future internet

MDPI

Article

Resilient Federated Learning for Vehicular Networks: A Digital Twin and Blockchain-Empowered Approach

Jian Li, Chuntao Zheng and Ziyao Chen *

School of Advanced Manufacturing, Guangdong University of Technology, Guangzhou 510000, China; lijian@gdut.edu.cn (J.L.); chuntao.zheng@gdut.edu.cn (C.Z.)
* Correspondence: chenzy@gdut.edu.cn

Abstract

Federated learning (FL) is a foundational technology for enabling collaborative intelligence in vehicular edge computing (VEC). However, the volatile network topology caused by high vehicle mobility and the profound security risks of model poisoning attacks severely undermine its practical deployment. This paper introduces DTB-FL, a novel framework that synergistically integrates digital twin (DT) and blockchain technologies to establish a secure and efficient learning paradigm. DTB-FL leverages a digital twin to create a real-time virtual replica of the network, enabling a predictive, mobility-aware participant selection strategy that preemptively mitigates network instability. Concurrently, a private blockchain underpins a decentralized trust infrastructure, employing a dynamic reputation system to secure model aggregation and smart contracts to automate fair incentives. Crucially, these components are synergistic: The DT provides a stable cohort of participants, enhancing the accuracy of the blockchain's reputation assessment, while the blockchain feeds reputation scores back to the DT to refine future selections. Extensive simulations demonstrate that DTB-FL accelerates model convergence by 43% compared to FedAvg and maintains 75% accuracy under poisoning attacks even when 40% of participants are malicious—a scenario where baseline FL methods degrade to below 40% accuracy. The framework also exhibits high resilience to network dynamics, sustaining performance at vehicle speeds up to 120 km/h. DTB-FL provides a comprehensive, cross-layer solution that transforms vehicular FL from a vulnerable theoretical model into a practical, robust, and scalable platform for next-generation intelligent transportation systems.

Keywords: vehicular edge computing (VEC); federated learning; digital twin; blockchain; security; incentive mechanism; resource optimization

Academic Editors: Oleksandr Kuznetsov, Cristian Randieri and Paolo Bellavista

Received: 25 September 2025
Revised: 31 October 2025
Accepted: 1 November 2025
Published: 3 November 2025

Citation: Li, J.; Zheng, C.; Chen, Z. Resilient Federated Learning for Vehicular Networks: A Digital Twin and Blockchain-Empowered Approach. *Future Internet* **2023**, *17*, 505. https://doi.org/10.3390/fi17110505

1. Introduction

Internet of Vehicles (IoV) and autonomous driving technologies have fundamentally transformed vehicular networks, giving rise to vehicular edge computing (VEC) [1]. VEC deploys computing and storage resources at the network edge—including roadside units (RSUs) and base stations—enabling vehicles to offload computation-intensive tasks locally. This paradigm shift supports latency-sensitive applications, such as real-time traffic prediction, autonomous navigation, and collaborative perception [2,3]. These applications rely on sophisticated AI models trained on vehicle-generated data that capture road conditions, traffic patterns, and driving behaviors.

Conventional AI model training aggregates data from all vehicles at a central cloud server [4]. However, this centralized approach faces critical limitations in VEC envi-

ronments. First, transmitting massive volumes of raw vehicular data incurs prohibitive communication overhead and latency, undermining the requirements of real-time applications. Second, centralized data collection raises severe privacy concerns, as vehicular data often contain sensitive information, including location trajectories and personal driving patterns [5].

Federated learning (FL) offers a compelling alternative by enabling collaborative model training without sharing raw data [6]. In FL, vehicles train models locally and exchange only model parameters (e.g., gradients or weights) with an edge server, which aggregates these updates to refine a global model [7]. This decentralized approach reduces communication overhead while preserving data privacy.

However, deploying FL in dynamic and untrusted VEC environments presents three fundamental challenges [8]:

Challenge 1: Network Dynamics and Heterogeneity: High-speed vehicle mobility causes intermittent connectivity and rapid topology changes. Moreover, vehicles exhibit substantial heterogeneity in computational resources, data distributions (non-independent and identically distributed or non-IID), and availability. Random participant selection can severely degrade model convergence and training stability [9].

Challenge 2: Security and Trust: The open nature of VEC environments exposes FL to various attacks. Malicious vehicles may inject poisoned model updates to compromise global model performance or launch Sybil attacks to gain disproportionate influence [10]. Without robust security mechanisms, the integrity of collaboratively trained models remains vulnerable.

Challenge 3: Incentive Mechanisms: FL participation consumes valuable vehicle resources, including computation, energy, and bandwidth. Rational vehicle owners require adequate compensation to contribute their resources; otherwise, insufficient participation and free-riding behaviors emerge, undermining the collaborative ecosystem [11].

To address these interrelated challenges, we propose DTB-FL, a novel framework that synergistically integrates digital twin (DT) and blockchain (BC) technologies with federated learning in VEC. Our framework leverages DT to create high-fidelity virtual replicas of the physical VEC network, capturing real-time vehicle dynamics, network conditions, and resource availability. These digital twins enable predictive analytics for proactive participant selection—the edge server selects vehicles predicted to maintain stable connectivity and sufficient resources throughout training rounds, thereby enhancing efficiency and convergence.

Simultaneously, we employ blockchain technology to establish a decentralized, transparent, and tamper-proof infrastructure for security and incentive management. Our reputation-based smart contract evaluates model update quality and assigns dynamic reputation scores to participants. These scores weight contributions during model aggregation, mitigating the impact of malicious updates. Additionally, the smart contract automates fair and transparent reward distribution, incentivizing honest and reliable participation.

The main contributions of this paper are as follows:

1. **Novel Three-Layer Architecture**: We design DTB-FL, the first framework to synergistically integrate digital twin, blockchain, and federated learning for addressing network dynamics, security threats, and incentive challenges in VEC environments.
2. **DT-Driven Participant Selection**: We develop a proactive selection algorithm that leverages digital twin predictive analytics to optimize vehicle selection, significantly improving FL convergence speed and stability under high mobility.
3. **Blockchain-Based Security and Incentives**: We implement a reputation management system and automated reward mechanism through smart contracts, ensuring model integrity while motivating sustained vehicle participation.

4. **Comprehensive Evaluation**: We demonstrate through extensive simulations that DTB-FL achieves superior model accuracy, training efficiency, and attack resilience compared to state-of-the-art baselines.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the system model and problem formulation; Section 4 details the DTB-FL framework design; Section 5 analyzes convergence and complexity; Section 6 evaluates performance through simulations; Section 7 concludes this paper.

DTB-FL is particularly suited for safety-critical vehicular applications such as cooperative perception, traffic prediction, and road hazard detection, where model trustworthiness is paramount. The increasing availability of powerful onboard computing resources and 5G/C-V2X communication infrastructure in modern connected vehicles [12] makes the deployment of DTB-FL increasingly practical.

2. Related Work

This section reviews three pillars relevant to DTB-FL: federated learning (FL) in vehicular edge computing (VEC), blockchain-enhanced security and accountability for FL, and digital twins (DTs) for prediction and resource management. We summarize core challenges—communication/latency constraints, mobility-driven client churn and non-IID data, and trust and incentives under adversarial participation—and position how DTB-FL integrates these elements, as shown in Table 1.

Table 1. Comparison of DTB-FL with representative related works.

Work	VEC	Blockchain	Digital Twin	Predictive Selection	Trust Mgmt	Incentive
Lu et al. [13]	✓	✗	✗	✗	✗	✗
Kang et al. [14]	✓	✓	✗	✗	Partial	✓
Ye et al. [11]	✓	✗	✗	Partial	✗	✗
Shayan et al. [15]	✗	✓	✗	✗	✓	✗
Wu et al. [16]	✓	✗	✗	Partial	✗	✗
HaghighiFard & Coleri [17]	✓	✗	✗	Partial	✗	✗
Madill et al. [18]	✗	✓	✗	✗	Partial	✗
Pan et al. [19]	✗	✗	✗	✗	✓	✗
Wang et al. [20]	✗	✗	✓	✗	✗	✗
Li et al. [21]	✓	✓	✓	Partial	Partial	✗
Yan et al. [22]	✓	✗	✗	✗	✗	✗
DTB-FL (Ours)	✓	✓	✓	✓	✓	✓

Note: Partial indicates limited or reactive implementation.

2.1. Federated Learning in Vehicular Networks

Federated learning is a key paradigm for collaborative intelligence in VEC and ITS systems [3,8,9,23]. Three recurring challenges dominate this space.

First, communication efficiency is critical because frequent uplinks from many vehicles stress spectrum and backhaul. Hierarchical aggregation via RSUs reduces long-haul traffic [13], and model compression/quantization further shrink payloads [24,25]. These techniques, however, can trade accuracy for bandwidth and may be sensitive to mobility-induced stragglers.

Second, data heterogeneity is inherent due to regional driving patterns and sensor diversity. Variants of FedProx and control-variate methods such as SCAFFOLD help stabilize convergence under non-IID data [26,27], but performance can still degrade when non-IID data coincide with volatile connectivity and intermittent clients.

Third, mobility-aware and reliability-aware client selection is pivotal. Reputation-aware selection improves stability [28], while learning-based selection reduces training

time at a given accuracy [29]. Recent vehicular/VEC work continues this line with stability- and mobility-aware vehicle selection and multi-hop, cluster-based hierarchical FL that jointly address mobility and participation churn [16,17]. In parallel, energy constraints in EVs call for battery-aware participation; FL for EV energy modeling demonstrates gains while respecting privacy [22]. Overall, most schemes still react to present conditions rather than anticipating near-future link/session viability, which can lead to mid-round dropouts and wasted computation. DTB-FL targets a unified design that uses DTs to predict participation viability and network states and blockchain to close the loop with auditable trust and incentives.

2.2. Blockchain for Secure Federated Learning

Blockchain augments FL with auditability, accountability, and incentives [30]. Classic approaches store evidence of model updates and validations on an immutable ledger, supporting traceability but raising scalability concerns [15,31]. Smart-contract incentives reward contribution quality and deter free-riding [14]. Reputation and voting-based defenses mitigate poisoning risks, albeit with latency overheads that can be problematic for time-sensitive VEC [15,32].

Recent work focuses on making blockchain–FL more scalable and trust-aware: Sharding reduces on-chain bottlenecks [18]; surveys and system designs consolidate best practices for privacy, accountability, and fairness [33,34]; privacy-preserving computation with homomorphic encryption is being optimized for FL pipelines [19]. Still, many frameworks treat client selection as orthogonal to trust and prediction. DTB-FL explicitly links DT-based predictive selection with blockchain-based validation and incentives: predicted reliability informs who joins; on-chain evidence and reputation feed back into later rounds.

2.3. Digital Twin for Network Management

Network digital twins (NDTs) mirror system state and dynamics, enabling what-if forecasts and proactive control [35]. In vehicular/edge systems, DTs support traffic and resource prediction [36], adaptive VEC management [37], and V2X optimization with MEC [38]. DTs are also combined with blockchain for secure, cooperative vehicular edge offloading, illustrating that DT-driven prediction and blockchain-based accountability can coexist in practice [21]. For FL, DTs can augment training by offloading heavy teacher models and enabling simulation-informed decisions for heterogeneous clients [20]. Security perspectives on DT-enabled federated edge learning further motivate joint proactive and reactive defenses [39]. Compared to these efforts, DTB-FL uses DTs not merely for monitoring or offline simulation but to forecast short-horizon participant viability and link quality in real time, feeding a predictive selector; blockchain then validates behavior and sustains incentives, closing a proactive–reactive loop.

2.4. Research Gap and Our Contribution

To the best of our knowledge, no existing framework simultaneously addresses all three challenges: predictive participant selection under mobility, scalable blockchain-based trust management, and adversary-resilient incentive alignment.

DTB-FL fills this gap through a unified architecture where DTs enable proactive participant profiling and network forecasting, blockchain provides tamper-evident validation and programmable incentives, and FL leverages these integrated signals to achieve secure and efficient training under realistic VEC constraints.

3. System Model and Problem Formulation

This section presents the DTB-FL system architecture, including the network model, federated learning process, digital twin construction, blockchain integration, and threat model. We then formulate the optimization problem that captures the trade-offs among training efficiency, security, and incentive provision. Figure 1 illustrates the overall system architecture, while Table 2 summarizes the key notations used throughout this paper.

Table 2. Key notations used in DTB-FL framework.

Symbol	Description
Network and Vehicle Parameters	
$\mathcal{V}$	Set of all vehicles, $\{1, 2, \ldots, N\}$
N	Total number of vehicles
$\mathcal{D}_i$	Local dataset of vehicle i
$r_{i,t}$	Uplink data rate of vehicle i at time t
$h_{i,t}$	Channel gain of vehicle i at time t
$f_{i,t}$	CPU frequency of vehicle i at time t
Federated Learning Parameters	
$\mathbf{w}_t$	Global model parameters at round t
$\mathbf{w}_{t+1}^i$	Local update from vehicle i
$F(\mathbf{w})$	Global loss function
$F_i(\mathbf{w})$	Local loss function of vehicle i
$\mathcal{S}_t$	Set of selected participants in round t
$\alpha_{i,t}$	Aggregation weight of vehicle i at round t
K	Number of selected vehicles per round
θ	Target accuracy threshold
Digital Twin and Blockchain	
$\mathbf{s}_{i,t}$	State vector of vehicle i's digital twin
$\hat{\mathbf{s}}_{i,t+\Delta t}$	Predicted state after time interval Δt
$\rho_{i,t}$	Reputation score of vehicle i at time t
$R_{i,t}$	Reward for vehicle i in round t
$\mathcal{C}_{\text{rep}}$	Reputation management smart contract
$\mathcal{C}_{\text{inc}}$	Incentive distribution smart contract
Security Parameters	
$\mathcal{V}_m$	Set of malicious vehicles
$\tilde{\mathbf{w}}_{t+1}^j$	Poisoned update from malicious vehicle j
ϵ	Maximum allowed malicious influence
$\rho_{\min}$	Minimum reputation threshold for participation
Road Network Parameters	
$\mathcal{M}$	Digital road map structure
$\mathcal{R}$	Set of valid road coordinates
$\mathcal{I}$	Set of intersection coordinates

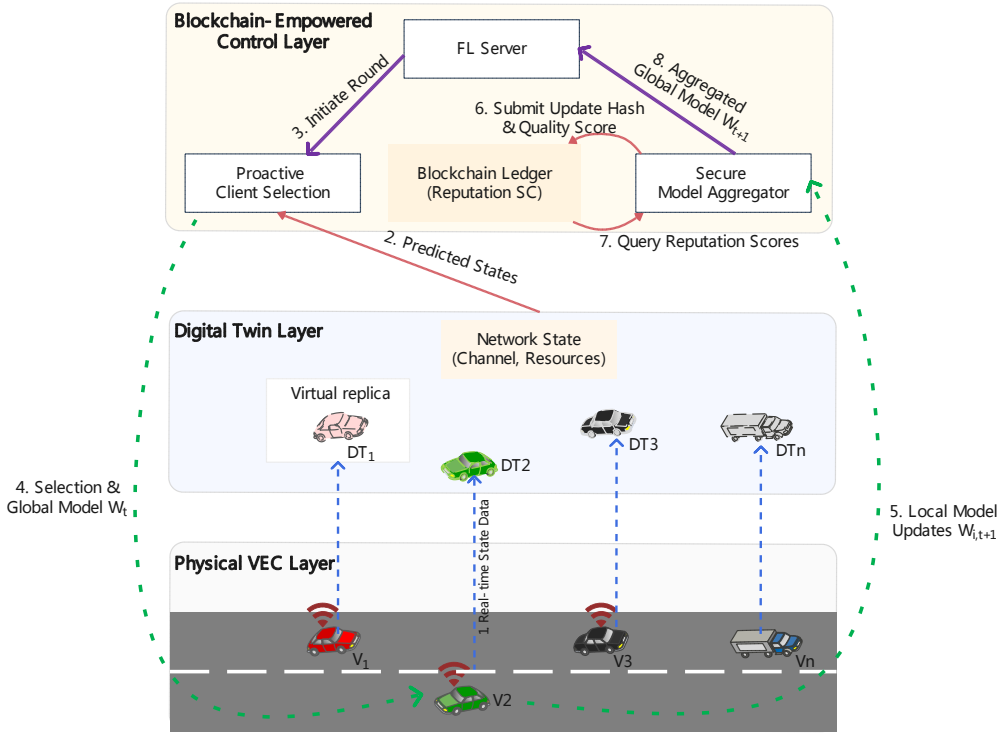

Figure 1. The DTB-FL framework architecture comprising three layers: physical layer with vehicles and RSUs, digital twin layer for predictive analytics, and blockchain-empowered control layer for secure aggregation and incentive management.

