IoT environment includes two Arduino Uno devices for sensing and collecting data, as well as a Raspberry Pi acting as an edge computing device to train the CNN locally and participate securely in FL.

4.2. Implementation Parameters

Table 1 summarizes the principal implementation parameters and software components used to deploy and evaluate the proposed SFL-VHDS framework. It complements the experimental methodology by providing the key configuration details required to facilitate reproducibility.

4.3. Evaluation Metrics Definition

For evaluating the performance of the proposed PPS-SFAA algorithm in terms of performance, security metrics, and integrity during the training phase, we embedded it in a decentralized SDS architecture and a VPN mesh.

4.3.1. Communication Integrity Rate Communication integrity measures the reliability of encrypted model-update transmission within the proposed FL communication infrastructure. In the SFL-VHDS framework, encrypted CNN model updates are exchanged between IoT clients, the SDS controller, and the aggregation server through the WireGuard VPN Mesh. Therefore, communication integrity is evaluated at the network communication layer by quantifying the ratio between successfully received TCP control

Table 1. Implementation Parameters of the Proposed SFL-VHDS Framework

Parameter	**Configuration**
Programming language	Python
FL framework	Flower
Deep learning framework	PyTorch
Local learning model	Convolutional Neural Network (CNN)
Optimizer	Adam
Communication rounds	20
Batch size	1000
Dataset partitioning	Dirichlet distribution ($\alpha = 0.5$) (Non-IID)
Homomorphic encryption library	OpenFHE-Python
Homomorphic encryption scheme	CKKS
Secure aggregation algorithm	PPS-SFAA
Decentralized storage platform	Ceph Software-Defined Storage (SDS)
Secure communication protocol	WireGuard VPN Mesh
Virtualization platform	Proxmox VE
Network simulator	NS-3

flags and the total number of transmitted TCP control flags during the secure model-update exchange process.

$$\text{Communication Integrity}(\%) = \frac{\text{TCP Flags Received}}{\text{TCP Flags Sent}} \times 100 \tag{1}$$

4.3.2. Accuracy Rate Accuracy measures the predictive performance of the global FL model after the completion of the training process. It is defined as the ratio between the number of correctly classified samples and the total number of evaluated samples.

$$\text{Accuracy}(\%) = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \times 100 \tag{2}$$

4.3.3. Resilience Rate Resilience evaluates the capability of the proposed SFL-VHDS framework to maintain continuous and secure FL operations under abnormal conditions, including communication disruptions and security threats. It is quantified according to the proportion of successfully completed secure training rounds compared with the total number of executed training rounds.

$$\text{Resilience}(\%) = \frac{\text{Successful Secure Training Rounds}}{\text{Total Training Rounds}} \times 100 \tag{3}$$

4.4. Dataset and Task

To assess the effectiveness of the SFL-VHDS framework, we employ the NGSIM dataset [28], which consists of a substantial amount of real-world freeway traffic trajectories. The decision to use the NGSIM dataset for evaluation is based on the consideration that connected vehicular networks can be viewed as one of the most challenging IoT scenarios, since they demand secure collaboration along with efficient communication among geographically dispersed vehicles for collaborative learning. Thus, the NGSIM dataset serves as a benchmark for measuring the communication efficiency, security, and collaborative learning performance of the SFL-VHDS framework. Before federated partitioning, the dataset is preprocessed to remove missing values and normalize the input features.

Since the SFL-VHDS framework considers communication and security for FL in general rather than for a particular application domain, the NGSIM dataset is considered a representative workload for evaluating model update exchange, decentralized model aggregation, communication latency, and learning efficiency in a realistic environment. In order to establish a realistic FL setting, the NGSIM vehicular dataset is divided among the participating clients according to a Dirichlet distribution with a concentration parameter of $\alpha = 0.5$ to generate non-IID local datasets. The proposed dataset partitioning approach introduces statistical heterogeneity among clients and reflects the characteristics of vehicular networks. A fixed Dirichlet concentration parameter of $\alpha = 0.5$ is used for all experiments.

The experiments are carried out using 10,000 NGSIM samples over 20 communication rounds, where in each client uses the PyTorch framework to train a CNN locally with a batch size of 1,000. The classification task involves four driving behavior classes: *Free-flow Driving*, which is characterized by fast-moving vehicles with minimal interaction; *Congested Driving*, which refers to congested traffic conditions with reduced vehicle speed; *Lane Keeping*, which indicates a stable vehicle trajectory within a single lane; and *Car Following*, in which the vehicle continuously adjusts its motion according to the preceding vehicle.

