PureFL: Trust-Weighted Federated Learning With Noise-Resilient Homomorphic Encryption for Blockchain-Based IoV Networks

Hope Leticia Nakayiza, Love Allen Chijioke Ahakonye, *Senior Member, IEEE*, Dong-Seong Kim, *Senior Member, IEEE*, and Jae Min Lee, *Member, IEEE*

***Abstract*—Federated learning (FL) in the Internet of Vehicles (IoV) networks is challenged by privacy vulnerabilities, model poisoning attacks, and the limitations of centralized aggregation, which threaten security and scalability. While homomorphic encryption (HE) provides an additional layer of privacy, existing schemes often suffer from cumulative noise growth, resulting in degraded model accuracy and increased inference latency, critical factors for real-time vehicular safety applications. This article presents PureFL, a novel FL framework that combines adaptive Cheon-Kim-Kim-Song (CKKS) HE with a custom permissioned blockchain (PureChain), utilizing a proof of authority and association (PoA^2) consensus mechanism. PureFL incorporates dynamic precision scaling to manage noise accumulation, ensuring computational efficiency without sacrificing privacy or accuracy. The framework also implements a trust-based aggregation mechanism based on cosine-similarity trust scoring, which dynamically weights client contributions according to historical reliability and anomaly detection, thereby enhancing resilience against model poisoning. Experimentation results on the CICIOV2024 and 5G-NIDD datasets demonstrate that PureFL achieves superior anomaly detection accuracy of 99.9%, reduces CKKS HE overhead via precision scaling, and the proposed PoA^2 consensus outperforms other consensus mechanisms in terms of throughput and latency, making it suitable for real-time IoV operations.**

Received 11 June 2025; revised 23 November 2025 and 26 January 2026; accepted 3 May 2026. Date of publication 8 May 2026; date of current version 9 July 2026. This work was supported in part by the Innovative Human Resource Development for Local Intellectualization Program through Institute for Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government [Ministry of Science and ICT (MSIT)] under Grant IITP-2026-RS-2020-II201612, 20%; in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education, Science and Technology (MEST) under Grant 2018R1A6A1A03024003, 20%; in part by MSIT, South Korea, through Information Technology Research Center (ITRC) Support Program under Grant IITP-2026-RS-2024-00438430, 20%; in part by the Basic Science Research Program through NRF funded by the Ministry of Education (MOE) under Grant RS-2025-25431637, 20%; and in part by the Regional Innovation System and Education [Regional Innovation System and Education (RISE)]-(Regional Growth Innovation Laboratory) Program through Gyeongbuk RISE Center, funded by the MOE, Gyeongsangbuk-do, Republic of Korea, under Grant 2026-rise-15-105, 20%. *(Corresponding author: Jae Min Lee.)*

Hope Leticia Nakayiza and Jae Min Lee are with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: hopeleticia@kumoh.ac.kr; ljmpaul@kumoh.ac.kr).

Love Allen Chijioke Ahakonye is with ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: loveahakonye@kumoh.ac.kr).

Dong-Seong Kim is with the Department of IT Convergence Engineering, and NSLab Company Ltd., Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: dskim@kumoh.ac.kr).

Digital Object Identifier 10.1109/JIOT.2026.3691349

***Index Terms*—Anomaly detection, blockchain, Cheon-Kim-Kim-Song (CKKS), federated learning (FL), homomorphic encryption (HE), Internet of Vehicles (IoV), privacy preservation.**

I. Introduction

THE Internet of Vehicles (IoV) represents a critical aspect of intelligent transportation systems (ITSs) [1], enabling vehicles to communicate with each other, with infrastructure, and with broader networks [2]. These communications utilize technologies such as the IEEE 802.11p standard [3] along with dedicated short-range communications (DSRC), which provide low-latency, reliable wireless connectivity to support real-time vehicle-to-everything interactions [4]. IoV seeks to enhance road safety, optimize urban traffic management, and deliver smart, convenient user experiences [5], [6], supporting services such as autonomous driving, infrastructure maintenance, logistics, and emergency response.

The IoV architecture in Fig. 1 features multiple layers for efficient and secure communication in vehicular networks [7]. The data acquisition layer enables real-time collection via vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) communication. The network layer ensures seamless connectivity using 5G/6G, Wi-Fi, DSRC, GPS, and LAN technologies. The fog layer manages tasks, performs low-latency analysis, and provides caching to reduce latency and computational load. The cloud layer supports large-scale data analysis and storage. The application layer provides services such as traffic management, route optimization, road safety, emergency response, and autonomous driving. Security management is integrated across all layers, addressing privacy, authentication, access control, and trust management [7].

Vehicles in the IoV face multiple security threats due to their real-time data exchanges, including data falsification attacks [8], where malicious entities inject false information such as incorrect speed, fake traffic data, or nonexistent obstacles, disrupting traffic flow and safety. Model poisoning attacks [8] occur during federated learning (FL), where adversaries manipulate model updates, compromising the global model and degrading system accuracy, resulting in misclassification of traffic data. Replay and inference attacks [8] exploit model update transmission to infer sensitive data, such as vehicle location or driving behavior, without accessing raw data. Sybil attacks [8] involve adversaries creating multiple identities to manipulate system aggregation. Man-in-the-middle attacks [8] intercept and alter critical messages between vehicles or

© 2026 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

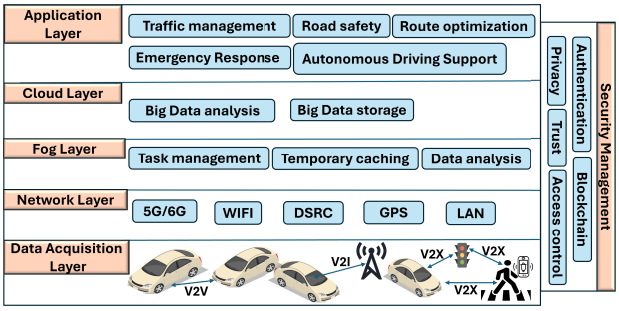


Fig. 1. Simplified IoV architecture.

infrastructure, potentially altering safety-critical data such as traffic lights or collision warnings, which could lead to accidents. These malicious activities hinder system performance, delay decision-making, and jeopardize safety, emphasizing the need for anomaly detection to preserve network integrity and prevent data breaches [7], [9].

Misbehavior detection in vehicular networks has gained significant attention recently due to the increasing reliance on ITS [10]. Robust intrusion detection systems (IDSs) are critical to counter potential cyber threats and anomalies within data exchanged in vehicle-to-infrastructure interactions [11]. To address the privacy challenges inherent in traditional machine learning-based IDSs, distributed paradigms such as FL enable clients to train models locally and share only encrypted updates for centralized aggregation [12], [13]. In IoV deployments, both vehicles and roadside units (RSUs) participate as FL clients, collaboratively refining a global model while keeping sensitive data, such as precise location and driving behavior, on-device [14]. This distributed approach enhances privacy and minimizes vulnerabilities by sharing only encrypted updates. However, ensuring the confidentiality and integrity of these model updates during transmission and aggregation necessitates the application of homomorphic encryption (HE), which enables secure computation over encrypted data without revealing underlying raw information [15].

HE is crucial for data privacy and model security in IoV environments, where sensitive information, such as location, behavior, and vehicular states, is transmitted [16]. HE allows computations on ciphertexts without decryption, maintaining confidentiality throughout the data lifecycle [17]. In FL, HE enables secure aggregation of client-side model updates in their encrypted state, preventing exposure of raw data even when interacting with untrusted fog nodes or cloud servers [18]. Various HE schemes, including Paillier [19], Brakerski/Fan-Vercauteren [20], and Cheon-Kim-Kim-Song (CKKS) [21], have been studied for FL-based IoV systems. Paillier-based schemes offer strong security but incur high communication overhead and poor scalability for large-scale vehicular networks. Although CKKS supports approximate arithmetic for machine learning, traditional implementations suffer from unbounded noise growth across aggregation rounds, necessitating early decryption or larger ciphertext modulus, which degrades performance and accuracy, especially in real-time applications such as anomaly detection and traffic control [15], [18].

Conventional centralized security frameworks are inadequate for dynamic IoV environments, primarily due to high vehicle mobility, constrained communication bandwidth, and stringent low-latency requirements [22]. These limitations have driven the adoption of blockchain-based solutions to address the shortcomings of single-point control and to enhance trust in distributed vehicular ecosystems [23]. Blockchain, as a decentralized and tamper-evident ledger, provides immutable record-keeping and secure transaction validation through cryptographic consensus mechanisms [24]. By eliminating the need for a central trusted authority, blockchain enhances transparency, supports secure identity authentication, ensures data provenance, and enforces privacy-preserving communication across untrusted parties [25].

Many blockchain implementations experience high latency due to complex consensus mechanisms, such as proof-of-work (PoW) and proof-of-stake (PoS), making them unsuitable for time-sensitive applications, including real-time anomaly detection in vehicular networks [7]. In addition, storing model updates directly on the blockchain in FL introduces scalability and storage issues [24]. These challenges require lightweight, low-latency blockchain architectures optimized for real-time systems, such as IoV. Pairing these architectures with off-chain storage solutions, such as the InterPlanetary File System (IPFS), ensures efficient, secure, and scalable handling of encrypted model updates, enabling responsive and privacy-preserving detection of misbehavior [24].

To address the discussed limitations, this study presents the PureFL framework with the following key contributions:

1) combination of HE with FL to ensure security and privacy preservation in misbehavior detection;
2) implementation of an anomaly detection mechanism at the vehicle layer to isolate and filter out malicious updates, preventing compromised data from reaching global aggregation;
3) integration of CKKS with dynamic precision scaling to manage noise accumulation and ciphertext packing to batch gradient components, preserving precision while reducing computational overhead;
4) incorporation of a trust-weighted aggregation strategy that dynamically adjusts each client's influence on the global model based on historical behavior and observed consistency to enhance resilience against adversarial updates and promote reliable participation in the FL process;
5) a custom permissioned PureChain network backed by the IPFS to provide a tamper-proof and transparent management of encrypted model updates and track client contributions so as to reward quality and honest behavior to encourage active participation in the FL process.

The remainder of this study is organized as follows. Section II reviews existing work on HE. Section III discusses the proposed framework, while Section IV highlights the experimental setup and results. Section V concludes this study.

II. Related Works

This section critically examines existing approaches to privacy-preserving misbehavior detection in IoV. The literature

has been organized into three areas: misbehavior detection mechanisms, HE techniques for privacy preservation, and blockchain integration for decentralized trust management. Through these, the technical gaps that motivate the PureFL framework have been identified.

A. Misbehavior Detection in IoV

Machine learning-based IDSs have shown promise in detecting data falsification and malicious behavior in IoV networks [33]. Recent approaches focus on improving detection accuracy, computational efficiency, and real-time responsiveness for time-critical IoV safety applications.

Several studies have proposed specialized architectures for anomaly detection. Zhang et al. [34] introduced an IDS that leverages Gaussian random incremental principal component analysis for dimensionality reduction, combined with an optimally weighted extreme learning machine. While effective for feature extraction, this approach operates in a centralized manner, creating privacy vulnerabilities when vehicles must share raw sensor data. Similarly, the edge intelligence approach by [35] employed auxiliary classifier generative adversarial networks to improve throughput and reduce latency and computational demands during real-time intrusion detection. However, this work does not address the privacy implications of data aggregation at edge nodes. To address energy efficiency concerns, Lihua [36] developed a two-phase contract model for secure, energy-saving V2V communication paths. However, the system lacks mechanisms to detect model poisoning attacks during collaborative learning. Wang et al. [37] presented a rapid misbehavior detection model based on broad learning with ridge regression and incremental learning for adaptive updates. Since privacy-preserving mechanisms were not considered for this study, deployment in privacy-sensitive IoV environments becomes challenging.

