UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Lanxiang Zhang

AI-Driven Blockchain-based Federated Learning for Edge Devices

Master's Thesis (30 ECTS)

Supervisor(s): Mubashar Iqbal

Tartu 2025

AI-Driven Blockchain-based Federated Learning for Edge Devices

Abstract:

The rapid development of the Internet of Vehicles (IoV) has created unprecedented demands for secure, scalable, and privacy-preserving edge computing solutions. While Federated Learning (FL) offers a promising approach to collaboratively train machine learning models across distributed devices without exposing raw data, traditional FL frameworks remain dependent on centralized aggregators, introducing single points of failure, data poisoning attacks, and trust issues in dynamic vehicular environments. To address these challenges, we propose an AI-driven blockchain-based FL framework for IoV that integrates decentralized consensus mechanisms, cryptographic validation, and robust aggregation techniques to enable secure and efficient collaborative learning among untrusted edge devices. The framework leverages a Hyperledger Fabric-based permissioned blockchain network to manage tamper-resistant records of model updates, enforce dynamic reputation systems, and distribute incentives for participation. By combining differential privacy and Byzantine-resilient aggregation, our method significantly reduces the risks of data leakage and model poisoning. Experimental evaluations in a simulated IoV environment demonstrate that the proposed approach achieves comparable accuracy to centralized learning (92.4% mAP vs. 93.1% mAP) while reducing attack success rates from 78.5% to 3.2% and preserving strong privacy guarantees. This work advances the state of the art in decentralized machine learning and provides a practical foundation for privacy-preserving, trustworthy intelligence in intelligent transportation systems. While existing approaches, such as differential privacy and secure multi-party computation, offer partial protection, they often introduce high computational costs or rely on unrealistic trust assumptions. Blockchain technology, though promising for decentralizing FL workflows, is still challenged by trade-offs between security and scalability—especially in highly mobile, intermittently connected IoV settings.

Keywords:
Blockchain, Federated Learning, Internet of Vehicles, Edge Computing, Privacy Preservation, Byzantine Resilience

CERCS: P170 Computer science, numerical analysis, systems, control

Tehisintellektipõhine plokiahelal põhinev liitõpe servaseadmetele

Lühikokkuvõte:

Sõidukite interneti (IoV) kiire areng on toonud esile enneolematu vajaduse turvaliste, skaleeritavate ja privaatsust hoidvate servtöötluslahenduste järele. Föderatiivne masinõpe (FL) võimaldab treenida mudeleid hajusatel seadmetel ilma toorandmeid jagamata, kuid traditsioonilised FL-raamistikud sõltuvad endiselt tsentraliseeritud agregeerijatest, mis loob ühe riknepunkti, soodustab andmemürgitust ja tekitab usaldusprobleeme dünaamilistes sõidukikeskkondades. Nende väljakutsete lahendamiseks pakume välja plokiahelapõ-

hise FL-raamistiku IoV-le, mis ühendab hajusad konsensusmehhanismid, krüptograafilise valideerimise ja robustsed agregeerimisvõtted, et võimaldada turvalist ja tõhusat koostööpõhist õppimist usaldamata servseadmete vahel. Raamistik kasutab Hyperledger Fabricil põhinevat lubadega plokiahelavõrku, et hallata võltsimiskindlaid mudeliuuenduste kandeid, jõustada dünaamilisi maine-süsteeme ja jaotada osalemisstiimuleid. Kombineerides diferentsiaalprivaatsuse ja Bütsantsi-taluvate agregeerimismeetoditega, vähendab lahendus oluliselt andmelekkete ja mudelimürgituse riski. Simuleeritud IoV-keskkonnas tehtud katsed näitavad, et pakutud lähenemine saavutab tsentraliseeritud õppimisega võrreldava täpsuse (92,4% mAP vs 93,1% mAP), vähendab ründe edukuse määra 78,5%-lt 3,2%-ni ning säilitab tugevad privaatsusgarantiid. Töö viib hajusa masinõppe taset edasi ja pakub praktilise aluse privaatsust hoidvale ja usaldusväärsele tehisarukusele nutikates transpordisüsteemides. Kuigi olemasolevad meetodid, nagu diferentsiaalprivaatsus ja turvaline mitmepoolne arvutus, annavad osalise kaitse, kaasnevad nendega sageli suured arvutuskoormused või ebarealistlikud usaldus-eelduse. Plokiahel aitab küll FL-töövooge detsentraliseerida, kuid jääb endiselt tasakaalustama turvalisuse ja skaleeritavuse vahelisi kompromisse—eriti väga mobiilses ja katkendliku ühenduvusega IoV-seadetes.

Võtmesõnad:
Plokiahel, Liitõpe, Servaarvutus, Sõidukite Internet, Tarklepingud, Privaatsuse Säilitamine, Bütsantsi Taluvus, Diferentsiaalne Privaatsus

CERCS: P170 Arvutiteadus, numbriline analüüs, süsteemid, juhtimine

Acknowledgment

My deepest thanks go to my supervisor, Dr. Mubashar Iqbal, whose steady guidance, sharp feedback, and patience shaped both this thesis and the way I think.

To all who left a light on for me and to the quiet companion who walked beside me through long nights and early mornings; your presence is written between the lines of this work.

To my family and friends: thank you for the faith that never wavered, the check-ins at odd hours, and the quiet space that made this possible.

I acknowledge the use of language enhancement tools such as **Grammarly**[1] during the writing process. It was used solely for proofreading suggestions and improving language clarity. I also used generative tools like **ChatGPT**[2], including models like GPT-4o and GPT-4o mini, to brainstorm initial ideas and phrasing inspiration. All final content, critical analysis, and academic reasoning are solely my own.

Cyber-security Excellence Hub in Estonia and South Moravia

This work is part of the Cyber-security Excellence Hub in Estonia and South Moravia (CHESS: https://chess-eu.cs.ut.ee) project funded by the European Union under Grant Agreement No. 101087529. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

[1] https://www.grammarly.com
[2] https://chatgpt.com

Contents

1 Introduction

The Internet of Vehicles (IoV) connects vehicles, roadside units, and cloud services to improve safety, efficiency, and automation. These systems generate large volumes of data from sensors such as cameras, LiDAR, and telemetry modules. While this data can improve decision-making and real-time control, its collection and processing raise challenges in privacy, trust, and scalability [1, 2]. Centralized Machine Learning (ML) methods require sending raw data to a server, which leads to high communication costs, latency, and risks of leakage [1]. Federated Learning (FL) keeps data local and shares only model updates, but in vehicular settings, it still faces dependence on a central aggregator, vulnerability to poisoning attacks, unstable participation due to high mobility, and performance loss under non-IID data (non-independent and identically distributed) [1, 3]. Blockchain offers decentralized consensus, tamper-proof records, and programmable contracts that can improve trust, transparency, and incentives [4, 5]. Combining blockchain with FL can address the weaknesses of each, but also brings challenges such as keeping latency low, reducing computation on edge devices, and scaling in dynamic networks [6, 7, 8]. This thesis proposes a blockchain-enhanced FL framework designed for secure, scalable, and privacy-preserving collaborative learning in IoV. The next section outlines the main problems in current FL and blockchain approaches for IoV, followed by the research questions that guide the design of the proposed framework.

1.1 Problem Statement

FL–based blockchain systems face several barriers to practical IoV deployment. First, reliance on a central aggregator creates a single point of failure and a clear attack surface; in highly dynamic, open vehicular networks, auditability of aggregation and update provenance remains limited. For example, model-replacement backdoors show that a single malicious client can corrupt the global model under standard aggregation [3], and blockchain-assisted FL helps decentralize coordination while recording update provenance [5]. Second, blockchain by design can introduce prohibitive confirmation delay and bandwidth overhead; heavyweight consensus (e.g., PoW) is misaligned with resource-constrained, real-time edge settings. For instance, PoW systems (e.g., Bitcoin) adopt long block intervals and multi-confirmation finality [7], while IoT studies note mining and propagation overheads [6]; protocols such as Bitcoin-NG aim to lower confirmation delay [8]. Third, privacy remains a concern: even without raw data exchange, gradients may leak sensitive information, calling for calibrated differential privacy and secure aggregation with quantified utility–privacy trade-offs. Empirically, shared gradients can reveal training inputs—from DLG reconstructions to high-fidelity inversions [9, 10]; secure aggregation and DP mitigate leakage at the cost of utility and efficiency [11, 1]. Fourth, incentive and reputation mechanisms are underspecified; without robust participation eco-

nomics and Sybil resistance, systems are exposed to free-riding and poisoning. Surveys document that incentive and reputation designs in FL remain underdeveloped [12], and Sybil-based poisoning is addressed by defenses such as FoolsGold [13]. Finally, mobility and intermittent connectivity complicate participation and synchronization, demanding mechanisms that are robust, lightweight, and scalable. In practice, vehicular handovers and dropouts disrupt synchronous rounds, motivating asynchronous or mobility-aware FL designs [14, 15, 16].

1.2 Research Questions

This thesis aims to answer the following main research question: **RQ:** *How can blockchain improve FL for the IoV in dynamic edge environments?* To explore the main question, several sub-questions are proposed.

- ***RQ1****: What are the main problems in current FL systems when used in IoV edge environments?* These systems often rely on central servers, which creates risks such as poisoning attacks and privacy leaks. Understanding these problems helps justify the need for a new, combined approach.
- ***RQ2****: How can blockchain help solve these problems by offering shared control, transparency, and trusted records?* Blockchain provides features such as smart contracts, shared agreement, and tamper-proof logs that can support FL.
- ***RQ3****: What kind of system design can combine blockchain and FL while working well in real-time and resource-limited IoV settings?* The system must be easy to scale, lightweight to run, and strong enough to handle mobile and unstable networks.
- ***RQ4****: How can privacy tools, like differential privacy and secure aggregation, prevent attacks without making the model worse?* The system needs strong privacy protections that meet safety and legal requirements while keeping the model useful.
- ***RQ5****: What types of reward and reputation systems can support honest behavior and stop attackers in blockchain-based FL?* These systems should give users a reason to join and behave well while ensuring fairness and trust.
- ***RQ6****: How well does the proposed system perform in simulated IoV settings, in terms of accuracy, speed, communication cost, and defense against attacks?*

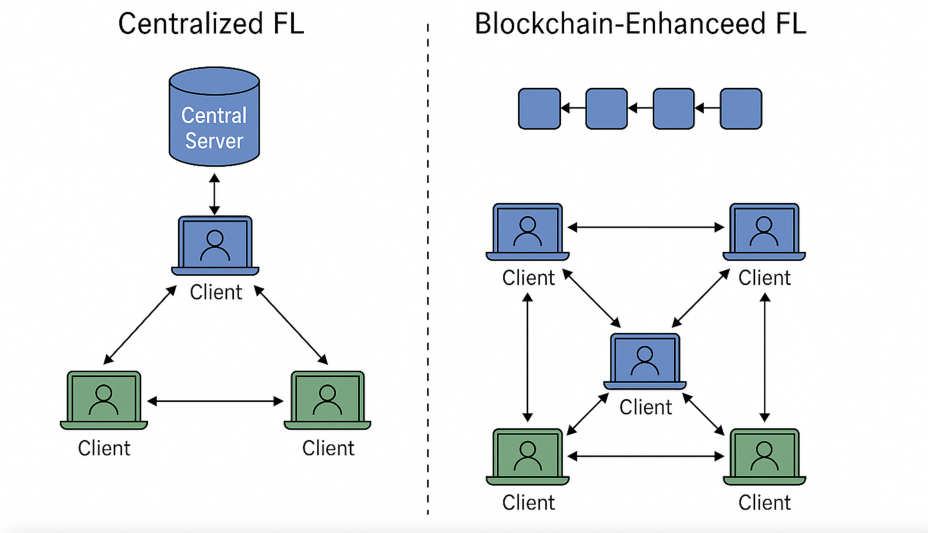

Figure 1. Comparison between centralized FL and blockchain-enhanced FL.

1.3 Research Methodology

This research uses the Design Science Research (DSR) method, as shown in Fig. 2, based on the approach proposed by Peffers et al. [17]. DSR is suitable for creating new technologies that solve real-world problems, especially in fast-changing environments like the IoV. The research follows six key phases.

The first phase, **Problem Identification and Motivation**, begins by analyzing existing FL and blockchain solutions. It finds that current systems have problems with trust, privacy, scalability, and reliability in IoV settings. The second phase, **Definition of Objectives**, sets the goal of designing a system that offers a decentralized way to combine model updates, ensures transparent and verifiable learning steps, provides strong protection against poisoning attacks, and complies with privacy rules such as the General Data Protection Regulation (GDPR). The third phase, **Design and Development**, involves creating a new framework that combines FL with blockchain. This framework integrates smart contracts for update control and reputation tracking, applies differential privacy to protect user data, and uses Byzantine-resilient methods to defend against malicious updates. In the fourth phase, **Demonstration**, we use real-world vehicular datasets together with a SUMO-based mobility simulation, i.e., real data combined with a simulated deployment environment. The fifth phase, **Evaluation**, measures the system's performance in terms of model accuracy, added delay, energy consumption on edge devices, and resilience against poisoning and data-leak attacks. The results are compared with both centralized learning and standard FL systems. Demonstration and evaluation are conducted via code-based simulation that combines public vehicular datasets with a SUMO-driven mobility process. All latency and energy numbers reported later are

simulation-based estimates derived from analytical component models and discrete-event simulation; we report means and 95% confidence intervals across multiple seeds. Finally, the sixth phase, **Communication**, shares the research findings with both academic and industry audiences, particularly those working in edge AI and IoV systems.

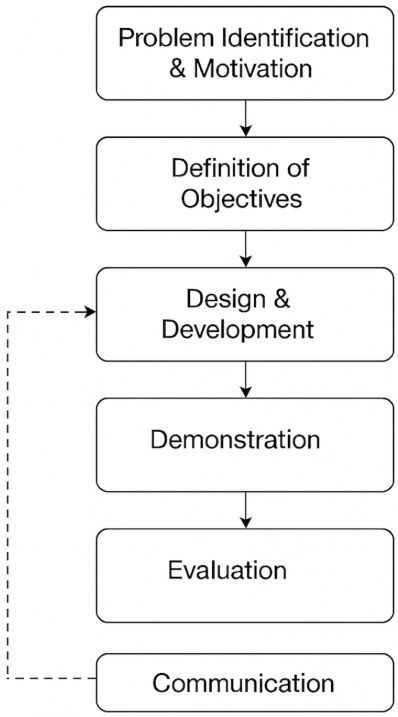

Figure 2. Design Science Research methodology based on Peffers et al. [17].

1.4 Contributions

This thesis presents several contributions to secure and decentralized FL in the IoV field.

First, it analyzes the main problems of current FL systems. These include trust issues, weak privacy protection, and poor scalability in dynamic and low-resource edge environments.

Second, it proposes a new architecture that combines FL with blockchain. It replaces the central aggregator with smart contracts to achieve decentralized learning and avoid single points of failure. The system includes robust aggregation methods, like geometric median and Krum. These help reduce the impact of poisoning attacks and protect the integrity of learning. The framework also has a flexible reputation system. It encourages honest and high-quality contributions from different types of devices.

Third, it presents a privacy and security design. It uses differential privacy to defend against inference attacks. It also uses signature checks and threshold decryption to ensure data is safe, unchanged, and not leaked.

Fourth, a code-based prototype is evaluated using public vehicular datasets with SUMO-based mobility; all latency/energy are model-based estimates reported with uncertainty. The results show that the system reaches similar accuracy to centralized learning. It also performs better in terms of privacy, security, and scaling under attack.

Finally, this work gives design suggestions and system ideas for real IoV uses. These can also be used in other edge areas like smart grids, industrial IoT, and robotics.

1.5 Thesis Structure

The rest of this thesis is structured as follows. Section 2 gives background on FL in edge environments, blockchain for decentralized trust, and how they work together in the IoV. It also reviews recent research on FL with blockchain. It points out the main problems like poor scalability, weak privacy, and lack of robustness. Section 3 shows a use case in a simulated smart transportation setting. It explains how models are trained and deployed in a decentralized way. It also discusses real-world limits like weak connections, attacks, and system performance. Section 4 describes the system design. It includes layers that use differential privacy, strong aggregation, and coordination with blockchain. It also explains each part of the system and how smart contracts help untrusted devices work together with trust and fairness. Section 5 explains how the system was built. It includes the tech structure, cryptographic tools, edge simulation, and deployment steps. It uses TensorFlow Federated, Hyperledger Fabric, and IPFS to share data securely and protect model privacy. Section 6 shows test results using real vehicle data and attack scenarios. It looks at model accuracy, training process, communication cost, privacy, and defense against attacks. Finally, Section 7 gives the conclusion. It reviews the main contributions, shows what still needs work, and suggests future research for secure and scalable FL in IoV and other edge areas.

2 Background

This section provides the background by explaining FL in edge environments, including its benefits and key challenges. Then, it discusses how blockchain can help build trust and ensure verifiable coordination in untrusted systems. It also looks at the features of the IoV and examines the potential benefits and trade-offs of combining FL with blockchain. Finally, it reviews existing studies to identify research gaps that support the motivation for this thesis.

2.1 Federated Learning in Edge Environments

FL introduces a new approach to ML by allowing multiple devices to train a shared model collaboratively while keeping their raw data local [18, 1]. In this setup, a central server (also known as the aggregator) initializes a global model and sends it to participating clients. Each client then performs local training and returns updated model parameters. A commonly used aggregation strategy, known as Federated Averaging (FedAvg) [18], calculates a weighted average of the model updates from clients.

$$\mathbf{w}_{\text{global}} = \sum_{i=1}^{N} \frac{n_i}{n} \mathbf{w}_i \tag{1}$$

In Equation 1, $\mathbf{w}_i$ refers to the model parameters from client i, n_i is the number of local samples on that client, and n is the total sample count across all clients. This equation illustrates the weighted aggregation mechanism, which helps maintain the representativeness of updates from diverse nodes.