3.1. Network Model

We consider a vehicular edge computing (VEC) network comprising an edge server (ES), multiple roadside units (RSUs), and N vehicles denoted by $\mathcal{V} = \{1, 2, \ldots, N\}$. The ES orchestrates the federated learning process through RSUs, which provide network connectivity to vehicles. Each vehicle $i \in \mathcal{V}$ possesses the following capabilities:

- A local dataset $\mathcal{D}_i$ generated by onboard sensors;
- Computational resources characterized by CPU frequency $f_{i,t}$ at time t;
- Communication capability with uplink data rate

$$r_{i,t} = B \log_2 \left(1 + \frac{P_i h_{i,t}}{\sigma^2}\right), \tag{1}$$

where B denotes the allocated bandwidth, P_i is the transmission power, $h_{i,t}$ is the time-varying channel gain, and σ^2 represents the noise power.

Vehicle mobility causes significant temporal variations in the channel gain $h_{i,t}$, leading to highly dynamic network conditions that complicate participant selection and resource allocation.

3.2. Federated Learning Model

The objective is to collaboratively train a global model parameterized by $\mathbf{w}$ that minimizes the weighted average loss across all vehicles:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|} F_i(\mathbf{w}), \tag{2}$$

where $F_i(\mathbf{w})$ represents vehicle i's local loss function evaluated on its dataset $\mathcal{D}_i$. The training process proceeds iteratively over multiple communication rounds as follows:

1. Participant Selection: At round t, the ES selects a subset of vehicles $\mathcal{S}_t \subseteq \mathcal{V}$ based on digital twin predictions of their future availability and channel conditions.
2. Model Distribution: The current global model parameters $\mathbf{w}_t$ are broadcast to all selected vehicles in $\mathcal{S}_t$.
3. Local Training: Each selected vehicle $i \in \mathcal{S}_t$ performs local stochastic gradient descent (SGD) for a fixed number of epochs to compute its model update $\mathbf{w}_{t+1}^i$.
4. Update Upload: Vehicles transmit their local model updates to the ES via their associated RSUs.
5. Secure Aggregation: The ES aggregates the received updates using blockchain-verified reputation-based weights:

$$\mathbf{w}_{t+1} = \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \cdot \mathbf{w}_{t+1}^i, \tag{3}$$

where $\alpha_{i,t}$ represents vehicle i's aggregation weight, which is determined by its reputation score maintained on the blockchain. The weights satisfy $\sum_{i \in \mathcal{S}_t} \alpha_{i,t} = 1$.

3.3. Digital Twin Layer

The digital twin layer maintains real-time virtual replicas of all vehicles to enable predictive participant selection and proactive resource management. Each vehicle's digital twin DT_i continuously tracks a comprehensive state vector:

$$\mathbf{s}_{i,t} = [\text{pos}_{i,t}, \text{vel}_{i,t}, h_{i,t}, f_{i,t}, \rho_{i,t}], \tag{4}$$

where $\text{pos}_{i,t} = (x_{i,t}, y_{i,t})$ represents the vehicle's 2D coordinate position, $\text{vel}_{i,t} = (v_x, v_y)$ denotes the velocity vector, $h_{i,t}$ is the channel gain, $f_{i,t}$ represents the computational capacity (CPU frequency), and $\rho_{i,t}$ is the blockchain-maintained reputation score. Crucially, these state parameters are obtained through infrastructure-based measurements conducted by RSUs and the ES, as detailed in Section 4.2.2. This infrastructure-centric approach ensures data trustworthiness despite the potentially untrusted nature of vehicles, preventing malicious vehicles from falsifying their reported states.

3.3.1. Map-Constrained Mobility Modeling

The digital twin layer incorporates a digital road map $\mathcal{M} = (\mathcal{R}, \mathcal{I})$, where $\mathcal{R}$ denotes the set of all valid coordinates on road segments, and $\mathcal{I}$ represents the set of coordinates at intersections. Vehicle positions are constrained to the road network topology, i.e., $\text{pos}_{i,t} \in \mathcal{R} \cup \mathcal{I}$. This map-based constraint enables realistic mobility prediction that conforms to actual road infrastructure rather than unconstrained two-dimensional motion.

3.3.2. Predictive State Estimation

The digital twin employs machine learning-based predictive models to forecast future vehicle states over a prediction horizon Δt, yielding the predicted state $\hat{\mathbf{s}}_{i,t+\Delta t}$. The prediction methodology comprises three complementary components:

Mobility Prediction: For short-term trajectory forecasting (time horizons of 5 to 30 s), an extended Kalman filter (EKF) predicts future positions by integrating current velocity estimates with kinematic models. The EKF prediction incorporates road network constraints through a projection operation: If the unconstrained prediction yields a position outside the valid road network ($\widehat{\text{pos}}_{i,t+\Delta t} \notin \mathcal{R} \cup \mathcal{I}$), the predicted position is projected onto the nearest valid road coordinate. For longer time horizons (beyond 30 s), the digital twin employs a map-matching algorithm that considers the following: (i) the vehicle's current road segment, (ii) its heading direction, and (iii) probabilistic turning behavior at intersections derived from historical traffic patterns.

Channel Prediction: An LSTM network trained on historical channel measurements forecasts future channel conditions. The training process leverages the spatial correlation between vehicle positions on specific road segments and observed channel quality. By learning location-dependent propagation patterns, the LSTM enables accurate channel prediction conditioned on the predicted future position $\widehat{\text{pos}}_{i,t+\Delta t}$.

Computational Resource Prediction: The digital twin employs ARIMA-based time-series analysis to predict computational resource availability patterns. The prediction model accounts for temporal factors such as time of day and contextual factors such as vehicle type (e.g., personal vehicles may have different usage patterns than commercial vehicles). Computational capacity predictions are validated against resource attestation mechanisms described in Section 4.2.2.

3.3.3. Prediction-Based Participant Selection

The map-aware predictive approach ensures that forecasted vehicle states $\hat{\mathbf{s}}_{i,t+\Delta t}$ remain physically plausible, thereby improving the reliability of proactive participant selection. The selection algorithm assigns lower priority to vehicles predicted to move into road segments with poor RSU coverage or to vehicles approaching intersections where trajectory prediction exhibits high variance (e.g., due to multiple possible turning directions). This prediction-driven strategy enhances training round completion reliability by selecting vehicles that are likely to maintain stable connectivity throughout the training round duration.

3.4. Blockchain Integration

A permissioned blockchain maintains decentralized trust and implements the incentive mechanism, providing tamper-proof record-keeping and automated smart contract execution. The blockchain layer supports three primary functions:

Reputation Management: A smart contract $\mathcal{C}_{\text{rep}}$ evaluates the quality of each vehicle's submitted model update and updates its reputation score accordingly:

$$\rho_{i,t+1} = \beta \cdot \rho_{i,t} + (1 - \beta) \cdot Q(\mathbf{w}_{t+1}^{i}), \tag{5}$$

where $Q(\mathbf{w}_{t+1}^{i}) \in [0, 1]$ measures the quality of vehicle i's update (detailed in Section 6), and $\beta \in [0, 1]$ is a momentum factor that balances historical reputation with recent performance. The reputation score is bounded to $\rho_{i,t} \in [0, 1]$, with higher values indicating more trustworthy participants.

Incentive Distribution: A smart contract $\mathcal{C}_{\text{inc}}$ automatically distributes cryptocurrency rewards to participating vehicles based on their contributions:

$$R_{i,t} = R_{\text{base}} + R_{\text{bonus}} \cdot \frac{\rho_{i,t} \cdot |\mathcal{D}_i|}{\sum_{j \in \mathcal{S}_t} \rho_{j,t} \cdot |\mathcal{D}_j|}, \tag{6}$$

where R_{base} is a fixed participation reward, and R_{bonus} is an additional performance-based reward distributed proportionally to each vehicle's reputation-weighted data contribution. This mechanism incentivizes both participation and high-quality contributions.

Update Verification: Before aggregation, the blockchain verifies the authenticity and integrity of model updates using cryptographic signatures and hash verification. Each update is cryptographically linked to its submitting vehicle's identity, preventing impersonation attacks and ensuring non-repudiation.

3.5. Threat Model

We consider a realistic threat model that reflects the security challenges in vehicular federated learning:

Trusted Entities: The edge server (ES) and roadside units (RSUs) are assumed to be trusted infrastructure components operated by reliable network providers. These entities correctly execute the protocol specifications and do not collude with adversaries.

Untrusted Vehicles: We assume that a subset $\mathcal{V}_m \subset \mathcal{V}$ of vehicles is controlled by adversaries, where $|\mathcal{V}_m| \leq \alpha N$ for some attack fraction $\alpha < 0.5$. Honest vehicles (those in $\mathcal{V} \setminus \mathcal{V}_m$) follow the protocol correctly.

Adversarial Capabilities: Malicious vehicles can execute the following attacks:

- **Model Poisoning**: A malicious vehicle $j \in \mathcal{V}_m$ can submit a crafted model update $\tilde{\mathbf{w}}_{t+1}^j$ designed to degrade global model performance or introduce backdoor triggers.
- **Sybil Attacks**: Adversaries may attempt to create multiple fake vehicle identities to gain disproportionate influence over the aggregation process.
- **Free-Riding**: Malicious vehicles may submit low-quality or random updates to minimize resource expenditure while attempting to claim rewards.

Adversarial Objective: The adversary seeks to maximize the attack's impact on model performance while evading detection by the reputation system. Formally, the adversary solves the following:

$$\begin{aligned} \max_{\{\tilde{\mathbf{w}}_{t+1}^j\}_{j\in\mathcal{V}_m}} \quad & \|F(\mathbf{w}_{\text{poisoned}}) - F(\mathbf{w}_{\text{clean}})\|_2 \\ \text{s.t.} \quad & Q(\tilde{\mathbf{w}}_{t+1}^j) \geq \tau_{\text{detect}}, \quad \forall j \in \mathcal{V}_m, \end{aligned} \tag{7}$$

where $\mathbf{w}_{\text{poisoned}}$ is the global model resulting from aggregating poisoned updates, $\mathbf{w}_{\text{clean}}$ is the model that would result from honest updates only, and τ_{detect} represents the quality threshold below which updates are flagged as suspicious.

3.6. Problem Formulation

Building on the models presented above, we formulate a multi-objective optimization problem that minimizes the system operator's total cost (comprising training time and reward payments) while maintaining model accuracy and security guarantees. The optimization operates over T training rounds:

$$\begin{aligned} \min_{\{\mathcal{S}_t\}_{t=1}^T, \{\alpha_{i,t}\}_{i,t}} \quad & \omega_1 \sum_{t=1}^{T} T_t + \omega_2 \sum_{t=1}^{T} C_t \\ \text{s.t.} \quad & \text{C1: } \text{Acc}(\mathbf{w}_T) \geq \theta, \\ & \text{C2: } |\mathcal{S}_t| = K, \quad \forall t \in \{1, \ldots, T\}, \\ & \text{C3: } \sum_{j \in \mathcal{V}_m \cap \mathcal{S}_t} \alpha_{j,t} \leq \epsilon, \quad \forall t, \\ & \text{C4: } \rho_{i,t} \geq \rho_{\min}, \quad \forall i \in \mathcal{S}_t, \forall t, \\ & \text{C5: } \sum_{i \in \mathcal{S}_t} \alpha_{i,t} = 1, \quad \alpha_{i,t} \geq 0, \quad \forall t, \end{aligned} \tag{8}$$

where the objective function components and constraints are defined as follows:

Objective Function: The first term T_t represents the duration of training round t, given by the following:

$$T_t = \max_{i \in \mathcal{S}_t} \left\{ T_{i,t}^{\text{comp}} + T_{i,t}^{\text{comm}} \right\}, \tag{9}$$

where $T_{i,t}^{\text{comp}} = \frac{C_i |\mathcal{D}_i|}{f_{i,t}}$ is vehicle i's computation time (C_i denotes the CPU cycles required per data sample), and $T_{i,t}^{\text{comm}} = \frac{S}{r_{i,t}}$ is the communication time for transmitting an update of size S bits. The second term $C_t = \sum_{i \in \mathcal{S}_t} R_{i,t}$ represents the total reward cost in round t.

The weight parameters $\omega_1, \omega_2 \geq 0$ balance the relative importance of minimizing training time versus operational costs.

Constraint C1 (Accuracy Requirement): The final trained model must achieve at least the target accuracy threshold θ on a held-out validation dataset.

Constraint C2 (Participation Level): Exactly K vehicles are selected in each round to maintain consistent aggregation quality and predictable resource consumption.

Constraint C3 (Malicious Influence Bound): The cumulative aggregation weight assigned to malicious vehicles (if any are inadvertently selected) must not exceed ϵ, limiting their potential to corrupt the global model. This constraint is enforced through reputation-based weight assignment.

Constraint C4 (Minimum Reputation): Only vehicles with reputation scores above threshold $\rho_{\min}$ are eligible for selection, excluding previously identified malicious actors.

Constraint C5 (Valid Aggregation Weights): The aggregation weights must form a valid probability distribution over the selected participants.

This formulation captures the fundamental trade-offs in vehicular federated learning: The system must balance training efficiency (minimizing time), operational cost (minimizing rewards), model quality (meeting accuracy targets), and security (limiting adversarial influence). The challenge lies in jointly optimizing the participant selection strategy $\{\mathcal{S}_t\}_{t=1}^{T}$ and aggregation weights $\{\alpha_{i,t}\}_{i\in\mathcal{S}_t,t=1}^{T}$ under dynamic vehicular network conditions, unpredictable mobility patterns, and the presence of strategic adversaries. Section 4 presents our digital twin-enabled and blockchain-secured solution to this optimization problem.

4. The Proposed DTB-FL Framework

Building on the technical foundations described in Section 3, we now present our digital twin and blockchain-empowered federated learning (DTB-FL) framework. The framework addresses three critical challenges through targeted technology deployment:

- **Digital Twin Targets Network Dynamics:** Predicts vehicle mobility and resource availability to proactively select stable participants.
- **Blockchain Targets Security and Trust:** Provides tamper-proof reputation management and automated incentive distribution.
- **Integration Targets Efficiency:** Synergistic operation where DT enhances BC's accuracy and BC enhances DT's trustworthiness

We first detail the framework architecture and integration principles and then elaborate each component's operation.