The CNN global model is trained collaboratively using the proposed PPS-SFAA method, which manages the entire FL process and ensures the secure transmission of model updates through OpenFHE homomorphic encryption while providing a low-latency communication network using the WireGuard VPN Mesh and the decentralized SDS infrastructure. In each communication round, only encrypted model updates are exchanged, whereas the training data remain stored locally on the clients without being shared, thereby preserving data privacy throughout the FL process. The model performance is evaluated on a held-out global test set using Accuracy.

5. Results and Discussion

5.1. End-to-End Performance Evaluation

The proposed SFL-VHDS framework deploys FL as a decentralized learning paradigm within a Software-Defined Storage (SDS) architecture. The SDS acts as an intelligent control layer that manages the virtualized storage environment while enforcing authentication, authorization, and access control policies. In particular, access control lists enable the precise identification of authorized IoT devices participating in the FL process. Consequently, only authenticated devices are permitted to access and exchange local and global model updates during each communication round.

Furthermore, the SDS-based access control mechanism ensures the secure exchange and storage of both locally generated model updates and globally aggregated parameters through the decentralized SDS infrastructure. Combined with the WireGuard VPN Mesh, this architecture establishes authenticated peer-to-peer communication channels that protect the confidentiality and integrity of model updates while enabling secure decentralized collaborative learning.

In addition, the proposed SFL-VHDS framework is architecturally designed to support scalable computing and storage through its decentralized Software-Defined Storage (SDS) infrastructure. The framework also enhances resilience against the representative attack scenarios considered in this study. Moreover, the secure exchange of encrypted model updates within the virtual shared storage enables efficient privacy-preserving aggregation while maintaining low communication latency and high communication integrity.

The latency values in Table 2 demonstrate the performance evaluation of the proposed PPS-SFAA framework, which is implemented within the SFL-VHDS architecture. Latency times reflect the amount of time needed for the completion of one round of FL; hence, the latency figures include all processing tasks, such as training the model locally, encrypting model weights using homomorphic encryption techniques, sending encrypted models through the WireGuard VPN, synchronizing processes at the SDS

Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Latency time (s)	31.86	32.01	31.10	31.04
Integrity rate (%)	97.87	98.27	98.09	98.12
Accuracy rate (%)	97.67	97.58	98.53	98.58
Resilience rate (%)	96.57	96.77	97.76	98.87
Active IoT devices	10	12	14	14
Epoch size	20	25	30	35
Training time (s)	2min42	2min53	2min43	2min45

Table 2. End-to-End Performance Evaluation of the Proposed PPS-SFAA Framework over the SFL-VHDS Architecture

level, aggregating the model, and globally distributing the model. The latency time, thus, is not only the time taken to complete homomorphic encryption but that of the whole process as well.

From the results, we observe that the proposed framework exhibits operational stability in a changing environment. As the number of training epochs increases from 20 to 35 and the number of IoT devices from 10 to 14, the latency remains stable, ranging between 31.86 s and 31.04 s. Additionally, the results reveal the efficiency of combining the SDS architecture with the WireGuard VPN mesh.

Additionally, it can be noted that the proposed PPS-SFAA scheme demonstrates a high capability for collaborative learning as the number of IoT devices grows, as shown by the increase in the resilience level from 96.57% to 98.87%, whereas the integrity level remains stable between 97.87% and 98.12%. Consequently, it can be stated that the secure communication scheme was successfully developed, as it preserves data integrity during the parameter exchange process in FL. Moreover, prediction accuracy remains stable and increases from 97.67% to 98.58% despite the greater number of participants and training epochs. Hence, the SFL-VHDS framework provides an effective balance among communication security, collaboration efficiency, and minimal end-to-end execution time.

5.2. Computational Overhead of Homomorphic Encryption

Despite the end-to-end performance results shown in Table 2 for the proposed PPS-SFAA framework, the overhead introduced by the homomorphic encryption scheme must also be quantified. In addition to the evaluation results discussed above, this section analyzes the computational and memory overhead caused by secure model updates in each FL cycle. The evaluation uses the NGSIM dataset, where distributed trajectory data of vehicular traffic are simulated to reflect collaborative learning among connected cars in a latency-sensitive environment. Consequently, these results provide further insights into the effects of homomorphic encryption within the SFL-VHDS framework.