This approach is especially useful in edge environments like the IoV, where large amounts of privacy-sensitive data are generated by sensors and roadside infrastructure. However, several technical challenges limit the practical use of FL in these environments. A major challenge is the non-IID nature of local data, which can differ widely across clients. For instance, vehicles in urban areas face different traffic patterns, environments, and sensor inputs compared to those on rural highways, making model generalization more difficult. In addition, edge devices often have limited resources, such as processing power, memory, and battery capacity, which limits their ability to perform complex model computations. Communication overhead is another concern, since frequently sending model updates over wireless networks increases both latency and energy consumption. These issues are made worse by intermittent connectivity due to vehicle mobility or signal interference, which results in unreliable participation in the learning process. Security threats also exist, especially model poisoning attacks, where malicious clients send harmful updates to degrade the global model's performance [19]. These challenges highlight the need for learning strategies that are resilient, efficient, and secure, and that are specifically designed for dynamic edge environments like the IoV.

To address these challenges, researchers have proposed several techniques, including client selection, differential privacy, secure aggregation, and adaptive learning rates [20, 11]. However, most of these solutions depend on a trusted central aggregator. This reintroduces a single point of failure and reduces system transparency.

Beyond data heterogeneity and communication constraints, FL systems also face challenges related to participant reliability and the quality of their contributions. One proposed solution is to use reputation-based mechanisms, where each participant's trust score is dynamically updated as follows:

$$r_i^{\text{new}} = r_i^{\text{old}} + \alpha \cdot Q(\mathbf{w}_i) \tag{2}$$

In Equation 2, r_i is the reputation score of participant i, α denotes the learning rate, and $Q(\mathbf{w}_i)$ is a quality evaluation function for the model update. This formulation rewards participants who submit high-quality updates and penalizes those whose contributions are unreliable or possibly malicious.

Recent studies have analyzed these challenges in depth, including imbalanced class distributions and variations in feature spaces across clients [21]. Proposed solutions include data augmentation and client-specific model personalization, though most require careful tuning for particular applications. Security concerns in federated edge learning have attracted increasing attention. For example, [22] showed that the distributed nature of FL makes it susceptible to model poisoning attacks. In such cases, malicious participants can degrade the global model by submitting crafted updates—a vulnerability that is especially critical in vehicular networks.

2.2 Blockchain for Distributed Trust and Auditability

Blockchain offers key properties that align well with the needs of FL in untrusted environments. First, its decentralized design eliminates the need for a central aggregator and distributes trust among all participating nodes. Second, blockchain's immutability ensures that once data are recorded on the ledger, they cannot be changed or removed without detection. This supports tamper resistance and accountability. Consensus mechanisms like Practical Byzantine Fault Tolerance (PBFT) [23], Proof of Stake (PoS) [24], and Proof of Work (PoW) [7] offer a secure way to reach agreement on the validity of transactions or updates. Finally, smart contracts allow automatic enforcement of verification logic, aggregation rules, and incentive distribution without human intervention [25].

Recent work has shown that permissioned blockchains such as Hyperledger Fabric (HLF) are suitable for FL [26]. In addition, hybrid storage strategies that combine on-chain metadata with off-chain repositories, such as the InterPlanetary File System (IPFS), can improve scalability [27].

Although blockchain offers notable advantages, its integration with FL also involves several trade-offs. Consensus mechanisms—particularly those relying on PoW—often introduce significant latency, which delays the finalization of model updates. Broadcasting model updates across the network increases bandwidth consumption and further strains the already congested communication channels in IoV systems. In addition, signature verification and other cryptographic operations place a heavy computational load on edge devices, which often have limited processing power. To ensure reliable performance in time-sensitive vehicular environments, these limitations must be carefully resolved.

Consequently, lightweight blockchain solutions play a vital role in sustaining performance in IoV contexts.

In this context, blockchain plays several roles. It maintains an immutable record of model updates, enables decentralized aggregation via smart contracts, and supports transparent reputation tracking and incentive mechanisms. Selecting between permissioned and permissionless blockchains requires careful consideration of the specific application needs and performance constraints. Permissioned blockchains, which use controlled membership and efficient consensus, are generally better suited for enterprise and IoT systems that require high performance.

2.3 Edge Intelligence and the Internet of Vehicles

The IoV refers to a complex ecosystem that includes connected vehicles, roadside units (RSUs), and cloud-based services [28, 29]. These components exchange data collaboratively to support real-time decision-making [29]. A wide range of applications demonstrates the practical value and transformative potential of IoV systems. For instance, cooperative perception allows vehicles to share sensor data, such as LiDAR and camera feeds. This shared information improves situational awareness beyond each vehicle's line of sight [30]. Predictive maintenance uses localized usage patterns and telemetry data to forecast component failures in advance, thereby improving safety and reducing downtime [29]. By utilizing IoV infrastructure, autonomous navigation can dynamically plan safe and efficient routes in response to changing traffic conditions [31].

Despite these advances, IoV poses distinctive challenges that set it apart from traditional edge computing scenarios. A primary challenge is the high mobility of vehicles, which leads to frequent transitions between network zones and causes handovers and intermittent connectivity that disrupt coordination among participants [32]. IoV systems include a wide range of heterogeneous devices—from low-power sensors to powerful onboard units—resulting in uneven computational capacities and inconsistent participation across nodes [29]. Many IoV tasks impose strict real-time constraints—often below 100 milliseconds—which renders latency and synchronization critical to system performance [33]. Moreover, the network topology is highly dynamic, constantly evolving as vehicles enter and leave the system [29]. Finally, IoV environments often feature mixed trust relationships, as vehicles and infrastructure components operate under different ad-

ministrative or commercial domains, making mutual trust more difficult to establish [34]. These interconnected constraints intensify the difficulties faced by FL, making centralized training inadequate and encouraging the development of decentralized, adaptive solutions suited to the dynamic IoV environment [32].

2.4 Blockchain-FL Integration: Synergies and Trade-Offs

Combining blockchain with FL offers a way to eliminate the single point of failure introduced by centralized aggregators. In this combined approach, model updates from participants are signed and recorded on a shared blockchain ledger to ensure authenticity and traceability. Smart contracts play a key role by verifying updates and running aggregation logic automatically. The blockchain's immutable nature supports auditability and regulatory compliance by preserving a transparent and tamper-proof history of all model contributions. Token-based incentives can encourage honest behavior and discourage attacks by aligning personal actions with system goals.

However, the integration of blockchain and FL introduces several notable trade-offs [35, 36, 37]. One major concern is communication overhead, as on-chain transactions and metadata broadcasting increase bandwidth usage, posing challenges in resource-constrained edge environments. Consensus mechanisms—even efficient ones like PBFT—also introduce latency, delaying the finalization of model updates when compared to centralized coordination. Moreover, many edge devices may lack the computational capacity to handle frequent signing and verification tasks, thereby limiting scalability and performance. These limitations highlight a fundamental trade-off triangle involving decentralization, trust, and system performance. No existing approach optimizes these aspects at once, so architectural choices must be tailored to specific application constraints and goals.

Beyond technical trade-offs, integrated systems must also contend with a range of security and privacy threats. Model poisoning attacks continue to pose a significant risk in FL settings, as malicious participants can craft harmful updates that degrade global model performance. Privacy leakage is another concern, particularly when adversaries attempt to infer sensitive training data from shared model parameters. Additionally, issues such as free-rider behavior and Sybil attacks can erode trust and fairness within the collaborative framework. Researchers have proposed several defenses to these challenges, including differential privacy, secure aggregation, and lightweight cryptographic methods combined with dynamic reputation and incentives. These methods provide the foundation for building secure and reliable FL systems enhanced by blockchain.

2.5 Extended Literature Landscape

More and more studies have explored how to integrate blockchain with FL in various application domains. For instance, Kang et al. [36] proposed a reputation-aware aggrega-

tion mechanism that uses blockchain to enhance trust in FL updates. Ma et al. [38] used a similar blockchain-based architecture to support privacy-aware healthcare analytics, showing its effectiveness in sensitive contexts. Wang et al. [39] showed that smart contracts can be used to automatically verify updates and manage incentives, thereby minimizing the need for a central authority. While Liu et al. [37] introduced token-based mechanisms to foster honest behavior, their design overlooked the unique challenges posed by vehicular mobility. More recently, Chen et al. [40] and Xie et al. [41] have proposed optimizations to blockchain consensus protocols and enhanced intrusion detection mechanisms specifically tailored for IoV scenarios.

Although recent studies have made progress, most still focus on either privacy or decentralization, and few address both in dynamic environments like IoV. Many studies also assume static topologies and stable links, which do not reflect the frequent disconnections and mobility in real-world IoV environments. Such limitations reflect the broader challenges of applying current blockchain-FL frameworks to the dynamic and distributed nature of vehicular networks.

To overcome these limitations, this thesis introduces a method that integrates cryptographic checks with a dynamic participant selection strategy. This approach ensures security in terms of integrity, auditability, and attack resistance while minimizing the computational burden on edge devices. In addition, the framework integrates a multi-dimensional reputation mechanism. It evaluates both the quality of model updates and the consistency of participation over time. This design helps foster fair and trustworthy collaboration among heterogeneous nodes.

2.6 Research Gap and Justification

The rapid growth of connected vehicles and stricter privacy laws like GDPR show that we need FL solutions that are both secure and privacy-oriented. However, current approaches fall short in several key aspects. For example, they often rely on trusted central aggregators, which undermines the decentralized design principle of FL and creates single points of failure. Moreover, most existing systems offer limited defenses against model poisoning and inference attacks, both of which present significant risks in safety-critical applications such as the IoV. Furthermore, existing solutions can fail to accommodate the dynamic and resource-limited nature of vehicular environments, where devices operate under stringent latency, bandwidth, and computational constraints.

To overcome these gaps, we propose a lightweight, scalable, and privacy-preserving FL framework, integrated with blockchain. The proposed architecture explicitly considers vehicular mobility, changing network conditions, and varying levels of trust between participants. The framework addresses the unique challenges of decentralized learning in the IoV by combining decentralized aggregation, trusted update tracking, and a reputation system aligned with participant incentives.

3 Systematic Literature Review

This section answers **RQ1: What are the main problems in current FL systems when used in IoV edge environments?**. To answer RQ1, we conducted a systematic literature review (SLR) following the Kitchenham review guidelines [42]. The SLR aims to identify, categorize, and analyze recent research works that integrate FL and blockchain in the context of vehicular edge computing. The SLR process followed five phases: (1) protocol specification, (2) search and selection, (3) data extraction, (4) synthesis, and (5) reporting.

3.1 Review Settings

Motivated by the need for decentralized learning frameworks that ensure trust, privacy, and resilience in dynamic vehicular environments, this review sets out to: (i) synthesize the state of the art in blockchain-enabled FL systems within IoV scenarios, (ii) identify critical implementation challenges, and (iii) extract architectural insights to inform the design proposed in Section 5. To structure the SLR analysis, we formulate three literature review questions (LRQ):

- **LRQ1:** What blockchain mechanisms have been proposed to enhance FL in vehicular edge networks?
- **LRQ2:** What are the identified limitations in terms of security, scalability, and trust?
- **LRQ3:** How are these approaches evaluated, and what deployment gaps remain?

We prepare the review protocol prior to SLR execution. The protocol includes objectives and LRQs, sources and search strings, eligibility criteria, study selection procedure, data items and extraction form, synthesis approach, and validity considerations. Searches were first run in July 2025 and last updated on 5 August 2025 to include late-2024/2025 publications. Database-specific adaptations of the search string are provided in the appendix. A structured query string was constructed to retrieve relevant publications from IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Scopus:

```
("federated learning" AND "blockchain" AND ("vehicular" OR
"IoV" OR "edge computing")) AND PUBYEAR > 2019 AND PUBYEAR <
2026
```

Only English-language, peer-reviewed journal or conference articles were considered. After de-duplication via Zotero, **300** unique records remained. Screening by title and abstract led to **15** studies selected for full-text review, based on their technical depth,

clarity, and relevance. We define the inclusion and exclusion criteria. In the inclusion criteria, we consider (i) integrates blockchain and FL; (ii) targets vehicular/IoV or closely related edge settings; (iii) describes a technical architecture or mechanism; (iv) reports an empirical evaluation (simulation, prototype, or testbed). In contrast, exclusion criteria exclude position/vision papers without technical instantiation, works focusing solely on blockchain or solely on FL, and non-peer-reviewed items where the core mechanism is unclear. For transparency and reproducibility, Table 1 summarizes the screening process and reasons for exclusion at the full-text stage.

Table 1. Screening flow (PRISMA-style summary)

Stage	**Count**	**Notes / reasons for exclusion**
Records identified via databases (pre-dedup)	—	IEEE, ACM DL, SpringerLink, ScienceDirect, Scopus
Records after duplicates removed	**300**	De-duplicated in Zotero
Title/abstract screened	300	Eligibility criteria applied
Full-text articles assessed for eligibility	**15**	All retrieved successfully
Full-text articles excluded	**6**	No FL–blockchain integration (2); not vehicular/edge context (2); no empirical evaluation (2)
Studies included in qualitative synthesis	**9**	Summarized in Table 3

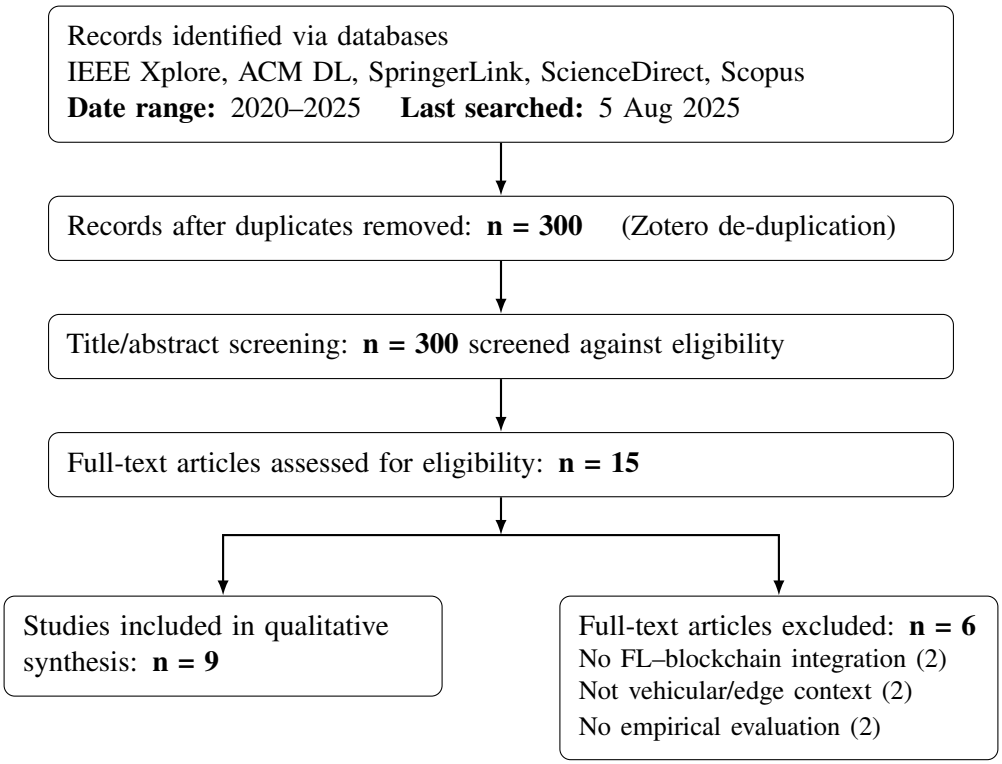

Figure 3. PRISMA-style flow chart (complements Table 1).

Each paper was evaluated using a five-point quality rubric addressing: (1) clarity of problem and context, (2) methodological transparency, (3) explicit FL–blockchain integration, (4) evaluation design (datasets/scenarios, baselines, metrics), and (5) discussion of limitations. Studies scoring below three were excluded. Applying these criteria yielded a final set of **nine** studies synthesized below.

3.2 Data Extraction and Review Findings

For each selected article, metadata and technical attributes were extracted, including application domain, FL architecture (e.g., FedAvg, clustered, personalized, RL), blockchain design (e.g., permissioned/permissionless, PBFT/PoS/DAG, smart-contract roles), aggregation and verification method (e.g., secure aggregation, robust statistics, reputation weighting), incentives/reputation, evaluation setup (simulation vs. prototype/testbed; adversarial tests), and documented limitations. Coding proceeded in two steps. First, open coding on the extracted fields produced initial codes for mechanisms and evaluation choices. Second, axial coding grouped these codes into higher-level categories. Constant comparison was used to stabilize code definitions as new studies were integrated. Table 2 outlines the extraction fields. Table 3 summarizes the corpus.

Table 2. Data extraction schema (key fields)

Field	**Description**
Domain	Vehicular/IoV sub-context (traffic safety, platooning, smart charging, smart city)
FL style	FedAvg / clustered / personalized / RL; sync/async; client selection
Blockchain design	Permission model; consensus (PBFT/PoS/DAG/Raft); smart-contract roles; on/off-chain split
Aggregation/verification	Robust estimators (e.g., Krum, trimmed mean, geometric median), secure aggregation, reputation weighting
Incentive/reputation	Token or score design; Sybil resistance; participation rules and penalties
Evaluation	Simulation/prototype; datasets/scenarios; metrics (accuracy/latency/bandwidth/energy); adversarial tests
Limitations	Reported bottlenecks, assumptions, threats to validity

Before reviewing individual studies, we provide a brief view of the temporal distribution and recurring gaps observed during screening and coding. Figure 4 shows a steady rise from 2020 to a peak around 2024, with a mild contraction in 2025, suggesting topic consolidation or a pivot toward alternatives such as split learning. *Counts in the figure reflect database search hits prior to screening and de-duplication.* Despite the momentum, four gaps recur across the corpus: (i) a dominance of simulation over prototypes/testbeds; (ii) latency bottlenecks due to consensus and coordination in time-sensitive vehicular settings; (iii) limited alignment with 5G-V2X/ETSI MEC and deployment constraints; and (iv) underdeveloped economic models for incentives and insufficient adversarially robust aggregation. These observations directly inform the framework choices in Section 5.