4.1. Framework Architecture

The key innovation of DTB-FL lies in the synergistic integration of digital twin, blockchain, and federated learning technologies within a carefully designed three-layer architecture.

4.1.1. System Components and Layers

As illustrated in Figure 1, our architecture comprises three main layers that work in concert to enable secure and efficient vehicular federated learning.

Physical Layer: This layer consists of real-world VEC components, including vehicles equipped with sensors and computational resources, roadside units (RSUs) providing wireless connectivity, and the edge server (ES) coordinating the learning process. Vehicles collect local data (e.g., traffic conditions, road images) and perform distributed model training without sharing raw data, preserving privacy while enabling collaborative intelligence.

Digital Twin Layer: Residing at the ES, this layer maintains a real-time virtual replica of the physical vehicular network. It captures dynamic vehicle states, including position,

velocity, channel quality, and computational capacity through continuous monitoring. Predictive models forecast future states over the training round duration, enabling proactive rather than reactive decision-making. This foresight is critical in highly dynamic VEC environments where vehicle mobility causes rapid topology changes.

Blockchain and Control Layer: This decentralized layer serves as the trust anchor for the entire system. A permissioned blockchain network, maintained collaboratively by the ES and RSUs, records all critical FL interactions in an immutable ledger. Smart contracts automate reputation score updates based on contribution quality and execute cryptocurrency-based reward distribution transparently. The ES acts as the central orchestrator, synthesizing information from both the DT and blockchain layers to manage participant selection, model aggregation, and security enforcement.

4.1.2. Component Roles and Interactions

Each technology serves a distinct role while contributing synergistically to system objectives.

Digital Twin as Predictive Intelligence: The DT layer maintains a synchronized virtual representation of the vehicular network, continuously monitoring vehicle states (position, velocity, channel quality, and computational capacity). Predictive models forecast future conditions over the training round horizon. Before each FL round, the DT evaluates which vehicles will likely maintain stable connectivity and sufficient resources throughout training. This predictive capability transforms reactive participant selection into proactive optimization, dramatically reducing training interruptions from vehicle mobility.

Blockchain as Trust Infrastructure: The blockchain layer provides decentralized trust through two key functions. First, it maintains an immutable record of all FL interactions—model submissions, quality evaluations, and reputation histories—creating an auditable trail preventing retroactive tampering. Second, smart contracts automatically execute reputation updates and reward distribution based on predefined rules, eliminating the need for a trusted central authority. This creates a transparent, tamper-proof system for managing trust in an inherently untrusted vehicular environment.

Federated Learning as Collaborative Intelligence: The FL layer orchestrates distributed training, leveraging insights from both DT and blockchain components. DT predictions enable participant selection optimizing for training completion probability, while blockchain reputation scores weight model aggregation to suppress malicious contributions. Critically, FL provides the feedback loop—actual training performance refines DT predictions, and gradient quality scores update blockchain reputations. This bidirectional flow enables FL to operate efficiently despite network dynamics and securely despite adversaries, achieving both performance and robustness.

The synergy emerges because each technology addresses a gap in the others: DT handles dynamic uncertainty (mobility and resources), BC handles trust uncertainty (malicious actors and incentives), and FL provides a collaborative learning substrate that benefits from both while feeding back performance data to improve their accuracy.

4.1.3. Bidirectional Integration

The three components interact through bidirectional flows that create a self-improving system through two mechanisms:

Feedforward Integration (DT/BC → FL): The DT provides *predictive guidance* by identifying vehicles predicted to maintain stable connectivity and adequate resources throughout training. The blockchain provides *trust weights* via reputation scores reflecting historical contribution quality. These inputs combine via the synergy utility function

(Equation (10)), which weights DT-predicted performance by blockchain-verified trust, optimizing both efficiency and security.

Feedback Integration (FL → DT/BC): FL performance metrics—actual completion times and quality scores—enable the DT to self-correct its predictions through online learning. Simultaneously, FL gradient analysis provides quality scores to blockchain smart contracts, updating reputations to reflect actual contribution value rather than claimed capabilities. The blockchain's accumulated reputation history enhances DT predictions by identifying consistently reliable vehicles, creating a virtuous cycle where past behavior informs future selection.

This bidirectional reinforcement creates two critical synergies.

DT Enhances Blockchain Security: Proactive DT-based selection creates a stable, reliable cohort each round. This stability acts as a noise filter for the reputation system—by minimizing dropouts and selecting high-performing participants, the DT ensures that honest gradients are more consistent in direction and magnitude. This homogeneity makes cosine similarity-based quality evaluation more effective at identifying malicious updates, which stand out as directional anomalies. Without DT guidance, random sampling would include unstable vehicles for which their dropouts or degraded performance create noise, masking adversarial behavior.

Blockchain Enhances DT Selection: The blockchain provides a critical trust metric feeding back to the digital twin. Historical trustworthiness is as important as predicted performance. Therefore, we incorporate reputation score $\rho_{i,t}$ into the DT utility function:

$$U_{i,t} = \left(\alpha \cdot \frac{\hat{r}_{i,t}}{R_{\max}} + (1-\alpha) \cdot \frac{\hat{f}_{i,t}}{F_{\max}} \right) \cdot \rho_{i,t}, \tag{10}$$

where $\hat{r}_{i,t}$ and $\hat{f}_{i,t}$ are the DT-predicted data rate and CPU frequency, $R_{\max}$ and $F_{\max}$ are normalization constants, and $\alpha \in [0,1]$ balances communication versus computation. This multiplicative integration ensures that low-reputation vehicles receive proportionally reduced utility scores regardless of predicted capability, preventing repeated selection of high-performing but malicious vehicles and hardening FL against persistent adversaries.

4.1.4. High-Level Workflow

The integrated system operates through six coordinated phases in each federated learning round (see Algorithm 2):

1. **Intelligent Participant Selection:** The ES combines blockchain reputation scores and DT-predicted vehicle states to select the top-K participants via the synergy utility function.
2. **Parallel Local Training:** Selected vehicles train on private data and upload model updates along with actual training times to the ES.
3. **Quality Evaluation:** The ES evaluates update quality using cosine similarity analysis, classifying contributions as valid or invalid based on the threshold $Q_{\min}^{\text{non-IID}}$.
4. **Reputation-Weighted Aggregation:** Valid updates are aggregated with weights proportional to reputation scores and dataset sizes, producing the new global model.
5. **Blockchain Updates and Incentives:** Smart contracts automatically update reputation scores based on quality evaluations and distribute cryptocurrency rewards accordingly.
6. **DT Predictor Refinement:** Actual training times are used to retrain the DT predictor periodically, improving prediction accuracy for future rounds.

This workflow creates a self-improving cycle: Accurate DT predictions enable better participant selection, leading to higher-quality updates, which refine reputation scores

and enhance future predictions. The detailed technical implementation of each phase is presented in Section 4.4.

4.2. Digital Twin-Based Participant Selection

To mitigate the effects of high mobility and resource heterogeneity in vehicular edge computing, we design a proactive participant selection mechanism that leverages the predictive capabilities of the digital twin. Rather than selecting vehicles based solely on current status snapshots, our approach identifies those predicted to remain suitable throughout the upcoming training round.

4.2.1. Utility-Based Selection Strategy

The selection is guided by the synergy utility function defined in Equation (10), which quantifies the suitability of vehicle i for round t. This function prioritizes vehicles with higher predicted communication rates and computational capabilities while incorporating trust via reputation scores, thereby minimizing expected round completion time (as formulated in T_t in Section 3).

Using the forecasted state $\hat{\mathbf{s}}_{i,t} = [\widehat{\text{pos}}_{i,t}, \widehat{\text{vel}}_{i,t}, \hat{h}_{i,t}, \hat{f}_{i,t}]$ from the DT, we estimate the future data rate $\hat{r}_{i,t}$ and CPU frequency $\hat{f}_{i,t}$ that the vehicle is expected to maintain during training. The parameters $R_{\max}$ and $F_{\max}$ represent the maximum achievable data rate and CPU frequency in the network for normalization, and $\alpha \in [0,1]$ balances communication and computation priorities based on whether the FL task is model-size-limited or computation-limited.

At the start of round t, the ES first filters vehicles by a minimum reputation threshold $\rho_{\min}$ to exclude consistently poor performers or adversaries. For remaining candidates, it computes $U_{i,t}$ and selects the top-K vehicles to form the participant set $\mathcal{S}_t$. This proactive strategy reduces training interruptions from vehicle dropouts and resource bottlenecks, enhancing both FL efficiency and stability. The procedure is formalized in Algorithm 1.

Algorithm 1 DT-Based Proactive Participant Selection

Require: Set of available vehicles $\mathcal{V}_t$, number of participants to select K, weighting factor α, reputation threshold $\rho_{\min}$
Ensure: Selected participant set $\mathcal{S}_t$
Initialize an empty list $\mathcal{L}$
for each vehicle $i \in \mathcal{V}_t$ **do**
  Retrieve current reputation $\rho_{i,t}$ from blockchain
  if $\rho_{i,t} < \rho_{\min}$ **then**
    continue ▷ *Exclude low-reputation vehicles*
  Retrieve the predicted state $\hat{\mathbf{s}}_{i,t}$ from the Digital Twin layer
  Estimate the predicted data rate $\hat{r}_{i,t}$ and CPU frequency $\hat{f}_{i,t}$
  Calculate the utility score using Equation (10):
  $$U_{i,t} = \left(\alpha \cdot \frac{\hat{r}_{i,t}}{R_{\max}} + (1-\alpha) \cdot \frac{\hat{f}_{i,t}}{F_{\max}} \right) \cdot \rho_{i,t}$$
  Add the tuple $(i, U_{i,t})$ to $\mathcal{L}$
Sort $\mathcal{L}$ in descending order based on $U_{i,t}$
Select the first K vehicles from the sorted list to form the set $\mathcal{S}_t$
return $\mathcal{S}_t$

Map-Aware Utility Computation: The utility function in Equation (10) leverages map-constrained predictions from the digital twin. For vehicle i traveling on road segment $r_i \in \mathcal{R}$, the DT predicts not only future channel quality but also the likelihood of maintaining RSU coverage based on the road trajectory and network topology. This coverage consideration is implicitly captured through the predicted channel quality $\hat{h}_{i,t}$: vehicles on road segments with poor RSU coverage or approaching coverage boundaries exhibit

degraded channel predictions, resulting in lower predicted data rates $\hat{r}_{i,t}$ and, consequently, lower utility scores. This geographic awareness prevents the selection of participants likely to experience disconnection mid-training, even if their current channel conditions appear favorable.

4.2.2. Infrastructure-Based State Estimation

A critical consideration in our framework is obtaining reliable state information from potentially untrusted vehicles—a fundamental trust paradox in security-critical systems. To resolve this, the digital twin layer relies primarily on infrastructure-based measurements performed by trusted RSUs and the edge server, rather than vehicle self-reports. This ensures the integrity of the state vector $\mathbf{s}_{i,t}$ even in the presence of malicious participants who might falsify their capabilities to gain selection.

Channel State Estimation ($h_{i,t}$ and $r_{i,t}$): RSUs continuously monitor the wireless channel quality of connected vehicles through standard physical layer measurements that cannot be manipulated by vehicles:

- Reference Signal Received Power (RSRP) and Signal-to-Interference-Plus-Noise Ratio (SINR) are measured directly at the RSU receiver based on uplink transmissions.
- The Channel Quality Indicator (CQI) is computed based on observed signal characteristics and channel reciprocity.
- The predicted data rate $\hat{r}_{i,t}$ is calculated using the Shannon capacity formula with infrastructure-measured parameters:

$$\hat{r}_{i,t} = B \log_2\left(1 + \text{SINR}_{i,t}^{\text{measured}}\right)$$

where B is the allocated bandwidth.

Computational Capacity Verification ($f_{i,t}$): While vehicles initially report their computational capabilities during registration, the ES employs a computational capacity attestation mechanism before each training round:

1. The ES sends a calibrated computational task τ_{test} with known complexity C_{test} CPU cycles (e.g., a standardized matrix multiplication benchmark).
2. Vehicle i must return the correct result within deadline t_{deadline}.
3. The verified computational frequency is estimated as follows:

$$\hat{f}_{i,t} = \frac{C_{\text{test}}}{t_{\text{actual},i}}$$

where $t_{\text{actual},i}$ is the actual completion time measured by the ES.

4. Vehicles failing to meet the deadline or returning incorrect results are excluded from selection.

Mobility Tracking ($\text{pos}_{i,t}$ and $\text{vel}_{i,t}$): Vehicle positions are determined through infrastructure-based localization, assuming synchronized RSUs via GPS or network time protocol:

- Multiple RSUs perform time difference of arrival (TDoA) measurements using vehicle uplink signals.
- Position is triangulated using at least three RSU measurements with sub-meter accuracy.
- Velocity is derived from consecutive position estimates: $\text{vel}_{i,t} = \frac{\text{pos}_{i,t} - \text{pos}_{i,t-\Delta t}}{\Delta t}$.

This infrastructure-centric approach ensures that the digital twin operates on trustworthy data, maintaining the integrity of the proactive selection mechanism even when vehicles are potentially malicious. The utility function in Equation (10) thus uses only

verified values $\hat{r}_{i,t}$ and $\hat{f}_{i,t}$, not self-reported claims, preventing adversaries from gaming the selection process through capability inflation.

4.3. Blockchain-Based Security and Incentives

To counter security threats from malicious participants and address incentive deficiencies in voluntary collaboration, we implement a comprehensive mechanism using a permissioned blockchain and smart contracts.

4.3.1. Reputation Management

We maintain a dynamic reputation score $\rho_{i,t} \in [0,1]$ for each vehicle i, reflecting its trustworthiness based on historical contribution quality. This score is stored on the blockchain and updated after each round via a smart contract. For an update $\mathbf{w}_{t+1}^i$ from vehicle $i \in \mathcal{S}_t$, the ES evaluates quality using cosine similarity of gradients. Let $\mathbf{g}_{t+1}^i = \mathbf{w}_{t+1}^i - \mathbf{w}_t$ denote the gradient (model update). The quality score $Q_{i,t}$ is computed as follows:

$$Q_{i,t} = \frac{1}{|\mathcal{S}_t| - 1} \sum_{j \in \mathcal{S}_t, j \neq i} \frac{\mathbf{g}_{t+1}^i \cdot \mathbf{g}_{t+1}^j}{\|\mathbf{g}_{t+1}^i\| \|\mathbf{g}_{t+1}^j\|}. \tag{11}$$

Under this metric, honest updates from vehicles with similar optimization objectives align closely in direction, yielding high $Q_{i,t}$ values (typically > 0.4), while malicious updates that inject adversarial perturbations diverge significantly, resulting in low scores. The smart contract updates reputation using an exponential moving average:

$$\rho_{i,t+1} = \beta \cdot \rho_{i,t} + (1 - \beta) \cdot Q_{i,t}, \tag{12}$$

where $\beta \in [0,1]$ is the decay factor that balances historical behavior with recent performance. We set $\beta = 0.5$ by default to provide sufficient inertia against temporary fluctuations while remaining responsive to sustained behavioral changes.

4.3.2. Cross-Edge Reputation Portability

To address vehicle mobility across edge server boundaries—a common occurrence in vehicular networks where vehicles may traverse multiple infrastructure coverage zones during their journeys—we implement a federated reputation management system that maintains reputation continuity across geographic regions.

