Round-Level Consensus for Energy-Efficient Federated Learning in Software-Defined Vehicles

Miraculous Udurume [1], Love Allen Chijioke Ahakonye [2], Jae Min Lee [1], Dong-Seong Kim [1] *
[1] IT-Convergence Engineering, *Kumoh National Institute of Technology*, Gumi, South Korea
* NSLab Co. Ltd., Gumi, South Korea, *Kumoh National Institute of Technology*, Gumi, South Korea
[2] ICT Convergence Research Center, *Kumoh National of Technology*, Gumi, South Korea
udurumemiraculous@gmail.com, (loveahakonye, ljmpaul, dskim)@kumoh.ac.kr

***Abstract*—Software-defined vehicles must learn at the edge under tight energy and latency budgets, yet many blockchain–FL stacks run consensus per client update, overworking validators and slowing rounds. We address this granularity mismatch by co-scheduling consensus with FL: finalize one block after each round of aggregation, converting $O(n)$ update commits into $O(1)$ round commits. Instantiated with PoA^2 and evaluated via energy–delay metrics (EDP/ED^{2}P) on four IoV/IDS datasets with $\{5, 10, 20\}$ clients and $\{10, 20, 30\}$ rounds, the policy preserves accuracy while improving efficiency. In CICIoV2024, the validator energy drops by 64–83% and updates/s increases from $+5\%$ (5 clients) to $+82\%$ (20 clients); BurST-ADMA, CICIDS2018, and VeReMi show 15–38% energy cuts. Gains scale with clients and rounds and remain robust under non-IID skew and client churn. Demonstrated with PoA^2 but protocol-agnostic, the rule integrates directly into existing FL pipelines for energy-efficient, blockchain-based client coordination.**

***Index Terms*—Edge computing, Energy efficiency, Federated learning, Intrusion detection, PoA^2 consensus, PureChain, Software-Defined Vehicles (SDV)**

I. Introduction

Software-defined vehicles (SDVs) replace fixed function electrical/electronic (E/E) stacks with edge-intelligent, software-driven platforms capable of continuous evolution [1]. By integrating vehicle-to-everything (V2X) communication, roadside units (RSUs), and mobile edge computing (MEC), SDVs enable service-oriented architectures and over-the-air (OTA) update pipelines for perception, control, and safety at fleet scale [2]. Looking ahead, sixth-generation (6G) roadmaps emphasize ultra-reliable low-latency communication (URLLC), massive connectivity, and energy-efficient edge learning, further motivating on-vehicle model training for safety-critical workloads [3]. These architectural and latency constraints render centralized training impractical at scale, favoring on-vehicle updates coordinated across many SDVs.

Federated learning (FL) addresses these constraints by training models collaboratively in distributed vehicular nodes while keeping raw data locally. By transmitting only gradient or weight updates, FL reduces backhaul usage and preserves privacy, aligning with bandwidth and latency limits in mobile edge settings [4], [5]. To cope with mobility and heterogeneity, recent designs incorporate adaptive client selection and aggregation policies aware of non-independent and identically distributed (non-IID) data, as well as intermittent participation [6]. However, FL alone does not provide verifiable or accountable update provenance among partially trusted stakeholders [7]. This leaves room for attacks such as model poisoning and coordinated misbehavior, motivating an auditable validation layer to certify contributions and outcomes [8], [9].

An auditable validation layer is naturally realized with blockchain ledgers, which offer immutable logging, decentralized verification, and programmable policy enforcement for model-update workflows [8], [9]. In vehicular FL, such ledgers can certify update provenance and aggregation outcomes among partially trusted stakeholders (vehicles, RSUs, validators), improving reliability under heterogeneous participation [10]. However, the cost of consensus often dominates both energy and latency; edge-oriented studies emphasize accounting for per-round expenditure in constrained devices [5], while surveys note that validator power draw and commit overhead remain under-addressed at scale [3]. Complementing these perspectives, early analyzes for autonomous vehicles catalog practical design challenges and validation pathways for blockchain-enabled FL [11]. These observations motivate aligning commit granularity with the FL round, validating once per round rather than per update; a policy we adopt in our design.