FL has emerged as a promising paradigm to enable collaborative model training while keeping data localized. The study by Jai Vinita and Vetriselvi [38] employed FL for Sybil attack detection in 6G IoV, using fuzzy logic for client selection. However, the framework does not protect model updates during transmission, leaving them vulnerable to inference attacks. To address trust issues in FL models, [25] proposed a blockchain-based validation system with enhanced PoW. However, the high computational overhead of PoW makes it impractical for real-time vehicular applications. Roy and Madria [39] designed a blockchain framework for misbehavior detection and traffic event validation, effective even in adversarial scenarios. In addition, Nakayiza et al. [40] proposed a resource-aware FL strategy that dynamically selects a subset of clients based on resource states to improve distributed denial-of-service (DDoS) detection in vehicle networks. However, the study did not consider privacy issues related to model updates.

Existing misbehavior-detection approaches either sacrifice privacy for performance by using centralized methods or fail to protect model updates during federated aggregation.

B. HE for Privacy Preservation in IoV

HE enables computation on encrypted data without decryption, making it particularly suitable for privacy-preserving FL in IoV. However, different HE schemes present distinct tradeoffs between security, computational efficiency, and noise management.

Prior work by Boudguiga et al. [41] used Paillier HE to protect vehicle data privacy during analysis by external service providers, developing a neural network to classify driver behavior. Costantino et al. [42] utilized the CKKS scheme to create a privacy-preserving mechanism for ITS that enables personalized services without exposing sensitive driver information. To further enhance security, [30] proposed implementing a distributed cryptosystem based on an additive HE (AHE) scheme that facilitates HE functionality even when data are encrypted with different keys. Karim and Rawat [43] introduced a postquantum privacy-focused HE system for safeguarding electronic toll transponder data, leveraging blockchain for secure data exchange within IoV and mobile edge networks. Zhou et al. [27] strengthened FL for fog computing by incorporating blinding and Paillier HE to defend against model attacks, improving training efficiency and model accuracy while addressing data imbalance and computational constraints. In addition, [16] presented a model that permits vehicle units to securely offload encrypted data to a central server using HE, enabling secure processing without decryption.

Despite these advancements, several gaps remain. Existing HE implementations do not account for the rapid noise growth and latency across layers that are introduced by large polynomial moduli or frequent rescaling.

C. Blockchain Integration With HE for FL

Blockchain technology provides decentralized, tamper-evident infrastructure for FL [23], addressing single-point-of-failure vulnerabilities in centralized aggregation. However, the choice of consensus mechanism critically affects system latency and throughput, which are key requirements for real-time IoV applications.

Several studies have explored blockchain-HE schemes to enhance privacy preservation across various real-time network applications. Yan et al. [26] introduced an HE approach based on the Paillier and Rivest–Shamir–Adleman (RSA) cryptosystems to ensure identity privacy without correlation, anonymity, and effective supervision within a blockchain-assisted edge computing system. Similarly, [28] introduced a lightweight FL framework that integrates blockchain with the Peikert–Vaikuntanathan–Waters (PVWs) HE scheme, enabling secure operations on integer vectors while safeguarding user privacy and minimizing resource consumption. A byzantine-robust FL framework in [29] leveraged blockchain for transparency, cosine similarity for detecting malicious gradients, and fully HE for secure aggregation.

For vehicular-specific applications, Xie et al. [31] proposed a blockchain-based authentication and management scheme for flying ad hoc network (FANET), using HE, smart contracts, and an identity consensus mechanism to enhance security, efficiency, and decentralized control. In a study on blockchain-based privacy-preserving FL for IoV, Wang et al. [32] integrated HE with an enhanced Multi-Krum aggregation

TABLE I
COMPARISON OF EXISTING STUDIES LEVERAGING BLOCKCHAIN AND HE WITH THE PROPOSED BPP-FL FRAMEWORK

Ref	Year	Employed HE Scheme	Target System	Federated Learning	Noise Management	Trust Management	Distributed Storage	Blockchain Integration	Decentralized Aggregation
[26]	2020	Paillier	Edge Computing	✗	✗	✗	✗	✓	✗
[27]	2020	Paillier	Fog Computing	✓	✗	✗	✗	✗	✗
[28]	2022	PVW	IoV	✓	✗	✗	✗	✓	✓
[29]	2022	CKKS	FL Networks	✓	✗	✓	✗	✓	✓
[30]	2022	AHE	Cloud Computing	✓	✗	✗	✗	✗	✗
[31]	2024	ElGamal	FANET	✗	✗	✗	✗	✓	✗
[32]	2024	Paillier	IoV	✓	✗	✓	✗	✓	✓
Ours (PureFL)	**2025**	**CKKS**	**IoV**	✓	✓	✓	✓	✓	✓

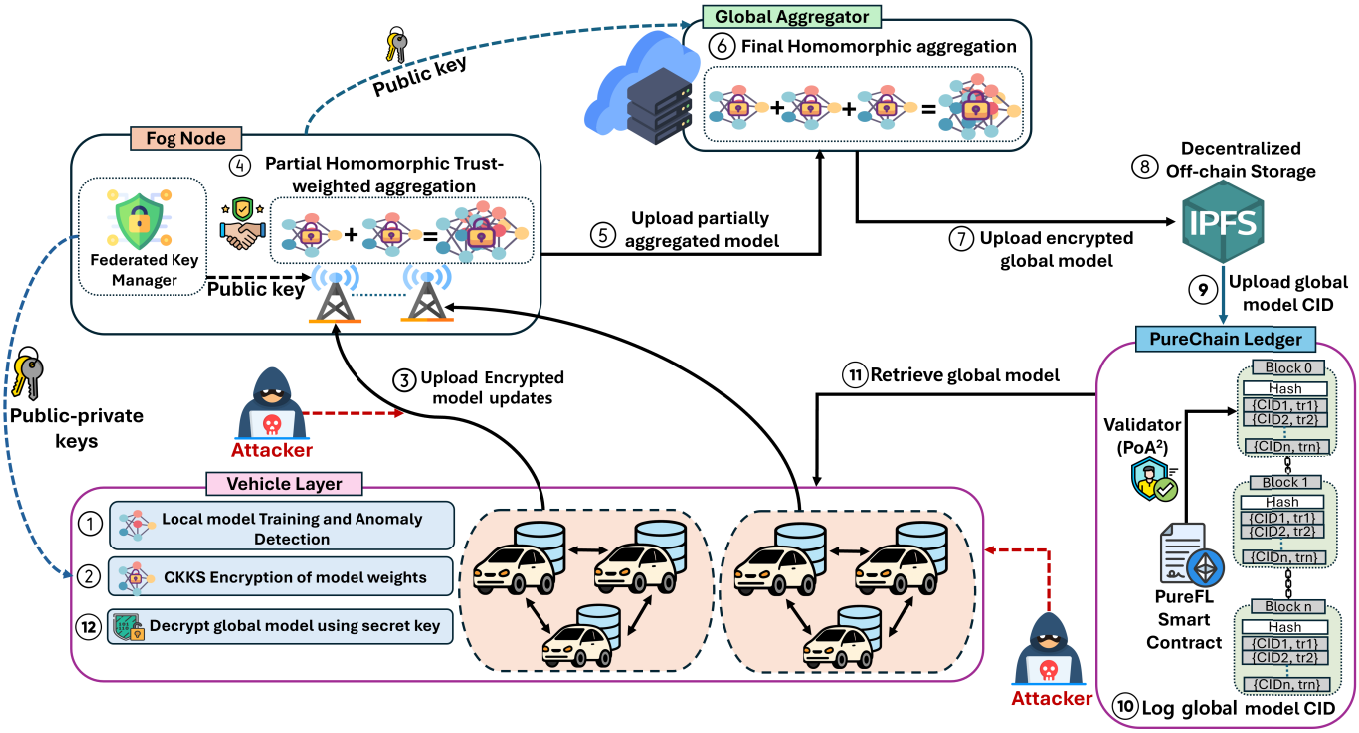


Fig. 2. General overview of the proposed PureFL framework and workflow.

for blockchain-based privacy-preserving FL in IoV. While the approach facilitated model aggregation and filtering at the ciphertext level, the study did not account for noise accumulation resulting from frequent model aggregation and its impact on the system.

Existing blockchain-integrated FL systems suffer from consensus-induced latency that violates real-time requirements for vehicular safety applications. Furthermore, various studies do not provide a comprehensive evaluation of both latency and throughput under increasing client loads for the employed blockchain network, a critical omission for assessing real-world scalability. The lack of off-chain storage integration creates additional scalability bottlenecks as model sizes and participant numbers increase.

Table I provides a comparative analysis of existing studies, highlighting key limitations that motivate the contributions of the PureFL framework in terms of scalability, trust management, and noise-resilient encryption.

III. BLOCKCHAIN-ASSISTED PRIVACY-PRESERVING FEDERATED IDS IN IOV

This section discusses the overall PureFL framework, integrating the proposed HE technique with blockchain in FL to ensure privacy, scalability, and real-time misbehavior detection within IoV. The general overview of PureFL's process flow is shown in Fig. 2.

A. Real-World Scenario: Privacy-Preserving Traffic Management in Smart Cities

A smart city with an ITS that integrates private vehicles, public transportation, and roadside infrastructure aims to enhance safety and optimize urban mobility. The traffic management center implements a collaborative anomaly detection system capable of identifying a wide range of threats, including cyberattacks, unsafe driver behavior, infrastructure malfunctions, and coordinated malicious activities such as

Sybil attacks or data falsification. Achieving high detection accuracy with machine learning requires input from diverse parties, including autonomous vehicle manufacturers with proprietary sensor data, everyday vehicle owners, public transportation operators with extensive route coverage, and various city infrastructure sensors.

However, traditional centralized machine learning approaches encounter several significant challenges. Private vehicle owners may be unwilling to upload raw telemetry data, GPS routes, speed logs, or driving habits, as this involves revealing personal locations and routines. Autonomous vehicle manufacturers may be reluctant to share raw sensor information or behavioral analyses due to the exposure of proprietary algorithms or the violation of privacy regulations governing data sharing in ITS, such as the General Data Protection Regulation (GDPR) [44]. In addition, centralized servers create single points of failure, making them attractive targets for cyberattacks that could compromise sensitive data from thousands of vehicles in a single breach.

To address these persistent barriers, a solution must enable privacy-preserving, collaborative anomaly detection using FL to support efficient data management and enable more informed traffic management decisions [44]. The solution should allow all contributors to participate without sacrificing data ownership or confidentiality. PureFL is designed to address these challenges as discussed throughout this work.

B. Threat Model

The proposed framework assumes a partially adversarial FL environment deployed across vehicular networks, where some participating clients may behave maliciously or unreliably. The threat model considers the following adversaries.

1) *Malicious Clients*: A subset of clients may perform model poisoning attacks by uploading manipulated updates or attempting to impersonate other vehicles.
2) *Curious Aggregators*: Fog and cloud servers are honest-but-curious and may attempt to infer sensitive client information from encrypted model updates or trust scores.
3) *Passive Eavesdroppers*: External adversaries may monitor communication channels to infer the content of model updates or client identities.
4) *Tampering Nodes*: Compromised fog or blockchain nodes may attempt to rewrite, replay, or delete trust logs and model update metadata.

The framework is designed to defend against these threats by employing CKKS HE for end-to-end confidentiality, cosine similarity-based trust scoring to filter low-quality or adversarial updates, and blockchain-backed IPFS logging to ensure the immutability and auditability of aggregation and trust history.