Reputation Blockchain Federation: Multiple edge servers in a geographic region (e.g., a metropolitan area or highway corridor) form a consortium blockchain network with shared ledger access. When vehicle i moves from edge server ES_a's coverage area to ES_b's area, the following occurs:

- ES_a records a reputation certificate $\mathcal{C}_i = \{\mathrm{ID}_i, \rho_{i,t}, t, \mathrm{sig}_{ES_a}\}$ on the shared blockchain before handover.
- ES_b retrieves this certificate from the blockchain and initializes the vehicle with $\rho_{i,0}^{ES_b} = \rho_{i,t}^{ES_a}$.
- The certificate includes a cryptographic signature sig_{ES_a}, preventing reputation tampering or forgery.

Hybrid Trust Initialization: For vehicles with limited interaction history at a particular edge server, we employ a hybrid model that combines transferred reputation, verification results, and conservative defaults:

$$\rho_{i,0}^{\text{new}} = \begin{cases} \rho_{i,t}^{\text{transferred}} & \text{if reputation certificate exists} \\ \rho_{\text{init}} + \Delta_{\text{attestation}} & \text{if capacity attestation verified} \\ \rho_{\text{min}} & \text{otherwise} \end{cases} \tag{13}$$

where $\rho_{\text{init}} = 0.5$ is the default neutral initialization for completely new vehicles, and $\Delta_{\text{attestation}} \in [0, 0.2]$ is a bonus for successfully passing computational verification (Section 4.2.2), providing initial trust to demonstrably capable vehicles.

Minimum Interaction Threshold: Our system is designed to provide security benefits even with limited interactions within a single edge server's coverage. Assuming 30 s training rounds and a typical RSU coverage of 2–3 km in urban environments, vehicles traveling at 40–60 km/h remain within an edge server's coverage for approximately 2–4 min, enabling 5–15 training rounds during a typical commute segment. The reputation mechanism operates effectively within this time frame:

- **Rounds 1–3**: Vehicle participates with basic verification (computational attestation) and neutral reputation.
- **Rounds 4+**: Reputation begins to differentiate honest from malicious behavior with statistical significance, as malicious vehicles consistently receive low quality scores.

This design ensures meaningful reputation establishment within typical vehicle dwell times, while the blockchain federation enables long-term trust accumulation across the vehicle's entire journey.

4.3.3. Secure Aggregation and Incentive Mechanism

Reputation scores enable secure model aggregation that mitigates the influence of malicious participants. We replace standard FedAvg with a trust-weighted approach:

$$\mathbf{w}_{t+1} = \sum_{i \in \mathcal{S}_t} \frac{\rho_{i,t} \cdot |\mathcal{D}_i|}{\sum_{j \in \mathcal{S}_t} \rho_{j,t} \cdot |\mathcal{D}_j|} \mathbf{w}_{t+1}^i. \tag{14}$$

This weights contributions by both data quantity and reputation quality, amplifying reliable inputs and marginalizing potential threats. Vehicles with low reputation (e.g., $\rho_{i,t} < 0.3$) contribute negligibly even if they possess large datasets, thereby limiting the influence of malicious actors on the global model without requiring explicit Byzantine detection.

The smart contract also distributes cryptocurrency rewards (e.g., tokens on the blockchain) post-round to incentivize sustained high-quality participation. Vehicle i's reward $R_{i,t}$ is calculated as follows:

$$R_{i,t} = R_{\text{total}} \cdot \frac{\rho_{i,t} \cdot |\mathcal{D}_i|}{\sum_{j \in \mathcal{S}_t} \rho_{j,t} \cdot |\mathcal{D}_j|}, \tag{15}$$

where R_{total} is the total reward pool for round t. This mechanism creates a virtuous cycle: vehicles that maintain high reputation scores through consistent honest behavior earn proportionally larger rewards, incentivizing long-term cooperation. The transparency of blockchain ensures that all participants can verify the fairness of reward distribution by auditing the smart contract's execution, preventing disputes and promoting long-term engagement in the federated learning ecosystem.

4.3.4. Design Rationale and Limitations

Rationale for Cosine Similarity: We employ cosine similarity for quality assessment due to its distinct advantages in the federated learning context:

1. **Scale Invariance:** Cosine similarity measures the angular alignment between gradient vectors independent of their magnitudes. This property is critical because vehicles possess datasets of varying sizes ($|\mathcal{D}_i|$), leading to natural variations in gradient magnitudes even among honest participants. By focusing on direction rather than magnitude, cosine similarity prevents penalizing vehicles simply due to smaller datasets while effectively detecting malicious gradients that point in adversarial directions [40].
2. **Interpretability:** The bounded range $[-1, 1]$ provides intuitive interpretation: Values near 1 indicate alignment (likely honest), values near 0 suggest independence, and negative values indicate opposition (potentially malicious). This facilitates straightforward threshold-based detection.
3. **Robustness to Data Heterogeneity:** In non-IID settings, honest gradients computed on different data distributions may have different magnitudes but should still point toward reducing the global loss. Cosine similarity captures this directional consistency while being robust to magnitude variations caused by statistical heterogeneity [41].
4. **Computational Efficiency:** The metric requires only inner products and norms, with $O(d)$ complexity where d is the model dimension, making it scalable for large neural networks in resource-constrained VEC environments.
5. **Established Effectiveness:** Cosine similarity and related angular metrics have been successfully employed in Byzantine-robust aggregation methods such as Krum [40] and FABA [42], demonstrating their effectiveness in identifying outlier gradients in adversarial FL settings.

Robustness to Non-IID Data: A critical consideration is distinguishing between legitimate gradient divergence due to non-IID data and malicious poisoning. Our reputation mechanism addresses this through two key insights:

1. **Statistical Differentiation:** While non-IID data causes gradient variance, malicious updates exhibit systematically different patterns. Honest gradients, despite data heterogeneity, maintain a positive correlation with the true optimization direction. In contrast, poisoned gradients from label-flipping attacks point in adversarial directions. Our experiments (detailed in Section 6) demonstrate that honest vehicles maintain $Q_{i,t} \in [0.4, 0.8]$ even under high non-IID conditions, while malicious vehicles executing label-flipping attacks consistently score $Q_{i,t} < 0.2$.
2. **Adaptive Normalization:** Rather than using fixed reputation penalties, we implement an adaptive mechanism that accounts for expected non-IID variance by establishing a minimum quality threshold:

$$Q_{\text{min}}^{\text{non-IID}} = 0.3 \tag{16}$$

This threshold represents the minimum similarity expected for honest vehicles under non-IID conditions, determined empirically in our experiments. Updates scoring below this threshold are flagged as potentially malicious and receive reputation penalties. This prevents unfair penalization of vehicles with rare data distributions while maintaining security.

Additionally, the reputation update in Equation (12) uses a decay factor $\beta = 0.5$ by default, providing inertia that prevents temporary divergence (due to local data batches or transient network conditions) from drastically affecting long-term reputation. This temporal smoothing is crucial for maintaining fairness under non-IID conditions.

Limitations Against Sophisticated Collusion: While the cosine similarity metric in Equation (11) is effective for identifying and isolating individual malicious actors who submit divergent updates, we acknowledge its limitations against more sophisticated, coordinated attacks. The primary vulnerability is the **collusion attack**, where a group of malicious vehicles (potentially Sybil identities controlled by a single adversary) submit carefully crafted malicious updates that are highly similar to each other. In such a scenario, these colluding attackers would achieve high pairwise similarity scores among themselves, artificially inflating their reputations and potentially penalizing honest participants whose updates may differ due to non-IID data.

However, the proposed DTB-FL framework is uniquely equipped to mitigate such advanced threats by leveraging the synergy between the digital and physical layers. Unlike purely software-based defense mechanisms, our framework can correlate digital behavior (gradient submissions) with physical-world context provided by the digital twin layer. The DT maintains real-time, high-fidelity data on vehicle states, including precise location, velocity, and trajectory. This enables a second layer of defense against collusion:

Physical Plausibility Check: The Edge Server can implement an advanced heuristic that flags a potential collusion ring if a cluster of vehicles simultaneously exhibits the following: (1) geographic co-location or anomalous, coordinated mobility patterns (e.g., traveling in a tight, unlikely convoy) and (2) submitted gradients showing high intra-cluster similarity while deviating significantly from the broader consensus of other participants. An honest, diverse set of vehicles is unlikely to produce such a strong spatio–digital correlation.

This cross-layer validation turns a critical vulnerability into a detectable anomaly. While a sophisticated attacker can forge digital gradients, faking coordinated physical locations and trajectories is substantially more difficult and costly. By incorporating this physical context into the trust evaluation, the ES can proactively penalize the reputation of the suspected colluding cluster, ensuring the integrity of the global model. While cosine similarity provides strong defense against individual Byzantine attackers, we address the remaining collusion threat through the novel integration of DT-based physical plausibility checks described above. This capability represents a promising direction for future enhancements of the reputation algorithm within the DTB-FL architecture.

4.4. Complete System Algorithm

Building on the participant selection strategy and security mechanisms, Algorithm 2 presents our complete DTB-FL training protocol that integrates DT-based prediction, blockchain verification, and adaptive reputation management. The algorithm operates in six coordinated phases per communication round: (1) intelligent participant selection using DT predictions and reputation scores, (2) parallel local training on selected vehicles, (3) blockchain-based quality evaluation of model updates, (4) reputation-weighted global aggregation, (5) smart contract-driven reputation updates with tokenized incentives, and (6) adaptive refinement of the DT predictor using actual training times. This design ensures that only high-quality contributions influence the global model while maintaining transparency and accountability through blockchain records.

Algorithm 2 DTB-FL Training Protocol

Require: Vehicles $\mathcal{V} = \{v_1, \ldots, v_N\}$, initial model $\mathbf{w}_0$, total rounds T, participants per round K, local epochs E, learning rate η, initial reputations $\rho_i^0 = 0.5$, reputation threshold $\rho_{\min}$, decay factor $\beta = 0.5$, blockchain $\mathcal{BC}$, round duration $\Delta T = 30$ s

Ensure: Final model $\mathbf{w}_T$, final reputations $\{\rho_i^T\}_{i=1}^N$

Initialization:
Deploy smart contracts for reputation management and incentive distribution on $\mathcal{BC}$
Initialize DT predictor with empty training set $\mathcal{D}_{\text{DT}} \leftarrow \emptyset$
for round $t = 1, \ldots, T$ **do**
  /* Phase 1: Intelligent Participant Selection */
  $\mathcal{V}_{\text{elig}}^t \leftarrow \{v_i \in \mathcal{V} : \rho_i^{t-1} \geq \rho_{\min}\}$
  for each $v_i \in \mathcal{V}_{\text{elig}}^t$ **do**
    Collect state: $\mathbf{x}_i^t = [\rho_i^{t-1}, \text{speed}_i^t, \text{location}_i^t, f_i^t, r_i^t]$
    Predict time: $\hat{T}_{i,t} \leftarrow \text{DT-Predictor}(\mathbf{x}_i^t, \text{history}_{i,t-h:t-1})$
    Compute utility using Equation (10): $U_{i,t} \leftarrow \left(\alpha \cdot \frac{\hat{r}_{i,t}}{R_{\max}} + (1-\alpha) \cdot \frac{\hat{f}_{i,t}}{F_{\max}}\right) \cdot \rho_i^{t-1}$
  $\mathcal{S}_t \leftarrow \text{TopK}(\{(v_i, U_{i,t})\}, K)$ ▷ *Select top-K vehicles*
  Broadcast global model $\mathbf{w}_{t-1}$ to vehicles in $\mathcal{S}_t$
  /* Phase 2: Parallel Local Training */
  for each $v_i \in \mathcal{S}_t$ **in parallel do**
    $\mathbf{w}_i \leftarrow \mathbf{w}_{t-1}$
    for epoch $e = 1, \ldots, E$ **do**
      for batch $\mathcal{B} \subset \mathcal{D}_i$ **do**
        $\mathbf{w}_i \leftarrow \mathbf{w}_i - \eta \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i; \mathcal{B})$ ▷ *Local SGD*
    $\Delta \mathbf{w}_i^t \leftarrow \mathbf{w}_i - \mathbf{w}_{t-1}$ ▷ *Compute gradient*
    Upload $\Delta \mathbf{w}_i^t$ and actual time $T_{i,t}^{\text{actual}}$ to edge server
  /* Phase 3: Quality Evaluation */
  for each $v_i \in \mathcal{S}_t$ **do**
    Compute quality score using Equation (11):
    $Q_{i,t} \leftarrow \frac{1}{|\mathcal{S}_t|-1} \sum_{j \in \mathcal{S}_t, j \neq i} \frac{\Delta \mathbf{w}_i^t \cdot \Delta \mathbf{w}_j^t}{\|\Delta \mathbf{w}_i^t\| \|\Delta \mathbf{w}_j^t\|}$
    $\text{isValid}_i^t \leftarrow \mathbb{I}(Q_{i,t} \geq Q_{\min}^{\text{non-IID}})$ ▷ $Q_{\min}^{non\text{-}IID} = 0.3$
  Submit evaluation record $\mathcal{E}_t = \{(v_i, Q_{i,t}, \text{isValid}_i^t)\}_{i \in \mathcal{S}_t}$ to blockchain
  /* Phase 4: Reputation-Weighted Aggregation */
  $\mathcal{S}_t^{\text{valid}} \leftarrow \{v_i \in \mathcal{S}_t : \text{isValid}_i^t = \text{True}\}$
  Aggregate using Equation (14):
  $\mathbf{w}_t \leftarrow \sum_{v_i \in \mathcal{S}_t^{\text{valid}}} \frac{\rho_i^{t-1} \cdot |\mathcal{D}_i|}{\sum_{v_j \in \mathcal{S}_t^{\text{valid}}} \rho_j^{t-1} \cdot |\mathcal{D}_j|} \cdot \Delta \mathbf{w}_i^t$
  /* Phase 5: Blockchain Reputation Update and Incentive Distribution */
  Trigger smart contract with input $\mathcal{E}_t$
  for each $v_i \in \mathcal{S}_t$ **do**
    if isValid_i^t **then**
      Update reputation using Equation (12): $\rho_i^t \leftarrow \beta \cdot \rho_i^{t-1} + (1-\beta) \cdot Q_{i,t}$
      Distribute reward using Equation (15):
      $R_{i,t} \leftarrow R_{\text{total}} \cdot \frac{\rho_i^t \cdot |\mathcal{D}_i|}{\sum_{v_j \in \mathcal{S}_t^{\text{valid}}} \rho_j^t \cdot |\mathcal{D}_j|}$
    else
      $\rho_i^t \leftarrow \max(\rho_{\min}, \rho_i^{t-1} - 0.1)$ ▷ *Penalize invalid update*
      $R_{i,t} \leftarrow 0$ ▷ *No reward*
  Record reputation updates $\{\rho_i^t\}_{i \in \mathcal{S}_t}$ and rewards $\{R_{i,t}\}_{i \in \mathcal{S}_t}$ on blockchain
  /* Phase 6: Update DT Predictor */
  $\mathcal{D}_{\text{DT}} \leftarrow \mathcal{D}_{\text{DT}} \cup \{(\mathbf{x}_i^t, T_{i,t}^{\text{actual}}) : v_i \in \mathcal{S}_t\}$
  if $t \bmod 10 = 0$ **then**
    Retrain DT predictor on accumulated dataset $\mathcal{D}_{\text{DT}}$ ▷ *Periodic retraining*
return $\mathbf{w}_T$, $\{\rho_i^T\}_{i=1}^N$

4.4.1. Algorithm Walkthrough

Initialization Phase: The edge server initializes the blockchain network by deploying smart contracts for reputation management and reward distribution. The DT predictor begins with an empty training dataset $\mathcal{D}_{\text{DT}}$ that will accumulate historical training records over time. All vehicles start with neutral reputation scores ($\rho_i^0 = 0.5$) to ensure fair initial participation opportunities.