The core gap is that many blockchain–FL integrations validate per client update, inflating the validator work, energy, and round latency [10]. This granularity is misaligned with the round-level aggregation of FL, creating structural inefficiency at the commit layer [5]. We address this with a co-scheduled consensus that finalizes a single PoA^2 block after each FL round. Collapsing per-update commits into a round-level commit reduces consensus from $O(n)$ to $O(1)$ blocks/round, yielding validator-efficient, latency-aware finality while retaining auditability (round hash/metadata; no raw features/labels). Our contributions are:

1) A blockchain–FL co-design that aligns block validation with FL rounds, replacing per-update commits with one round-level commit ($O(n) \rightarrow O(1)$).
2) An energy–latency modeling framework (EDP/ED^{2}P) capturing validator power, client participation, and non-IID severity.
3) Empirical validation in vehicular IDS benchmarks (CICIoV2024, VeReMi, BurST-ADMA, CICIDS2018): accuracy is preserved; validator energy drops by up to 83% and throughput increases by up to 82% on CICIoV2024

at larger client counts, with 15–38% energy reductions observed on the other datasets.

The remainder of this paper is organized as follows: Section II reviews related work on blockchain–FL in vehicular systems. Section III presents the system model. Section IV discusses the experimental setup, evaluation, and analysis. Section V concludes with future directions.

II. Background and Related Studies

In vehicular-edge deployments, the governing constraints: latency, bandwidth, and privacy, favor local edge learning orchestrated across vehicles and RSUs rather than a centralized cloud [3], [5]. FL meets these requirements by collaboratively training models at distributed vehicular nodes while keeping raw data local; By exchanging only gradients or weights, FL reduces backhaul and preserves privacy within mobile-edge budgets [4], [5]. To remain effective under mobility and heterogeneity, recent designs add resource-aware scheduling and adaptive client selection, with aggregation policies that account for non-IID data and intermittent participation [6]. For sensitive telemetry, privacy-preserving pipelines (e.g. homomorphic encryption) protect updated contents without exporting raw streams [6]. However, communication and synchronization bottlenecks persist in latency-sensitive intelligent transportation systems, and stochastic participation continues to affect convergence stability at scale, motivating per-round budgeting of energy and communication and the addition of an auditable validation layer to certify update provenance and outcomes [3], [5].

Blockchain-assisted federated learning employs distributed ledgers and smart contracts to ensure immutable logging, decentralized verification, and policy orchestration under partial trust [8]. In vehicular contexts, this concept enhances reliability among heterogeneous participants and supports energy-aware consensus for latency-sensitive workloads [10]. Recent systematizations outline design parameters across consensus, identity, incentives, and storage configurations [12], [13]. However, asynchronous and fully decentralized variants often suffer from high communication or validation overhead when committing per client update [14], [15]. To mitigate validator energy and latency costs, we adopt round-synchronous finality—one blockchain commit per FL round—aligning consensus granularity with learning cadence while preserving auditability and model integrity.

These observations expose a system gap: many BCFL prototypes validate at the transaction/update level, mismatching FL's round-level semantics and inflating validator workload, energy, and end-to-end round time [14], [15]. Previous vehicular BCFL evidences the coupling between consensus and energy [10], while edge FL studies advocate a per-round accounting for device constraints [5]. We co-schedule consensus with training rounds, one ledger commit per round, to match the unit of commitment to learning cadence and close this systems gap.