The measurements reported in Table 3 correspond to the average values obtained over all FL rounds. Encryption, decryption, and secure aggregation times were measured directly from the OpenFHE implementation integrated into the proposed framework using the high-resolution Python performance timer (`time.perf_counter()`). To evaluate the memory footprint introduced by homomorphic encryption, the serialized size of the plaintext model updates generated by PyTorch was compared with the serialized ciphertexts produced by OpenFHE after encryption. The memory overhead is computed as

$$\text{Memory Overhead}(\%) = \frac{S_{\text{enc}} - S_{\text{plain}}}{S_{\text{plain}}} \times 100, \tag{4}$$

where S_{plain} and S_{enc} denote the serialized sizes of the plaintext and encrypted model updates, respectively.

Table 3 quantifies the computational and memory overhead introduced by the OpenFHE-based homomorphic encryption mechanism. Unlike the end-to-end latency reported in Table 2, these measurements isolate the cryptographic operations performed during each FL round. Encryption

Table 3. Computational and Memory Overhead of the Homomorphic Encryption Mechanism in the Proposed PPS-SFAA Framework

Metric	**Value**
Encryption time per model update (ms)	175.20
Decryption time of the aggregated model (ms)	176.53
Secure aggregation time (ms)	3006.20
Plain model update size (MB)	6.84
Encrypted model update size (MB)	18.52
Memory overhead (%)	170.8

and decryption operations take an average of 175.20 ms and 176.53 ms, respectively, meaning that safeguarding local model updates imposes a negligible computational cost relative to the total duration of approximately 31 s required to perform a full FL round. Similarly, performing secure aggregation takes an estimated 3.01 s, which accounts for only a small fraction of the total duration while maintaining the privacy of the transferred model updates.

In addition, the memory cost of encrypting model updates rises from 6.84 MB to 18.52 MB, representing an increase of around 170.8%. This is due to ciphertext expansion, which is a characteristic feature of lattice-based homomorphic encryption algorithms such as OpenFHE. However, despite this overhead, the system maintains good performance in terms of latency and model accuracy. Hence, the memory cost introduced by the encryption process remains sufficiently low.

Under this approach, computations involving encryption and decryption based on OpenFHE, as well as secure aggregation, are performed using Raspberry Pi edge devices and edge/cloud computing systems, whereas Arduino Uno boards are used exclusively for data collection. This ensures that resource-intensive computations are handled by high-resource devices, thereby making the approach feasible.

From the analysis carried out in terms of computation and memory overhead, it is evident that the SFL-VHDS architecture has effectively been able to strike a balance between communication security, privacy, and computational efficiency in connected vehicle IoT systems.

5.3. Threat Model

The proposed SFL-VHDS system operates within a distributed IoT environment comprising connected vehicles that utilize FL technology to cooperatively train a global CNN while preserving their local data privacy. Consequently, the SFL-VHDS system architecture adopts a semi-trusted FL paradigm consisting of the following elements: IoT clients, an aggregation server, an SDS controller, and a decentralized SDS storage layer, all interconnected via WireGuard VPN mesh technology. Specifically, the aggregation server is assumed to be *honest-but-curious*, meaning it complies with the FL procedure while attempting to derive sensitive information from the received model updates.

To prevent these kinds of attacks, the PPS-SFAA protocol incorporates homomorphic encryption from the OpenFHE library for model update confidentiality, a WireGuard VPN mesh to secure communication channels, and an SDS controller to ensure authentication, authorization, integrity checking, and access control prior to secure aggregation. Based on the threat model presented above, three attack scenarios were selected to assess the security of the proposed SFL-VHDS approach.

- **Data Poisoning:** The corrupted IoT client purposely tampers with its local NGSIM training dataset prior to the local CNN training process in order to generate malicious updates for the global model.
- **Model Inversion:** The honest-but-curious attacker attempts to reverse-engineer sensitive data from the model updates shared during the FL process between the IoT devices and the aggregation server.

- **Integrity Attack:** The external attacker attempts to tamper with or inject fabricated data while the client IoT devices and the aggregator server exchange model data for training the global model.

The attack scenarios were implemented using an external workstation running the NS-3 simulator. Dedicated attack scripts were executed against the proposed SFL-VHDS architecture during the FL process to emulate malicious client behavior, inference attempts, and communication attacks. The resilience rate represents the percentage of FL rounds that successfully maintain acceptable learning performance under attack, whereas the integrity rate denotes the percentage of model updates successfully verified and protected against unauthorized modification before secure aggregation.