C. Intrusion Detection at the Vehicle Layer

Each vehicle independently trains a machine learning model on its local data to identify misbehavior patterns. Vehicles perform local anomaly detection before submitting updates to ensure that sensitive raw data never leaves the vehicle and discard the model update if an anomaly or attack is detected. It also prevents compromised data from entering the FL process, thereby safeguarding the integrity of the global model.

The intrusion detection model employed the long short-term memory (LSTM) networks. The LSTM cell consists of three primary gates: forget, input, and output gates, along with a cell state (C_t) that helps in retaining long-term dependencies. The LSTM gate structure is discussed as follows.

1) Forget gate f_t in (1) determines which data to discard from the previous cell state

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

where x_t is the input vector at time t and h_{t-1} represents the hidden state from the previous timestep. W_f is the weight matrix for the forget gate, b_f is the bias vector for the forget gate, and σ is the sigmoid activation function that outputs values in $0, 1$, depicting the data fraction to retain

2) Input gate i_t decides the new data update into the cell state as follows:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \tilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \end{aligned} \tag{2}$$

where i_t is the input gate at time t and σ is the sigmoid activation function. W_i and b_i are the weight matrix and bias for the input gate, and $\tilde{C}_t$ represents the candidate cell state. tanh is the hyperbolic tangent activation function, and W_C and b_C are the weight matrix and bias for the candidate cell state. h_{t-1}, x_t concatenates the previous hidden state and current input.

3) Cell state update C_t combines the previous cell state (scaled by the forget gate) and the new data in the following equation:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3}$$

where $\odot$ depicts elementwise multiplication and C_{t-1} is the previous cell state.

4) Output gate o_t controls the output of the LSTM cell and combines the current cell state and the hidden state as shown in the following equation:

$$\begin{aligned} o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t \odot \tanh\left(C_t\right) \end{aligned} \tag{4}$$

where W_o represents the Weight matrix for the output gate, b_o is the bias vector for the output gate, and h_t is the current hidden state, the LSTM cell's output.

D. CKKS for Privacy Preservation

HE ensures secure aggregation of model updates in federated IoV systems, preventing inference attacks on vehicle model updates that inherently contain information about driving patterns and behavior. The CKKS scheme [21], a lattice-based cryptosystem relying on the ring learning with errors (RLWEs) problem [45], which is widely believed to be secure against quantum attacks [46], is implemented in this study. Unlike integer-based homomorphic schemes, Paillier [19] and BFV [20], CKKS supports approximate arithmetic on

encrypted real-valued data [15], making it ideal for machine learning tasks involving floating-point computations [47]. CKKS implementation facilitates trustless computation across distributed fog nodes by allowing mathematical operations to be performed directly on encrypted data, eliminating the risk of data exposure at the aggregator by never requiring decryption during aggregation.

Algorithm 1 CKKS Key Generation

Require: Polynomial degree N, Moduli $[b_1, \ldots, b_k]$, initial scale Δ
Ensure: CKKS context $\mathcal{C}$, keys (pk_v, sk_v), keygen time t_{keygen}
Initialization:
procedure CKKS_SETUP(N, $[b_1, \ldots, b_k]$, Δ)
  $\mathcal{C} \leftarrow$ CKKS context with degree N and modulus chain $[b_1, \ldots, b_k]$
  Set global scale Δ
  Generate public-private key pair (pk_v, sk_v) for encryption and decryption
  Generate Galois keys for ciphertext rotations
  Generate evaluation key for linearization
  $t_{\text{keygen}} \leftarrow$ Measure and record key generation time
  return $\mathcal{C}$, pk_v, sk_v, t_{keygen}
end procedure

In PureFL, a designated fog node serves as the federated key manager and is responsible for generating a CKKS public–secret key pair, (pk_v, sk_v). The public key pk_v is distributed to all vehicle participants, fog aggregators, and the final global aggregator to enable HE and aggregation. Each vehicle also receives the corresponding secret key, sk_v, which is securely stored and used solely for decrypting the final aggregated global model. The fog and global aggregators operate only on encrypted data and never require access to the secret key, preserving end-to-end privacy throughout the FL process. Algorithm 1 details the key generation process.

E. CKKS Encryption With Dynamic Precision Scaling

CKKS enables approximate arithmetic homomorphic operations on real and complex numbers by encoding plaintext data with a scaling factor Δ. Sequential homomorphic operations and encoding processes introduce noise, as a small error, which gradually degrades precision [48]. Therefore, maintaining numerical precision throughout the CKKS computation chains remains a key challenge. If not controlled, this noise can eventually render decryption impossible or degrade the accuracy of decrypted results. Dynamic precision scaling mitigates this challenge by adaptively adjusting Δ across computation levels to balance computational efficiency and accuracy, mitigating noise accumulation and approximation errors.

Before transmitting model updates to the fog layer, vehicles encrypt nonmalicious model updates m using the public key pk_v. The encryption function over the polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$ is mathematically represented in the following equation:

$$c = \text{Enc}(m) = \left[\lfloor \Delta_0 \cdot m \rceil + e + pk_v \cdot u\right]_q \tag{5}$$

where $\mathcal{R}_q$ represents the polynomial ring with modulus q, Δ_0 denotes the initial scaling factor, which encodes $m \in \mathbb{R}$ into $\mathcal{R}_q$, and e is small error term added for RLWE security. u is a random vector, and q is the ciphertext modulus, which controls the tradeoff between security and noise. Last, $\lfloor \cdot \rceil$ denotes coordinatewise rounding to the nearest integer.

Chunkwise encryption is implemented to handle model updates in smaller portions and to parallelize encryption and decryption, both of which are vital for large models. This approach reduces encryption and decryption times by processing smaller segments of the model update per encryption operation, thereby optimizing resource usage and latency. To enhance the traditional CKKS, dynamic precision scaling is applied. After each homomorphic multiplication, the ciphertext is rescaled by a level-dependent prime s_j from the modulus chain $[b_1, \ldots, b_k]$ in the following equation:

$$\Delta_i = \frac{\Delta_{i-1}}{s_j} \tag{6}$$

where Δ_i represents the scaling factor after the ith operation, s_j is the scaling adjustment factor after each operation j, and j accounts for noise generated during that operation. This preserves significant bits while reducing noise growth, enabling deeper computations during multilayered model aggregation.

Algorithm 2 Dynamic Precision Scaling for CKKS Encrypted Model Updates

Require: Model update vector U of length n, chunk size c, base scale Δ_0, total depth D
Ensure: Encrypted model update chunks $\mathcal{U}_{enc} = \{c_1, c_2, \ldots, c_k\}$
Initialize CKKS context with parameters $(N, \{\mathsf{mod}_i\}, \Delta_0)$
Generate public and secret keys for encryption/decryption
Flatten model update $U \leftarrow \text{flatten}(U)$
Divide U into $k = \lceil n/c \rceil$ chunks: $U = \{u_1, u_2, \ldots, u_k\}$ **for** *each chunk index* $i = 1$ *to* k **do**
  end
  Compute dynamic scale: $\Delta_i \leftarrow \text{ScaleDecay}(\Delta_0, i, D)$
Encrypt chunk: $c_i \leftarrow \text{CKKS\_Encrypt}(u_i, \Delta_i)$
Estimate noise level: $\eta_i \leftarrow \text{EstimateNoise}(c_i, \Delta_i)$
Append c_i to $\mathcal{U}_{enc}$
return $\mathcal{U}_{enc}$

Algorithm 2 details the CKKS encryption and dynamic scaling adjustment processes.

F. Trust-Weighted Aggregation of Encrypted Model Updates

Traditional federated averaging (FedAvg) assumes uniform client reliability, which is unrealistic in vehicular networks where participants exhibit varying trustworthiness and data quality. Our proposed framework implements trust-weighted homomorphic aggregation to compute trust scores and directly aggregate weighted models on encrypted model updates $\mathbf{U}_{enc}$, eliminating privacy vulnerabilities associated with temporary decryption. The trust-weighted aggregation process is discussed in the following.

1) Homomorphic Trust Score Computation: Each client i sends an encrypted model update $\mathbf{u}_i^t \in \mathbf{U}_{enc}$ to the partial aggregation fog node in round t. To evaluate client reliability, the partial aggregation fog node computes a trust score based on the cosine similarity between each encrypted update and an encrypted reference update.

The trust score S_i^t for client i is computed in the following equation:

$$S_i^t = \text{HE\_CosineSim}\left(\Delta_i^t, \Delta_{\text{ref}}^t\right) \tag{7}$$

where Δ_i^t is the encrypted update from client i in round t and Δ_{ref}^t is the encrypted reference update computed as the homomorphic mean of all client updates.

The homomorphic cosine similarity is implemented as

$$\text{dot}_i^t = \sum k = 1^d \mathbf{u}_{enc,i,k}^t \odot \mathbf{u}_{enc,\text{ref},k}^t \tag{8}$$

$$\text{norm}_i^t = \text{PolyApprox}\left(\sqrt{\sum k = 1^d \left(\mathbf{u}_{enc,i,k}^t\right)^2}\right) \tag{9}$$

$$\text{norm}_{\text{ref}}^t = \text{PolyApprox}\left(\sqrt{\sum_{k=1}^d \left(\mathbf{u}_{enc,\text{ref},k}^t\right)^2}\right) \tag{10}$$

$$S_i^t = \text{PolyApprox}\left(\frac{\text{dot}_i^t}{\text{norm}_i^t \odot \text{norm}_{\text{ref}}^t}\right) \tag{11}$$

where $\odot$ denotes CKKS homomorphic multiplication and PolyApprox$(\cdot)$ indicates polynomial approximation for nonlinear operations. Since all computations are carried out in the encrypted domain using polynomial approximations, the process inherently introduces approximation errors.

2) Homomorphic Memory-Based Trust Scoring: Memory-based scoring is applied to ensure stability over time and penalize inconsistent clients by rewarding consistent reliability and discouraging short-term cheating. The updated trust score $\hat{S}_i^t$ is a weighted combination of the current round's score and the previous score, as shown in the following equation:

$$\hat{S}_i^t = \alpha \odot S_i^t + (1-\alpha) \odot \hat{S}_i^{t-1} \tag{12}$$

where $\alpha \in [0,1]$ controls how quickly new behavior affects trust. Lower α values emphasize long-term consistency over recent performance.

3) Homomorphic Trust Weight Normalization: Since cosine similarity yields values in $[-1,1]$, trust scores are mapped to positive weights using a homomorphic affine transformation in the following equation:

$$w_i^t = \text{PolyApprox}\left(\max\left(\frac{\hat{S}_i^t + 1}{2}, \epsilon\right)\right) \tag{13}$$

where ϵ is a small positive constant to prevent zero weights.

4) Homomorphic Weighted Aggregation: The global model update ($\mathcal{G}_{enc}$) is computed via trust-weighted homomorphic aggregation, which sums up the encrypted trust-weighted model updates at the global aggregator. The weighted aggregation leverages CKKS scalar multiplication to directly apply trust weights to encrypted updates in the following equation:

$$\mathcal{G}_{enc}{}^t = \sum_{i=1}^N \widetilde{\Delta}_i^t = \sum_{i=1}^N \hat{S}_i^t \odot \text{Enc}\left(\Delta_i^t\right) \tag{14}$$

where $\odot$ denotes CKKS homomorphic scalar multiplication, preserving both privacy and dynamic scaling properties established in the encryption phase.