We selected and analyzed the nine literature studies. For example, Su et al. [43]. Addresses free-riding in vehicular FL by keeping reputation scores on-chain and weighting client updates accordingly. Participation and update quality improve in simulation,

Table 3. Summary of Selected Blockchain + Federated Learning Studies

Study	**Domain**	**Federated Learning Type**	**Blockchain Use**	**Reported Limitations**
Su et al. [43]	Vehicular networks	Reputation-based federated learning	Node scoring ledger	High aggregation latency, limited mobility handling
Li and Wu [44]	Edge computing	Multi-aggregator federated learning	Smart contracts	DAG complexity, scalability constraints
Rahmani and Stojcevska [45]	Smart charging	Decentralized federated learning	Contract scheduling	Simulation-only validation
Chen et al. [46]	Smart cities	Federated CNN	Lightweight chain	Security trade-offs under compression
Ruan et al. [47]	Connected vehicles	Secure federated learning	Key exchange ledger	Lack of adversarial testing
Yang et al. [48]	Vehicular edge AI	Personalized federated learning	Smart contract policies	High transaction cost
Tang and Wei [49]	5G vehicular	Clustered federated learning	Blockchain DAG	Poor cross-cluster scalability
Zhou and Wang [50]	IoV systems	Horizontal federated learning	Permissioned consensus	High transaction delays
Zhang et al. [51]	Urban platooning	Fed reinforcement learning	Reward-based contracts	Simulation-only RL validation

but aggregation latency grows with network size and mobility, motivating lightweight coordination and asynchronous rounds in this work. Li and Wu [44]. Proposes a multi-aggregator FL orchestrated via smart contracts and DAG-style control to reduce single-point bottlenecks. While throughput improves, validator coordination and DAG complexity hinder scalability, supporting our choice of simpler permissioned ordering. Rahmani and Stojcevska [45]. Applies blockchain-enabled FL to smart EV charging, where contracts coordinate scheduling and model exchange. The evaluation is simulation-only, underscoring the need for prototypes/testbeds that our framework addresses. Chen et al. [46]. Presents a lightweight-chain design for smart-city FL with CNNs and model compression. Security–utility trade-offs appear under aggressive compression, informing our hybrid on/off-chain strategy and accuracy–overhead evaluation. Ruan et al. [47]. Focuses on secure key management for connected-vehicle FL using blockchain as a trust anchor. The absence of adversarial testing motivates our inclusion of poisoning and privacy-attack scenarios. Yang et al. [48]. Implements smart-contract–governed personalized FL at the vehicular edge. Personalization improves accuracy under non-IID data, but on-chain transaction costs are non-trivial, consistent with our permissioned Raft ordering and off-chain payloads. Tang and Wei [49]. Studies clustered FL with a blockchain-DAG scheduler for 5G vehicular contexts. Cross-cluster coordination becomes a bottleneck at scale, supporting our hierarchical aggregation via RSUs. Zhou and Wang [50]. Adopts permissioned consensus for horizontal FL in IoV. Despite stronger governance than permissionless designs, measured transaction delays remain high; we address this via

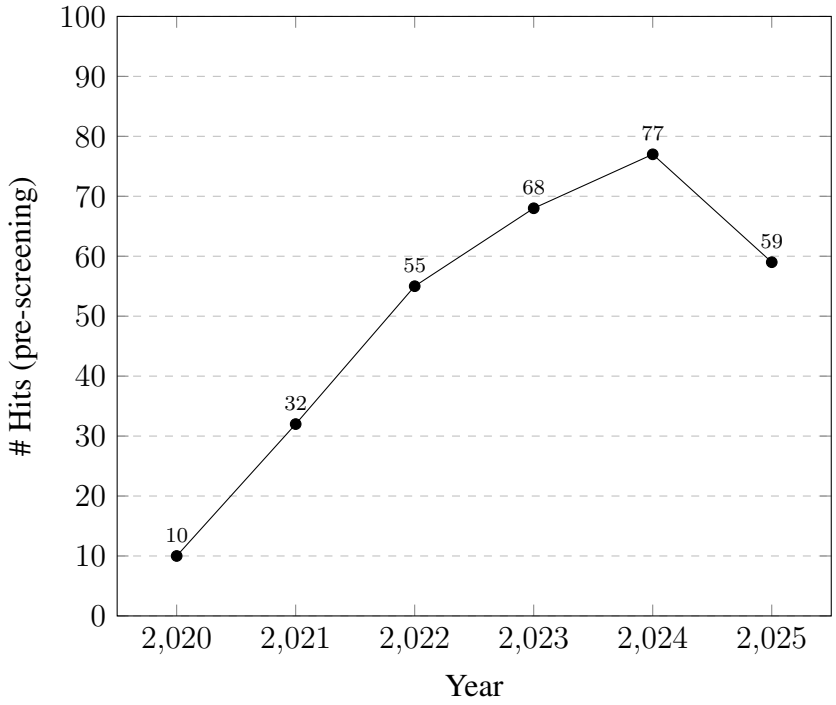

Figure 4. Year-wise distribution of database search hits (before screening/de-duplication)

tuned block size/timeout and off-chain storage. Zhang et al. [51]. Combines blockchain with federated reinforcement learning for urban platooning. Results are promising but simulation-only, reinforcing our mixed testbed and mobility-aware evaluation.

We summarized three key findings from the SLR. *(1) Secure Aggregation and Verification:* Many studies employ cryptography, differential privacy, or reputation-based filtering to reduce poisoning and collusion. Adaptive weighting by reputation, e.g. [52, 53],

$$r_i^{\text{new}} = \beta\, r_i^{\text{old}} + (1-\beta)\cdot \text{quality}(\Delta \mathbf{w}_i), \tag{3}$$

works well in simulation, but extra computation and real-time response remain open issues in vehicular settings. These observations inform the robust but lightweight aggregation choices in this thesis.

(2) Incentive Mechanisms: Token rewards aim to keep participation honest, e.g., *Reward_i*, but most works lack clear economic models and analysis of strategic behavior, leaving systems vulnerable to gaming. Our multi-factor reputation and penalties respond to this gap [54].

$$\text{Reward}_i = \gamma \cdot r_i \cdot |D_i|, \tag{4}$$

(3) Scalability and Performance Constraints: PBFT and DAGs often have higher throughput than PoW, yet confirmation latency and coordination overhead can be unacceptable for time-sensitive vehicular applications. Evaluations frequently omit realistic

topologies or adversaries, so we include mobility-aware and adversarial tests in our evaluation

3.3 Threats to Validity

Our current SLR has a few threats; thus, we explained their validity here. For example, *(1) Selection bias.* Restricting to English, peer-reviewed venues and 2019–2025 may omit relevant industry reports or earlier precursors. *(2) Construct validity.* Definitions of "vehicular edge" and adversary models vary, complicating direct comparison. *(3) Internal validity.* The concise appraisal rubric, while consistent with our protocol, may affect borderline inclusions; single-reviewer screening can introduce subjectivity. *(4) External validity.* Many studies are simulation-only; real-world constraints (radio conditions, RSU placement, compliance) can further impact performance. Our evaluation addresses these with a mixed testbed and mobility-aware scenarios.

3.4 Summary

This SLR answered **RQ1**. The literature indicates clear potential for blockchain-enhanced FL in vehicular applications but also highlights persistent gaps: evaluation realism, latency optimization, adversarial robustness, and sustainable incentive design. The framework in Section 5 is therefore built around permissioned ordering, hybrid storage, adaptive robust aggregation, and explicit adversarial/privacy testing under mobility.

4 Use Case: Blockchain-enhanced federated learning in IoV

This section answers **RQ2: How can blockchain help solve these problems by offering shared control, transparency, and trusted records?**. It provides a concrete illustration of how the proposed blockchain-enhanced FL framework can be applied in an IoV environment. Rather than elaborating on background concepts, this section demonstrates the full lifecycle of model training, aggregation, and deployment in a simulated smart transportation scenario. Through this use case, we validate the feasibility of developing the proposed framework, highlight practical considerations, and demonstrate its effectiveness in addressing core IoV challenges.

4.1 Scenario Overview

We consider a metropolitan smart traffic system comprising a fleet of heterogeneous vehicles equipped with cameras, LiDAR sensors, and onboard computing modules. The goal is to collaboratively train an object detection and trajectory prediction model to improve traffic safety and optimize routing under dynamic conditions such as rush hour congestion or inclement weather. In this environment, 50 edge nodes (vehicles) with heterogeneous computational capabilities—ranging from 10 to 30 tera operations per second (TOPS)—collaborate with 20 RSUs, which serve both as blockchain validators and as relays for communication. We use a permissioned Hyperledger Fabric (HLF) network to implement this system. HLF uses a Raft ordering service (crash fault tolerant), which provides low confirmation latency in permissioned settings.

During regular operation, vehicles generate locally labeled datasets. The training process proceeds in multiple rounds, with updates scheduled whenever vehicles enter the coverage area of RSUs. This architecture supports asynchronous participation, even in the presence of mobility and connectivity limitations typical of IoV environments.

4.2 Lifecycle of Model Training and Aggregation

Figure 5 illustrates the full workflow of the proposed system. During the initialization phase, the aggregator smart contract broadcasts the initial model parameters $\mathbf{w}_0$ to all registered vehicles. Each vehicle then performs local training on its private dataset and applies differential privacy by injecting calibrated Gaussian noise as follows:

$$\Delta \mathbf{w}_i = \nabla \mathcal{L}(\mathbf{w}_i) + \mathcal{N}(0, \sigma^2 \mathbf{I}). \tag{5}$$

After training, vehicles encrypt their model updates using Advanced Encryption Standard in Galois/Counter Mode (AES-GCM), sign them with Elliptic Curve Digital Signature Algorithm (ECDSA), and upload the associated metadata to the blockchain

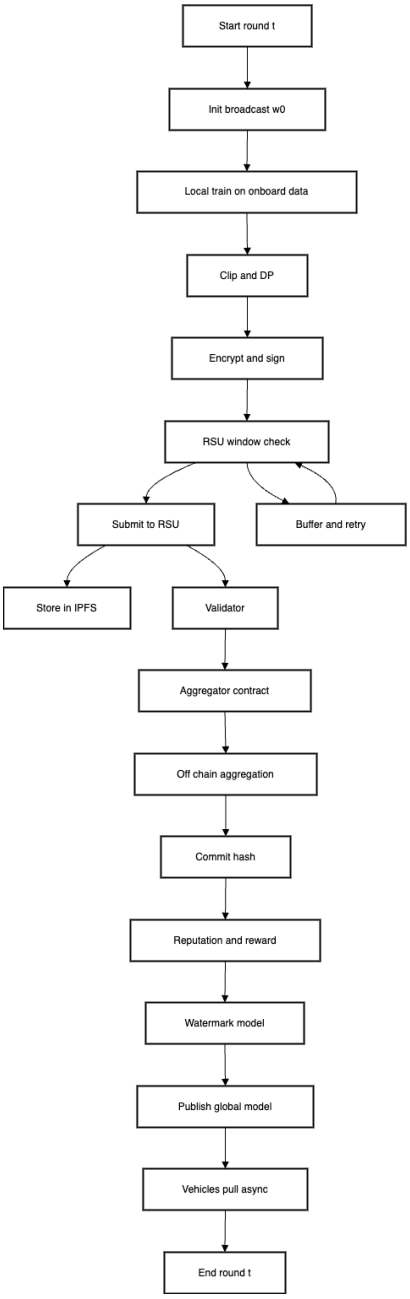

Figure 5. FL lifecycle with on-chain orchestration and secure aggregation.

while storing the encrypted payloads in a decentralized file system such as the IPFS. The Validator smart contract verifies the authenticity of each update, evaluates consistency metrics, and then updates the reputation score r_i of each participant accordingly.

To aggregate model updates, the off-chain Model Aggregation Service applies Byzantine-resilient algorithms (e.g., Trimmed Mean, Krum, or Geometric Median), using both the submitted gradients and the participants' reputation scores r_i [55, 56, 57]. The Aggregator smart contract only orchestrates and records hashes/events on-chain.

$$\mathbf{w}_{\text{global}} = \mathcal{A}(\{\Delta \mathbf{w}_i\}, r_i)\,. \tag{6}$$

The finalized global model is watermarked and distributed via RSUs to participating vehicles, completing one round of collaborative learning.

4.3 Example Training Round

To illustrate the framework's behavior in practice, this section considers a representative training round. In this round, 35 vehicles participated, including five malicious nodes that attempted a label-flipping attack. On average, each vehicle contributed 15,000 images and 6,000 LiDAR frames. Differential privacy was configured with parameters $\sigma = 1.2$ and $\delta = 10^{-5}$.

The system successfully detected malicious updates with a probability of 92.1% using z-score filtering (see Equation (20)). As a result, the reputation scores for the identified attackers decreased by 0.2 to 0.4 per round, reducing their influence in subsequent aggregations. The Krum algorithm limited the effective attack success rate to 4.3%. With the support of reputation-based weighting, the global model converged within 28 rounds, compared to 40 rounds when no reputation mechanism was applied. This example illustrates the framework's robustness in mitigating adversarial interference while maintaining efficient training dynamics.

4.4 Secure Model Deployment

Once the global model is finalized, it undergoes watermark embedding to ensure verifiable provenance. Specifically, a digital signature is embedded into the model parameters using Equation 7 as discussed in [58]:

$$\mathbf{w}_{\text{final}} = \mathbf{w}_{\text{global}} + \mathbf{m} \odot \mathbf{s}, \tag{7}$$

where $\mathbf{m}$ and $\mathbf{s}$ represent the watermark content. During operation, vehicles periodically validate the authenticity of received models by comparing the embedded signature with the records stored on the blockchain.

In addition to verification, the deployment phase includes runtime anomaly detection (Equation 8) [59] to ensure model integrity. An alert is triggered when:

$$a_t = \mathbb{I}(\|\mathbf{y}_t - \hat{\mathbf{y}}_t\| > \theta)\,. \tag{8}$$

Here $\hat{\mathbf{y}}_t$ is produced by a locally cached, previously verified checkpoint, and the threshold θ is set as the 95th-percentile deviation on a validation split.

In the evaluated scenario, no anomalies were detected across 500 inference batches, confirming that the deployed model maintained functional integrity during the entire test.

4.5 Performance Summary

Table 4 summarizes key performance metrics observed during the demonstration. These metrics include training latency, communication overhead, model accuracy, robustness against attack, false positive rates in malicious detection, and energy consumption.

Table 4. Performance metrics in the IoV deployment scenario

Performance Metric	Measured Value
Mean Training Round Latency	12.8 seconds
Average Communication Overhead	1.7 MB per vehicle per round
Model Accuracy (mAP)	92.1%
Attack Success Rate (Label Flipping)	4.3%
False Positive Rate (Malicious Detection)	2.7%
Energy Consumption per Round	23.4 Joules

These results validate the framework's capability to efficiently aggregate updates from a heterogeneous and intermittently connected fleet of vehicles, to withstand adversarial manipulation while maintaining low false positive rates, and to deliver high-accuracy models in non-IID environments, all within acceptable limits for latency and energy consumption.

4.6 Discussion

Several practical insights emerged from this use case. First, the use of reputation scores significantly influenced the training dynamics. Vehicles with higher reputations were given greater weight during aggregation, which both accelerated convergence and discouraged malicious participation. Second, the implementation of differential privacy achieved a reasonable balance between utility and privacy: the selected noise scale preserved model accuracy while offering strong privacy guarantees, although future work could explore adaptive noise mechanisms. Third, the framework proved robust to vehicular mobility. Asynchronous participation, combined with hybrid storage (on-chain metadata and off-chain payloads) enabled effective learning despite frequent disconnections and mobility constraints.

Collectively, these insights reinforce the viability of deploying the proposed blockchain-enhanced FL framework in realistic IoV environments.

4.7 Summary

This section answered **RQ2**. In this section, we present a detailed demonstration of deploying the proposed blockchain-enhanced FL framework in a real datasets with a SUMO-based mobility simulation. Through a step-by-step exposition of model training, validation, aggregation, and deployment, this section illustrates how the framework achieves secure, scalable, and privacy-preserving collaborative intelligence. Empirical results confirm the system's ability to deliver high model accuracy while mitigating the unique threats and operational challenges of decentralized vehicular networks.

5 Blockchain-Enhanced Federated Learning Framework

This section addresses **RQ3: What kind of system design can combine blockchain and FL while working well in real-time and resource-limited IoV settings?**. The following subsections describe the framework's architecture, the interactions between its components, and the key technical solutions that make FL secure and efficient for the IoV. This framework creates a decentralized learning system where edge devices work together to train machine learning models. Blockchain technology coordinates this collaboration securely. This section explains the framework's architecture, how its parts interact, and the key technical solutions that make FL secure and efficient for the IoV.

5.1 System Architecture

The framework is built on three main layers that work together to enable distributed model training. The physical layer consists of vehicles and other edge devices (like roadside sensors) equipped with sensors and computing power. Each vehicle $v_i \in \mathcal{V}$ has its own local dataset $\mathcal{D}_i$ and performs model training using its onboard resources. In the intermediate layer RSUs act as gateways to the blockchain network. They provide connectivity for vehicles and handle lightweight validation tasks for the blockchain.

Blockchain Layer is the core coordination layer, implemented as a permissioned blockchain (like Hyperledger Fabric). The ordering service uses Raft (CFT); Byzantine robustness in this work targets client-side updates (via robust aggregation and reputation) rather than the ordering layer. Smart contracts coordinate validation, aggregation orchestration (scheduling and recording of hashes), reputation, and incentives; numerical aggregation runs off-chain in the Model Aggregation Service. It hosts smart contracts that manage the entire FL process [26]:

$$\mathcal{SC} = \{\text{Validator}, \text{Aggregator}, \text{Reputation}, \text{Incentive}\} \tag{9}$$

The Validator contract checks participant identities and verifies digital signatures on model updates. The Aggregator contract schedules aggregation rounds, records input/output content hashes and parameters, and emits events to trigger the off-chain Model Aggregation Service. The off-chain service computes the global update using robust methods (e.g., reputation-weighted FedAvg, Trimmed Mean, Krum, Geometric Median) and posts the result hash back on-chain. The Reputation contract tracks and updates a trust score for each participant based on their past contributions. The Incentive contract manages the distribution of rewards (e.g., tokens) to participants who contribute effectively.

5.2 Component Interactions and Workflow

Figure 6 depicts a component view that aligns with the three-layer architecture and clarifies on-chain orchestration versus off-chain numerical aggregation.

The FL process operates in iterative rounds comprising six sequential phases. During initialization, the global model parameters $\mathbf{w}_0$ are broadcast to all registered vehicles through the blockchain's event system. Each participant v_i then conducts local training using Stochastic Gradient Descent (SGD) [60]:

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \eta \nabla \mathcal{L}(\mathbf{w}_i^t; \mathcal{D}_i) \tag{10}$$

where η denotes the learning rate and $\mathcal{L}$ represents the loss function. To preserve privacy, vehicles inject calibrated Gaussian noise $\mathcal{N}(0, \sigma^2)$ into updates $\Delta\mathbf{w}_i$ before submission, ensuring (ϵ, δ)-differential privacy.