Algorithm 1 Energy-Aware Client Selection at RSU (Step 2)

Require: Candidate SDVs $\mathcal{C}$, residual energy E_i, link quality q_i, uplink rate R_i, update size M (bits), compute energy e_i^{comp}, round budget B_{round}, min/max cohort $K_{\min}, K_{\max}$, fairness history h_i, weights $\alpha, \beta, \gamma, \delta$, safety factor $\eta \in (0,1)$, small $\varepsilon > 0$
Ensure: Selected set $\mathcal{S} \subseteq \mathcal{C}$

```
Normalize: Ê_i ← (E_i − min E)/(max E − min E), q̂_i ← (q_i − min q)/(max q − min q), ĥ_i ← (h_i − min h)/(max h − min h)
for all i ∈ C do
    e_i^tx ← (M/R_i) P_tx;   e_i ← e_i^tx + e_i^comp
    eligible_i ← (e_i ≤ η E_i)
    u_i ← α Ê_i + β q̂_i + γ(1 − ĥ_i) − δ e_i/(max_j e_j + ε)
end for
Sort C by u_i (desc.; tie ↑ E_i)
S ← ∅,  E_used ← 0
for all i in sorted C do
    if eligible_i and |S| < K_max and E_used + e_i ≤ B_round then
        S ← S ∪ {i};  E_used ← E_used + e_i
    end if
end for
if |S| < K_min then                              ▷ guarantee progress
    for all i in sorted C \ S do
        if eligible_i then
            S ← S ∪ {i}
        end if
        if |S| = K_min then
            break
        end if
    end for
end if
return S
```

III. System Methodology

We co-schedule FL with the ledger: clients train locally, RSUs aggregate, and the chain seals one PoA^2 block per round rather than per update. Round-level commits, energy-aware client selection, and backlog-aware multi-RSU aggregation yield scalable, resource-efficient security for SDVs without altering models or features.

A. Edge Layer: Local Training

SDVs perform on-device updates on non-IID IDS logs and vehicular telemetry; raw data never leaves the vehicle; only gradients are shared. Each node reports residual energy, uplink rate, and link quality to enable energy-aware scheduling and avoidance of stragglers. The result is a pool of candidate updates with resource telemetry, which feeds the client selection.

B. RSU Layer: Energy-Aware Client Selection

The RSU serves as the first decision point, selecting the SDVs for the current FL round under an explicit energy budget B_{round}. Algorithm 1 shows candidates ranking using a multi-objective utility over residual energy, link quality,