A global aggregator receives the partial aggregated results from different fog nodes and performs the final aggregation of the encrypted model updates using pk_v. This approach ensures that clients with low trust contribute less to the global model, and all trust computations and aggregation steps are performed on encrypted data, eliminating the need for decryption at the aggregator and preserving client privacy throughout. Polynomial approximations for nonlinear operations are chosen to balance computational efficiency and accuracy within the CKKS noise budget. Since trust evaluation and aggregation are both homomorphic, the system is resistant to inference and poisoning attacks, as adversaries cannot manipulate or observe plaintext updates or trust scores.

Algorithm 3 Trust Score and Model Update Logging Smart Contract

Input: Participant address p, trust score S, global model CID CID, round r
Output: On-chain trust log, model log

Initialization: Authorize FL participants p;
Function `LogTrustScore(`p, S, r`)`**:**
  TrustScore$[p][r] \leftarrow S$;
Function `GetTrustScore(`p, r`)`**:**
  return TrustScore$[p][r]$;
Function `LogGlobalModel(`CID, r`)`**:**
  ModelLog$[r] \leftarrow CID$;
Function `GetGlobalModel(`r`)`**:**
  return ModelLog$[r]$;

G. Trust Score and Model Update Logging to the Blockchain

The study implements decentralized trust-score management and model-update storage. It employs PureChain [49], a custom permissioned blockchain network that uses the proof of authority and association (PoA^2) consensus mechanism, as shown in Fig. 3. The PoA^2 consensus mechanism enhances traditional PoA by introducing an idle-redundant signer mechanism that automatically replaces inactive signers. PoA^2 ensures uninterrupted consensus, mitigates the risk of downtime due to validator or miner failures, and maintains consistent transaction throughput under dynamic network conditions [50]. The proposed PureFL framework implements three signers (miners) that mine PureFL transactions and one redundant signer, which is automatically activated to replace any signer that becomes unresponsive or fails to function correctly. The quality of each vehicle's participation, represented as trust scores, is transparently recorded on the PureChain network. In addition, the fully aggregated global model is stored on the IPFS, and its content identifier (CID) is recorded on the blockchain. This separation allows encrypted model updates to be stored off-chain, significantly reducing blockchain storage and computational overhead. Algorithm 3 details the smart contract functions required for this implementation.

By integrating HE with blockchain, the proposed system provides a dual layer of security: HE preserves the confidentiality of model content, while the blockchain provides an

immutable, transparent, and auditable record of trust scores and model updates. This is vital for safety-critical vehicular network applications, where both data privacy and operational integrity are paramount.

H. Decryption of the Global Model at the Client

Vehicles retrieve the global model from IPFS using its CID, which is stored on the blockchain. Each vehicle decrypts the global model using its private key, sk_v, and continues participating in the FL process.

Algorithm 4 Encrypted Global Model Decryption

Require: Aggregated ciphertext chunks $\mathcal{G}_{enc} = \{g_1, g_2, \ldots, g_k\}$, secret key sk

Ensure: Flattened decrypted global model $\mathcal{G}_{plain}$

Initialize empty list $\mathcal{G}_{plain} \leftarrow [\,]$ **for** *each chunk* $g_i \in \mathcal{G}_{enc}$ **do**

    end

    Decrypt g_i using secret key: $g_i^{dec} \leftarrow$ CKKS_Decrypt(g_i, sk)

Append decrypted vector g_i^{dec} to $\mathcal{G}_{plain}$

Flatten $\mathcal{G}_{plain}$ into a single vector

return $\mathcal{G}_{plain}$

The accuracy of the decrypted value depends on both the scaling factor Δ and the noise accumulated during encryption and homomorphic operations. This enables secure, privacy-preserving processing and safeguarding of sensitive vehicular data across the network. Algorithm 4 illustrates the decryption process.

I. Mitigating Data Leakage

Data leakage is a challenge in both FL and HE frameworks. The PureFL framework addresses data leakage at various stages of the pipeline, as summarized in the following.

1) Local Anomaly Detection as Preupload Filter: Each vehicle performs intrusion detection on its local data before participating in federated training. If attacks are detected, the vehicle discards its model update rather than uploading it. This mechanism prevents indirect data leakage, where an adversary poisons a vehicle's local training data to manipulate its model updates and reveal information about other participants. By filtering compromised updates at the source, PureFL prevents leaked or malicious data from propagating through the federated system. In addition, only encrypted gradient chunks are transmitted, ensuring that no raw features, labels, or intermediate activations leave the vehicle.

2) End-to-End CKKS Encryption: All client model updates are encrypted using CKKS before transmission. This means that model updates encrypted with CKKS never exist in plaintext outside the originating vehicle. All participating nodes operate only on ciphertexts and never access plaintext model parameters, thereby preventing gradient inversion and model reconstruction attacks. In addition, trust is computed on homomorphically encrypted updates to ensure that neither the fog layer nor any other participant has access to plaintext trust values or model updates, eliminating leakage through trust-weight manipulation.

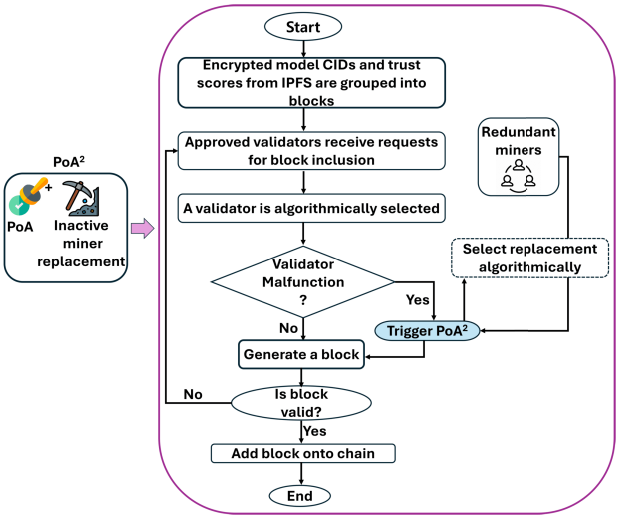


Fig. 3. Workflow of validator selection and redundancy handling in PoA^2 consensus for secure model update logging in PureFL [49].

3) Aggregator Privacy Through Homomorphic Computation: Fog and global aggregators operate exclusively on encrypted data, never requiring decryption to perform weighted aggregation. This eliminates the honest-but-curious aggregator threat, where aggregators might attempt to infer sensitive information from plaintext model updates. Traditional centralized FL systems that decrypt updates for aggregation expose vehicles to privacy risks even when aggregators are not explicitly malicious. PureFL's homomorphic trust-weighted aggregation ensures that trust score updates and model combination all occur in the encrypted domain, preventing aggregators from accessing or inferring vehicle-specific information.

4) IPFS Hash Storage on the Blockchain: No sensitive data are stored on-chain. The blockchain layer records only the IPFS CID of encrypted aggregated models and encrypted trust logs. Model CIDs are cryptographic hashes of encrypted models stored on IPFS. Therefore, accessing the underlying model requires both the CID and the secret key sk_v for decryption. Blockchain observers can verify participation and trust evolution without learning anything about individual vehicles' data distributions or model contents.

5) Client-Side Final Decryption Only: The global model retrieved from IPFS can only be decrypted by legitimate vehicles holding the secret key. Thus, no intermediary node can observe plaintext model parameters.

Through these strategies at each stage of the PureFL pipeline, potential leakage of model updates, trust scores, or training data from local training through encrypted aggregation to final model distribution is eliminated.

IV. Experimentation and Result Discussion

This section analyzes the experimental results to investigate the robustness and reliability of the proposed system.

TABLE II

ATTACK DISTRIBUTION IN CICIOV2024 AND 5G-NIDD DATASETS. ✓ INDICATES ATTACK PRESENCE; ✗ INDICATES ATTACK ABSENCE

Attack Description	CICIoV2024	5G-NIDD
Denial of Service (DoS)	✓	✓
Spoofing (gas, speed, RPM, steering_wheel)	✓	✗
Distributed Denial of Service (DDoS)	✗	✓
UDP Flood	✗	✓
SYN Flood	✗	✓
HTTP Flood	✗	✓
UDP Scan	✗	✓
TCP Connect Scan	✗	✓
SYN Scan	✗	✓
ICMP Flood	✗	✓

A. Data Scenarios and Preprocessing

The study utilizes the CICIoV2024 [51] and 5G-NIDD datasets [52]. The CICIoV2024 dataset contains false messages injected via a 2019 Ford Car's CAN Bus protocol, which covers all ECUs. It has 153 input features and four response classes. The 5G-NIDD dataset comprises data from a 5G testbed, including attack scenarios such as denial-of-service attacks and port scans. It has 46 input features and nine response classes. Table II shows the distribution of attacks in both datasets.

Data preprocessing is crucial in model learning and involves various steps [53]. Mean imputation replaced missing values with the average observed value. Categorical variables were one-hot encoded, and finally, features were normalized to a standard range. The preprocessed dataset is split into two stages: subsets depending on the number of clients, and then, each subset is further split into training and validation sets; 10% of each client's partition is allocated to the validation set, and 90% is allocated to the training set. A separate test set is created from the entire dataset for evaluation, without further partitioning.

B. Simulation Environment Configurations

The experimentation was done in Python on PyCharm, on a system with an Intel[1] Core[2] i5-8500 CPU @ 3.00 GHz, 32-GB RAM, and an NVIDIA GeForce GTX 1050, running Windows 11 Pro. The FL system was developed using the Flower framework version 1.11.0 within a PyTorch environment of version 2.4.1. The TenSEAL library, built on top of Microsoft SEAL, was used to implement CKKS HE. A custom blockchain network, PureChain [49], was employed with smart contracts written in Solidity. PureFL leverages web3.py to facilitate communication between smart contracts, the FL environment, and the PureChain network. Simulation parameters considered for experimentation are shown in Table III.

The proposed framework implements a scaling strategy in which Δ_i decreases with the operation depth d to control noise accumulation during homomorphic operations. A minimum scale threshold $\Delta_0/2^{20}$ is considered to prevent precision underflow in deep computations.

[1]Registered trademark.
[2]Trademarked.

TABLE III

SIMULATION PARAMETERS

Parameter	Value
FL Clients	60
FL Rounds	20
LSTM Layers	2
Hidden Size	128
Batch Size	32
Learning Rate	0.001
Activation Function	Adam
Loss Function	Cross-Entropy
Base Scale	2^{40}
Polynomial modulus degrees (n)	8192, 16384, 32768
Coefficient modulus	[60, 40, 40, 40, 40, 60]
Chunk Size	4096
Scaling strategy	exponential
Reduction Factor	0.7
Minimum scale	2^{20}
Paillier modulus size (Comparison)	2048

C. Analysis of Component-Specific Contributions

PureFL combines intrusion detection, HE, trust-weighted aggregation, and blockchain logging to jointly provide robustness, confidentiality, and accountability in IoV FL. The integrated components target different guarantees, and therefore, contributions are reflected differently. HE primarily protects the confidentiality of updates, and blockchain logging provides tamper-evident auditability. In contrast, robustness under adversarial participation is mainly driven by intrusion detection and trust-weighted aggregation, which reduce the influence of unreliable or malicious updates.