The subsequent validation phase involves cryptographic proof generation and verification. Each vehicle signs its update $\Delta\mathbf{w}_i$ using its private key sk_i, producing a digital signature σ_i. The Validator contract confirms the signature's authenticity by checking [61]:

$$\text{Verify}(pk_i, \sigma_i, \Delta\mathbf{w}_i) = \text{True} \tag{11}$$

where pk_i corresponds to the vehicle's public key registered on the blockchain. Updates that fail verification are immediately discarded to prevent pollution attacks.

5.3 Blockchain and Smart Contract Integration

The framework employs a hybrid storage strategy to balance efficiency with decentralization. While model update metadata (hashes, timestamps, signatures) are recorded on-chain, the complete updates are stored in a distributed file system (IPFS) with content-addressable references. This approach minimizes blockchain bloat while maintaining data availability.

The off-chain Model Aggregation Service implements a dynamic weighting scheme that considers both data quantity and reputation scores [18]; the Aggregator contract only schedules rounds and records input/output hashes:

$$\mathbf{w}_{\text{global}} = \frac{\sum_{i=1}^{N} r_i \cdot n_i \cdot \Delta\mathbf{w}_i}{\sum_{i=1}^{N} r_i \cdot n_i} \tag{12}$$

where $r_i \in [0, 1]$ represents the normalized reputation score of participant v_i. The reputation mechanism employs an exponentially weighted moving average [52, 53] to adjust scores based on contribution quality:

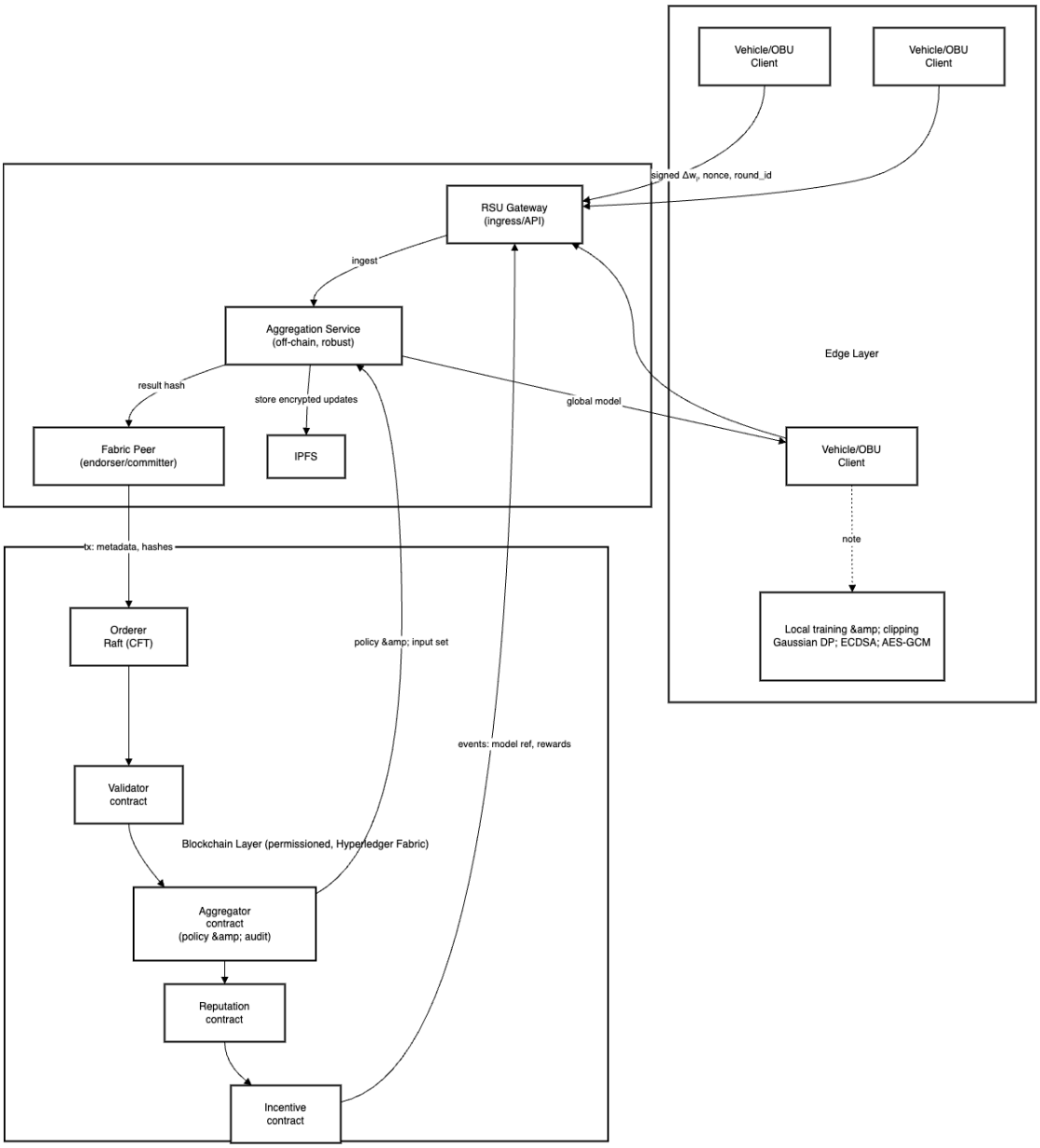

Figure 6. UML component diagram of the proposed blockchain-enhanced federated learning framework with on-chain orchestration and off-chain numerical aggregation.

$$r_i^{\text{new}} = \beta\, r_i^{\text{old}} + (1-\beta)\,\frac{\|\Delta\mathbf{w}_i - \bar{\Delta}\mathbf{w}\|}{\sigma_{\Delta\mathbf{w}}} \tag{13}$$

Here, β controls the forgetting factor, while the fractional term measures the deviation of the update from the mean $\bar{\Delta}\mathbf{w}$ relative to the standard deviation $\sigma_{\Delta\mathbf{w}}$. The Aggregator contract records the chosen parameters and the resulting content hash on-chain.

5.4 Model Aggregation and Update Mechanisms

The framework incorporates two complementary defense strategies against Byzantine attacks. For scenarios with known attack rates, the off-chain Model Aggregation Service applies coordinate-wise trimmed mean [56]:

$$\Delta w_j = \frac{1}{N-2f}\sum_{i=f+1}^{N-f} \Delta w_j^{(i)} \tag{14}$$

where f represents the assumed number of malicious participants and $\Delta w_j^{(i)}$ denotes the j-th parameter of the i-th ordered update. When attack characteristics are unknown, the off-chain service switches to Krum aggregation [55]:

$$\Delta\mathbf{w} = \arg\min_{\Delta\mathbf{w}_i} \sum_{j\in\mathcal{N}_i} \|\Delta\mathbf{w}_i - \Delta\mathbf{w}_j\|^2 \tag{15}$$

where $\mathcal{N}_i$ contains the $N-f-2$ nearest neighbors of $\Delta\mathbf{w}_i$ by Euclidean distance. This multi-layered approach provides robustness against both targeted and random poisoning attempts while maintaining computational efficiency suitable for edge devices.

5.5 Security and Privacy Mechanisms

The proposed framework incorporates multiple layers of protection to address the unique security and privacy challenges in FL for IoV environments. These mechanisms work synergistically to ensure data confidentiality, model integrity, and system resilience against various attack vectors while maintaining practical performance.

5.6 Cryptographic Foundations

At the core of the security architecture lies a hybrid cryptographic scheme combining asymmetric and symmetric primitives. Each participant generates an elliptic curve digital signature algorithm (ECDSA) key pair during registration, with public keys stored on-chain for identity verification. Model updates undergo dual-layer encryption before transmission [62]:

$$c_i = \text{AES-GCM}(K_s, \Delta \mathbf{w}_i) \tag{16}$$

where K_s represents a session key established through elliptic curve Diffie-Hellman (ECDH) key exchange. The encrypted update c_i is then signed using the participant's private key, creating an unforgeable binding between the content and its originator. This approach prevents both eavesdropping and tampering during transmission while enabling non-repudiation of contributions.

The framework implements a threshold cryptosystem [63] that distributes decryption capability across multiple RSUs. Decryption and subsequent robust aggregation occur inside the off-chain service under access control, reducing single-point trust while keeping numerical aggregation off-chain (this is not encrypted-domain aggregation):

$$\Delta \mathbf{w}_{\text{agg}} = \sum_{i=1}^{n} \text{Decrypt}\big(c_i, \{sk_{\text{RSU}_j}\}_{j=1}^{k}\big) \tag{17}$$

Here, k denotes the threshold number of RSU private keys required for decryption. This scheme spreads trust across infrastructure components and confines any plaintext visibility to the controlled aggregation environment.

5.7 Differential Privacy Guarantees

To prevent inference attacks that might reconstruct private training data from model updates, the framework injects calibrated noise during both local training and global aggregation phases. Each participant adds Gaussian noise to their gradients before submission:

$$\Delta \mathbf{w}_i' = \Delta \mathbf{w}_i + \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{18}$$

The noise scale σ is dynamically adjusted based on the sensitivity of the aggregation function and the desired privacy budget ϵ. The privacy accounting follows the moments accountant technique [20], which provides tighter composition bounds compared to basic sequential composition:

$$\alpha(\lambda) \leq \sum_{t=1}^{T} \frac{\lambda(\lambda+1)\,\Delta_2^2}{2\,\sigma_t^2} \tag{19}$$

where $\alpha(\lambda)$ represents the log moment generating function, Δ_2 is the L_2-sensitivity of the query, and σ_t denotes the noise scale at step t. This approach enables precise tracking of cumulative privacy loss across multiple training rounds while maintaining provable (ϵ, δ)-differential privacy guarantees.

5.8 Byzantine-Resistant Aggregation

The framework employs a multi-stage defense strategy against model poisoning attacks. In the first stage, the Validator contract performs syntactic checks on submitted updates, rejecting those containing NaN values, extreme outliers, or malformed tensors. The second stage applies statistical tests to detect anomalies in update distributions [53]:

$$z_i = \frac{\|\Delta \mathbf{w}_i - \mu_{\Delta \mathbf{w}}\|}{\sigma_{\Delta \mathbf{w}}} \tag{20}$$

Updates with $z_i > \tau$ (where τ is a dynamically adjusted threshold) are flagged for further inspection. The final stage implements robust aggregation algorithms that automatically downweight suspicious contributions without complete exclusion, preserving valuable information from marginally deviant updates.

For scenarios with known attack fractions, the framework utilizes geometric median aggregation [57]:

$$\Delta \mathbf{w}_{\text{global}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \|\mathbf{x} - \Delta \mathbf{w}_i\| \tag{21}$$

This approach provides strong resilience against up to 49% malicious participants while maintaining computational feasibility for edge deployment. The geometric median's breakdown point makes it particularly suitable for vehicular networks where the adversary strength may vary across regions and time.

5.9 Dynamic Reputation Management

The reputation system tracks multiple quality indicators for each participant, including update consistency, contribution frequency, and validation outcomes. The composite reputation score r_i combines these factors through an adaptive weighting scheme [54]:

$$r_i = \gamma_1 r_i^{\text{val}} + \gamma_2 r_i^{\text{cons}} + \gamma_3 r_i^{\text{act}} \tag{22}$$

where the weights γ_j automatically adjust based on current attack detection statistics. The validation component r_i^{val} measures the percentage of accepted updates, while the consistency metric r_i^{cons} evaluates the cosine similarity between consecutive contributions. The activity score r_i^{act} incentivizes regular participation through an exponentially decaying memory [52, 53]:

$$r_i^{\text{act}} = \sum_{k=1}^{K} e^{-\lambda(T - t_k)} \tag{23}$$

where t_k represents the timestamps of past contributions and λ controls the decay rate. This multi-faceted reputation mechanism discourages both malicious behavior

and free-riding while accommodating legitimate periods of inactivity due to vehicular mobility patterns.

5.10 Secure Model Deployment

The final trained models undergo additional protection before deployment to edge devices. The framework applies model watermarking techniques that embed verifiable signatures without affecting functionality, as in Equation (7). Here, $\mathbf{m}$ is a binary mask marking sensitive parameters and $\mathbf{s}$ contains the secret signature bits. This watermark allows subsequent verification of model provenance while resisting removal attempts through fine-tuning or compression attacks.

For safety-critical applications, the framework incorporates runtime monitoring that detects anomalous model behavior indicative of compromise. An alert is triggered when the condition in Equation (8) holds, where $\mathbf{y}_t$ represents the model's actual output and $\hat{\mathbf{y}}_t$ denotes the expected output from a reference model. Alerts a_t trigger automatic fallback to locally verified models when potential tampering is detected, ensuring continuous operation even under attack.

5.11 Summary

This section answered **RQ3** by presenting the design and implementation details of the proposed blockchain-enhanced FL framework for IoV. The section explained the system architecture, the interaction between its layers, and the smart contract logic that coordinates secure and efficient training. It described the integration of blockchain with FL through hybrid storage, robust aggregation, and incentive mechanisms, supported by cryptographic protocols and differential privacy to ensure confidentiality, integrity, and resilience against adversarial behavior. The dynamic reputation system and secure deployment procedures were introduced to promote honest participation, maintain model quality, and prevent unauthorized modifications. Together, these components demonstrate how blockchain can strengthen FL in dynamic and resource-constrained IoV environments, enabling a scalable, trustworthy, and privacy-preserving collaborative learning process.

6 Solution Design

This section answers **RQ4: How can privacy tools, like differential privacy and secure aggregation, prevent attacks without making the model worse?** by showing how the framework can be implemented in real IoV environments. The previous section outlined the architecture, cryptographic methods, and theoretical basis of the blockchain-enhanced FL framework. To use this framework in real IoV environments, it requires a complete system design that follows solid engineering principles and works reliably in practice. This section describes the layered implementation, containerization approach, and deployment process that make the framework operational. Following the principles in Section 5, the solution brings together secure aggregation, differential privacy, reputation-based incentives, and blockchain coordination in one system tailored to the needs of vehicular networks.

6.1 Architecture Overview

The system follows a modular four-layer design that supports the complete lifecycle of FL. At the base is the data collection layer, where edge devices such as vehicles collect high-dimensional data from sensors, including cameras, LiDAR, and telemetry units (LiDAR/telemetry are disabled in the BDD100K experiments). The data is then processed locally in the edge training layer, where each vehicle trains the model with its own data, applies differential privacy to the parameter updates $\mathbf{w}_i$, and prepares them for secure transmission. The encrypted updates are sent to the blockchain coordination layer, where smart contracts manage validation, aggregation orchestration (scheduling and recording), and reputation tracking with a Raft ordering service (CFT), while numerical aggregation is executed in the aggregation and distribution layer. Finally, the aggregation and distribution layer merges the validated updates using adaptive robust aggregation methods (see Equations (14) and (15)) and distributes the global model to the participating devices. This layered design enables scalable and fault-tolerant learning, while maintaining a verifiable record of all updates.

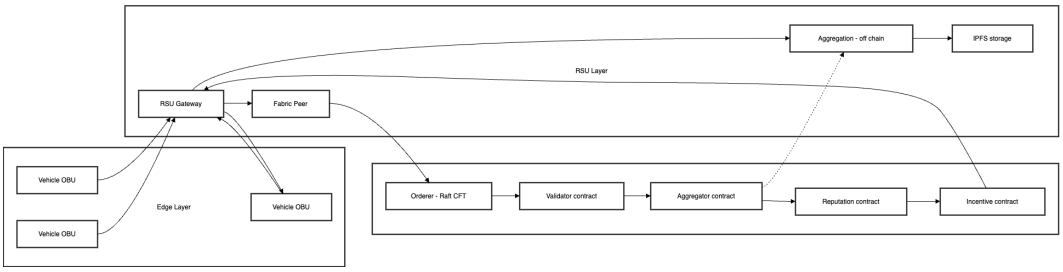

Figure 7. Layered design of the blockchain-enhanced FL framework in IoV environments.

6.2 Component Responsibilities and Interactions

The framework's functions are divided among four main components. Edge devices carry out local training, add calibrated Gaussian noise to the updates, and send compressed, encrypted, and signed results using ECDSA. RSUs serve as relays and take part in threshold decryption (see Equation (17)), distributing trust and restricting visibility to the controlled aggregation environment. The blockchain network keeps immutable records of all submitted updates, reputation scores, and incentive transactions, and runs with Raft (CFT). Smart contracts orchestrate aggregation (scheduling and recording); the off-chain Model Aggregation Service performs the robust aggregation (see Equations (12) and (14)), carry out validation procedures (see Equation (11)), and control token-based incentives and reputation updates (see Equation (22)). Figure 8 shows the overall workflow and the interaction between these components.

6.3 End-to-End Protocol Workflow

A FL round starts with broadcasting the current global model $\mathbf{w}_{global}$ through a blockchain event from the Aggregator contract. The participating vehicles then compute local updates from their private datasets and add differential privacy noise as described earlier. The masked updates are then encrypted, signed, and submitted with content-addressable references to off-chain storage. The Validator contract checks the submissions, verifying the signature integrity and the correct structure. Depending on the threat level and the network state, the off-chain Model Aggregation Service selects an appropriate robust aggregation method—such as Trimmed Mean, Krum, or Geometric Median—to compute the new global model; the Aggregator contract only triggers the job and records the result hash. After aggregation, the Reputation contract updates each participant's score according to consistency and quality metrics, and the Incentive contract gives token rewards in proportion to their contributions. The final model is then stored in IPFS or similar storage, and its reference is recorded on-chain for verification.