Phase 1: Intelligent Participant Selection: At each round t, the edge server filters vehicles with a reputation that exceeds the minimum threshold ($\rho_i^{t-1} \geq \rho_{\min}$) to exclude consistently poor performers. For each eligible vehicle v_i, the server collects current state information including reputation, mobility status, and device capabilities (verified through infrastructure-based measurements as described in Section 4.2.2) and then uses the DT predictor to estimate training time $\hat{T}_{i,t}$. The utility score $U_{i,t}$ computed via Equation (10) balances contribution quality (via reputation) against training efficiency (via predicted communication and computation capabilities), with hyperparameter α controlling the communication–computation trade-off. The top-K vehicles with highest utility scores are selected, and the current global model $\mathbf{w}_{t-1}$ is broadcast to them.

Phase 2: Parallel Local Training: Selected vehicles independently train on their local datasets for E epochs using mini-batch stochastic gradient descent. Each vehicle v_i records its actual training and communication duration $T_{i,t}^{\text{actual}}$ and computes the model update $\Delta\mathbf{w}_i^t = \mathbf{w}_i - \mathbf{w}_{t-1}$, which represents the gradient direction. Both the gradient update and timing information are uploaded to the edge server, with the latter serving as ground truth labels for refining the DT predictor.

Phase 3: Quality Evaluation: The edge server evaluates each received update by computing its cosine similarity with all other updates using Equation (11). This measures how well the update aligns with the consensus direction. Updates are marked as valid if their quality score $Q_{i,t}$ meets the minimum threshold $Q_{\min}^{\text{non-IID}} = 0.3$, which accounts for expected variance in non-IID settings, as discussed in Section 4.3.4. The evaluation record $\mathcal{E}_t$ containing all quality scores and validity flags is submitted to the blockchain as a transaction, ensuring tamper-proof documentation of each vehicle's contribution quality.

Phase 4: Reputation-Weighted Aggregation: Only valid updates participate in global aggregation according to Equation (14), with each weighted proportionally to the product of the vehicle's reputation score and dataset size. The normalization ensures that weights sum to one. This reputation-based weighting amplifies contributions from consistently reliable vehicles while reducing the influence of those with lower trust scores, thereby improving the robustness of the global model against low-quality or malicious updates.

Phase 5: Blockchain Reputation Update and Incentive Distribution: A smart contract automatically executes upon receiving the evaluation record $\mathcal{E}_t$. For vehicles with valid updates, reputation is updated using the exponential moving average formula in Equation (12) with decay factor $\beta = 0.5$, providing stability while adapting to recent performance. Rewards are distributed according to Equation (15), proportional to reputation-weighted data contributions. Invalid updates trigger a reputation penalty ($\rho_i^t = \max(\rho_{\min}, \rho_i^{t-1} - 0.1)$) and zero reward to discourage malicious behavior. All reputation updates and reward distributions are recorded on the blockchain, providing transparent and verifiable history that participants can audit.

Phase 6: Adaptive DT Update: The actual training time $T_{i,t}^{\text{actual}}$ collected from this round is paired with their corresponding state vector $\mathbf{x}_i^t$ and added to the DT training dataset $\mathcal{D}_{\text{DT}}$. Every 10 rounds, the DT predictor is retrained on the accumulated dataset, allowing it to adapt to changing network conditions, vehicle mobility patterns, and workload distributions. This continuous learning ensures prediction accuracy improves over time, leading to better participant selection decisions in future rounds.

The algorithm completes after T communication rounds, returning the final global model $\mathbf{w}_T$ and the mature reputation score $\{\rho_i^T\}_{i=1}^N$ that reflect each vehicle's long-term contribution quality.

4.4.2. Implementation Considerations

DTB-FL deployment is feasible given current infrastructure trends. The blockchain uses permissioned PBFT consensus providing sub-second finality, with hybrid on-chain/off-chain storage—only reputation scores and evaluations are stored on-chain, while gradients use conventional channels. The DT maintains lightweight state representations for candidate pools (50–100 vehicles) rather than full simulations, making computation manageable. Modern 5G networks (100+ Mbps peak rates) and MEC infrastructure co-located with base stations [43] provide sufficient bandwidth and acceptable latency (sub-10 ms).

Industry initiatives demonstrate readiness: the MOBI consortium [44] brings together major automakers and blockchain providers for standardized vehicular blockchain applications, while 3GPP V2X standards [12] provide protocols supporting DTB-FL data exchange requirements.

5. Convergence and Complexity Analysis

To formally ground the performance of DTB-FL, this section provides a theoretical analysis of its convergence guarantees and computational complexity. We demonstrate that despite the presence of malicious actors and a dynamic network, DTB-FL converges to a near-optimal solution, and we quantify the overhead associated with its advanced features.

5.1. Convergence Analysis

Our convergence analysis builds upon the foundational proofs for FedAvg, but extends them to account for the two core mechanisms of DTB-FL: reputation-weighted aggregation and proactive participant selection.

5.1.1. Assumptions

We make the following standard assumptions, which are common in the FL convergence literature:

Assumption 1 (L-Smoothness)**.** *The global loss function $F(\mathbf{w})$ and all local loss functions $F_i(\mathbf{w})$ are L-smooth. That is, for any $\mathbf{w}_1$ and $\mathbf{w}_2$,*

$$||\nabla F(\mathbf{w}_1) - \nabla F(\mathbf{w}_2)|| \leq L||\mathbf{w}_1 - \mathbf{w}_2||$$

Assumption 2 (μ-Strong Convexity)**.** *The global loss function $F(\mathbf{w})$ is μ-strongly convex. For any $\mathbf{w}_1$ and $\mathbf{w}_2$,*

$$F(\mathbf{w}_1) \geq F(\mathbf{w}_2) + \langle \nabla F(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{\mu}{2}||\mathbf{w}_1 - \mathbf{w}_2||^2$$

Assumption 3 (Bounded Gradient Variance)**.** *The variance of the stochastic gradients for any local model is bounded:*

$$\mathbb{E}[||\nabla F_i(\mathbf{w}; \xi) - \nabla F_i(\mathbf{w})||^2] \leq \sigma^2$$

where ξ is a data sample drawn from $\mathcal{D}_i$. For simplicity in this sketch, we assume one full local epoch, so the local update is based on the full local gradient $\nabla F_i(\mathbf{w})$.

Assumption 4 (Bounded Malicious Influence)**.** *Our blockchain-based reputation system ensures that the total aggregation weight assigned to malicious participants is bounded. Let $\mathcal{V}_m$ be the set of malicious vehicles. The reputation mechanism guarantees that for any round t,*

$$\sum_{j \in \mathcal{V}_m \cap \mathcal{S}_t} \alpha_{j,t} \leq \epsilon$$

where $\alpha_{j,t}$ is the aggregation weight from Equation (11), and ϵ is a small constant representing the maximum tolerable malicious influence. The system is designed to drive $\rho_{j,t} \to 0$ for malicious nodes, thus ensuring that this condition holds.

Assumption 5 (Bounded Adversarial Perturbation)**.** *A malicious vehicle $j \in \mathcal{V}_m$ submits a poisoned update $\tilde{\mathbf{w}}_{t+1}^j$. The gradient derived from this update can be modeled as $\tilde{\mathbf{g}}_t^j = \nabla F_j(\mathbf{w}_t) + \mathbf{e}_{j,t}$, where the adversarial perturbation $\mathbf{e}_{j,t}$ is bounded, i.e., $||\mathbf{e}_{j,t}|| \leq G_{adv}$.*

The proactive participant selection enabled by the **digital twin** does not alter the mathematical formulation of the aggregation step itself, but it critically reinforces the assumptions. By selecting vehicles predicted to have stable connectivity, it minimizes participant dropouts. This ensures that the effective set of participants K is consistent, reducing variance and preventing the destabilizing effects of stragglers, which standard FL analyses often have to ignore or simplify.

5.1.2. Proof Sketch

Our goal is to bound the optimality gap $\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)]$. We start by analyzing the progress in a single round t. The global model update is as follows:

$$\mathbf{w}_{t+1} = \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{w}_{t+1}^i = \mathbf{w}_t - \eta \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{g}_t^i$$

where $\mathbf{g}_t^i = \nabla F_i(\mathbf{w}_t)$ for honest clients, and $\mathbf{g}_t^i = \nabla F_i(\mathbf{w}_t) + \mathbf{e}_{i,t}$ for malicious clients.

Let us analyze the expected distance to the optimal model $\mathbf{w}^*$:

$$\begin{aligned} \mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] &= \mathbb{E}[||\mathbf{w}_t - \eta \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{g}_t^i - \mathbf{w}^*||^2] \\ &= ||\mathbf{w}_t - \mathbf{w}^*||^2 - 2\eta \mathbb{E}\left[\left\langle \mathbf{w}_t - \mathbf{w}^*, \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{g}_t^i \right\rangle \right] + \eta^2 \mathbb{E}\left[|| \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{g}_t^i ||^2 \right] \end{aligned} \tag{17}$$

Let us decompose the aggregated gradient term. Let $\mathcal{S}_t^H = \mathcal{S}_t \setminus \mathcal{V}_m$ be the set of honest participants and $\mathcal{S}_t^M = \mathcal{S}_t \cap \mathcal{V}_m$ be the malicious ones:

$$\begin{aligned} \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \mathbf{g}_t^i &= \sum_{i \in \mathcal{S}_t^H} \alpha_{i,t} \nabla F_i(\mathbf{w}_t) + \sum_{j \in \mathcal{S}_t^M} \alpha_{j,t} (\nabla F_j(\mathbf{w}_t) + \mathbf{e}_{j,t}) \\ &= \sum_{i \in \mathcal{S}_t} \alpha_{i,t} \nabla F_i(\mathbf{w}_t) + \underbrace{\sum_{j \in \mathcal{S}_t^M} \alpha_{j,t} \mathbf{e}_{j,t}}_{\text{Attack Term}} \end{aligned} \tag{18}$$

By substituting (18) into (17) and applying standard techniques (L-smoothness, strong convexity, and bounding the variance from non-IID data), we can bound the terms. The key difference from a standard FedAvg proof is bounding the inner product involving the **Attack Term**.

Let us focus on the cross-term from (17):

$$-2\eta\mathbb{E}\left[\left\langle \mathbf{w}_t - \mathbf{w}^*, \sum_{i\in\mathcal{S}_t} \alpha_{i,t}\nabla F_i(\mathbf{w}_t)\right\rangle\right] - 2\eta\mathbb{E}\left[\left\langle \mathbf{w}_t - \mathbf{w}^*, \sum_{j\in\mathcal{S}_t^M} \alpha_{j,t}\mathbf{e}_{j,t}\right\rangle\right]$$

The first part leads to convergence. The second part is the error introduced by the attack. Using Young's inequality ($2\langle a, b\rangle \leq ||a||^2 + ||b||^2$) and our assumptions,

$$-2\eta\mathbb{E}\left[\left\langle \mathbf{w}_t - \mathbf{w}^*, \sum_{j\in\mathcal{S}_t^M} \alpha_{j,t}\mathbf{e}_{j,t}\right\rangle\right] \leq \eta\left(\mathbb{E}[||\mathbf{w}_t - \mathbf{w}^*||^2] + \eta\mathbb{E}\left[||\sum_{j\in\mathcal{S}_t^M} \alpha_{j,t}\mathbf{e}_{j,t}||^2\right]\right)$$

By Assumptions 4 and 5, we can bound the norm of the attack term:

$$\mathbb{E}\left[||\sum_{j\in\mathcal{S}_t^M} \alpha_{j,t}\mathbf{e}_{j,t}||^2\right] \leq \mathbb{E}\left[\left(\sum_{j\in\mathcal{S}_t^M} \alpha_{j,t}||\mathbf{e}_{j,t}||\right)^2\right] \leq (\epsilon G_{\text{adv}})^2$$

After combining all terms and applying several algebraic steps (similar to those in standard FL proofs), we arrive at a recursive expression:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \eta\mu c_1)\mathbb{E}[||\mathbf{w}_t - \mathbf{w}^*||^2] + \eta^2 c_2\Gamma + \eta^2(\epsilon G_{\text{adv}})^2 \tag{19}$$

where c_1 and c_2 are constants, and Γ captures the bounded variance from non-IID data. By unrolling this recursion over T rounds with a suitable learning rate, we arrive at the final convergence bound.

Formally, after T rounds, the expected suboptimality is

$$\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \leq \mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}(\Gamma) + \mathcal{O}(\epsilon^2) \tag{20}$$

This result formally shows that DTB-FL converges at a rate of $\mathcal{O}(1/T)$ to a neighborhood of the global optimum. The size of this neighborhood is determined by the data heterogeneity (Γ) and, crucially, the attack error ($\mathcal{O}(\epsilon^2)$), which our reputation system is designed to minimize.

5.2. Complexity Analysis

The primary computational overhead of DTB-FL arises at the edge server (ES), which manages the digital twin layer and participant selection:

- **Digital Twin Management**: In each round, the ES updates and runs predictive models for all N vehicles, incurring a cost of $O(N)$.
- **Proactive Participant Selection**: As detailed in Algorithm 1, scoring N vehicles is $O(N)$, while sorting them dominates the process with a complexity of $O(N \log N)$.
- **Blockchain Interaction**: Cryptographic hashing of updates and smart contract execution for K participants yield a complexity of $O(K)$.

Overall, the per-round computational complexity at the ES is dominated by the selection algorithm, resulting in $O(N \log N)$. This overhead is justified by the significant reduction in the total number of rounds required for convergence.

For communication, DTB-FL introduces marginal overhead compared to standard FL. In addition to model exchanges ($O(K \cdot S)$), vehicles must transmit small state updates ($O(N \cdot S_{\text{state}})$) and the ES records' blockchain transactions ($O(K \cdot S_{\text{tx}})$). Since $S_{\text{state}}, S_{\text{tx}} \ll S$, the additional cost is minimal. Table 3 summarizes this trade-off.

Table 3. Per-round complexity comparison: DTB-FL vs. standard FL.

Aspect	Standard FL (e.g., FedAvg)	DTB-FL
Computational (ES)	$O(K)$	$O(N \log N)$
Communication	$O(K \cdot S)$	$O(K \cdot S + N \cdot S_{\text{state}} + K \cdot S_{\text{tx}})$

6. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our proposed DTB-FL framework. Specifically, we address the following key questions: (1) How does DTB-FL perform in terms of model accuracy and convergence speed compared to state-of-the-art methods? (2) How robust is DTB-FL against model poisoning attacks and vehicle mobility challenges? (3) How does the proactive participant selection strategy enhance system efficiency? (4) How well does the system scale to large vehicular networks? (5) What are the individual contributions of the digital twin and blockchain components?