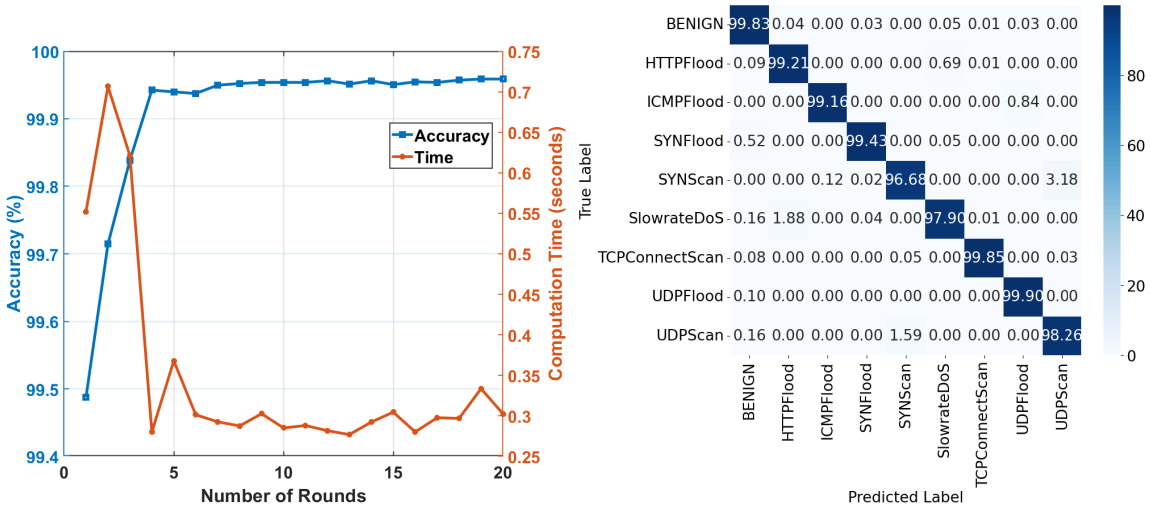
Table IV presents a qualitative ablation comparison across representative configurations. This comparison highlights that removing trust-weighting most directly weakens poisoning resilience; removing intrusion detection reduces update-quality assurance at the client level; removing blockchain logging eliminates traceability without materially affecting predictive performance; and removing HE breaks confidentiality even if robustness mechanisms remain enabled. This analysis makes explicit which components drive robustness versus privacy and accountability, and which components introduce the primary system overhead. Further evaluation of these components is presented in Sections IV-D–IV-H.

D. Federated IDS Model Performance

Fig. 4(a) demonstrates the model's efficient convergence, achieving 99.9% accuracy within five rounds in the 5G-NIDD data scenario. Computation time per round decreases sharply initially, stabilizing at 0.3 s, reflecting computational efficiency due to reduced gradient updates. The confusion matrix in Fig. 4(b) highlights precise classification with significant diagonal values (99.83% for benign traffic and 99.90% for UDPFlood). Minimal misclassification, such as 1.88% of SlowrateDoS traffic being identified as HTTPFlood and 3.18% of SYNScan traffic as UDPScan, suggests overlapping feature spaces that could benefit from refined feature engineering or training.

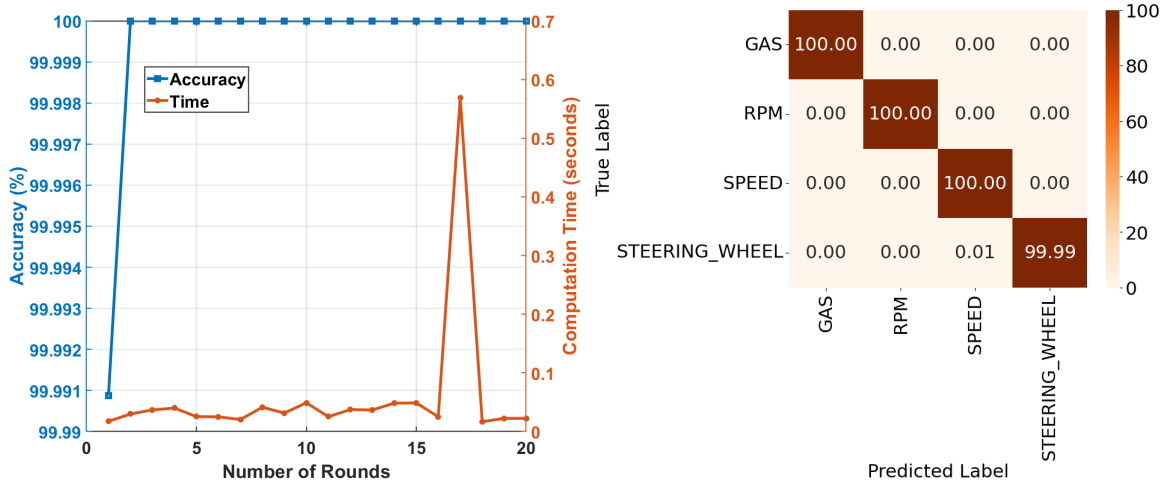
TABLE IV
COMPONENTWISE QUALITATIVE ABLATION COMPARISON OF THE PUREFL FRAMEWORK

Configuration	IDS	Trust Weighting	Homomorphic Encryption	Blockchain Integration	Robustness to Poisoning	Privacy Preservation	Auditability	Overhead
FedAvg	✗	✗	✗	✗	Very Low	✗	✗	Very Low
HE-FL only (CKKS + FedAvg)	✗	✗	✓	✗	Low	Strong	✗	Low
PureFL without IDS	✗	✓	✓	✗	Medium	Strong	✓	High
PureFL without Trust	✓	✗	✓	✓	Low	Strong	✓	Medium
PureFL without Blockchain	✓	✓	✓	✗	High	Strong	✗	Medium
PureFL (Integrated)	✓	✓	✓	✓	**High**	**Strong**	✓	**High**



(a) (b)

Fig. 4. Demonstration of the model performance on the 5G-NIDD data scenario. (a) Accuracy over computation time at 20 rounds. (b) Confusion matrix of the model performance.



(a) (b)

Fig. 5. Illustration of the model performance on the CICIoV2024 data scenario. (a) Accuracy over computation time at 20 rounds. (b) Confusion matrix of the model performance.

Similarly, the visualizations in Fig. 5(a) demonstrate a highly optimized classification model with notable performance in the CICIoV2024 data scenario. Accuracy (black) reaches 100% early, indicating rapid convergence and strong predictive capability. Computation time (orange) remained consistent except for a spike at the 15th round, possibly due to an anomaly like a bottleneck or model update. Consequently, the confusion matrix in Fig. 5(b) confirms outstanding

TABLE V
TIME EVALUATION ACROSS VARYING POLYMODULUS DEGREES

Scheme	Dataset	CICIoV2024			5G-NIDD		
	Key Length	8192	16384	32768	8192	16384	32768
Static CKKS	Key Generation Time (s)	0.563	1.914	24.634	0.446	3.153	26.127
	Encryption Time (s)	1.862	5.036	154.059	1.238	9.177	156.899
	Decryption Time (s)	1.548	3.2271	48.517	1.258	5.338	51.656
Proposed dynamic CKKS with precision scaling	Key Generation Time (s)	0.487	0.8775	23.892	0.557	4.036	25.656
	Encryption Time (s)	0.783	3.942	62.391	0.885	2.477	74.643
	Decryption Time (s)	0.427	1.656	13.525	0.453	1.5176	16.744

TABLE VI
TIME COMPARISON OF THE PROPOSED SCALED CKKS WITH PAILLIER HE SCHEME ON THE CICIOV2024 AND 5G-NIDD DATASETS

DATASET	SCHEME	Key Generation Time	Encryption Time	Decryption Time
CICIoV2024	Paillier	11.5305	76512.9758	22119.57
	Scaled CKKS	12.892	62.391	13.525
5G-NIDD	Paillier	6.9499	75603.9713	22015.9268
	Scaled CKKS	20.656	75.643	16.744

classification accuracy (100% for most classes and 99.99% for STEERING_WHEEL with 0.01% misclassification). Minimal errors and consistently high accuracy highlight the model's robustness though the computation spike and minor misclassification warrant minor investigation.

E. CKKS With Dynamic Precision Scaling Evaluation

Key generation, encryption, and decryption times were evaluated for the proposed framework using polynomial modulus degrees (n) of 8192, 16 384, and 32 768 with a coefficient modulus of $[60, 40, 40, 40, 40, 60]$ both with and without the application of dynamic precision scaling. Table V shows that computation overhead increases with an increase in n due to the growing complexity of homomorphic computations and ciphertext size, highlighting a tradeoff between security and performance. In addition, even with increasing modulus sizes, the efficiency of the dynamic CKKS outperforms static CKKS because the dynamic precision scaling mechanism effectively manages noise budget utilization, reducing unnecessary rescaling operations and optimizing parameter selection based on real-time noise levels.

Comparison experimentation results as shown in Table VI between CKKS ($n = 32\,768$) and the basic Paillier modulus size ($n = 2048$) demonstrate that while Paillier achieved faster key generation time, CKKS significantly outperforms it in encryption and decryption, making it ideal for performance-critical applications. It confirms the superiority of scaled CKKS with a significantly high polynomial modulus degree over Paillier, even with the basic 2048 modulus size.

Fig. 6 compares encryption and decryption times between existing HE studies and the proposed approach in PureFL, illustrating that the proposed dynamic precision scaling with the CKKS scheme significantly reduces encryption and decryption times, achieving the lowest recorded values for both operations. This highlights the improved efficiency of our approach compared to previous work.

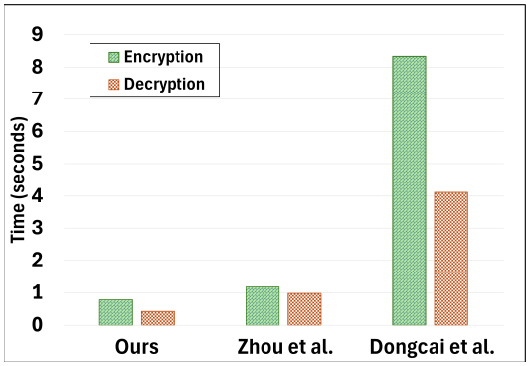


Fig. 6. Comparison graph showing encryption and decryption time of existing studies and the proposed approach.

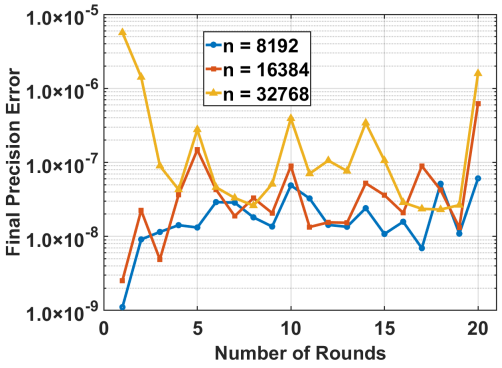


Fig. 7. Final precision error for the 5G-NIDD dataset.

F. Error Analysis and Precision Management

In the PureFL framework, trust-weighted aggregation requires evaluating noise accumulation and precision errors from both homomorphic addition and scalar multiplication during model aggregation. While homomorphic addition in CKKS is relatively noise-efficient, scalar multiplication, necessary for applying trust weights, leads to higher noise growth. Each aggregation round involves multiplying encrypted model updates by trust weights and summing the resulting ciphertexts. The approximate CKKS encoding causes scale reduction and noise propagation with each operation, which, if not managed, can degrade decryption accuracy and the fidelity of the aggregated global model.

1) Final Precision Error: This quantifies the deviation between the expected plaintext weighted sum of model updates and the decrypted result after trust-weighted aggregation. As shown in Figs. 7 and 8, the error remained low, indicating that CKKS successfully preserved model fidelity in both data scenarios. However, a trend of increasing error with larger n was observed, with $n = 32\,768$ yielding the largest deviation. This is attributed to deeper modulus chains and increased sensitivity to rounding errors due to larger ciphertexts. Nonetheless, even at $n = 32\,768$, the decrypted values did not significantly deviate from the true values, and the precision error remained within acceptable bounds for anomaly detection use cases.