Table 4. Security Evaluation of the Proposed SFL-VHDS Framework Under Representative Threat Scenarios

Threat Scenario	**Defense Mechanism**	**Integrity (%)**	**Resilience (%)**
Data Poisoning	PPS-SFAA + SDS Controller Validation	97.58	97.53
Model Inversion	OpenFHE Homomorphic Encryption + WireGuard VPN Mesh	98.57	96.68
Integrity Attack	WireGuard VPN Mesh + SDS Controller	97.85	97.58

The experimental results presented in Table 4 demonstrate that the proposed SFL-VHDS framework maintains high resilience and integrity across all evaluated attack scenarios. The integration of the PPS-SFAA secure aggregation algorithm, OpenFHE homomorphic encryption, the WireGuard VPN Mesh, and the SDS controller effectively protects the confidentiality and integrity of exchanged CNN model updates while preserving the reliability of the collaborative learning process. These results indicate that the proposed framework provides effective protection against the evaluated data poisoning, model inversion, and communication integrity attacks while maintaining stable FL performance in latency-sensitive connected vehicular IoT environments.

The present evaluation considers representative attack scenarios, including data poisoning, model inversion, and communication integrity attacks. While these scenarios provide a comprehensive assessment of the proposed framework under realistic security threats, the experimental scope does not include more sophisticated Byzantine attacks involving colluding malicious clients or energy-aware adversarial evaluations.

5.4. Comparative Evaluation Against Secure FL Architectures

For comparison, a baseline ProxFed model without the SDS and WireGuard VPN Mesh security layer, together with the four secure FL architectures discussed in the related work, were implemented and evaluated under identical experimental conditions. All compared architectures employ the same NGSIM dataset, CNN model, training configuration, communication environment, and evaluation metrics to ensure a fair and unbiased comparison. Unlike the proposed SFL-VHDS framework, which integrates a Software-Defined Storage (SDS) architecture, a WireGuard VPN Mesh, and OpenFHE-based homomorphic encryption to secure collaborative learning, the baseline frameworks adopt different communication and security mechanisms. This comparative evaluation enables the impact of the proposed architecture on communication latency, training efficiency, and security-related performance to be assessed.

The baseline methods were selected because they represent fully secure FL architectures that address all the goals of the SFL-VHDS system in terms of communication security, decentralized coordination, and model aggregation. Hence, the experiment will focus on the most relevant secure FL architectures.

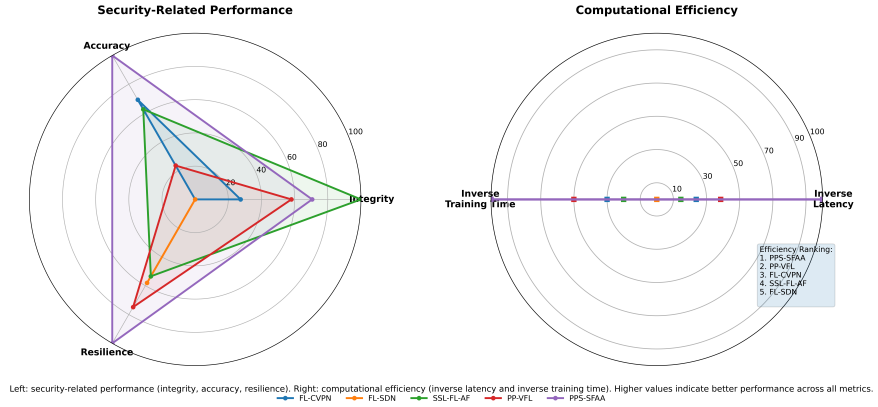


Figure 3. Multi-Panel Radar Visualization of Security and Computational Efficiency in Secure FL Architectures

Figure 3 presents a comparative analysis of the secure FL architectures evaluated in this paper via a multivariate radar diagram, which provides insight into the performance of the algorithms in terms of both security and computational efficiency. Security-oriented performance metrics—specifically communication integrity, resilience, and model accuracy—are used, while computational efficiency is measured through inverse latency and inverse training time.