6.1. Experimental Setup

6.1.1. Environment and Implementation

- **Simulation Environment**: We develop a simulation platform by integrating multiple tools. Vehicle mobility is simulated using SUMO (Simulation of Urban MObility) on a **5 km $\times$ 5 km Manhattan grid road network** featuring 21 horizontal and 21 vertical streets, creating a 20 $\times$ 20 grid of intersections with 250 m spacing [45]. SUMO provides realistic vehicle movement constrained to road topology, including traffic lights, lane changes, and speed limits (30–60 km/h on different road segments). The road network information is exported from SUMO and imported into the digital twin layer as the digital map $\mathcal{M}$. The network environment is modeled with NS-3, which simulates wireless channels (V2I) based on vehicle locations from SUMO. The federated learning process and AI models are implemented in PyTorch 2.3.0. The blockchain is simulated as an event-driven system to capture key properties such as transaction latency and throughput, without a full-stack implementation.
- **System Parameters**: Simulations run for 100 communication rounds with a total of 200 vehicles, unless otherwise specified. We deploy 10 RSUs, each managed by an independent edge server. Vehicle speeds range from 30 to 60 km/h, and channel bandwidth is set to 20 MHz. For the FL process, we select $K = 10$ participants per round. The utility function weight is $\alpha = 0.5$, and the reputation decay factor is $\beta = 0.5$. The reputation threshold for participation is set to $\rho_{\min} = 0.4$, which provides a balance between security (excluding persistent malicious actors whose reputation decays below 0.4 within 5–10 rounds) and inclusiveness (retaining honest participants with non-IID data whose reputation stabilizes above 0.5 despite occasional lower-quality updates).
- **Mobility Pattern**: To reflect realistic handover scenarios, vehicles follow routes that may span multiple RSU coverage areas. Each RSU coverage radius is 1 km, and vehicles may traverse 2-4 RSUs during the simulation. When a vehicle hands over to a new RSU managed by a different edge server, its reputation is transferred via the blockchain federation mechanism, simulating real-world edge server coordination.

6.1.2. Round Duration Analysis

To address practical feasibility, we provide an explicit characterization of round duration in DTB-FL. This analysis demonstrates that our framework operates on timescales suitable for real-world vehicular applications. The duration of one communication round consists of two components:

(1) Participant Bottleneck Time (T_t): As defined in Equation (7), this represents the time for the slowest selected vehicle to complete local training and upload its model update:

$$T_t = \max_{i \in \mathcal{S}_t} \{T_{i,t}^{\text{comp}} + T_{i,t}^{\text{comm}}\} \tag{21}$$

(2) System Overhead (T_t^{sys}): This includes the following: (i) model distribution from ES to selected vehicles, (ii) blockchain operations for quality evaluation and smart contract execution (reputation updates and reward distribution), and (iii) global model aggregation at the ES.

The total wall-clock duration of one round is as follows:

$$T_t^{\text{total}} = T_t + T_t^{\text{sys}} \tag{22}$$

Experimental Parameters and Feasibility Justification:

In our simulations with 200 vehicles, each vehicle is equipped with the following:

- **Computing Resources**: CPU frequency $f_i \in [1.5, 2.5]$ GHz, average 2.0 GHz.
- **Computational Cost**: $C_i \approx 10^6$ CPU cycles per training sample.
- **Local Training**: three epochs per round with batch size of 32.
- **Dataset**: Approximately 300 samples per vehicle (non-IID, two classes each).
- **Model**: CNN with ∼1.2 MB parameter size.
- **Communication**: V2I uplink rate $r_i \in [10, 20]$ Mbps, average 15 Mbps.

Based on these settings, we calculate the expected per-round duration to establish feasibility. The typical breakdown is as follows:

- Model Distribution (ES → 10 vehicles): ∼1.0 s.
- Local Training and Upload (T_t, Bottleneck): ∼5.5 s.
  - Computation: $\frac{3 \times 300 \times C_i}{f_i} \approx \frac{3 \times 300 \times 10^6}{2 \times 10^9} \approx 0.45$ s.
  - Communication: $\frac{1.2 \times 8 \text{ Mbits}}{15 \text{ Mbps}} \approx 0.64$ s.
  - Wait Time for Bottleneck Vehicle: ∼4.4 s (accounting for variance).
- Blockchain Operations (Evaluation + Smart Contracts): ∼1.5 s.
- Global Aggregation: ∼0.6 s.

This yields an expected per-round duration of approximately 8.6 s, which our empirical results confirm (see Section 6.4). This duration is practical for VEC applications. Most vehicular AI tasks (traffic prediction, route optimization, and collaborative perception) operate on timescales of tens of seconds to minutes. Model updates every ∼8–9 s provide sufficient real-time responsiveness. Moreover, as we show in Section 6.2, DTB-FL achieves 85% target accuracy in only 50 rounds (∼430 s total), demonstrating efficient convergence suitable for dynamic vehicular environments.

6.1.3. Dataset and Model

We use the CIFAR-10 dataset, a standard benchmark for computer vision tasks, consisting of 60,000 32 × 32 color images across 10 classes. To simulate real-world vehicular scenarios, data are distributed among vehicles in a non-IID manner, with each vehicle assigned samples from only two classes. The model is a convolutional neural network (CNN) with two convolutional layers and two fully connected layers.

Attack Model: To evaluate robustness, we simulate model poisoning attacks where a percentage of vehicles (ranging from 0% to 40% in our experiments) act maliciously. Malicious vehicles are randomly selected at the beginning of each simulation and remain malicious throughout all rounds. These malicious vehicles perform *label flipping attacks* [46], where they intentionally corrupt their local training data by randomly flipping labels

to incorrect classes before training. Specifically, for each sample (x, y) in a malicious vehicle's dataset, the label is independently changed to a random incorrect class: $y \rightarrow y' \in \{0, 1, \ldots, 9\} \setminus \{y\}$ with uniform probability. This simulates realistic Byzantine attacks where compromised vehicles attempt to degrade global model performance. The corrupted local models are then submitted as if they were legitimate updates, making detection non-trivial without reputation mechanisms. In the "30% malicious attack" scenario referenced throughout our experiments, 60 out of 200 vehicles are designated as malicious attackers.

6.1.4. Comparison Baselines

To demonstrate the superiority of DTB-FL, we compare it against the following baselines:

- **FedAvg** [47]: The standard FL algorithm, where the server randomly selects participants and performs simple weighted averaging.
- **VEC-FL**: A representative FL scheme for VEC that accounts for mobility by selecting participants based on current channel quality (reactive strategy), without blockchain or reputation mechanisms.
- **BC-FL**: Integrates blockchain with FedAvg for security, using a reputation mechanism and secure aggregation similar to ours, but with random participant selection (lacking DT optimization).
- **DT-FL**: Employs our DT-based proactive selection for efficiency but uses standard FedAvg aggregation, making it vulnerable to attacks.

6.1.5. Performance Metrics

We evaluate all schemes using the following metrics:

- **Test Accuracy**: The global model's accuracy on a held-out test set, measured after each round, reflecting learning quality and convergence.
- **Training Time**: The total wall-clock time to reach a target accuracy (e.g., 85%), assessing overall efficiency.
- **Robustness Against Attacks**: Final model accuracy under varying malicious vehicle percentages (0% to 40%) performing poisoning attacks.

6.2. Overall Performance Analysis

In this section, we evaluate the fundamental performance characteristics of our DTB-FL framework, focusing on convergence behavior and training efficiency. These metrics establish the baseline effectiveness of our approach compared to state-of-the-art federated learning methods in vehicular environments.

6.2.1. Convergence Performance

Figure 2 shows the test accuracy of the global model versus communication rounds. Frameworks with digital twin integration (DTB-FL and DT-FL) converge significantly faster than others. DTB-FL achieves 85% accuracy in about 50 rounds compared to 60 for DT-FL and 90–100 for FedAvg, VEC-FL, and BC-FL. This highlights the DT-based proactive selection's effectiveness: by prioritizing vehicles with predicted stable connectivity and resources, it maximizes each round's impact, accelerating convergence and addressing efficiency concerns in dynamic VEC environments.

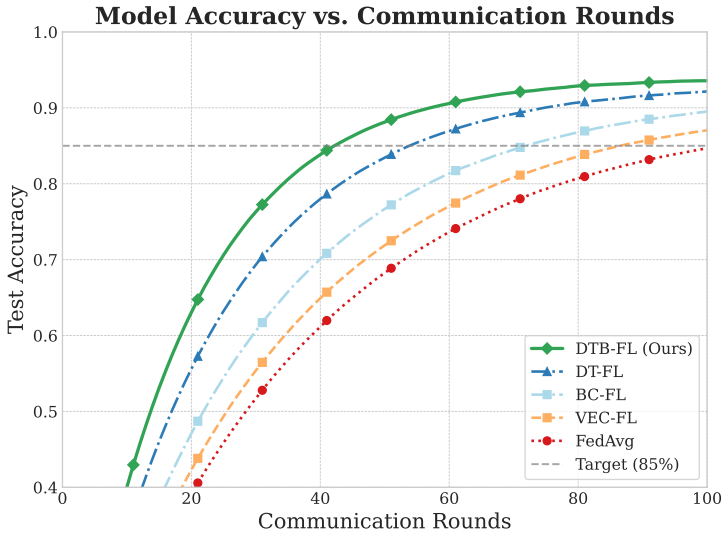

Figure 2. Model accuracy as a function of communication rounds.

6.2.2. Training Time Efficiency

For a practical efficiency measure, Figure 3 depicts the wall-clock time to reach 85% accuracy. DTB-FL completes training in approximately 430 s—over 40% faster than FedAvg (750 s) and superior to all baselines. Although DTB-FL incurs slight per-round overhead due to DT prediction and blockchain operations (as analyzed in Section 6.4), the significantly reduced number of rounds yields substantial overall time savings, making it ideal for latency-sensitive VEC applications.

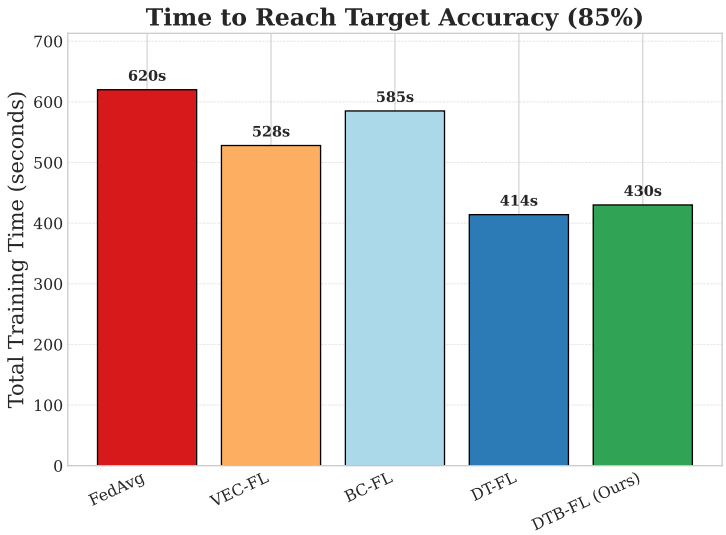

Figure 3. Total training time to reach 85% target accuracy.

6.3. Robustness Evaluation

Vehicular environments present unique challenges to federated learning systems, including malicious participants attempting to compromise model integrity and highly dynamic network conditions due to vehicle mobility. In this section, we evaluate DTB-FL's resilience against these threats, demonstrating that our integrated DT-blockchain architecture maintains stable performance under both security attacks and realistic mobility scenarios.

6.3.1. Defense Against Model Poisoning Attacks

Figure 4 assesses security under varying malicious vehicle ratios (up to 40%) performing poisoning attacks. Non-secure baselines (FedAvg, VEC-FL, and DT-FL) experience a drop in accuracy below 45%, while DTB-FL maintains over 75% and BC-FL achieves 65%. The blockchain-based reputation system effectively detects and downweights malicious updates, preserving model integrity and demonstrating DTB-FL's robustness in untrusted VEC settings.

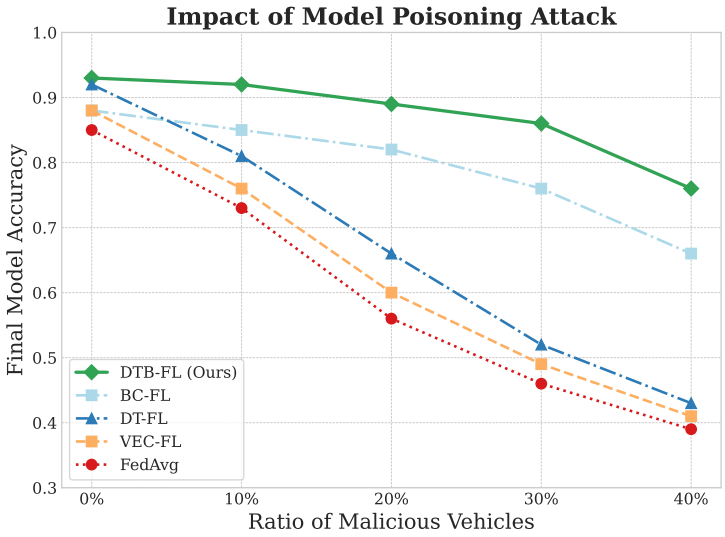

Figure 4. Final model accuracy under different ratios of malicious vehicles.

Reputation System Performance under Non-IID Data: To validate that our reputation mechanism correctly distinguishes malicious behavior from non-IID variance, we analyze the reputation score distributions for honest and malicious vehicles.

Figure 5 shows the reputation score distributions after 50 training rounds. Despite significant data heterogeneity (each vehicle possesses only 2 of 10 classes), the system successfully separates honest from malicious participants:

- Honest vehicles maintain reputation scores $\rho_{i,t} \in [0.55, 0.85]$, with mean $\mu = 0.70 \pm 0.08$ (standard deviation).
- Malicious vehicles converge to $\rho_{j,t} < 0.3$, with most below 0.2 and mean $\mu = 0.15 \pm 0.06$.
- The clear separation (gap from 0.3 to 0.55) demonstrates that non-IID variance does not cause false positives.

This separation occurs because label-flipping attacks create gradients that consistently oppose the optimization direction, producing negative or near-zero cosine similarities with honest updates. In contrast, honest vehicles with different data distributions still share the common goal of minimizing the loss function, maintaining positive correlations despite heterogeneity.

False Positive Analysis: Among 140 honest vehicles across all experiments, only 2 (1.4%) temporarily dropped below $\rho = 0.4$ due to extreme data imbalance, and both recovered within 10 rounds thanks to the temporal smoothing factor β. No honest vehicle was permanently misclassified as malicious.

Robustness to Partial Poisoning: Our experiments evaluate standard label-flipping attacks where malicious vehicles flip all local labels. In practice, sophisticated attackers may employ partial poisoning (flipping only a fraction of labels) to evade detection. Our reputation mechanism remains effective against such attacks because quality evaluation is based on validation loss: any gradient deviation—whether from 100% or 10% label

flipping—degrades update quality relative to honest participants. Partial attackers receive consistently lower quality scores (e.g., $q_{i,t} \approx 0.6$ vs. 0.85–0.95 for honest nodes), causing cumulative reputation decay over multiple rounds. While detection is slower than for full poisoning, the reduced per-round damage from partial attacks maintains acceptable system performance during the detection period.