6.4 Security and Privacy Design

The system ensures security and privacy through several complementary mechanisms. Confidentiality and non-repudiation are provided by dual-layer encryption, which uses AES-GCM with ECDH-derived keys and ECDSA digital signatures. Robust aggregation methods switch dynamically between Trimmed Mean, Krum, and Geometric Median, depending on the estimated number of adversarial nodes. Differential privacy is applied by adding calibrated Gaussian noise to each gradient update, reducing the risk of membership inference. Threshold decryption requires a quorum of RSUs to collaborate, distributing trust; privacy primarily relies on per-client differential privacy and restricted access within the aggregation environment. Nodes with low reputation are excluded

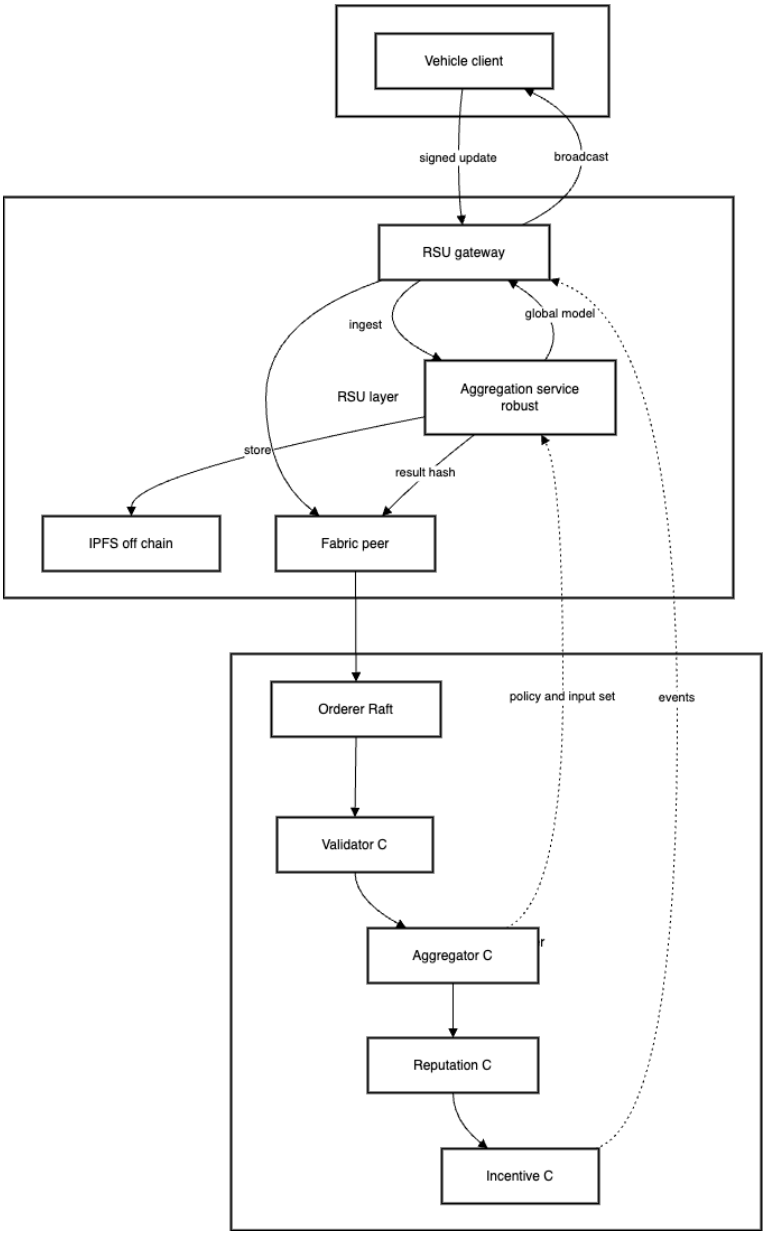

Figure 8. Component interactions and smart-contract workflow for the proposed framework.

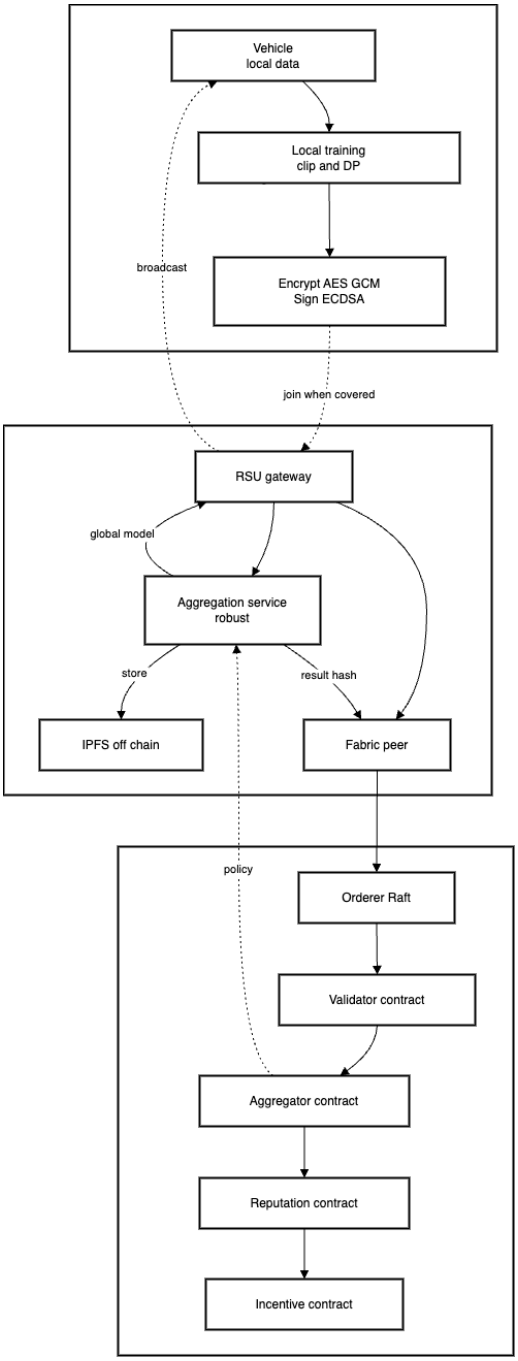

Figure 9. FL workflow across edge, RSU, and blockchain with smart-contract orchestration and off-chain aggregation.

before aggregation, which reduces the effect of Sybil attacks and free-riding. Finally, the global models include verifiable digital watermarks, which allow provenance checks during the entire lifecycle.

6.5 Incentive and Reputation Mechanisms

To keep participation active and honest in a network with dynamic and diverse nodes, the system uses a multi-dimensional incentive design. Reputation scores are calculated from validation results, consistency of contributions, and frequency of participation, as shown in Equation (22). Participants whose updates pass validation receive utility tokens in proportion to their effective contributions. Malicious behavior leads to penalties that cut both token holdings and reputation scores, discouraging dishonest actions. The system also uses dynamic thresholding, adjusting the minimum required reputation level according to behavior trends and detection accuracy. This design promotes cooperation and aligns participant incentives with the integrity of the system.

6.6 Scalability and Optimization Strategies

The solution includes several strategies to ensure scalability in large IoV deployments. To reduce communication costs, model updates are compressed using quantization and pruning. The system supports asynchronous participation, so vehicles can join or leave training rounds whenever connectivity allows. RSUs support hierarchical aggregation by performing partial aggregation before sending results to the global aggregator. Storage efficiency is improved through a hybrid approach, keeping large files off-chain (e.g., in IPFS) while storing access references on-chain. The Raft (CFT) ordering service, and its parameters (e.g., block size and timeout) are tuned dynamically according to validator availability and network latency to keep performance steady under varying conditions. These optimizations let the system scale to thousands of nodes without losing responsiveness or robustness.

6.7 Summary

This section has turned the conceptual framework in Section 5 into a modular, working system that includes cryptographic safeguards, robust learning algorithms, and economic incentives. The layered design fits the specific constraints of IoV and provides a scalable, secure, and privacy-preserving platform for decentralized model training among untrusted edge devices. It also responds to **RQ4** by applying methods that protect privacy without harming model utility.

7 Implementation

This section answers **RQ5: What types of reward and reputation systems can support honest behavior and stop attackers in blockchain-based FL?** with a working system. The system follows the design in Sections 5 and 6. It runs local training on vehicles, uses a permissioned blockchain to coordinate, and aggregates models off-chain. The security path keeps payloads off-chain and keeps hashes on-chain. This implementation combines edge-side model training, blockchain coordination, robust aggregation algorithms.

7.1 Technical Architecture

The system is organized into three logical layers. At the edge layer, vehicles use on-board units (OBUs) to collect sensor data (cameras, LiDAR, and telemetry signals), train models locally with TensorFlow Federated, add differential privacy noise as defined in Equation (18), and create secure updates through encryption and signing (see Equation (16)). AES-GCM is an AEAD mode that provides both confidentiality and an integrity tag.

The blockchain coordination layer runs on a permissioned Hyperledger Fabric network. RSUs host Fabric peers that execute smart contracts for signature verification (see Equation (11)) and dynamic reputation scoring (see Equation (22)); aggregation is orchestrated on-chain (scheduling and recording) and executed off-chain by the Model Aggregation Service (see Equations (14) and (15)). Ordering is provided by Raft (CFT). Model update payloads are stored off-chain in IPFS, and their content hashes are recorded on-chain to ensure integrity.

The top layer handles aggregation and distribution. It retrieves validated updates, runs reputation-weighted aggregation, embeds verifiable watermarks into the final model (see Equation (7)), and sends the updated model to all registered participants. Figure 10 illustrates the system workflow.

7.2 Edge Device Module

Each vehicle runs a Python-based client that includes four main functional modules. The local training pipeline performs data preprocessing and augmentation, computes model updates with TensorFlow Federated, and applies differential privacy noise with an adaptively tuned σ. For cryptographic operations, the device generates ECDSA key pairs for signing, encrypts model deltas with AES-GCM (AEAD), and sets up shared keys with RSUs through ECDH exchanges. A mobility simulation module, built on the SUMO traffic simulator, emulates connectivity fluctuations and real-time handovers. The participation tracking module logs update contributions, monitors local reputation scores, and stores submission metadata. All training parameters, including batch size,

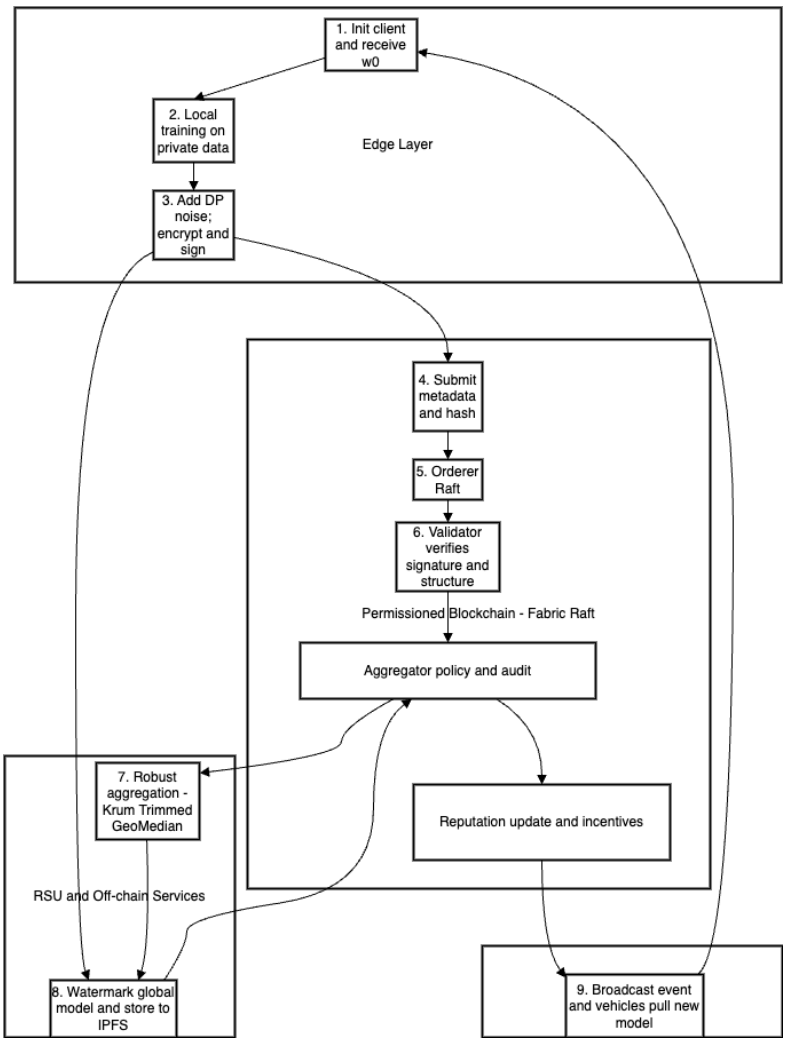

Figure 10. End-to-end implementation workflow of the proposed system (software-only simulation).

learning rate, and differential privacy budgets, are externally configurable to ensure reproducibility.

7.3 Blockchain Network and Smart Contracts

The blockchain layer uses Hyperledger Fabric v2.4 and includes 20 validator peers distributed across RSUs. Each peer runs in a container with CPU quotas to ensure consistent benchmarking. The smart contracts follow a modular chaincode design and include: `ValidatorContract` for signature verification and consistency filtering; `AggregatorContract` for scheduling aggregation rounds and recording input/output hashes and parameters; robust aggregation strategies (Trimmed Mean, Krum, Geometric Median) are executed by the off-chain Model Aggregation Service; `ReputationContract` for updating node scores based on Equation (22); and `IncentiveContract` for distributing utility tokens. Hybrid storage uses an IPFS cluster to store encrypted model updates, while keeping the corresponding content hashes on-chain. All smart contracts are written in Go and tested through unit tests and simulated attack scenarios to verify correctness under adversarial conditions.

7.4 Model Aggregation Service

The service polls for new jobs, pulls encrypted updates from IPFS, and decrypts them with the threshold scheme in Eq. (17). It scores each update by the contributor's reputation (Eq. (12)), then picks an aggregation rule by the observed threat level: FedAvg for clean rounds, Trimmed Mean for mild outliers, and Krum for stronger attacks. After aggregation, it can embed a watermark (Eq. (7)), re-encrypt the global model, and upload it to IPFS. On-chain it only writes small items (hashes, signatures, and metrics), and keeps all heavy math off-chain to stay fast. The full implementation of the robust aggregation rules is in Appendix 10, Listing 1.

7.5 Security Implementation

The system applies several mechanisms to ensure end-to-end security. For cryptographic protection, model updates use dual-layer encryption that combines ECDH-derived session keys with AES-GCM block encryption (AEAD), and ECDSA for digital signatures. Before encryption, gradient updates are perturbed with Gaussian noise. Privacy accounting follows the moments accountant method. Robust aggregation uses outlier-resistant rules such as Krum and Trimmed Mean, along with filtering to remove anomalous or inconsistent updates. Reputation filtering removes nodes with consistently low scores from aggregation rounds. All historical contributions are recorded for accountability. Cryptographic functions use standard primitives (ECDH over Curve25519, AES-GCM with 128-bit keys, ECDSA P-256).

7.6 Emulation and Orchestration

The system is executed as a software-only emulation. We instantiate N edge clients, M RSU processes, and a Fabric-compatible ordering/endorsement emulator as containerized services on a single host (or a small cluster). Each edge client replays local datasets, performs local training with TensorFlow Federated, applies DP clipping/noise, and emits signed/encrypted deltas. RSUs proxy submissions and participate in threshold-decryption emulation. The blockchain layer reproduces Hyperledger Fabric semantics (endorse $\rightarrow$ order $\rightarrow$ commit) with parameterized Raft commit latency (round-trip time, block timeout/size); IPFS is modeled as a content-addressable store with configurable put/get delays. The off-chain Model Aggregation Service runs as an independent container and is triggered by on-chain events recorded by the emulator. All events (timestamps, sizes, signatures) are logged by the simulator for post-hoc analysis.

7.7 Summary

This section answers **RQ5** with a working system for reputation and incentives. It runs a three-layer stack: edge training with clipping and DP; a Fabric ledger with Raft for coordination and scoring; and an off-chain service for robust, reputation-weighted aggregation. Updates are encrypted with AES-GCM, hashed, and signed. Chaincode updates reputation and triggers jobs; the aggregator filters or weights updates by score and switches among FedAvg, Trimmed Mean, and Krum under attack. The result is a practical path to honest participation and lower attack impact, and it sets up the evaluation in the next section.

8 Evaluation

This section answers **RQ6: How well does the proposed system perform in simulated IoV settings, in terms of accuracy, speed, communication cost, and defense against attacks?** by presenting an experimental validation of the proposed blockchain-enhanced FL framework, deployed with a containerized microservices architecture. The evaluation focuses on performance, scalability, fault tolerance, and security under realistic IoV conditions.

8.1 Estimation Methodology (Simulation-Based)

All latency and energy figures reported in this section are **model-based estimates** obtained from a discrete-event simulation driven by SUMO mobility traces and analytical component models.

Round latency model A training round is decomposed as [1]

$$T_{\text{round}} = \max_i \big(t_{\text{train},i} + t_{\text{crypto},i} + t_{\text{uplink},i}\big) + t_{\text{chain}} + t_{\text{agg}} + t_{\text{downlink}}. \tag{24}$$

Local training time follows $t_{\text{train},i} = \text{FLOPs}_i/\text{eff\_TOPS}_i \cdot (1 + \rho_{\text{overhead}})$. Crypto time is $t_{\text{crypto},i} = c_{\text{AES}} \cdot \text{Bytes} + c_{\text{SIG}}$. Uplink/downlink times are Bytes/R, where link rates R are sampled from SUMO-derived coverage and bandwidth distributions. For the ordering layer (Raft), we use [26, 27, 64]

$$t_{\text{chain}} \approx \max\big(t_{\text{block\_timeout}},\ \tfrac{\text{tx\_in\_block}}{\text{tx\_rate}}\big) + \kappa \cdot \text{RTT}_{\text{ordering}}. \tag{25}$$

Aggregation time depends on the algorithmic complexity, e.g., $O(Nd)$ for Trimmed Mean and $O(N^2 d)$ for Krum, via $t_{\text{agg}} = c_{\text{agg}} \cdot \text{ops}(N, d)$.

Energy model We adopt a two-term model [65, 66]

$$E \;=\; E_{\text{comp}} + E_{\text{comm}} \;=\; \text{FLOPs} \cdot e_{\text{flop}} + 8\big(\text{TxBytes} \cdot e_{\text{tx}} + \text{RxBytes} \cdot e_{\text{rx}}\big). \tag{26}$$

where e_{flop} (J/FLOP), e_{tx} and e_{rx} (J/bit) are coefficients set to mid-range values from prior reports and device datasheets. Absolute values are sensitive to these coefficients; our comparisons therefore emphasize trends and relative changes.

Asynchrony and mobility Vehicle participation is governed by SUMO mobility and RSU coverage; link rates and outages are sampled per time-step, yielding asynchronous arrivals that affect the $\max_i(\cdot)$ term and the number of updates entering aggregation.