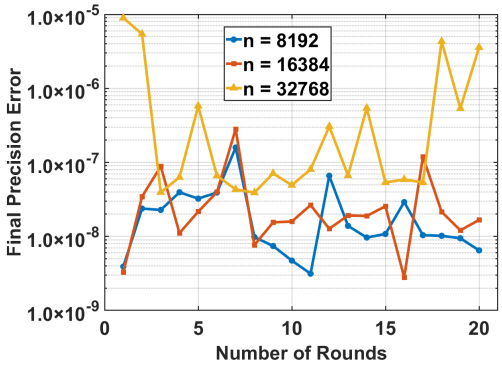


Fig. 8. Final precision error for the CICIoV2024 dataset.

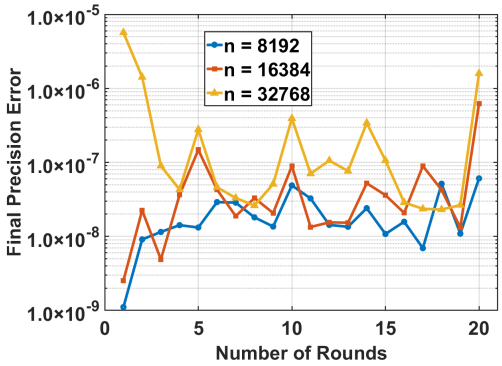


Fig. 9. Average aggregation noise for the 5G-NIDD dataset.

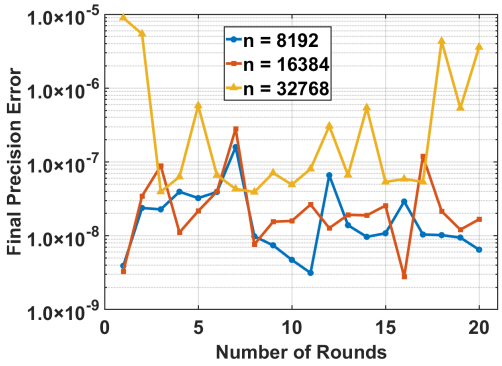


Fig. 10. Average aggregation noise for the CICIoV2024 dataset.

2) Average Aggregation Noise: This reflects the mean deviation introduced per aggregation step. It reflects the noise growth from homomorphic operations applied to encrypted model updates across multiple clients and rounds. Figs. 9 and 10 show that $n = 32\,768$ experienced the highest average noise, followed by $n = 16\,384$ and $n = 8192$, consistent with the trend observed in the final precision error. This highlights the tradeoff between security (higher n) and noise resilience, as more complex ciphertext operations increase the cumulative noise budget consumption.

3) Maximum Aggregation Noise: This represents the worst case noise observed during any aggregation step in each round. As shown in Figs. 11 and 12, despite fluctuations, all maximum noise values remained below 1×10^{-8}, demonstrating the robustness of the scaling and encryption parameters.

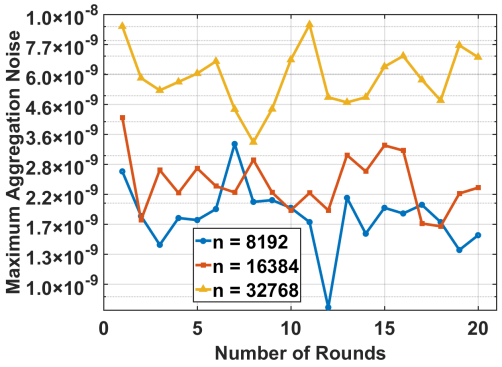


Fig. 11. Maximum aggregation noise for the 5G-NIDD dataset.

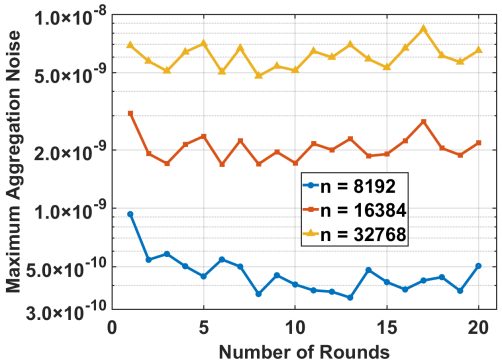


Fig. 12. Maximum aggregation noise for the CICIoV2024 dataset.

This confirms that the global scale and coefficient modulus configurations prevented excessive noise buildup, even under large model sizes with an increasing number of FL rounds.

Overall, these results validate that trust-weighted aggregation with CKKS can achieve low-noise, high-precision global model computation across multiple rounds of FL. While small errors in a single round may be tolerable, unmanaged noise across multiple rounds can lead to misaligned gradients, poor global model convergence, and potential inference errors. PureFL, therefore, mitigates these limitations and achieves balance, preserving accuracy without introducing runtime bottlenecks, making it suitable for secure, trust-based FL in vehicular network environments.

G. Robustness of Trust-Weighted Aggregation Under Adversarial Conditions

The resilience of the proposed trust-weighted aggregation mechanism under various adversarial conditions is evaluated. The evaluation substantiates the security analysis and validation of the proposed approach, particularly for the trust-weighted aggregation. For each attack type and ratio of malicious clients (0%, 10%, 30%, and 50%), the experiments are repeated for multiple trust-update parameters $\alpha \in \{0.2, 0.5, 0.8\}$ using the cosine similarity. The classes of adversarial clients are simulated as follows.

1) *Sybil Attack*: Tests the system's ability to handle a large number of malicious clients masquerading as honest participants.

TABLE VII
TRUST-WEIGHTED AGGREGATION PERFORMANCE UNDER ADVERSARIAL THREATS

Attack Type	Attack Ratio	$\alpha = 0.2$			$\alpha = 0.5$			$\alpha = 0.8$		
		MTA	**ASR**	**CI**	**MTA**	**ASR**	**CI**	**MTA**	**ASR**	**CI**
backdoor	**0%**	0.996	0	0.9973	0.996	0	0.998	0.996	0	0.999
	10%	0.996	0.023	0.779	0.996	0.022	0.486	0.996	0.022	0.194
	30%	0.997	0.024	0.778	0.997	0.0245	0.486	0.997	0.024	0.194
	50%	0.988	0.032	0.765	0.988	0.031	0.478	0.988	0.032	0.191
sybil	**0%**	0.997	0	0.997	0.996	0	0.998	0.996	0	0.999
	10%	0.992	0.08	0.797	0.990	0.10	0.498	0.988	0.12	0.199
	30%	0.980	0.15	0.798	0.975	0.18	0.498	0.970	0.22	0.199
	50%	0.960	0.25	0.60	0.950	0.30	0.40	0.940	0.35	0.20
random noise	**0%**	0.996	0	0.997	0.996	0	0.998	0.996	0	0.999
	10%	0.996	0.011	0.7888	0.996	0.0106	0.493	0.996	0.0106	0.197
	30%	0.997	0.0125	0.788	0.997	0.0125	0.492	0.997	0.0125	0.197
	50%	0.987	0.020	0.774	0.987	0.020	0.483	0.987	0.020	0.193
scaled attack	**0%**	0.996	0	0.9973	0.996	0	0.998	0.996	0	0.999
	10%	0.992	0.45	0.45	0.990	0.50	0.35	0.988	0.55	0.25
	30%	0.980	0.55	0.35	0.975	0.60	0.25	0.970	0.65	0.15
	50%	0.965	0.65	0.25	0.955	0.70	0.15	0.945	0.75	0.08
label flip	**0%**	0.996	0	0.997	0.996	0	0.998	0.996	0	0.999
	10%	0.990	0.08	0.78	0.988	0.10	0.48	0.985	0.12	0.20
	30%	0.975	0.15	0.72	0.970	0.18	0.45	0.965	0.22	0.18
	50%	0.950	0.25	0.60	0.935	0.30	0.38	0.920	0.35	0.15

2) *Random Noise Attack*: Malicious clients upload arbitrarily perturbed noisy updates to corrupt the global model.
3) *Label Flip Attack*: Malicious clients attempt to degrade the global model's performance by training on deliberately mislabeled data, leading to model poisoning.
4) *Scaled Attack*: Malicious clients scale their model updates to appear benign, directly challenging the similarity-based detection by dominating the aggregation.
5) *Backdoor Attack*: Malicious clients embed a specific pattern into their updates so that the global model misclassifies a trigger pattern while keeping main accuracy high.

Table VII presents the average performance of 60 clients over one round, which were evaluated with the performance metrics in the following.

1) *Main-Task Accuracy (MTA)*: MTA is the average cosine similarity between benign client updates and the aggregated global update. Values close to 1 indicate that the global model is highly aligned with honest clients (good overall performance), while values near 0 indicate that the model has been severely corrupted.
2) *Attack Success Rate (ASR)*: ASR quantifies the extent to which the malicious objective influences the global model. A high ASR (near 1) indicates a successful attack, and the global model was strongly influenced by it. A low ASR (near 0) indicates that the attack has little effect.
3) *Consensus Integrity (CI)*: CI reflects the quality of the trust mechanism itself, and it is the difference between the average trust of benign and malicious clients. When $CI > 0$, benign clients are more trusted than attackers, proving that the trust mechanism is behaving correctly. When $CI < 0$, attackers are more trusted than benign clients, and the trust mechanism is being fooled.

The trust update is governed by the parameter α in (12), which controls the balance between historical trust and current-round similarity. As shown in Table VII, α has a pronounced effect on consensus integrity. Smaller values ($\alpha = 0.2$) make trust highly responsive to recent behavior, yielding larger CI gaps and more aggressive down-weighting of malicious clients, at the cost of potentially greater variability for honest clients whose updates occasionally deviate from the consensus. In contrast, larger values ($\alpha = 0.8$) make trust more persistent over time. MTA remains very stable, but trust adapts more slowly to newly malicious behavior, past behavior dominates, and the CI is compressed, limiting how much attacker trust can be reduced within a single round.

As α increases from 0.2 to 0.8, the CI value systematically decreases. For the backdoor attack scenario at 50% attack ratio, for example, the CI decreases from 0.765 at $\alpha = 0.2$ to 0.191 at $\alpha = 0.8$. This suggests that giving more weight to historical trust ($\alpha = 0.8$) is less effective than prioritizing current-round similarity ($\alpha = 0.2$). Because the adversarial updates are consistently different from benign ones, a high reliance on current similarity is sufficient to filter them out based on their present behavior. Relying on historical trust allows a minor, yet measurable, degradation of consensus over time, even if the ASR remains low.

Overall, these results indicate that, under realistic adversarial scenarios, PureFL's trust-weighted aggregation can maintain model integrity and substantially enhance robustness against malicious clients. In addition, suitable choices of α allow a tradeoff between responsiveness and stability.

H. Blockchain Performance and Scalability Evaluation

The performance and scalability of PureChain, based on the PoA^2 consensus mechanism, are evaluated under increasing client batch sizes (10–60 clients) within the PureFL framework. For comparison, we include Hyperledger Fabric (Kafka-based) and Ethereum Sepolia (PoS-based) networks.

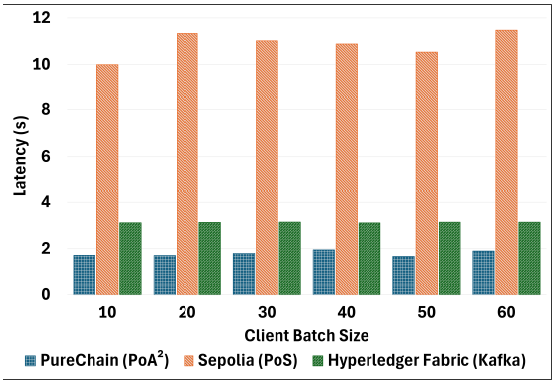


Fig. 13. PureChain latency in comparison to other blockchain networks in PureFL.