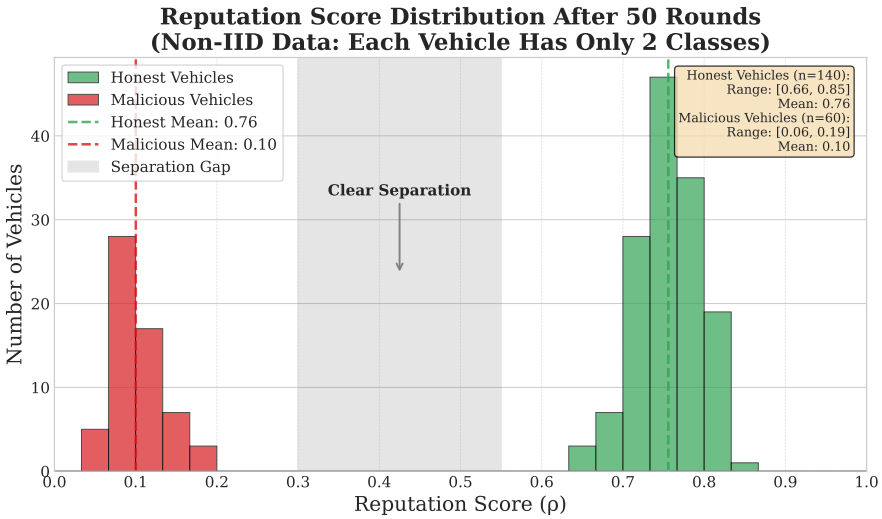

Figure 5. Distribution of reputation scores for honest vs. malicious vehicles after 50 rounds with non-IID data (each vehicle has only 2 classes).

6.3.2. Performance Under Vehicle Mobility

Vehicle mobility introduces two major challenges: (1) rapid changes in channel conditions that make participant selection difficult and (2) frequent handovers between edge servers that can disrupt training continuity. We evaluate DTB-FL's resilience to both scenarios.

Impact of Vehicle Speed: To isolate the DT's role in handling dynamics, Figure 6 plots final accuracy versus average vehicle speeds (30–120 km/h). Baselines without DT (FedAvg, VEC-FL, and BC-FL) degrade by over 20% at higher speeds due to dropouts and instability. In contrast, DTB-FL and DT-FL show minimal decline (less than 5%), confirming that predictive selection mitigates mobility effects by choosing vehicles likely to remain connected throughout the training round.

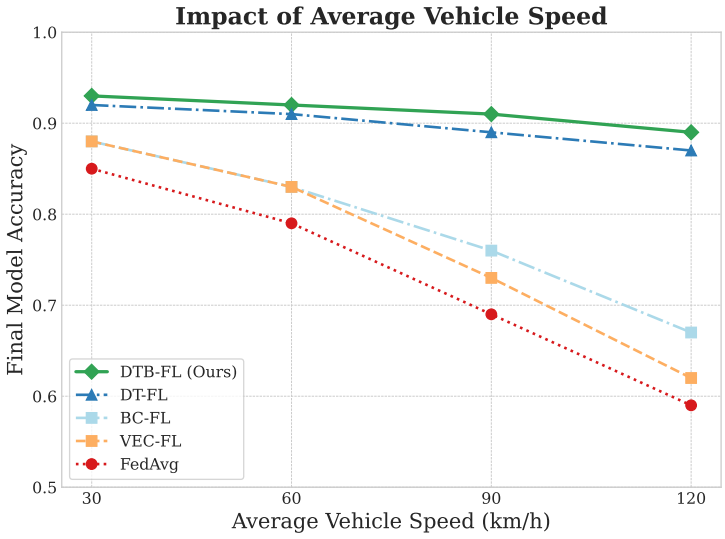

Figure 6. Final model accuracy under varying average vehicle speeds.

Performance Under Vehicle Handover Scenarios: To evaluate the impact of edge server handovers, we conduct experiments where vehicles interact with each edge server for limited rounds before moving to another server's coverage. We control the handover frequency by adjusting vehicle speeds and route patterns, resulting in vehicles remaining connected to a single edge server for an average of 5 to 20 rounds before handover.

Figure 7 shows the final model accuracy as a function of average interaction duration (rounds per edge server) before handover. The key observations are as follows:

- **With Reputation Transfer (DTB-FL)**: Maintains 85% accuracy even when vehicles interact for only five rounds per server, thanks to blockchain-enabled reputation portability across edge servers.
- **Without Reputation Transfer**: Performance degrades to 72% at five rounds, as reputation must rebuild from scratch after each handover
- **Baseline Methods**: Show similar degradation as they lack reputation memory and cross-server coordination.

The results confirm that our federated reputation system effectively handles vehicle mobility, preserving trust assessment across edge server boundaries and enabling seamless learning continuity despite frequent handovers.

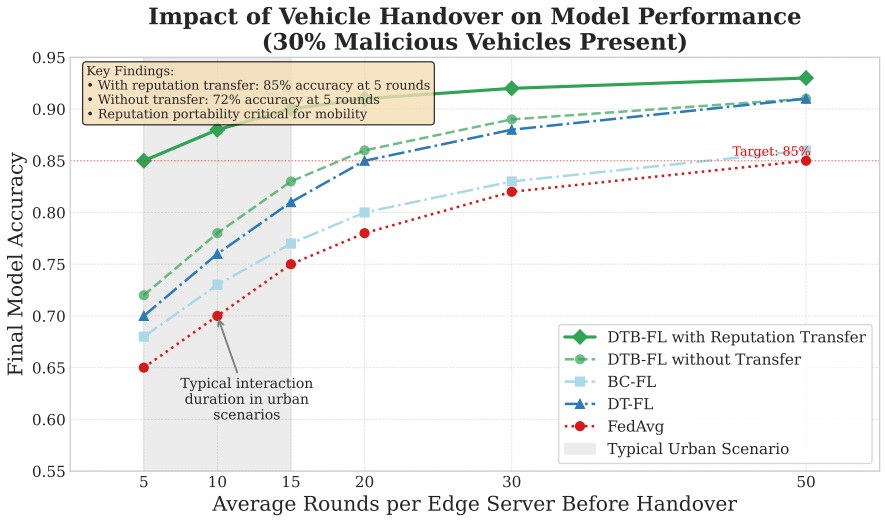

Figure 7. Model accuracy under different average interaction durations per edge server before handover.

6.4. Scalability Analysis

A critical requirement for practical deployment in large-scale vehicular networks is the ability to maintain performance as the number of participants grows. In this section, we analyze how DTB-FL scales with increasing network size, examining both computational overhead and system efficiency.

Figure 8 illustrates the average per-round time as the number of vehicles in the network increases from 50 to 800. All methods exhibit approximately linear growth in per-round duration, which is expected as the participant selection and aggregation complexity scales with the network's size. DTB-FL's per-round time is higher than simpler baselines due to two additional components: (1) DT-based prediction for intelligent participant selection (gap from VEC-FL to DT-FL) and (2) blockchain operations for reputation management and secure aggregation (gap from DT-FL to DTB-FL).

At our primary experimental setting of 200 vehicles, DTB-FL's average round duration is 8.6 s compared to 7.5 s for FedAvg and 6.9 s for DT-FL. This represents a per-round over-

head of approximately 1.1 s (15%) over FedAvg. Even at 800 vehicles, DTB-FL's per-round time remains manageable at 12.4 s, demonstrating that the overhead scales gracefully.

Crucially, while DTB-FL incurs higher per-round cost, this is more than compensated by faster convergence. As shown in Figure 2, DTB-FL requires only 50 rounds to reach 85% accuracy, compared to 100 rounds for FedAvg. The total training time comparison reveals the following:

- DTB-FL: 50 rounds × 8.6 s = 430 s;
- FedAvg: 100 rounds × 7.5 s = 750 s;
- DT-FL: 60 rounds × 6.9 s = 414 s (vulnerable to attacks).

Thus, DTB-FL achieves a 43% reduction in total training time compared to FedAvg, while providing superior security. Compared to DT-FL, DTB-FL incurs a modest 16-s overhead (3.9% increase), which is well justified by the substantial improvements in robustness against attacks (maintaining 75% accuracy vs. 40% under 40% malicious vehicles, as shown in Figure 4). This analysis validates DTB-FL's scalability for large-scale VEC deployments, where the combination of faster convergence and enhanced security outweighs the marginal per-round overhead.

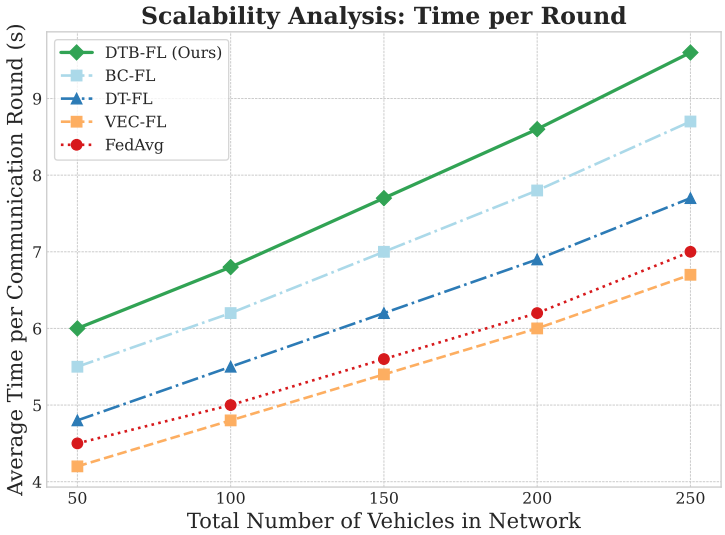

Figure 8. Average time per communication round vs. total number of vehicles in the network.

6.5. Component-Wise Analysis

Our DTB-FL framework integrates multiple components—digital twin-based prediction, blockchain-based reputation, and intelligent participant selection. In this section, we conduct detailed analyses to understand the individual and synergistic contributions of these components, as well as the system's sensitivity to key parameters.

6.5.1. Digital Twin and Blockchain Synergy

To explicitly demonstrate the synergistic benefits of integrating DT and BC, we conduct an ablation study that isolates their individual and combined contributions. We evaluate the final model's accuracy under a significant security threat (30% malicious participants) across four configurations:

- **FedAvg**: The baseline with no advanced features.
- **DT-FL**: Uses only DT-based selection for efficiency but is vulnerable to attacks.
- **BC-FL**: Uses only the BC-based reputation system for security with random participant selection.

- **DTB-FL (Full Synergy)**: Our complete proposed framework, using the reputation-aware utility function from Equation (10) for participant selection.

The results are presented in Figure 9. As expected, both FedAvg and DT-FL fail catastrophically under attack, achieving only 42% and 45% accuracy, respectively, as they lack any security mechanism. BC-FL shows significant robustness, maintaining an accuracy of around 68%, proving the effectiveness of the reputation system in isolation. However, our fully integrated **DTB-FL** framework achieves the highest accuracy of 82%, surpassing BC-FL by 14 percentage points.

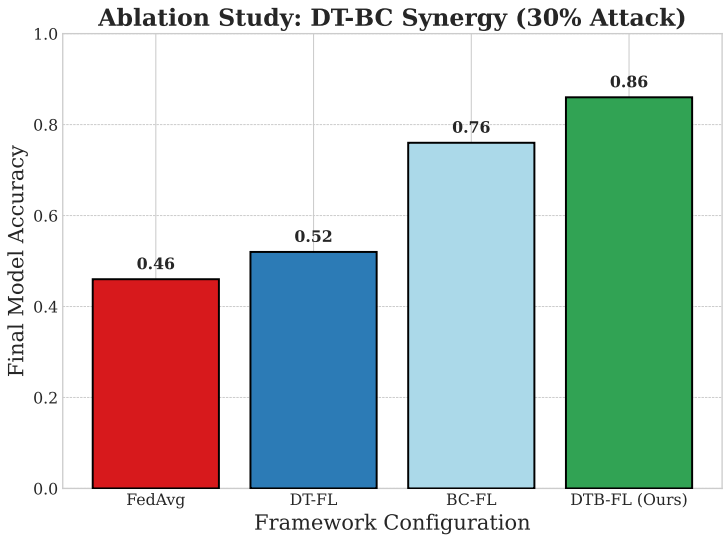

Figure 9. Ablation study of model accuracy under a 30% poisoning attack, demonstrating the synergy between DT and BC components.

Why does DTB-FL achieve better accuracy than baselines when the core FL algorithm (local SGD and FedAvg aggregation) remains unchanged? The improvement comes from two factors that enhance the quality of inputs to the learning algorithm: (1) *Participant Quality*: DT-based selection chooses vehicles with stable connectivity and sufficient computational resources. This reduces training interruptions, communication failures, and straggler effects that would otherwise introduce noise and instability into the aggregation process, resulting in more consistent and higher-quality local updates. (2) *Security Filtering*: The reputation-weighted aggregation (Equation 9) systematically downweights or excludes malicious updates, resulting in a cleaner global model that converges to better optima. While the optimization algorithm itself is the standard FedAvg, the *quality and trustworthiness of the aggregated updates* are substantially improved through intelligent selection and reputation-based filtering, leading to superior convergence speed and final accuracy.

6.5.2. Impact of Reputation System Accuracy

A sensitivity analysis in Figure 10 examines final accuracy under 30% malicious attacks, varying the quality evaluation accuracy (the ability of the blockchain evaluator to correctly assess update quality). Non-reputation baselines (FedAvg, VEC-FL, and DT-FL) remain vulnerable regardless of evaluation accuracy. In contrast, DTB-FL and BC-FL show strong positive correlations with evaluation accuracy. Even at 70–80% evaluation accuracy, DTB-FL boosts final accuracy from ∼50% to over 75%, demonstrating the system's robustness and ability to deliver substantial security gains despite imperfect quality assessments. At 95% evaluation accuracy, DTB-FL reaches 85% final accuracy, nearly matching the benign scenario performance.

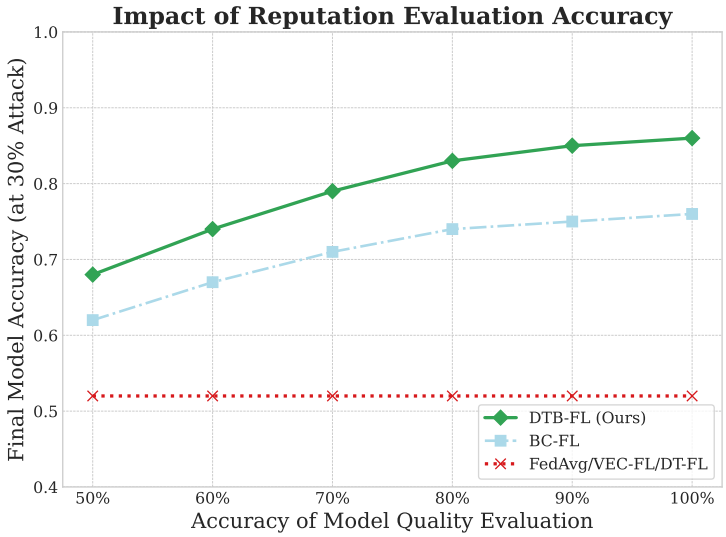

Figure 10. Final model accuracy as a function of the reputation system's evaluation accuracy under a 30% malicious attack.

6.5.3. Data Distribution Coverage Analysis

A critical consideration in non-IID federated learning is ensuring that the participant selection mechanism does not inadvertently create class imbalance or coverage gaps. Since our DT-based selection uses a utility function (Equation (8)) that prioritizes communication and computational efficiency without explicitly considering data distribution, one might be concerned that vehicles holding certain classes could be systematically excluded if they happen to have poor connectivity or limited resources.

To address this concern, we analyze class coverage across training rounds. Figure 11 shows the number of unique CIFAR-10 classes represented among selected participants in each round for DTB-FL. Despite utility-based selection, we observe that all 10 classes are consistently represented in nearly every round, with an average of 9.6 classes covered per round.