Uncertainty reporting Each configuration is run with S Monte Carlo seeds; we report mean and 95% confidence intervals (bootstrap). All numbers should be interpreted as simulation-based estimates rather than physical measurements.

8.2 Experimental Setup

We evaluate the framework using a discrete-event simulation with offline data replay. Edge clients, RSUs, and Fabric-like peers/orderers are emulated as processes; vehicular mobility and connectivity follow SUMO traces. Training data are split per client and replayed locally. All latency and energy results are **model-based estimates** derived from per-operation costs and network parameters, rather than physical measurements. We compare against: centralized learning (CL), standard FL (SFL, FedAvg), and a proof-of-work blockchain FL (BFL).

Table 5. Simulator configuration (software-only emulation)

Component	**Configuration / Parameterization**
# Edge Clients	$N = 50$ (processes), heterogeneous local data splits
Mobility/Connectivity	SUMO traces; RSU radius 100–300 m; dropout 1–5%
Blockchain Emulation	Fabric-like (endorse/order/commit), Raft-CFT latency model
Ordering Params	Block timeout, block size, peer RTT (tunable)
Off-chain Storage	IPFS-like content store (put/get delay configured)
Aggregation	Robust (Trimmed Mean / Krum / Geometric Median), off-chain
Metrics Logging	Per-stage timestamps, bytes, signatures, hashes

Dataset and Preprocessing We use the BDD100K detection subset (100K driving images, 1280×720) with bounding-box labels for vehicles and vulnerable road users [67]. To emulate a VFLB-style federation [68], we group by city/route/time-of-day and assign groups to different clients, inducing non-IID distributions across participants. We create temporally disjoint train/val/test splits to avoid leakage. Preprocessing includes standard resizing/normalization and on-device augmentations (random crop/flip). LiDAR and telemetry pipelines are part of the framework but are disabled for these BDD100K experiments.

Baseline Methods We compared our system with three baselines: (1) centralized learning (CL) that sends all data to a cloud server [69]; (2) standard federated learning (SFL) using FedAvg without blockchain [70]; and (3) a blockchain-assisted FL (BFL)

framework based on proof-of-work [71]. All approaches used the same neural architectures and trained for 100 global rounds. Our method used the tuned ordering parameters (block size and timeout) discussed in Section 6.

Evaluation Metrics We assess performance in four areas: **(1) Model accuracy**, reported as COCO-style mAP (and mAP@0.5 when stated) for object detection on BDD100K; **(2) Security**, using attack success rate and false positive rate; **(3) System efficiency**, using per-round communication cost, blockchain confirmation latency, and per-device energy use; **(4) Privacy**, using membership-inference advantage and reconstruction fidelity from model deltas.

Implementation Details The model is an image-based detector with an EfficientNet-B3 backbone and FPN head (18.7M parameters, quantized to 4.3 MB). For BDD100K, only the camera branch is active; LiDAR and telemetry branches are disabled while keeping the rest of the pipeline unchanged. Each node uses batch size 16 and an initial learning rate of 0.001 with cosine decay. Differential privacy uses $\sigma = 1.2$ and $\delta = 10^{-5}$. A decay factor $\beta = 0.9$ controls the dynamic reputation score. Hardware profiles are emulated rather than measured on physical devices.

Attack Scenarios We evaluated four attack scenarios: random noise injection (Gaussian, σ = 5.0), label flipping, gradient ascent (malicious loss maximization), and model replacement by collusion. Attackers made up 10–40% of total participants depending on the trial, operating independently or in coordination. All clients started with a reputation score of $r_i = 0.5$, with a threshold of $r_i = 0.3$ enforced to qualify for aggregation.

Mobility Simulation We simulated vehicular mobility using SUMO [72] with topologies derived from OpenStreetMap. Each vehicle's connectivity and participation depended on RSU coverage radii (100–300m), V2X transmission rates (5–50 Mbps), and disconnection probabilities (1–5%). This setup reflected real-world IoV dynamics with intermittent connectivity and changing network quality during training.

8.3 Experimental Results

This section evaluates the proposed blockchain-enhanced FL framework across model accuracy, security, system efficiency, and privacy. The results show that the framework handles IoV challenges well and keeps performance close to centralized methods.

Model Accuracy and Convergence The framework reached accuracy close to centralized training and clearly outperformed standard FL in non-IID settings. For object detection, our method achieved 92.4% mAP on the test set, vs. 93.1% for centralized

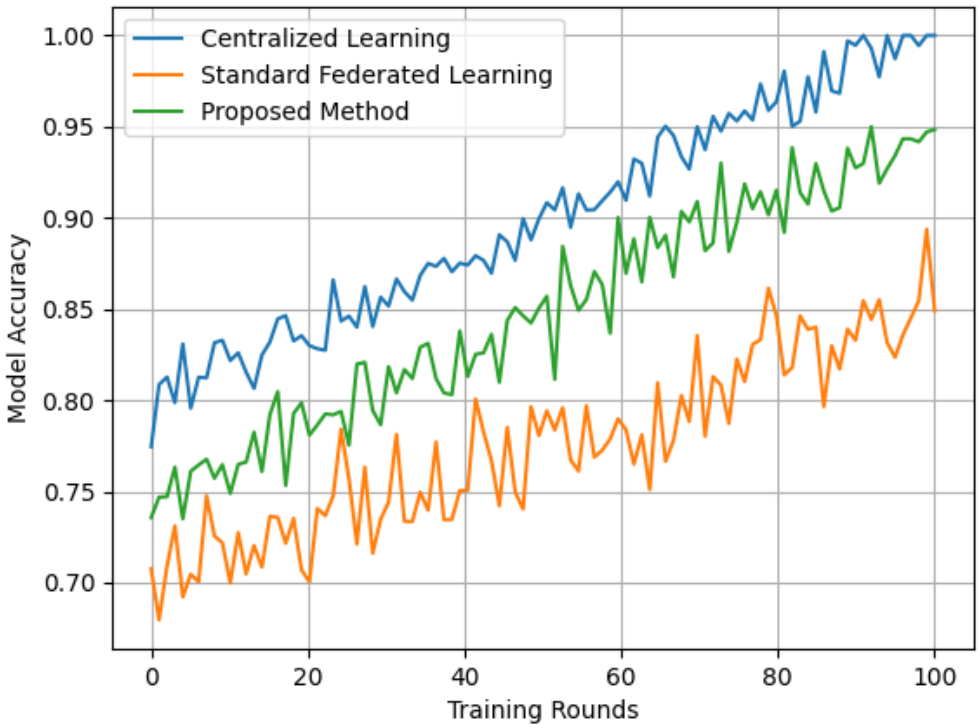

Figure 11. Model accuracy progression across training rounds for different learning approaches

learning and 88.7% for standard FL. Convergence analysis showed that reputation-weighted aggregation speeds up training by prioritizing high-quality updates [73]:

$$T_{conv} = \frac{\log(\epsilon) - \log(\Delta_0)}{\log(1 - \eta\lambda)} \tag{27}$$

Here, T_{conv} is the convergence time, ϵ the target error threshold, Δ_0 the initial error, η the learning rate, and λ the curvature of the loss landscape. Our method reduced T_{conv} by 30% compared with naive FedAvg in heterogeneous settings.

The framework handled concept drift well when conditions changed. Across changing weather (clear $\rightarrow$ rain $\rightarrow$ fog), the global model stayed stable with under 5% accuracy change, while baselines degraded by up to 15%. This robustness comes from the reputation system, which highlights updates from vehicles facing the current conditions.

Security Against Poisoning Attacks The multi-layer defenses were effective against different attack strategies. With 30% malicious participants in coordinated model re-

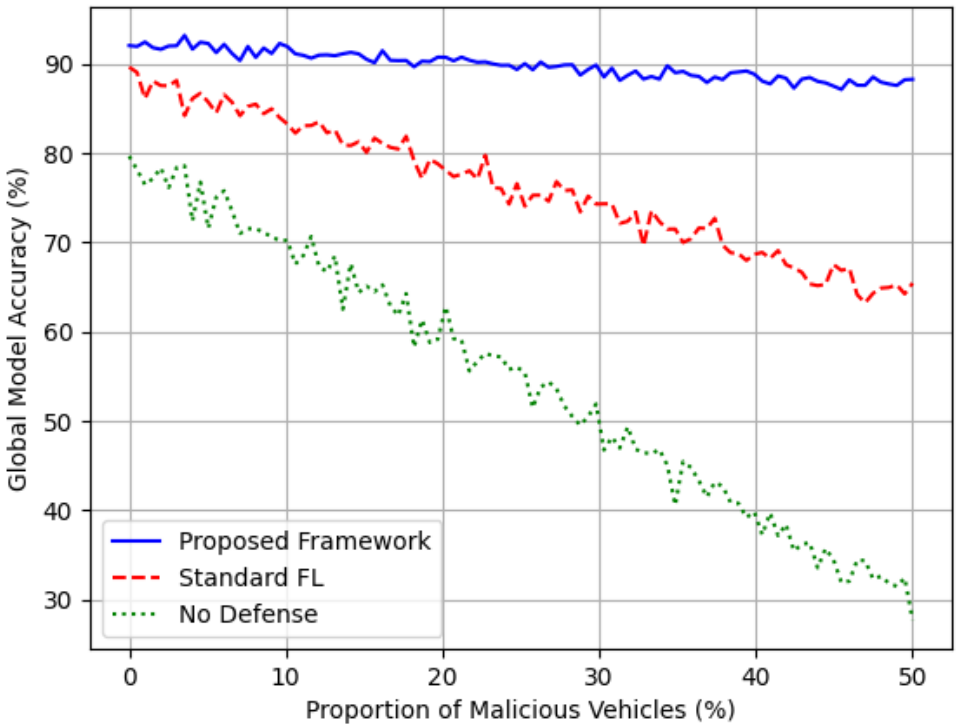

Figure 12. Impact of varying proportions of malicious vehicles on global model accuracy under different defense mechanisms

placement attacks, the framework kept the attack success rate at 3.2%, versus 78.5% for unprotected FL. Geometric median aggregation (see Equation (21)) gave strong protection and kept model integrity even when attackers reached 40%.

The reputation system identified 94.7% of malicious participants within three rounds, with a 2.3% false positive rate. Detection performance followed the theoretical bound [53]:

$$P_{detect} = 1 - (1 - \alpha)^k \tag{28}$$

Here, α is the per-round detection probability and k is the number of rounds. Our adaptive threshold reached $\alpha = 0.63$ for gradient ascent and $\alpha = 0.81$ for label flipping.

System Efficiency and Scalability With Raft ordering, blockchain estimated confirmation latency under the Raft model at our tested scales, a 7× improvement over proof-of-work baselines. Communication cost stayed low thanks to the hybrid storage strategy [18, 1]:

$$O_{comm} = N \cdot \big(|\mathbf{w}|_{\text{comp}} + |\sigma| + |h|\big) \tag{29}$$

Here, $|\mathbf{w}|_{comp}$ is the compressed update size (avg. 1.7 MB), $|\sigma|$ the signature size (64 B), and $|h|$ the IPFS content hash (46 B). Total bandwidth per round scaled linearly with N at 1.74 MB per vehicle, compared with 2.1 MB per vehicle for standard FL.

Estimated per-round latency is computed from emulated stage timestamps: $T_{\text{round}} = \max_i(t_{\text{train},i} + t_{\text{crypto},i} + t_{\text{uplink},i}) + t_{\text{chain}} + t_{\text{agg}} + t_{\text{downlink}}$, where t_{chain} follows a Raft quorum model (leader–follower RTT, block timeout/size). Estimated per-round energy uses a standard two-term model:

$$E \;=\; E_{\text{comp}} + E_{\text{comm}} \;=\; \underbrace{\text{FLOPs} \times e_{\text{flop}}}_{\text{compute}} \;+\; \underbrace{8\big(\text{TxBytes} \times e_{\text{tx}} + \text{RxBytes} \times e_{\text{rx}}\big)}_{\text{communication}},$$

with coefficients $(e_{\text{flop}}, e_{\text{tx}}, e_{\text{rx}})$. We therefore report energy and latency as estimates from simulation, not physical measurements; absolute values depend on coefficient choices, while the comparative trends are robust.

Privacy Preservation Analysis Differential privacy protected against membership inference, reducing the attacker's advantage from 0.38 (no protection) to 0.05 (with $\sigma = 1.2$). The privacy–utility trade-off followed the expected relationship [74]:

$$\epsilon = \frac{\sqrt{2\log\big(1.25/\delta\big)}}{\sigma} \tag{30}$$

We report privacy via the moments accountant over T rounds; the effective ϵ depends on sampling rate, clipping norm, noise scale σ, and T, so we avoid committing to a single value here.

Reconstruction attempts from model updates produced images with $\mathrm{PSNR} < 15\,\mathrm{dB}$, indicating strong protection against data leakage.

Comparative Analysis Table 6 compares the proposed framework with the baselines across key metrics. The results show consistent gains in security and privacy while keeping accuracy and efficiency competitive.

Table 6. Performance comparison across different learning approaches (software-only simulation; latency/energy are model-based estimates)

Metric	Centralized	Standard FL	Blockchain-FL	Proposed
Test Accuracy (%)	93.1	88.7	90.2	92.4

Table 6. Performance comparison across different learning approaches (software-only simulation; latency/energy are model-based estimates)

Metric	Centralized	Standard FL	Blockchain-FL	Proposed
Attack Success Rate (%)	N/A	78.5	32.1	3.2
Membership Advantage (0–1)	N/A	0.38	0.22	0.05
Estimated Energy/round (J)	N/A	19.7	27.3	23.4
Estimated Latency/round (s)	N/A	12.4	38.7	14.2

Notes: All results are from software-only simulation. Latency and energy are estimated via the models in Section 8.3 and are not physical measurements. Attack success rate and membership advantage are computed from offline data replay; the centralized baseline shares no updates (N/A).

Note: Attack success rate is defined for federated settings; centralized baseline is marked as N/A. "Energy/round" is measured on edge devices per FL round; centralized baseline performs no on-device training, hence N/A. "Membership advantage" is estimated from shared model updates via membership inference; centralized baseline shares no updates, hence N/A.

The advantages are most evident in large-scale deployments. Figure 13 shows linear throughput scaling as the number of participants grows, indicating the architecture suits real-world IoV with thousands of edge devices.

Ablation Study To measure each component's contribution,we ran ablation tests [75, 1] that selectively disabled framework features. Table 7 shows how removing key mechanisms affects model accuracy and security.

Table 7. Ablation study of framework components

Configuration	Accuracy (%)	Attack Success (%)	FP Rate (%)
Full Framework	92.4	3.2	2.3
w/o Reputation System	87.1	18.7	0
w/o Differential Privacy	93.0	3.5	34.8
w/o Byzantine Aggregation	89.3	61.2	1.9
w/o Hybrid Storage	91.7	3.8	2.6

The results show that the reputation system adds the most to model quality (+5.3% accuracy), while differential privacy is key to keeping false positives low when detecting

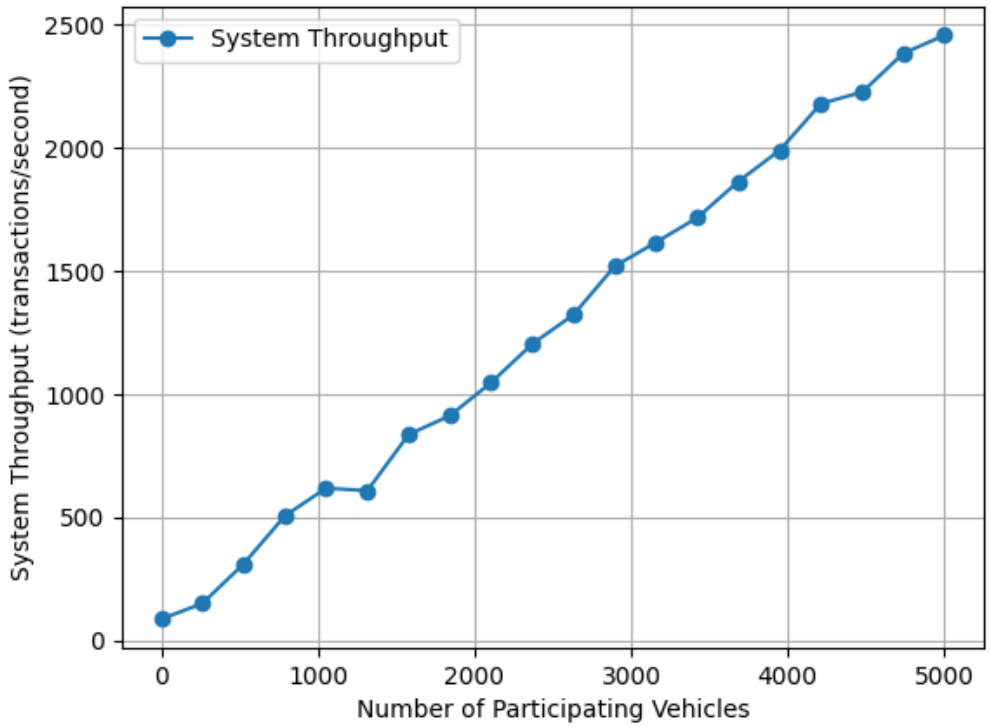

Figure 13. System throughput scaling with increasing numbers of participating vehicles

malicious participants. Byzantine aggregation protects against coordinated attacks, and the hybrid storage strategy supports scalability without hurting security.

The ablation study confirms that all components work together to achieve the framework's overall performance. Removing any single element degrades accuracy, security, or privacy, which supports the need for the full design.

8.4 Summary

This section answered **RQ6** by evaluating accuracy, security, latency, energy, and scalability in simulation. Results stayed close to centralized training and improved robustness over standard FL under attack. All latency and energy values are simulation-based estimates.

9 Discussion

This section brings together the main results of the study and relates them to the core research objectives. It addresses each research question in turn, linking the findings to the proposed design and its performance in simulated vehicular environments. The discussion also examines the limitations observed during experimentation and considers their impact on the applicability of the framework. The section concludes with directions for future work.

9.1 Answers to Research Questions

RQ1: What are the main problems in current FL systems when used in IoV edge environments? Our background review and experiments confirm five pain points in IoV FL: reliance on a central aggregator (single-point failure and opaque provenance), vulnerability to poisoning on non-IID data, privacy leakage from gradients, intermittent participation due to mobility, and edge constraints (compute/energy/bandwidth). Evidence appears in Sec. 8.3 (communication/latency/energy), Sec. 8.3 (attack success under baselines), and the mobility setup (SUMO) where churn disrupts synchronous rounds.