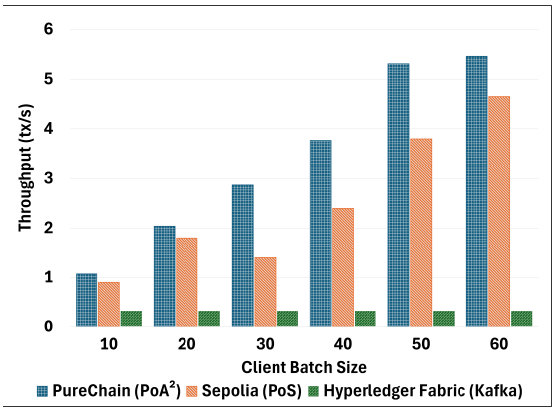


Fig. 14. PureChain throughput in comparison to other blockchain networks in PureFL.

Latency is measured as the average time required to upload trust scores and record hashes of encrypted model updates on the blockchain. As illustrated in Fig. 13, PureChain consistently exhibits the lowest latency across client batch sizes, remaining under 2 s. In contrast, Hyperledger Fabric exhibits a minimum of 3.11 s, while Ethereum Sepolia shows significantly higher latency, starting at 9.99 s. These results demonstrate that PureChain provides a more scalable, real-time-compatible consensus layer for secure model update logging in vehicular FL systems.

Throughput represents the number of encrypted model updates processed and hashed as transactions per second (tx/s). Fig. 14 reinforces PureChain's high scalability in comparison to the other blockchain networks. PureChain achieves an increase in throughput from $1.07tx/s$ at ten clients to $5.46tx/s$ at 60 clients, while Ethereum Sepolia 0.9 tx/s at ten clients to $4.64tx/s$ at 60 clients. Hyperledger Fabric maintains a constant $0.32tx/s$ across all client batch sizes.

Latency and throughput variations across blockchain platforms arise from the limitations of their consensus protocols. Sepolia, based on PoS, requires multiple block confirmations for finality, with strict constraints on block sizes and transaction throughput to prevent spam and maintain fairness, thereby limiting scalability under heavy loads. Hyperledger Fabric, which uses a Kafka-based ordering service, encounters centralized bottlenecks, where Kafka's batching and queuing mechanisms introduce latency and compromise high-throughput performance. In contrast, the PureChain network, utilizing PoA^2 consensus, benefits from an idle redundant signer that replaces inactive validators, ensuring continuous consensus and high transaction throughput even under dynamic conditions.

TABLE VIII

COMPARISON OF THE PROPOSED POA^2 IN PUREFL WITH EXISTING APPROACHES: ✗ INDICATES NO EVALUATION

Approach	Consensus	Offchain Storage	Latency (s)	Throughput (tx/s)
Xiaoyan et al. [26]	Raft	✗	19.13	✗
PBFL [29]	*Not specified*	✗	✗	✗
BUAMH [31]	Identity consensus mechanism	✗	3.4	✗
BPFL [32]	Practical Byzantine Fault Tolerance (PBFT)	✗	3.6	✗
PureFL	**PoA^2**	**IPFS**	**1.79**	**3.42**

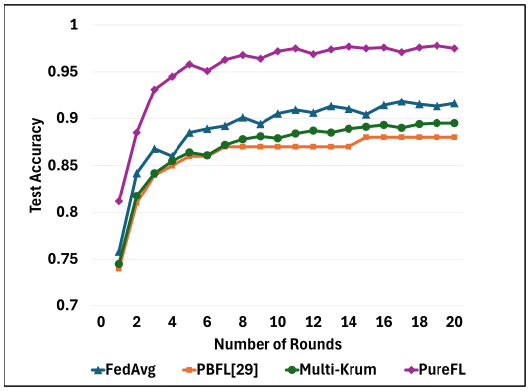


Fig. 15. Accuracy evolution of FL frameworks under a 30% label-flip attack.

We assess the robustness of the PoA^2 consensus mechanism in PureFL against existing blockchain-integrated FL approaches using HE. As shown in Table VIII, PureFL with PoA^2 achieves the lowest latency compared to other consensus mechanisms. Importantly, most prior works did not evaluate blockchain throughput, a critical gap in their analyses. Our study, however, provides a comprehensive evaluation of both latency and throughput, offering a more complete understanding of blockchain performance in FL scenarios.

I. End-to-End Performance Evaluation

Fig. 15 presents the end-to-end accuracy evolution of different FL frameworks under a 30% label-flip poisoning attack over 20 communication rounds. Vanilla FL (FedAvg) and the distance-based robust aggregation method, Multi-Krum [32], are evaluated on the same dataset as PureFL. In addition, PBFL [29], which uses cosine similarity to identify malicious gradients, is included as a reference baseline to contextualize performance relative to existing work.

FedAvg exhibits limited robustness due to the lack of a defense mechanism, leading to slower convergence and lower steady-state accuracy. Multi-Krum and PBFL improve robustness by filtering anomalous updates. However, their convergence remains slower, and their final accuracy remains below that of the proposed PureFL approach. In contrast, PureFL achieves the fastest convergence and steady accuracy,

reaching approximately 98% accuracy under adversarial participation. These results demonstrate that PureFL maintains high predictive performance while simultaneously providing privacy and auditability guarantees.

J. Theoretical Security Analysis of PureFL

The security of the proposed PureFL framework is grounded on two cryptographic assurances: the cryptographic strength of the CKKS HE scheme under the RLWE assumption, and the tamper-evident, auditable record-keeping provided by the PureChain blockchain. The following analysis details both the theoretical and practical assurances offered by these components.

1) Approximate Semantic Security Under RLWE: The CKKS scheme in PureFL ensures IND-CPA security based on the computational hardness of RLWE, which is considered quantum-resistant. Model updates are encrypted with raw noise during encoding to prevent the disclosure of information about the model weights or data. Even if ciphertexts are intercepted, the RLWE assumption guarantees no information leakage without the private key. PureFL also utilizes ciphertext packing and chunkwise encryption to introduce semantic variation, thereby mitigating risks associated with comparison or inference attacks. Decryption occurs solely on the client side, ensuring end-to-end confidentiality by keeping the secret key out of the reach of fog and cloud nodes.

2) Dynamic Noise and Precision Management: In CKKS, homomorphic operations, particularly multiplication, introduce noise that reduces ciphertext scale, potentially exceeding the noise budget and causing decryption errors or precision loss. Dynamic precision scaling mitigates this issue by adjusting the scaling factor after each operation, thereby maintaining the scale within bounds for correct decryption throughout the aggregation rounds. Homomorphic scalar multiplication by a public weight w_i preserves IND-CPA security, ensuring no additional information leakage beyond $\text{Enc}(\Delta_i^t)$. The CKKS security proof, under the RLWE assumption, guarantees that no polynomial-time adversary can distinguish between encryptions of chosen plaintexts. Thus, confidentiality and privacy are preserved, preventing both peer clients and aggregators from inferring private updates during the aggregation process.

3) Blockchain Integrity and Decentralized Auditability: PureChain supports the integrity and auditability of the FL process. Encrypted model updates and trust scores are stored on IPFS, with their CIDs recorded on PureChain. This architecture separates storage from verification, ensuring that every log entry, comprising the tuple (CID_r, S_i^t) for round r and client i, is tamper-evident, auditable, and cryptographically linked to the client's identity. The use of blockchain smart contracts guarantees that all updates and trust scores are transparently recorded and verifiable by authorized participants, without overburdening the blockchain with large data payloads.

4) Summary of Security Guarantees: The combined use of RLWE-based CKKS encryption, dynamic precision scaling, and blockchain-backed logging ensures confidentiality by protecting raw data and model updates from intermediaries. It maintains aggregation integrity by mitigating inference and poisoning attacks through trust-weighted aggregation and encrypted computation. In addition, auditability and tamper-resistance are achieved through immutably logged trust scores and update histories, ensuring transparent accountability. These features make the PureFL framework resilient against adversarial behaviors, such as eavesdropping, model poisoning, and log tampering, thereby ensuring secure and real-time operation in vehicular networks.

V. Conclusion

This study introduces PureFL, a privacy-preserving FL framework designed for IoV. PureFL integrates CKKS HE with dynamic precision scaling and blockchain-based decentralized trust management to ensure secure and efficient collaborative learning. By leveraging CKKS encryption alongside a permissioned blockchain, the framework enables privacy-preserving aggregation of encrypted model updates without relying on a central server, thereby eliminating single points of failure. Extensive experiments on the CICIoV2024 and 5G-NIDD datasets validate the framework's effectiveness in anomaly detection, achieving near-perfect classification accuracy with minimal misclassification across diverse attack scenarios. The integration of dynamic precision scaling significantly enhances the efficiency of CKKS operations by reducing noise growth, leading to faster and more stable encryption and decryption compared to traditional methods. Security analysis by evaluating the robustness of PureFL's trust-weighted aggregation of encrypted model updates under adversarial conditions shows that the system is effectively resilient against adversarial clients without degrading the performance of honest participants. Furthermore, blockchain performance evaluation demonstrates PureChain's scalability and real-time readiness. Under increasing client loads, PureChain outperformed Hyperledger Fabric and Ethereum Sepolia in both latency and throughput, affirming its suitability for time-sensitive IoV environments. While the results highlight PureFL's potential in enabling secure, decentralized, and scalable FL for vehicular networks, the integration of a fully realized PoA^2-based incentive mechanism will be explored to promote reliable client participation and long-term system sustainability.

References

[1] M. Alam, J. Ferreira, and J. Fonseca, "Introduction to intelligent transportation systems," in *Intelligent Transportation Systems: Dependable Vehicular Communications for Improved Road Safety*. Cham, Switzerland: Springer, 2016, pp. 1–17.

[2] S. Adnan Yusuf, A. Khan, and R. Souissi, "Vehicle-to-everything (V2X) in the autonomous vehicles domain—A technical review of communication, sensor, and AI technologies for road user safety," *Transp. Res. Interdiscipl. Perspect.*, vol. 23, Jan. 2024, Art. no. 100980.

[3] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 760–776, Feb. 2019.

[4] M. Chowdhury, A. Apon, and K. Dey, "Security and privacy solutions," in *Data Analytics for Intelligent Transportation Systems*. Amsterdam, The Netherlands: Elsevier, 2017.

[5] H. L. Nakayiza, L. Ahakonye, D.-S. Kim, and J. M. Lee, "Machine learning algorithms for detecting intra-vehicular data falsification," in *Proc. Symp. The Korean Inst. Commun. Inf. Sci. Summer Conf. (KICS)*, Jun. 2024, pp. 1088–1089.

[6] D. Man, F. Zeng, J. Lv, S. Xuan, W. Yang, and M. Guizani, "AI-based intrusion detection for intelligence Internet of Vehicles," *IEEE Consum. Electron. Mag.*, vol. 12, no. 1, pp. 109–116, Jan. 2023.

[7] H. L. Nakayiza, L. A. C. Ahakonye, D.-S. Kim, and J.-M. Lee, "Blockchain-enhanced feature engineered data falsification detection in 6G in-vehicle networks," *IEEE Internet Things J.*, vol. 12, no. 15, pp. 30036–30048, Aug. 2025.

[8] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Mar. 2021.

[9] A. A. Elkhail, R. U. D. Refat, R. Habre, A. Hafeez, A. Bacha, and H. Malik, "Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses," *IEEE Access*, vol. 9, pp. 162401–162437, 2021.

[10] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Novel hyper-tuned ensemble random forest algorithm for the detection of false basic safety messages in Internet of Vehicles," *ICT Exp.*, vol. 9, no. 1, pp. 122–129, Feb. 2023.