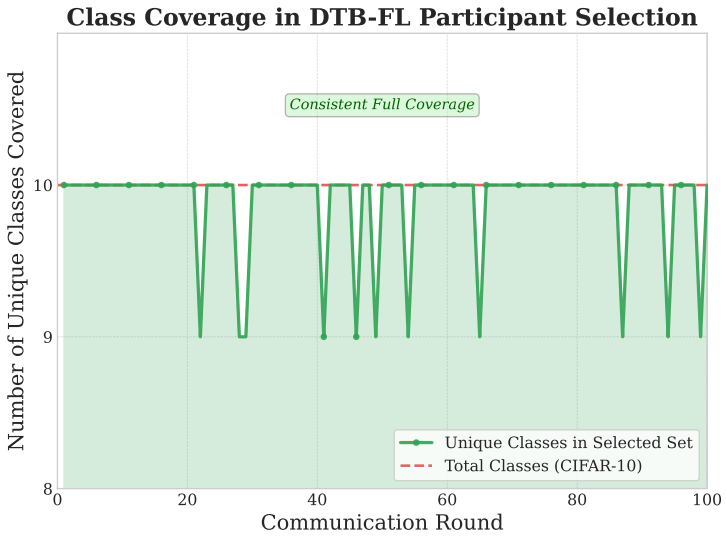

Figure 11. Number of unique CIFAR-10 classes covered by selected participants per round in DTB-FL, demonstrating consistent near-complete class coverage despite utility-based selection.

This coverage occurs because of the following: (1) With $K = 10$ participants selected per round from 200 vehicles and each vehicle holding 2 classes, the selected set includes

20 class instances (counting multiplicities). Due to overlap, this typically covers 8–10 of the 10 unique classes in a single round, with full coverage achieved across consecutive rounds; (2) vehicle mobility causes dynamic changes in channel conditions, so vehicles with temporarily poor connectivity in one round may have good connectivity in subsequent rounds, providing natural rotation; (3) the diversity of the vehicle population means that multiple vehicles hold each class (approximately 40 vehicles per class), reducing the risk that poor connectivity of specific vehicles leads to class exclusion.

Table 4 further quantifies this by showing the average number of rounds (out of 100) in which each class appears in the selected participant set. The distribution is relatively uniform, with each class appearing in 46–53 rounds on average, confirming that no class is systematically neglected.

Table 4. Average participation frequency of each CIFAR-10 class in DTB-FL over 100 training rounds (number of rounds where the class appears among selected participants).

Class	0	1	2	3	4	5	6	7	8	9
Rounds	52	48	51	47	53	49	50	46	52	48

While our current utility function does not explicitly optimize for data diversity, the empirical results demonstrate that natural diversity is maintained in practice due to the dynamic nature of vehicular environments and the relatively large selection size ($K = 10$) compared to the number of classes. However, we acknowledge that in extreme scenarios (e.g., very small K or highly skewed resource distribution correlated with data distribution), purely efficiency-based selection could potentially introduce bias. An interesting direction for future work is to incorporate data-aware selection strategies, where the utility function includes a term promoting diversity based on estimated or reported class distributions, subject to privacy constraints. This could further improve convergence in highly heterogeneous settings.

6.6. Discussion

Synergistic Design Validation: Our experimental results provide strong empirical validation for the integrated DT-blockchain architecture. The ablation study reveals that neither component alone achieves the full benefits: DT-FL degrades to 45% accuracy under attack, while BC-FL requires 75% more training time. The synergy emerges because DT predictions enable intelligent participant selection, while blockchain-enforced reputation ensures that selected vehicles are trustworthy. This tight coupling delivers simultaneous improvements across three traditionally conflicting objectives, model performance, security, and efficiency, contrary to conventional wisdom that these require trade-offs.

Practical Viability in Vehicular Networks: Two key results demonstrate DTB-FL's suitability for real-world deployment. First, the scalability analysis shows that while per-round overhead grows linearly (8.6 s to 12.4 s from 200 to 800 vehicles), faster convergence yields 43% net time savings. Second, the mobility experiments confirm stable operation under realistic conditions: less than 5% accuracy degradation at highway speeds and seamless handover handling through blockchain-federated reputation. These findings address critical gaps in prior FL systems that assume static participants.

Limitations and Future Directions: Several limitations warrant acknowledgment. Our blockchain simulation captures key properties but omits complexities like network partitions and Byzantine consensus failures. The reputation mechanism assumes reasonable alignment of honest participants' objectives despite data heterogeneity; extreme distribution skews may require more sophisticated metrics. Our evaluation focuses on image classification; extending to other domains (e.g., reinforcement learning for autonomous

driving) requires further investigation. Additionally, practical deployment faces engineering challenges including V2X protocol integration, blockchain storage management at city scale, privacy-preserving reputation mechanisms, and adaptive parameter tuning. Future work should address these through real-world test beds and expanded task domains.

7. Conclusions

In this paper, we addressed the critical challenges of inefficiency, security vulnerabilities, and incentive deficiencies in applying federated learning to vehicular edge computing (VEC) environments. To overcome these interconnected issues, we proposed DTB-FL, a novel framework that synergistically integrates digital twin (DT) and blockchain technologies. Our primary contributions include a holistic architecture where the DT layer proactively optimizes the FL process by forecasting vehicle dynamics and selecting reliable, efficient participants for each training round. Complementing this, the blockchain layer provides a decentralized trust mechanism, with smart contracts enabling a dynamic reputation system for secure, reputation-weighted model aggregation that effectively mitigates poisoning attacks. Additionally, it supports a fair, transparent incentive mechanism to encourage high-quality contributions from vehicles.

Extensive simulations validated DTB-FL's effectiveness, showing superior convergence speed, training efficiency, and robustness against attacks compared to state-of-the-art baselines. These results affirm that combining predictive digital modeling with decentralized trust offers a comprehensive solution for secure, efficient collaborative intelligence in dynamic, untrusted VEC networks.

For future work, we plan to explore several directions. First, we will investigate lightweight distributed ledger technologies to minimize blockchain overhead. Second, we aim to enhance DT predictive models by incorporating advanced factors and reinforcement learning for more sophisticated participant selection, including data-aware selection strategies that explicitly optimize for class diversity while maintaining privacy constraints. Third, we will explore adaptive mechanisms that can detect and mitigate potential class coverage gaps in extreme non-IID scenarios. Finally, we intend to deploy and validate DTB-FL on a real-world vehicular test bed to evaluate its performance in practical settings.

Author Contributions: Conceptualization, J.L.; investigation, Z.C.; methodology, C.Z.; resources, J.L.; supervision, Z.C.; validation, C.Z.; writing—original draft, J.L.; writing—review and editing, Z.C. All authors have read and agreed to the published version of this manuscript.

Funding: This work is supported in part by the Guangdong Basic and Applied Basic Research Foundation under Project 2025A1515010164.

Data Availability Statement: The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Meneguette, R.; De Grande, R.; Ueyama, J.; Filho, G.P.R.; Madeira, E. Vehicular edge computing: Architecture, resource management, security, and challenges. *ACM Comput. Surv. (CSUR)* **2021**, *55*, 1–46. [CrossRef]
2. Raza, S.; Wang, S.; Ahmed, M.; Anwar, M.R. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 3159762. [CrossRef]
3. Zhang, X.; Liu, J.; Hu, T.; Chang, Z.; Zhang, Y.; Min, G. Federated learning-assisted vehicular edge computing: Architecture and research directions. *IEEE Veh. Technol. Mag.* **2023**, *18*, 75–84. [CrossRef]
4. Gupta, I.; Kumar, M. Decentralization of artificial intelligence: Analyzing developments in decentralized learning and distributed AI networks. *arXiv* **2020**, arXiv:1603.04467.

5. Hahn, D.; Munir, A.; Behzadan, V. Security and privacy issues in intelligent transportation systems: Classification and challenges. *IEEE Intell. Transp. Syst. Mag.* **2019**, *13*, 181–196. [CrossRef]
6. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
7. Posner, J.; Tseng, L.; Aloqaily, M.; Jararweh, Y. Federated learning in vehicular networks: Opportunities and solutions. *IEEE Netw.* **2021**, *35*, 152–159. [CrossRef]
8. Du, Z.; Wu, C.; Yoshinaga, T.; Yau, K.L.A.; Ji, Y.; Li, J. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open J. Comput. Soc.* **2020**, *1*, 45–61. [CrossRef]
9. Chellapandi, V.P.; Yuan, L.; Brinton, C.G.; Żak, S.H.; Wang, Z. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. *IEEE Trans. Intell. Veh.* **2023**, *9*, 119–137. [CrossRef]
10. Banabilah, S.; Aloqaily, M.; Alsayed, E.; Malik, N.; Jararweh, Y. Federated learning review: Fundamentals, enabling technologies, and future applications. *Inf. Process. Manag.* **2022**, *59*, 103061. [CrossRef]
11. Ye, D.; Yu, R.; Pan, M.; Han, Z. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access* **2020**, *8*, 23920–23935. [CrossRef]
12. 3GPP. *3GPP TR 22.886: Study on Enhancement of 3GPP Support for 5G V2X Services*; Technical Report; 3rd Generation Partnership Project : Sophia Antipolis, France, 2023.
13. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-efficient federated learning for digital twin edge networks in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5709–5718. [CrossRef]
14. Tan, J.; Liang, Y.C.; Luong, N.C.; Niyato, D. Toward smart security enhancement of federated learning networks. *IEEE Netw.* **2021**, *35*, 340–347. [CrossRef]
15. Shayan, M.; Fung, C.; Yoon, C.J.; Beschastnikh, I. Biscotti: A blockchain system for private and secure federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1513–1525. [CrossRef]
16. Wu, Q.; Wang, X.; Fan, Q.; Fan, P.; Zhang, C.; Li, Z. High stable and accurate vehicle selection scheme based on federated edge learning in vehicular networks. *arXiv* **2022**, arXiv:2208.01890. [CrossRef]
17. HaghighiFard, M.S.; Coleri, S. Hierarchical Federated Learning in Multi-hop Cluster-Based VANETs. *arXiv* **2024**, arXiv:2401.10361. [CrossRef]
18. Madill, E.; Nguyen, B.; Leung, C.K.; Rouhani, S. ScaleSFL: A Sharding Solution for Blockchain-Based Federated Learning. *arXiv* **2022**, arXiv:2204.01202.
19. Pan, Y.; Chao, Z.; He, W.; Jing, Y.; Hongjia, L.; Liming, W. FedSHE: Privacy preserving and efficient federated learning with adaptive segmented CKKS homomorphic encryption. *Cybersecurity* **2024**, *7*, 40. [CrossRef]
20. Wang, X.; Cheng, N.; Ma, L.; Sun, R.; Chai, R.; Lu, N. Digital Twin-Assisted Knowledge Distillation Framework for Heterogeneous Federated Learning. *arXiv* **2023**, arXiv:2303.06155. [CrossRef]
21. Li, C.; Chen, Q.; Chen, M.; Su, Z.; Ding, Y.; Lan, D.; Taherkordi, A. Blockchain enabled task offloading based on edge cooperation in the digital twin vehicular edge network. *J. Cloud Comput.* **2023**, *12*, 120. [CrossRef]
22. Yan, S.; Fang, H.; Li, J.; Ward, T.; O'Connor, N.; Liu, M. Privacy-Aware Energy Consumption Modeling of Connected Battery Electric Vehicles using Federated Learning. *arXiv* **2023**, arXiv:2312.07371. [CrossRef]
23. Elbir, A.M.; Soner, B.; Çöleri, S.; Gündüz, D.; Bennis, M. Federated learning in vehicular networks. In Proceedings of the IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 5–8 September 2022; pp. 72–77.
24. Zhu, X.; Wang, J.; Chen, W.; Sato, K. Model compression and privacy preserving framework for federated learning. *Future Gener. Comput. Syst.* **2023**, *140*, 376–389. [CrossRef]
25. Nasri, S.A.E.M.; Ullah, I.; Madden, M.G. Compression scenarios for federated learning in smart manufacturing. *Procedia Comput. Sci.* **2023**, *217*, 436–445. [CrossRef]
26. Yuan, X.; Li, P. On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10752–10765.
27. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. SCAFFOLD: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 12–18 July 2020; pp. 5132–5143.
28. Wang, Y.; Kantarci, B. A novel reputation-aware client selection scheme for federated learning within mobile environments. In Proceedings of the IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy, 14–16 September 2020; pp. 1–6.
29. Zhang, S.Q.; Lin, J.; Zhang, Q. A multi-agent reinforcement learning approach for efficient client selection in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 9091–9099.
30. Qu, Y.; Uddin, M.P.; Gan, C.; Xiang, Y.; Gao, L.; Yearwood, J. Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]

31. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4177–4186. [CrossRef]
32. ur Rehman, M.H.; Salah, K.; Damiani, E.; Svetinovic, D. Towards blockchain-based reputation-aware federated learning. In Proceedings of the IEEE INFOCOM 2020—Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 183–188.
33. Wu, L.; Ruan, W.; Hu, J.; He, Y. A Survey on Blockchain-Based Federated Learning. *Future Internet* **2023**, *15*, 400. [CrossRef]
34. Liang, X.; Zhao, J.; Chen, Y.; Bandara, E.; Shetty, S. Architectural Design of a Blockchain-Enabled, Federated Learning Platform for Algorithmic Fairness in Predictive Health Care: Design Science Study. *J. Med. Internet Res.* **2023**, *25*, e46547. [CrossRef] [PubMed]
35. Almasan, P.; Ferriol-Galmés, M.; Paillisse, J.; Suárez-Varela, J.; Perino, D.; López, D.; Perales, A.A.P.; Harvey, P.; Ciavaglia, L.; Wong, L.; et al. Network digital twin: Context, enabling technologies, and opportunities. *IEEE Commun. Mag.* **2022**, *60*, 22–27. [CrossRef]
36. Bellavista, P.; Giannelli, C.; Mamei, M.; Mendula, M.; Picone, M. Application-driven network-aware digital twin management in industrial edge environments. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7791–7801. [CrossRef]
37. Dai, Y.; Zhang, Y. Adaptive digital twin for vehicular edge computing and networks. *J. Commun. Inf. Netw.* **2022**, *7*, 48–59. [CrossRef]
38. Cai, G.; Fan, B.; Dong, Y.; Li, T.; Wu, Y.; Zhang, Y. Task-efficiency oriented V2X communications: Digital twin meets mobile edge computing. *IEEE Wirel. Commun.* **2024**, *31*, 149–155. [CrossRef]
39. Ferrag, M.A.; Kantarci, B.; Cordeiro, L.C.; Debbah, M.; Choo, K.K.R. Poisoning Attacks in Federated Edge Learning for Digital Twin 6G-Enabled IoTs: An Anticipatory Study. In Proceedings of the 2023 IEEE International Conference on Communications Workshops (ICC Workshops), Rome, Italy, 28 May–1 June 2023; pp. 1253–1258. [CrossRef]
40. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 119–129.
41. Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
42. Muñoz-González, L.; Co, K.T.; Lupu, E.C. Byzantine-resilient decentralized stochastic gradient descent. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 3619–3623.
43. Patel, M.; Naughton, B.; Chan, C. Multi-Access Edge Computing: A Survey on Deployment Strategies and Use Cases. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1589–1625.
44. MOBI. MOBI: Mobility Open Blockchain Initiative, 2024. Available online: https://dlt.mobi (accessed on 31 October 2025).
45. Krajzewicz, D.; Hertkorn, G.; Rössel, C.; Wagner, P. SUMO (Simulation of Urban MObility)—An open-source traffic simulation. In Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM 2002), Sharjah, United Arab Emirates, 28–30 September 2002; pp. 183–187.
46. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In Proceedings of the European Symposium on Research in Computer Security, Guildford, UK, 14–18 September 2020; pp. 480–501.
47. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the convergence of FedAvg on non-IID data. *arXiv* **2019**, arXiv:1907.02189.