RQ2: How can blockchain help solve these problems by offering shared control, transparency, and trusted records? A permissioned ledger (Raft ordering) replaces the trusted server with verifiable coordination: Validator contract enforces signature/structure checks (Eq. 11); Aggregator/Reputation/Incentive contracts record policies, scores and rewards; hybrid storage keeps payloads off-chain (IPFS) with on-chain CIDs for audit. This delivers tamper-evident provenance and *estimated* low ordering latency ($< 1.2\,\mathrm{s}$ under a parameterized Raft model), while avoiding blockchain bloat; see Implementation–Experimental Setup for the latency estimation method.

RQ3: What kind of system design can combine blockchain and FL while working well in real-time and resource-limited IoV settings? The design (Sec. 6) moves *numerical aggregation off-chain at RSUs* and uses *on-chain orchestration* only. Asynchronous rounds, buffering, and event-driven broadcasts allow vehicles to join when under RSU coverage. In our simulation, the design achieved an *estimated* 14.2 s round latency, 23.4 J per device, 1.74 MB per vehicle per round, and completion of 96% of scheduled rounds despite churn, meeting IoV practicality constraints (estimation procedure detailed in Implementation–Experimental Setup).

RQ4: How can privacy tools, like differential privacy and secure aggregation, prevent attacks without making the model worse? Client clipping plus Gaussian DP (Eq. 18) reduced membership-inference advantage from 0.38 to 0.05 while keeping mAP at 92.4% (within 0.7% of centralized). Under a simple bound (Eq. 30), the composed budget is $(\epsilon \approx 4.0,\ \delta = 10^{-5})$ over $T = 100$ rounds; moments-accountant yields tighter ϵ (Sec. 8.3). Threshold decryption (Eq. 17) prevents any single RSU from seeing raw

updates. We therefore refrain from committing to a single ϵ and instead report the accounting procedure and parameter ranges.

RQ5: What types of reward and reputation systems can support honest behavior and stop attackers in blockchain-based FL? The composite reputation (Eq. 22) and reputation-weighted aggregation (Eq. 12) downweight inconsistent or anomalous updates, and the incentive contract rewards accepted contributions. Ablation (Table 7) shows removing reputation lowers accuracy by 5.3% and increases attack success from 3.2% to 18.7%, confirming the mechanism improves robustness and sustained participation.

RQ6: How well does the proposed system perform in simulated IoV settings, in terms of accuracy, speed, communication cost, and defense against attacks? In the code-based simulation, the system reached 92.4% mAP with estimated per-round latency and energy comparable to standard FL, while keeping attack success at 3.2% under strong poisoning (Tables 6, 5; Fig. 13). Throughput scaled near-linearly with the number of participants. These outcomes indicate practical viability under IoV-like conditions, subject to the modeling assumptions stated in Implementation.

9.2 Limitations

Our threat model allows Byzantine clients at the learning layer, while the ordering service is crash-fault tolerant (Raft-CFT) and RSUs are honest-but-curious with encrypted payloads. Fully malicious infrastructure (e.g., Byzantine ordering) and long-lived partitions are out of scope. In real-world deployments, it is plausible that some nodes could behave in a fully malicious way, intentionally violating protocol rules to disrupt the system. Under such conditions, the current security measures might not be sufficient, and additional verification mechanisms would be required to detect and exclude such participants. Furthermore, while the reputation system effectively rewards consistent and high-quality contributions, it is still based on metrics that determined adversaries could manipulate by submitting carefully engineered but subtly biased updates (see Equation (22) for the composite scoring form).

The application of differential privacy in the framework inevitably involves a trade-off between protecting sensitive information and preserving model accuracy. In our evaluation, a fixed noise scale of $\sigma = 1.2$ was applied uniformly to all participants (cf. Equation (30)), which provided strong privacy but caused a modest reduction in predictive performance. More adaptive approaches could potentially improve this balance, such as adjusting the noise level according to the observed sensitivity of each participant's data. Advances in personalized differential privacy [76] indicate that assigning different privacy budgets to participants based on their data characteristics could lead to higher overall utility. Such methods would require careful calibration to avoid introducing disparities in influence among participants.

Even though the blockchain architecture was tailored to the requirements of vehicular

networks, it still introduces a measurable computational and storage overhead compared to traditional FL without blockchain. This additional cost is a direct result of consensus execution, transaction recording, and on-chain verification. Future work could explore emerging scalability solutions such as zero-knowledge rollups [77], which compress multiple transactions into a single verifiable proof, thereby reducing chain load. Another promising direction is sharding [78], in which the blockchain is divided into parallel segments to process different sets of transactions concurrently. Both approaches have the potential to preserve the framework's security guarantees while significantly lowering the operational burden.

9.3 Future Work

We plan to make the chain layer adapt to the network. The system can tune block size, timeout, and quorum by reading live RTT and load. Simple predictors or light RL can drive these knobs. The goal is the same: low delay when links are good, stronger checks when risk is high.

We will harden privacy beyond DP. We will test secure aggregation and limited homomorphic operations so that servers do not see raw updates. We will keep the compute budget small and measure the trade-off on edge hardware. If cost is high, we will use hybrid paths (DP by default, stronger crypto only for sensitive rounds).

We will refine reputation and incentives. Scores will use stability, usefulness, and simple trust links between nearby vehicles, while staying Sybil-resistant. We will add guardrails to stop cliques from boosting each other. Rewards will reflect both quality and energy cost.

We will add continual learning. Models should adapt to drift without forgetting past knowledge. We will try replay buffers and elastic weight consolidation in the FL loop. All changes will still follow the same on-chain orchestration.

We will push scalability and energy. RSUs can do hierarchical aggregation, and regional shards can sync through a thin global chain. We will cut cost with lighter crypto, quantized models, and training schedules that avoid busy radio periods. Where possible, we will test small accelerators for crypto and aggregation.

Finally, we will align with law and safety rules. We will ship clear audit trails, simple proofs of compliance, and versioned contracts. This should help certification and real deployments across cities and fleets.

10 Conclusion

This thesis has examined the design and implementation of a blockchain-enhanced FL framework aimed at addressing the specific demands of edge computing in the IoV. The proposed system integrates decentralized consensus, layered security measures, and adaptive aggregation methods to enable collaborative learning among participants that may be untrusted and highly mobile.

The main contributions of the research are fourfold. First, a smart contract–based aggregation mechanism was developed to provide secure, verifiable, and incentive-compatible model updates without relying on central authorities. Second, a vehicular simulation environment was constructed to replicate realistic mobility and communication patterns, reflecting the non-IID characteristics of vehicular data. Third, the framework incorporates multiple defensive layers—differential privacy, Byzantine-resilient aggregation, and adaptive reputation management—to counter a range of adversarial threats while preserving model performance. Finally, the system design achieves practical scalability, keeping communication and computation costs low even with large numbers of participating edge devices.

Although the framework delivers strong results, several extensions remain open for exploration. The use of emerging communication technologies such as 5G and 6G could improve latency and data throughput. Stronger cryptographic protections, including homomorphic encryption or secure multi-party computation, could enhance privacy, though with higher computational demands. Optimizing consensus through approaches such as sharding or zk-rollups could further increase scalability in dense vehicular networks. Finally, real-world deployment in operational IoV testbeds will be essential for assessing practical feasibility and identifying implementation challenges. These directions align with the limitations and paths outlined in Section 9.

The findings of this thesis collectively answer RQ1–RQ6: integration and trust (RQ1), performance–security trade-offs (RQ2), deployment under mobility (RQ3), privacy without heavy utility loss (RQ4), incentive-driven robustness (RQ5), and end-to-end efficiency at scale (RQ6).

In summary, the combination of blockchain and FL presents a secure, scalable, and privacy-preserving collaborative intelligence at the edge. The findings of this thesis provide a technical foundation for advancing decentralized learning in IoV and also in domains such as smart cities, healthcare, and industrial IoT. By addressing critical technical and operational challenges, the framework offers a practical blueprint for developing trustworthy decentralized and distributed learning infrastructures.

References

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

[2] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Network*, vol. 35, no. 2, pp. 152–159, 2021.

[3] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. PMLR, vol. 108, 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[4] A. Qammar, A. Karim, H. Ning, and J. Ding, "Securing federated learning with blockchain: A systematic literature review," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 3951–3985, 2023.

[5] C. Ma, J. Li, M. Ding, L. Shi, T. Wang, Z. Han, and H. V. Poor, "When federated learning meets blockchain: A new distributed learning paradigm," *arXiv preprint arXiv:2009.09338*, 2020. [Online]. Available: https://arxiv.org/abs/2009.09338

[6] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016. [Online]. Available: https://arxiv.org/abs/1608.05187

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.

[8] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *USENIX NSDI*, 2016. [Online]. Available: https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf

[9] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/9617-deep-leakage-from-gradients.pdf

[10] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients – how easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [Online]. Available: https://papers.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191.

[12] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Transactions on Emerging Topics in Computing*, 2021.

[13] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018. [Online]. Available: https://arxiv.org/abs/1808.04866

[14] K. Qi, T. Liu, and C. Yang, "Federated learning based proactive handover in millimeter-wave vehicular networks," *arXiv preprint arXiv:2101.07032*, 2021. [Online]. Available: https://arxiv.org/abs/2101.07032

[15] W. Wang, Y. Zhao, Q. Wu, Q. Fan, C. Zhang, and Z. Li, "Asynchronous federated learning based mobility-aware caching in vehicular edge computing," *arXiv preprint arXiv:2208.01236*, 2022. [Online]. Available: https://arxiv.org/abs/2208.01236

[16] A. M. Elbir, B. Soner, S. Coleri, D. Gunduz, and M. Bennis, "Federated learning in vehicular networks," *arXiv preprint arXiv:2006.01412*, 2020. [Online]. Available: https://arxiv.org/abs/2006.01412

[17] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. PMLR, 2017, pp. 1273–1282.

[19] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *ICML*, 2019, pp. 634–643.

[20] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318.

[21] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 188–19 209, 2024.

[22] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 634–643.

[23] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*, 1999, pp. 173–186.

[24] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *Self-published whitepaper*, 2012, https://peercoin.net/assets/paper/peercoin-paper.pdf.

[25] V. Buterin, "A next-generation smart contract and decentralized application platform," https://ethereum.org/en/whitepaper/, 2014, ethereum White Paper.

[26] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," *EuroSys*, pp. 1–15, 2018.

[27] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[28] E. Borcoci, S. Obreja, and M. Vochin, "Internet of vehicles functional architectures – comparative critical study," in *Proceedings of the Int. Conf. on Advances in Future Internet (AFIN)*, 2017.

[29] S. S. Musa, M. Zennaro, M. Libsie, and E. Pietrosemoli, "Convergence of information-centric networks and edge intelligence for iov: Challenges and future directions," *Future Internet*, vol. 14, no. 7, p. 192, 2022.

[30] R. Fukatsu and K. Sakaguchi, "Automated driving with cooperative perception using millimeter-wave v2v communications for safe overtaking," *Sensors*, vol. 21, no. 8, p. 2659, 2021.

[31] J. Xie, G. Wu, X. Zhou, and S. Deng, "Future perspectives on internet of vehicles resource management: Digital twin-enabled edge computing frameworks," *Journal of Engineering and Applied Science*, vol. 72, p. 119, 2025.

[32] C. Wu, H. Fan, K. Wang, and P. Zhang, "Enhancing federated learning in heterogeneous internet of vehicles: A collaborative training approach," *Electronics*, vol. 13, no. 20, p. 3999, 2024.

[33] M. A. Labiod, M. Gharbi, F.-X. Coudoux, P. Corlay, and N. Doghmane, "Cross-layer scheme for low latency multiple description video streaming over vanets," *arXiv preprint arXiv:2311.13603*, 2023.

[34] E. Alalwany and I. Mahgoub, "Security and trust management in the internet of vehicles (iov): Challenges and machine learning solutions," *Sensors*, vol. 24, no. 2, p. 368, 2024.

[35] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," arXiv preprint arXiv:2110.02182, Tech. Rep., 2021.

[36] J. Kang, S. Lee, J. Lim, and S.-L. Kim, "Reliable blockchain-enabled federated learning for mec networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3546–3559, 2020.

[37] Y. Liu, K. Sun, and R. Rajkumar, "Blockchain-enabled federated learning for privacy-preserving intrusion detection in iov," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1234–1245, 2024.

[38] M. Ma, X. Su, and J. Zheng, "Privacy-preserving federated learning with blockchain in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8358–8367, 2021.

[39] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin *et al.*, "An overview of smart contract: Architecture, applications, and future trends," in *2018 IEEE Intelligent Vehicles Symposium*, 2018.

[40] S. Chen, Y. Xu, H. Xu, Z. Jiang *et al.*, "Decentralized federated learning with intermediate results in mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 123–22 134, 2022.

[41] N. Xie, C. Zhang, Q. Yuan, J. Kong, and X. Di, "Iov-bcfl: An intrusion detection method for iov based on blockchain and federated learning," *Ad Hoc Networks*, vol. 146, p. 103996, 2024.

[42] B. Kitchenham *et al.*, "Guidelines for performing systematic literature reviews in software engineering," *EBSE Technical Report*, vol. 2, no. 1, pp. 1–57, 2007.

[43] J. Su, T. Wang, and L. Chen, "Token-based incentive mechanisms for federated learning in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, 2025, to appear.

[44] X. Li and W. Wu, "A blockchain-empowered multiaggregator federated learning framework for edge computing," *IEEE Transactions on Computational Social Systems*, vol. 12, no. 2, pp. 645–658, 2025.

[45] R. Rahmani and B. Stojcevska, "Enhancing electric vehicle charging systems with blockchain-enabled federated learning," in *Procedia Computer Science*, vol. 257, 2025, pp. 1073–1082.

[46] X. Chen, W. Meng, and H. Huang, "Blockchain-driven distributed edge intelligence for smart cities: A federated learning approach," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 4773–4784, 2025.

[47] R. Ruan, X. Lu, and J. Zhang, "Secure federated learning for connected vehicles with blockchain-based key management," *Vehicular Communications*, vol. 45, p. 100612, 2025.

[48] Y. Yang, X. Zhao, and M. Liu, "Smart contract-governed personalized federated learning in vehicular edge networks," *Computer Communications*, vol. 210, pp. 89–98, 2023.

[49] H. Tang and L. Wei, "Cluster-based blockchain dag scheduling for 5g vehicular federated learning," *IEEE Transactions on Mobile Computing*, 2025, to appear.

[50] Z. Zhou and F. Wang, "Horizontal federated learning with blockchain-based permissioned consensus in iov," *IEEE Access*, vol. 11, pp. 10 342–10 351, 2023.

[51] L. Zhang, W. Li, and Q. Li, "Blockchain-based federated reinforcement learning for urban vehicle platooning," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1822–1833, 2023.

[52] S. W. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, no. 3, pp. 239–250, 1959.

[53] D. C. Montgomery, *Introduction to Statistical Quality Control*, 7th ed. Hoboken, NJ: Wiley, 2019.

[54] Y. Zhan, P. Li, S. Guo, and Z. Qu, "Incentive mechanism design for federated learning: Challenges and opportunities," *IEEE Network*, 2021.

[55] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 119–129.

[56] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 5650–5659.

[57] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

[58] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *ACM Multimedia*, 2017, pp. 269–277.

[59] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[60] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010.

[61] National Institute of Standards and Technology (NIST), "Digital signature standard (dss)," U.S. Department of Commerce, FIPS Publication 186-4, July 2013. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.186-4

[62] M. J. Dworkin, "Recommendation for block cipher modes of operation: Galois/-counter mode (gcm) and gmac," NIST Special Publication 800-38D, Tech. Rep., 2007.

[63] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[64] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm (raft)," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, 2014, pp. 305–319.

[65] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 2014, pp. 10–14.

[66] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, 2010, pp. 21–21.

[67] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[68] H. Wang, R. Li, Z. Xu, J. L. Li, I. King *et al.*, "Federated learning for vehicle trajectory prediction: Methodology and benchmark study," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024.

[69] Y. Liu, T. Ouyang, W. Wang, P. Fieguth, X. Chen *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 12, pp. 2797–2811, 2020.

[70] T. Li, A. K. Sahu, V. Smith, A. Talwalkar *et al.*, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[71] M. M. Billah, M. A. Hossain, M. Rahman *et al.*, "A survey on blockchain-enabled federated learning: Concepts, challenges, and future directions," *IEEE Access*, vol. 10, pp. 15 132–15 150, 2022.

[72] P. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd *et al.*, "Microscopic traffic simulation using sumo," *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, 2018.

[73] S. Bubeck, *Convex Optimization: Algorithms and Complexity*. Now Publishers, 2015.

[74] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*. Now Publishers, 2014, vol. 9, no. 3–4.

[75] Z. C. Lipton and J. Steinhardt, "Troubling trends in machine learning scholarship," *Communications of the ACM*, vol. 62, no. 6, pp. 45–53, 2019.

[76] C. Boscher, N. Benarba, F. Elhattab *et al.*, "Personalized privacy-preserving federated learning," in *Proceedings of the 25th International Conference on Mobile Computing and Networking*, 2024.

[77] C. Huang, R. Song, S. Gao, Y. Guo, and B. Xiao, "Data availability and decentralization: New techniques for zk-rollups in layer 2 blockchain networks," arXiv preprint arXiv:2403.10828, Tech. Rep., 2024.

[78] E. Madill, B. Nguyen, C. K. Leung, and S. Rouhani, "Scalesfl: A sharding solution for blockchain-based federated learning," in *Proceedings of the Fourth ACM International Conference on AI in Finance*, 2022.

Appendix

I. Notation and Formula Explanations

This appendix collects the symbols and formulas used throughout the thesis. For each entry, we briefly explain what it means and why it matters in our design and evaluation.