From the experimental results, it is clear that the proposed SFL-VHDS framework outperforms all other architectures tested. This superior performance is driven by the PPS-SFAA approach, which manages client authentication, secure model update exchanges, data integrity checks, and privacy-preserving aggregation during the FL process. Consequently, it emerged as the most resilient (98.87%), most accurate (98.58%), and most secure (98.12% communication integrity) framework tested. Among the baseline frameworks, SSL-FL-AF demonstrated strong security performance, whereas FL-SDN exhibited poor model accuracy despite maintaining an acceptable level of resilience.

Regarding computational efficiency, the SFL-VHDS framework achieves the lowest communication latency and the shortest training time compared to other frameworks. The computational efficiency of FL-SDN is the worst because of high communication latency. On the other hand, the FL-CVPN and SSL-FL-AF frameworks show average performance in computational efficiency. The computational efficiency of PP-VFL is relatively good; however, it is not as good as the proposed SFL-VHDS framework. Generally, it is clear that using the PPS-SFAA algorithm along with Software-Defined Storage, WireGuard VPN Mesh, and OpenFHE homomorphic encryption is highly efficient.

6. Conclusion and Future Work

In this paper, we review representative secure FL architectures and propose the SFL-VHDS framework, a secure FL framework intended to enhance the confidentiality of the model update exchange and aggregation process in distributed vehicular IoT networks. The integration of the PPS-SFAA secure aggregation algorithm, OpenFHE homomorphic encryption, the WireGuard VPN Mesh, and the SDS architecture enables secure collaborative learning with minimal communication latency. Based on the experimental results obtained using the NGSIM dataset, the proposed framework is feasible and effective, achieving an accuracy of 98.58% with a latency of 31.04 s under non-IID data distribution conditions while maintaining high communication integrity.

In future work, research will focus on improving communication latency and reducing computational complexity by integrating lightweight secure communication protocols, such as Datagram Transport Layer Security (DTLS), with enhanced AAA schemes for constrained IoT devices. Additionally, future studies will explore the scalability of the proposed SFL-VHDS architecture through large-scale deployments involving hundreds to thousands of IoT devices. This research will examine the deployment of multiple SDS controllers, SDS storage nodes, and multi-cloud infrastructures to evaluate the impact of large-scale participation on communication latency, secure aggregation complexity, resource consumption, and overall system performance. Finally, future work will evaluate the implications of non-IID data heterogeneity using various Dirichlet concentration parameters.