[11] J. Alsamiri and K. Alsubhi, "Federated learning for intrusion detection systems in Internet of Vehicles: A general taxonomy, applications, and future directions," *Future Internet*, vol. 15, no. 12, p. 403, Dec. 2023.

[12] T. Yuan, L. Chen, Y. Jiang, H. Chen, W. Gong, and Y. Gu, "Resource management and optimization in Internet of Vehicles for hierarchical federated learning," *IEEE Access*, vol. 12, pp. 158174–158188, 2024.

[13] D. M. Manias and A. Shami, "Making a case for federated learning in the Internet of Vehicles and intelligent transportation systems," *IEEE Netw.*, vol. 35, no. 3, pp. 88–94, May 2021.

[14] P. Rani et al., "Federated learning-based misbehavior detection for the 5G-enabled Internet of Vehicles," *IEEE Trans. Consum. Electron.*, vol. 70, no. 2, pp. 4656–4664, Feb. 2024.

[15] H. L. Nakayiza, L. A. C. Ahakonye, D.-S. Kim, and J. M. Lee, "Homomorphic encryption for privacy-preserving misbehavior detection in the Internet of Vehicles," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2025, pp. 0320–0324.

[16] B. D. Manh, C.-H. Nguyen, D. T. Hoang, and D. N. Nguyen, "Homomorphic encryption-enabled federated learning for privacy-preserving intrusion detection in resource-constrained IoV networks," in *Proc. IEEE 100th Veh. Technol. Conf. (VTC-Fall)*, Oct. 2024, pp. 1–6.

[17] J.-P. Hubaux, *Homomorphic Encryption*. Cham, Switzerland: Springer, 2023, pp. 35–39.

[18] X. Sun, F. R. Yu, P. Zhang, W. Xie, and X. Peng, "A survey on secure computation based on homomorphic encryption in vehicular Ad hoc networks," *Sensors*, vol. 20, no. 15, p. 4253, Jul. 2020.

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1999, pp. 223–238.

[20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *Electron. Colloq. Comput. Complex.*, vol. 18, p. 111, Jan. 2011.

[21] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2017, pp. 409–437.

[22] J. Grover, "Security of vehicular ad hoc networks using blockchain: A comprehensive review," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100458.

[23] H. Nakayiza, L. Ahakonye, D.-S. Kim, and J. M. Lee, "Blockchain-enabled intrusion detection system for distributed vehicular networks," in *Proc. Symp. Korean Inst. Commun. Inf. Sci. (KICS)*, Nov. 2024, pp. 1–11.

[24] A. Zainudin, M. A. P. Putra, R. N. Alief, R. Akter, D.-S. Kim, and J.-M. Lee, "Blockchain-inspired collaborative cyber-attacks detection for securing metaverse," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18221–18236, May 2024.

[25] F. Ayaz, Z. Sheng, D. Tian, M. Nekovee, and N. Saeed, "Blockchain-empowered AI for 6G-enabled Internet of Vehicles," *Electronics*, vol. 11, no. 20, p. 3339, Oct. 2022.

[26] X. Yan, Q. Wu, and Y. Sun, "A homomorphic encryption and privacy protection method based on blockchain and edge computing," *Wireless Commun. Mobile Comput.*, vol. 2020, no. 1, 2020, Art. no. 8832341.

[27] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10782–10793, Nov. 2020.

[28] D. Du, W. Zhao, L. Wei, S. Lu, and X. Wu, "A lightweight homomorphic encryption federated learning based on blockchain in IoV," in *Proc. IEEE Smartworld, Ubiquitous Intell. Comput., Scalable Comput. Commun., Digit. Twin, Privacy Comput., Metaverse, Auto. Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, Dec. 2022, pp. 1001–1007.

[29] Y. Miao, Z. Liu, H. Li, K.-K.-R. Choo, and R. H. Deng, "Privacy-preserving Byzantine-robust federated learning via blockchain systems," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2848–2861, 2022.

[30] J. Park and H. Lim, "Privacy-preserving federated learning using homomorphic encryption," *Appl. Sci.*, vol. 12, no. 2, p. 734, Jan. 2022.

[31] H. Xie, J. Zheng, T. He, S. Wei, and C. Hu, "A blockchain-based ubiquitous entity authentication and management scheme with homomorphic encryption for FANET," *Peer-Peer Netw. Appl.*, vol. 17, no. 2, pp. 569–584, Mar. 2024.

[32] N. Wang et al., "A blockchain-based privacy-preserving federated learning scheme for Internet of Vehicles," *Digit. Commun. Netw.*, vol. 10, no. 1, pp. 126–134, 2024.

[33] A. Alfardus and D. B. Rawat, "Machine learning-based anomaly detection for securing in-vehicle networks," *Electronics*, vol. 13, no. 10, p. 1962, May 2024.

[34] K. Zhang et al., "Intrusion detection model for Internet of Vehicles using GRIPCA and OWELM," *IEEE Access*, vol. 12, pp. 28911–28925, 2024.

[35] J. Li, S. Zhang, H. Zhu, Y. Chen, and J. Liu, "An efficient vehicular intrusion detection method based on edge intelligence," in *Proc. 27th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2024, pp. 2999–3004.

[36] L. Lihua, "Energy-aware intrusion detection model for Internet of Vehicles using machine learning methods," *Wireless Commun. Mobile Comput.*, vol. 2022, May 2022, Art. no. 9865549.

[37] X. Wang, Y. Zhu, S. Han, L. Yang, H. Gu, and F.-Y. Wang, "Fast and progressive misbehavior detection in Internet of Vehicles based on broad learning and incremental learning systems," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4788–4798, Mar. 2022.

[38] L. Jai Vinita and V. Vetriselvi, "Federated learning-based misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles," *Ad Hoc Netw.*, vol. 144, May 2023, Art. no. 103153.

[39] A. Roy and S. K. Madria, "BLAME: A blockchain-assisted misbehavior detection and event validation in VANETs," in *Proc. 22nd IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2021, pp. 69–78.

[40] H. L. Nakayiza, L. A. Chijioke Ahakonye, D.-S. Kim, and J. M. Lee, "Resource-aware adaptive federated learning for enhanced DDoS detection in vehicular ad hoc networks," in *Proc. 15th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2024, pp. 1262–1267.

[41] A. Boudguiga, O. Stan, A. Fazzat, H. Labiod, and P.-E. Clet, "Privacy preserving services for intelligent transportation systems with homomorphic encryption," in *Proc. ICISSP*, 2021, pp. 684–693.

[42] G. Costantino, M. De Vincenzi, F. Martinelli, and I. Matteucci, "A privacy-preserving solution for intelligent transportation systems: Private driver DNA," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 258–273, Jan. 2023.

[43] H. Karim and D. B. Rawat, "TollsOnly please—Homomorphic encryption for toll transponder privacy in Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2627–2636, Feb. 2022.

[44] European Automobile Manufacturers' Association. (May 2025). *Position Paper: Connected Vehicle Data Sharing*. [Online]. Available: https://www.acea.auto/files/ACEA_Position_paper_Connected_vehicle_data_sharing.pdf

[45] H. Zhang, M. Shang, H. Liu, and D. Zhang, "Lattice-based multi-key homomorphic encryption scheme without common random strings," *Symmetry*, vol. 17, no. 5, p. 722, 2025.

[46] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci.*, Oct. 2011, pp. 97–106.

[47] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[48] A. Kholod, Y. Polyakov, and M. Schlottke-Lakemper, "Secure numerical simulations using fully homomorphic encryption," *Comput. Phys. Commun.*, vol. 318, Jan. 2026, Art. no. 109868.

[49] D.-S. Kim, I. S. Igboanusi, L. A. Chijioke Ahakonye, and G. O. Anyanwu, "Proof-of-authority-and-association consensus algorithm for IoT blockchain networks," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2025, pp. 1–6.

[50] H. L. Nakayiza, L. Allen Chijioke Ahakonye, D.-S. Kim, and J.-M. Lee, "PureTrust: A soulbound token-based blockchain framework for incentive-driven trust management in V2X networks," *IEEE Internet Things J.*, vol. 13, no. 7, pp. 13734–13752, Apr. 2026.

[51] E. C. P. Neto et al., "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet Things*, vol. 26, Jul. 2024, Art. no. 101209.

[52] S. Samarakoon et al., "5G-NIDD: A comprehensive network intrusion detection dataset generated over 5G wireless network," 2022, *arXiv:2212.01298*.
[53] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "SCADA intrusion detection scheme exploiting the fusion of modified decision tree and chi-square feature selection," *Internet Things*, vol. 21, Mar. 2023, Art. no. 100676.

Hope Leticia Nakayiza received the Bachelor of Software Engineering (B.Eng.) degree from Kangnam University, Yongin, South Korea, in 2023. She is currently pursuing the master's degree in IT-convergence engineering with the Networked Systems Laboratory, IT-Convergence Engineering Department, Kumoh National Institute of Technology, Gumi, South Korea.

Her current research interests include artificial intelligence, blockchain applications, and optimization for Internet of Vehicles security.

Love Allen Chijioke Ahakonye (Senior Member, IEEE) received the B.Sc. degree in mathematics/computer science from the University of Port Harcourt, Port Harcourt, Nigeria, in 2001, the M.Sc. degree in information technology from the Federal University of Technology, Owerri, Nigeria, in 2016, and the Ph.D. degree in IT-convergence engineering from the Networked System Laboratory, IT-Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea, in 2024.

She has over a decade of experience in Nigerian Oil and Gas Sector as a Network and System Administrator from 2002 to 2016. In 2017, she briefly worked as a Logistics Superintendent at Nigerian Petroleum Development Company, Abuja, Nigeria, until 2019. She is also a Post-Doctoral Researcher at the ICT Convergence Research Center (ICT-CRC), Kumoh National Institute of Technology. Her research interest is artificial intelligence (AI) applications to the Internet of Things (IoT) and industrial IoT, Supervisory Control and Data Acquisition (SCADA), and industrial control systems (ICSs) for intrusion/anomaly detection, cybersecurity, fault detection, XAI, and manufacturing execution systems.

Dr. Ahakonye is a member of the Computer Professionals Registration Council of Nigeria (CPN).

Dong-Seong Kim (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003.

From 1994 to 2003, he worked as a full-time Researcher at ERC-ACI, Seoul National University. From March 2003 to February 2005, he was a Post-Doctoral Researcher at the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. Between 2007 and 2009, he was a Visiting Professor with the Department of Computer Science, University of California, Davis, Davis, CA, USA. He was the Dean of IACF from 2019 to 2022. He is currently a Professor with the Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea. He is also the Director of KIT Convergence Research Institute and ICT Convergence Research Center (ITRC and NRF Advanced Research Center Program), supported by Korean Government at Kumoh National Institute of Technology, and the Director of NSLab Company Ltd. His primary research interests include real-time IoT and smart platforms, industrial wireless control networks, networked embedded systems, Fieldbus, metaverse, and blockchain.

Dr. Kim is a Senior Member of ACM.

Jae Min Lee (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2005.

From 2005 to 2014, he was a Senior Engineer at Samsung Electronics, Suwon, South Korea. From 2015 to 2016, he was a Principal Engineer at Samsung Electronics. Since 2017, he has been an Associate Professor with the School of Electronic Engineering and the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk, South Korea. Since 2024, he has been the Director of the Smart Defense Logistics Innovation Convergence Research Center (ITRC Program). He is the Executive Director of Korean Institute of Communications and Information Sciences (KICS), Seoul. His current main research interests are smart IoT convergence applications, industrial wireless control networks, UAVs, metaverse, and blockchain.