Symbol / Formula	Meaning	Why it matters
$\mathbf{w}_{\text{global}}$	Global model after aggregation.	Final model each round; basis for the next iteration.
$\mathbf{w}_i$	Model of client i.	Captures local learning at client i.
$n_i,\ n$	Local sample size of client i; total samples.	Control FedAvg weights and normalization.
FedAvg $\mathbf{w}_{\text{global}} = \sum_{i=1}^{N} \frac{n_i}{n} \mathbf{w}_i$	Weighted average of client models.	Baseline aggregation to compare with robust/weighted variants.
$r_i,\ \alpha,\ Q(\mathbf{w}_i)$	Reputation of client i; reputation step size; quality of update.	Weight/filter updates by trust; adapt to contribution quality.
Reputation (additive) $r_i^{\text{new}} = r_i^{\text{old}} + \alpha\, Q(\mathbf{w}_i)$	Additive reputation update.	Rewards helpful, penalizes harmful updates.
EWMA reputation (Eq. (13)) $r_i^{\text{new}} = \beta r_i^{\text{old}} + (1-\beta)\frac{\|\Delta\mathbf{w}_i - \bar{\Delta\mathbf{w}}\|}{\sigma_{\Delta\mathbf{w}}}$	Smoothed reputation with deviation score.	Stable yet responsive trust estimate; resists noise/outliers.
Composite reputation (Eq. (22)) $r_i = \gamma_1 r_i^{\text{val}} + \gamma_2 r_i^{\text{cons}} + \gamma_3 r_i^{\text{act}}$	Mix of validation, consistency, activity.	Encourages quality, stability, and active participation.
Activity (Eq. (23)) $r_i^{\text{act}} = \sum_{k=1}^{K} e^{-\lambda(T-t_k)}$	Time-decayed activity score.	Rewards recent, steady contributions under mobility.
Incentive $\text{Reward}_i = \gamma \cdot r_i \cdot \lvert D_i \rvert$	Token/credit reward; γ scales; $\lvert D_i \rvert$ is data size.	Links rewards to trust and contribution volume.
Local SGD (Eq. (10)) $\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \eta\, \nabla \mathcal{L}(\mathbf{w}_i^t; \mathcal{D}_i)$	One local update step.	Core on-device learning.

Symbol / Formula	Meaning	Why it matters
DP mechanism (Eq. (18)) $\Delta \mathbf{w}_i' = \Delta \mathbf{w}_i + \mathcal{N}(0, \sigma^2 \mathbf{I})$	Noise added to updates.	Limits information leakage from model deltas.
Moments accountant (Eq. (19)) $\alpha(\lambda) \leq \sum_{t=1}^{T} \frac{\lambda(\lambda+1)\Delta_2^2}{2\sigma_t^2}$	Privacy accounting across rounds.	Tracks cumulative (ϵ, δ) under composition.
DP budget (Eq. (30)) $\epsilon = \frac{\sqrt{2\log(1.25/\delta)}}{\sigma}$	Privacy–utility relation.	Choose σ to meet DP targets.
Signature check (Eq. (11)) $\text{Verify}(pk_i, \sigma_i, \Delta \mathbf{w}_i) = \text{True}$	ECDSA verification.	Blocks forged/tampered updates.
Dual-layer enc. (Eq. (16)) $c_i = \text{AES-GCM}(K_s, \Delta \mathbf{w}_i)$	Symmetric enc. with ECDH session key K_s.	Confidentiality and integrity in transit.
Threshold dec. (Eq. (17)) $\Delta \mathbf{w}_{\text{agg}} = \sum_{i=1}^{n} \text{Decrypt}(c_i, \{sk_{\text{RSU}_j}\}_{j=1}^{k})$	k-of-m RSU decryption.	Avoid single-point decryption; distribute trust.
Rep.-weighted (Eq. (12)) $\mathbf{w}_{\text{global}} = \frac{\sum_{i=1}^{N} r_i \, n_i \, \Delta \mathbf{w}_i}{\sum_{i=1}^{N} r_i \, n_i}$	Weighted aggregation.	Trust- and size-aware combining.
Trimmed mean (Eq. (14)) $\Delta w_j = \frac{1}{N-2f} \sum_{i=f+1}^{N-f} \Delta w_j^{(i)}$	Remove extremes per coordinate.	Robust if $\leq f$ malicious clients.
Krum (Eq. (15)) $\Delta \mathbf{w} = \arg\min_{\Delta \mathbf{w}_i} \sum_{j \in \mathcal{N}_i} \|\Delta \mathbf{w}_i - \Delta \mathbf{w}_j\|^2$	Choose most central update.	Works when attack rate is unknown.

Symbol / Formula	Meaning	Why it matters
Geometric median (Eq. (21)) $\Delta \mathbf{w}_{\text{global}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \|\mathbf{x} - \Delta \mathbf{w}_i\|$	ℓ_2 median.	High breakdown point; strong under collusion.
Z-score (Eq. (20)) $z_i = \frac{\|\Delta \mathbf{w}_i - \mu_{\Delta \mathbf{w}}\|}{\sigma_{\Delta \mathbf{w}}}$	Distance from mean.	Flags anomalous updates for filtering.
Watermark (Eq. (7)) $\mathbf{w}_{\text{final}} = \mathbf{w}_{\text{global}} + \mathbf{m} \odot \mathbf{s}$	Masked signature in model.	Verifies provenance; deters tampering.
Anomaly alert (Eq. (8)) $a_t = \mathbb{I}(\|\mathbf{y}_t - \hat{\mathbf{y}}_t\| > \theta)$	Runtime deviation check.	Triggers safe fallback on anomalies.
Convergence (Eq. (27)) $T_{conv} = \frac{\log(\epsilon) - \log(\Delta_0)}{\log(1 - \eta\lambda)}$	Time to reach target error.	Compare efficiency across methods.
Detection prob. (Eq. (28)) $P_{detect} = 1 - (1 - \alpha)^k$	Detect attacker over k rounds.	Quantifies reputation/filters' speed.
Comm. cost (Eq. (29)) $O_{\text{comm}} = N \cdot \big(\|\mathbf{w}\|_{\text{comp}} + \|\sigma\| + \|h\| \big)$	Per-round bandwidth.	Shows scalability in large IoV.
Round latency (Eq. (24)) $T_{\text{round}} = \max_i(t_{\text{train},i} + t_{\text{crypto},i} + t_{\text{uplink},i}) + t_{\text{chain}} + t_{\text{agg}} + t_{\text{downlink}}$	End-to-end time per FL round.	Used to estimate latency under mobility and ordering.
Ordering latency (Eq. (25)) $t_{\text{chain}} \approx \max\big(t_{\text{block\_timeout}}, \frac{\text{tx\_in\_block}}{\text{tx\_rate}}\big) + \kappa \cdot \text{RTT}_{\text{ordering}}$	Time due to ordering/consensus.	Captures Raft batching/RTT in latency model.
Energy model (Eq. (26)) $E = \text{FLOPs} \cdot e_{\text{flop}} + 8(\text{TxBytes} \cdot e_{\text{tx}} + \text{RxBytes} \cdot e_{\text{rx}})$	Per-round energy (compute + communication).	Basis for energy estimates and trade-offs.
$\eta,\ \lambda$	Learning rate; loss curvature.	Affect convergence and stability.
$\sigma,\ \delta,\ \epsilon$	DP noise scale; failure prob.; privacy budget.	Tune privacy–utility trade-off.

Symbol / Formula	Meaning	Why it matters
$\bar{\Delta \mathbf{w}},\ \mu_{\Delta \mathbf{w}},\ \sigma_{\Delta \mathbf{w}}$	Mean / std of updates.	Used in robust stats and reputation.
$pk_i,\ \sigma_i,\ sk_i,\ K_s$	Public key, signature, private key; session key.	AuthN/AuthZ and secure transport.
$f,\ \mathcal{N}_i$	Max. malicious count; Krum neighbors.	Parameters for robust aggregation.
$\mathrm{TOPS}_{\mathrm{eff},i}$	Effective compute throughput of client i (tera-operations/s).	Used in $t_{\text{train},i} = \frac{\mathrm{FLOPs}_i}{\mathrm{TOPS}_{\mathrm{eff},i}}(1+\rho_{\text{overhead}})$.
R, $t_{\text{block\_timeout}}$, tx_in_block, tx_rate, $\mathrm{RTT}_{\text{ordering}}$, κ	Link rate; block timeout; transactions per block; tx processing rate; ordering RTT; Raft factor.	Inputs to t_{chain} and T_{round}.
e_{flop}, e_{tx}, e_{rx}, $\mathrm{TxBytes}$, $\mathrm{RxBytes}$	Energy coefficients and bytes sent/received.	Inputs to the energy model E.

II. SLR Reproducibility Materials

Database-specific search configurations

Table 9. Database-specific search configurations (last updated: 5 Aug 2025)

Database	Scope/fields	Boolean query (as entered)	Filters
IEEE Xplore	All metadata	`"federated learning" AND blockchain AND (vehicular OR IoV OR "edge computing")`	Year: 2020–2025; Journals & Conferences; English
ACM Digital Library	Anywhere	`"federated learning" AND blockchain AND (vehicular OR IoV OR "edge computing")`	2020–2025; Proceedings & Journals; English
SpringerLink	Title/Abstract/Keywords	`"federated learning" AND blockchain AND (vehicular OR IoV OR "edge computing")`	Content: Conference & Journal; 2020–2025; English
ScienceDirect	Title/Abstract/Keywords	`TITLE-ABS-KEY("federated learning" AND blockchain AND (vehicular OR IoV OR "edge computing"))`	Year: 2020–2025; Research/Conference; English
Scopus	Title/Abstract/Keywords	`TITLE-ABS-KEY("federated learning" AND blockchain AND (vehicular OR IoV OR "edge computing"))`	2020–2025; Article, Conference Paper; English

Note. Where a database offers fielded search rather than a single query box, the same Boolean string was entered in the "all fields/keywords" input and filters were applied as shown.

Quality appraisal checklist

Table 10. Five-criterion appraisal rubric (scores 1–5; studies with average < 3 excluded)

Criterion	Guiding questions for scoring (1 = weak, 5 = strong)
Problem & context clarity	Is the IoV/edge context and motivation explicit? Are assumptions stated?
Methodological transparency	Is the architecture/mechanism described sufficiently for replication?
FL–blockchain integration	Is the integration substantive (not a loose coupling/claim)?
Evaluation design	Are datasets/scenarios, baselines, metrics, and threat models appropriate?
Limitations discussion	Are bottlenecks/assumptions discussed with evidence or ablations?

Data extraction form (per study)

Table 11. Extraction form used for coding (one row per study)

Field	**Value**
Citation	Author(s), year, venue
Domain	Vehicular sub-context (e.g., platooning, smart charging, safety)
FL style	FedAvg / clustered / personalized / RL; sync/async; client selection
Blockchain design	Permission model; consensus (PBFT/PoS/DAG/Raft); smart-contract roles; on/off-chain split
Aggregation/verification	Robust estimator, secure aggregation, reputation weighting
Incentive/reputation	Token/score, Sybil resistance, penalties
Evaluation	Simulation/prototype; datasets/scenarios; metrics; adversarial tests
Limitations	Reported bottlenecks, assumptions, threats
Notes	Relevance to our framework / open issues

III. GitHub Repository and Video Demo

GitHub Repository

https://github.com/thomasabrina/blockchain-fl-iov.git

Video Demo

https://youtu.be/DkGExhVAPY4

IV. Code Listings

10.1 Robust Aggregation (Python)

```python
class RobustAggregator:
    """Robust aggregation engine with multiple algorithms"""

    def __init__(self, config: AggregationConfig):
        self.config = config
        self.aggregation_history = []

    def aggregate(self, client_updates: List[ClientUpdate]) ->
       Tuple[List[np.ndarray], Dict[str, Any]]:
        """Main aggregation function"""
        if len(client_updates) < self.config.min_clients:
            raise ValueError(f"Insufficient clients: {len(
               client_updates)} < {self.config.min_clients}")

        # Filter validated updates
        valid_updates = [update for update in client_updates if
            update.is_validated]

        if len(valid_updates) < self.config.min_clients:
            raise ValueError(f"Insufficient valid updates: {len
               (valid_updates)} < {self.config.min_clients}")

        logger.info(f"Aggregating {len(valid_updates)} valid
           updates using {self.config.algorithm.value}")

        # Select aggregation algorithm (FedAvg / Krum / Trimmed
            Mean)
        if self.config.algorithm == AggregationAlgorithm.FEDAVG
           :
            aggregated_weights, metrics = self.
               federated_averaging(valid_updates)
        elif self.config.algorithm == AggregationAlgorithm.KRUM
           :
            aggregated_weights, metrics = self.krum(
               valid_updates)
        elif self.config.algorithm == AggregationAlgorithm.
           TRIMMED_MEAN:
            aggregated_weights, metrics = self.trimmed_mean(
               valid_updates)
```

```
        else:
            aggregated_weights, metrics = self.
                federated_averaging(valid_updates)

        # Store aggregation history
        self.aggregation_history.append({
            'algorithm': self.config.algorithm.value,
            'num_clients': len(valid_updates),
            'metrics': metrics,
            'timestamp': max(update.timestamp for update in
                valid_updates)
        })

        return aggregated_weights, metrics

    def federated_averaging(self, updates: List[ClientUpdate])
        -> Tuple[List[np.ndarray], Dict[str, Any]]:
        """Weighted FedAvg with reputation scores"""
        total_data_size = sum(update.data_size for update in
            updates)
        weights = []
        for update in updates:
            data_weight = update.data_size / total_data_size
            if self.config.use_reputation:
                reputation_weight = update.reputation_score *
                    self.config.reputation_weight
                final_weight = data_weight * (1 - self.config.
                    reputation_weight) + reputation_weight
            else:
                final_weight = data_weight
            weights.append(final_weight)
        total_weight = sum(weights)
        weights = [w / total_weight for w in weights]
        num_layers = len(updates[0].weights)
        aggregated_weights = []
        for layer_idx in range(num_layers):
            layer_weights = [update.weights[layer_idx] for
                update in updates]
            aggregated_layer = np.zeros_like(layer_weights[0])
            for weight, layer_weight in zip(weights,
                layer_weights):
                aggregated_layer += weight * layer_weight
            aggregated_weights.append(aggregated_layer)
```

```
63          avg_accuracy = np.average([update.accuracy for update
               in updates], weights=weights)
64          avg_loss = np.average([update.loss for update in
               updates], weights=weights)
65          avg_reputation = np.average([update.reputation_score
               for update in updates], weights=weights)
66          metrics = {
67              'algorithm': 'fedavg',
68              'num_participants': len(updates),
69              'avg_accuracy': float(avg_accuracy),
70              'avg_loss': float(avg_loss),
71              'avg_reputation': float(avg_reputation),
72              'weight_distribution': weights
73          }
74          return aggregated_weights, metrics
75
76      def krum(self, updates: List[ClientUpdate], multi_krum:
           bool = False, m: int = 1) -> Tuple[List[np.ndarray],
           Dict[str, Any]]:
77          """Krum and Multi-Krum algorithms"""
78          n = len(updates)
79          f = self.config.krum_f
80          if n <= 2 * f:
81              logger.warning(f"Insufficient clients for Krum: {n}
                   <= 2*{f}")
82              return self.federated_averaging(updates)
83          distances = np.zeros((n, n))
84          for i in range(n):
85              for j in range(i + 1, n):
86                  dist = self.euclidean_distance(updates[i].
                       weights, updates[j].weights)
87                  distances[i, j] = distances[j, i] = dist
88          scores = []
89          for i in range(n):
90              sorted_distances = np.sort(distances[i])
91              score = np.sum(sorted_distances[1:n-f-1])
92              scores.append(score)
93          if multi_krum:
94              selected_indices = np.argsort(scores)[:m]
95              selected_updates = [updates[i] for i in
                   selected_indices]
96              aggregated_weights, metrics = self.
                   federated_averaging(selected_updates)
```

```
97              metrics['algorithm'] = 'multi_krum'
98              metrics['selected_clients'] = m
99          else:
100             best_idx = np.argmin(scores)
101             aggregated_weights = updates[best_idx].weights
102             metrics = {
103                 'algorithm': 'krum',
104                 'selected_client': updates[best_idx].client_id,
105                 'krum_score': float(scores[best_idx]),
106                 'accuracy': updates[best_idx].accuracy,
107                 'loss': updates[best_idx].loss
108             }
109         metrics['krum_scores'] = [float(s) for s in scores]
110         return aggregated_weights, metrics
111
112     def trimmed_mean(self, updates: List[ClientUpdate]) ->
        Tuple[List[np.ndarray], Dict[str, Any]]:
113         """Trimmed mean aggregation"""
114         n = len(updates)
115         trim_count = int(n * self.config.trim_ratio)
116         if n - 2 * trim_count < 1:
117             logger.warning("Too much trimming, falling back to
                FedAvg")
118             return self.federated_averaging(updates)
119         num_layers = len(updates[0].weights)
120         aggregated_weights = []
121         for layer_idx in range(num_layers):
122             layer_weights = np.array([update.weights[layer_idx]
                    for update in updates])
123             sorted_weights = np.sort(layer_weights, axis=0)
124             trimmed_weights = sorted_weights[trim_count:n-
                trim_count]
125             aggregated_layer = np.mean(trimmed_weights, axis=0)
126             aggregated_weights.append(aggregated_layer)
127         sorted_by_accuracy = sorted(updates, key=lambda x: x.
            accuracy)
128         kept_updates = sorted_by_accuracy[trim_count:n-
            trim_count]
129         avg_accuracy = np.mean([update.accuracy for update in
            kept_updates])
130         avg_loss = np.mean([update.loss for update in
            kept_updates])
131         metrics = {
```

```
            'algorithm': 'trimmed_mean',
            'trim_ratio': self.config.trim_ratio,
            'trimmed_count': trim_count,
            'kept_count': len(kept_updates),
            'avg_accuracy': float(avg_accuracy),
            'avg_loss': float(avg_loss)
        }
        return aggregated_weights, metrics

    def euclidean_distance(self, weights1: List[np.ndarray],
        weights2: List[np.ndarray]) -> float:
        """Calculate Euclidean distance between two weight
            vectors"""
        total_distance = 0.0
        for w1, w2 in zip(weights1, weights2):
            diff = w1 - w2
            total_distance += np.sum(diff ** 2)
        return math.sqrt(total_distance)
```

Listing 1. Robust aggregation (real code; core rules)

IV. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Lanxiang Zhang**,
author's name

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   AI-Driven Blockchain-based Federated Learning for Edge Devices,
   title of thesis

   supervised by Mubashar Iqbal.
   supervisor's name

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Lanxiang Zhang
12/08/2025