REFERENCES

1. A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, *Federated learning for resource-constrained IoT devices: Panoramas and state-of-the-art*, arXiv preprint arXiv:2002.10610, 2020.
2. M. J. C. S. Reis, E. B. Ashary, B. M. Alghamdi, R. Aboalela, N. Alsaadi, and K. H. Allehaibi, *Advanced artificial intelligence with federated learning framework for privacy-preserving cyberthreat detection in IoT-assisted sustainable smart cities*, Scientific Reports, vol. 15, p. 4470, 2025.
3. Q. Duan, and Z. Lu, *Edge cloud computing and federated–split learning in Internet of Things*, Future Internet, vol. 16, no. 7, p. 227, 2024.
4. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, *Fog computing and edge intelligence for Internet of Things: Architectures, applications, and challenges*, IEEE Communications Surveys & Tutorials, vol. 26, no. 2, pp. 1120–1158, 2024.
5. Y. Li, J. Chen, and Z. Wang, *Edge–fog–cloud continuum for IoT: Reducing latency while preserving data privacy in distributed learning systems*, IEEE Internet of Things Journal, vol. 12, no. 3, pp. 2456–2472, 2025.
6. M. Tawfik, A. H. Abdelhaliem, and I. S. Fathi, *Transforming IoT Security through Large Language Models: A Comprehensive Systematic Review and Future Directions*, Statistics, Optimization & Information Computing, vol. 14, no. 2, pp. 1018–1044, 2025.
7. M. R. War, Y. Singh, Z. A. Sheikh, and P. K. Singh, *Review on the Use of Federated Learning Models for the Security of Cyber-Physical Systems*, Scalable Computing: Practice and Experience, vol. 26, no. 1, pp. 16–33, 2025.
8. S. Sharma, S. Sharma, and T. Choudhury, *Enhancing Cloud Data Security Through an Encrypted and Efficient Connection Model Based on Blockchain Technology*, Scalable Computing: Practice and Experience, vol. 26, no. 2, pp. 517–530, 2025.
9. P. Kairouz, H. B. McMahan, B. Avent, et al., *Advances and open problems in federated learning*, Foundations and Trends in Machine Learning, vol. 17, no. 2–3, pp. 1–210, 2024.
10. J. Wang, M. Chen, and H. V. Poor, *Federated learning in IoT systems: Architecture, aggregation mechanisms, and privacy guarantees*, IEEE Internet of Things Journal, vol. 12, no. 5, pp. 3890–3908, 2025.
11. I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, and P. Watters, *Secure federated learning over encrypted communication channels for IoT systems: A VPN-based architecture*, IEEE Access, vol. 12, pp. 91234–91252, 2024.
12. B. Alotaibi, and N. Ansari, *Secure authentication and access control for federated learning in smart city IoT environments*, IEEE Internet of Things Journal, vol. 11, no. 15, pp. 15234–15248, 2024.
13. J. Li, X. Zhang, and D. Kim, *Software-defined networking for federated learning: Traffic control and latency optimization in IoT environments*, Computer Networks, vol. 242, p. 110255, 2024.
14. T. H. Nguyen, D. Kim, and C. S. Hong, *Software-defined networking for intelligent federated learning: Control-plane optimization and traffic management*, IEEE Communications Surveys & Tutorials, vol. 26, no. 4, pp. 3102–3128, 2024.
15. P. Kumar, and A. Singh, *Centralized SDN control for adaptive traffic engineering in IoT and edge-based federated learning systems*, Computer Networks, vol. 243, p. 110276, 2024.
16. P. Kumar, R. Gupta, and A. Singh, *Secure federated learning in SDN-IoT architectures: Local training and privacy guarantees*, Computer Networks, vol. 244, p. 110289, 2024.
17. R. Zhang, Y. Li, and X. Wang, *SDN-enabled federated learning with intelligent routing and adaptive aggregation strategies*, IEEE Internet of Things Journal, vol. 12, no. 8, pp. 7201–7218, 2025.
18. Y. Zhang, X. Li, and L. Sun, *Lightweight authentication mechanisms for secure federated learning in edge and IoT systems*, Computer Networks, vol. 246, p. 110338, 2024.
19. S. Mishra, and S. Chatterjee, *Lightweight cryptographic authentication for federated learning-enabled IoT networks*, Computer Networks, vol. 247, p. 110367, 2024.
20. S. Li, J. Chen, and Q. Yang, *Privacy-preserving federated learning via homomorphic encryption and secure aggregation*, IEEE Communications Surveys & Tutorials, vol. 26, no. 5, pp. 3601–3630, 2024.
21. H. Wang, M. Chen, and H. V. Poor, *Blockchain-enabled verifiable aggregation for secure federated learning under adversarial IoT environments*, IEEE Transactions on Network Science and Engineering, vol. 12, no. 4, pp. 512–530, 2025.
22. T. H. Nguyen, D. Kim, and C. S. Hong, *Communication efficiency and security challenges in federated learning across edge and cloud environments*, IEEE Communications Surveys & Tutorials, vol. 26, no. 6, pp. 3801–3830, 2024.

23. H. Wang, M. Chen, and H. V. Poor, *Latency-aware federated learning in hierarchical cloud-edge architectures*, IEEE Transactions on Wireless Communications, vol. 24, no. 4, pp. 3201–3218, 2025.
24. R. Zhang, Y. Li, and X. Wang, *Scalability and security challenges in federated learning for massive IoT deployments*, IEEE Internet of Things Journal, vol. 12, no. 11, pp. 10120–10138, 2025.
25. V. Mothukuri, R. M. Parizi, S. Pouriyeh, et al., *Security threats in federated learning: Data poisoning, model inversion, and integrity attacks*, Journal of Network and Computer Applications, vol. 231, p. 104078, 2024.
26. S. Li, J. Chen, and Q. Yang, *Software-defined storage for distributed edge computing: Architecture and challenges*, IEEE Communications Surveys & Tutorials, vol. 26, no. 7, pp. 4101–4128, 2024.
27. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, *Federated optimization in heterogeneous networks (FedProx): A robust approach to non-IID federated learning*, IEEE Communications Surveys & Tutorials, vol. 26, no. 6, pp. 3901–3928, 2024.
28. R. Zhang, X. Wang, and J. Luo, *Deep learning-based traffic modeling using NGSIM vehicle trajectory datasets*, IEEE Internet of Things Journal, vol. 12, no. 15, pp. 14010–14028, 2025.