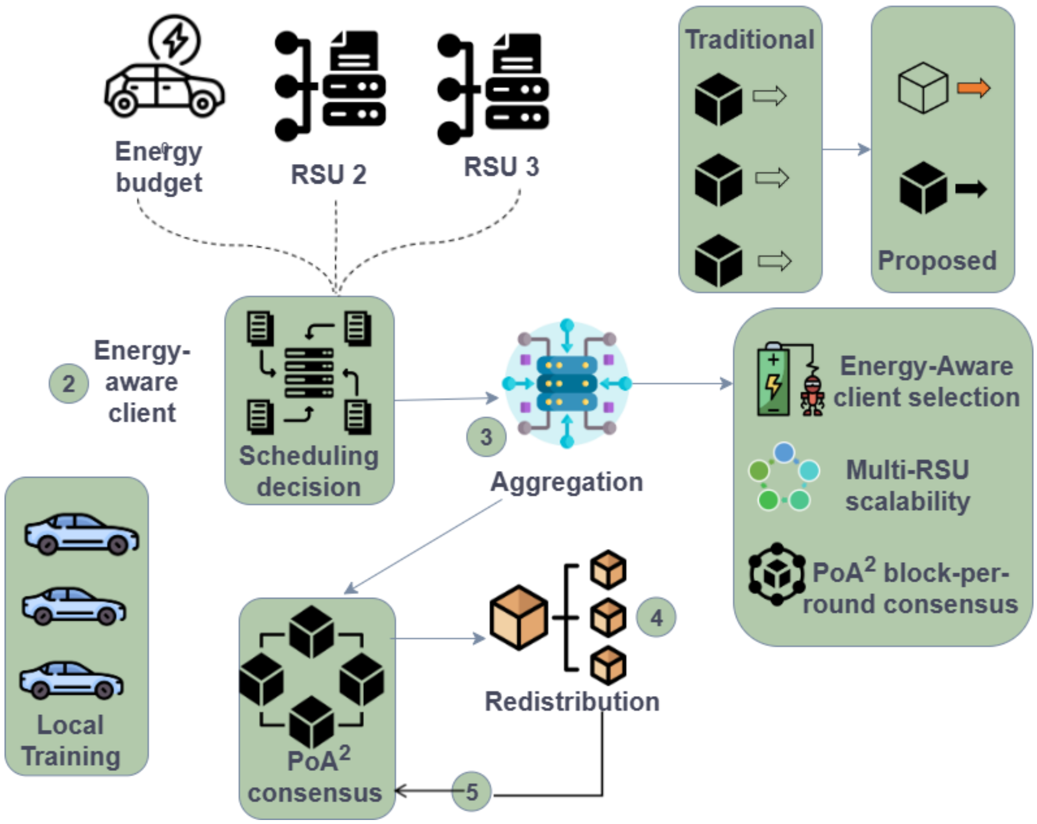

Fig. 1: Proposed system model of Energy-Aware FL

fairness history, and estimated cost. The top-ranked vehicles are admitted subject to a knapsack-like budget constraint, which yields the selected set $\mathcal{S}$. This prioritizes capable nodes, enforces fair participation, and keeps the round energy within limits.

C. RSU Layer: Aggregation and Coordination

Selected client updates are aggregated at RSUs as in Algorithm 2, using a sample-weighted FedAvg variant with per-update clipping and compression to bound communication. When the backlog exceeds a threshold, updates are offloaded to neighboring RSUs for multi-RSU coordination. The RSU emits a consolidated Δw plus metadata (cohort size, compression ratio, energy stats, hashes). This package feeds the blockchain layer for validation and commit.

D. PoA^2 Layer: Blockchain Consensus

The blockchain layer provides governance, traceability, and round-level authentication. Instead of sealing one block per client update, the validators commit *one block per FL round*. A scheduled leader proposes a block that includes the round identifier, the hash of the aggregated update produced in Algorithm 2, and energy/participation metadata. Validators attest; When the quorum is reached within a timeout, the block is appended and broadcast. This *block-per-round* coupling reduces the validator energy while preserving an immutable audit trail.

Algorithm 2 RSU Aggregation & Scheduling (Step 3)

Require: Selected clients $\mathcal{S}$, local updates $\{(w_i, n_i)\}$, compression rate $c \in (0,1]$, cross-RSU queue $\mathcal{Q}$, backlog limit $L_{\max}$
Ensure: Aggregated update Δw and metadata m

for all $i \in \mathcal{S}$ **in parallel do**
  $w_i \leftarrow \mathsf{Clip}(w_i, \tau)$; $w_i \leftarrow \mathsf{Compress}(w_i, c)$
end for
$N \leftarrow \sum_{i \in \mathcal{S}} n_i$
FedAvg: $\Delta w \leftarrow \sum_{i \in \mathcal{S}} \frac{n_i}{N} w_i$
if $|\mathcal{Q}| > L_{\max}$ **then** ▷ multi-RSU load balancing
  $\mathsf{OffloadToPeerRSU}(\Delta w)$
end if
$m \leftarrow \{\text{round\_id}, |\mathcal{S}|, N, c, \text{hash}(\Delta w), \text{energy\_stats}\}$
return $(\Delta w, m)$

E. Redistribution Layer: Model Delivery

After PoA^2 validation, the committed global model is redistributed to participating SDVs, closing the FL loop. This delivery immediately seeds the next local round under tight energy and bandwidth budgets. The combination of redistribution to commit per round block ensures that only the model approved by consensus propagates. Round-level integrity and traceability are preserved without additional overhead. Figure 1 summarizes the four-stage pipeline from client training to global model redistribution.

IV. Experimentation and Result Discussion

A. Datasets Description

We evaluated four datasets: BurST-ADMA [16], VeReMi [17], CICIoV2024 [18], and CICIDS2018 [19], covering industrial IoT traffic, V2X misbehavior, smart-city IoV flows, and a general-purpose IDS baseline. Continuous features are standardized (z-score), and categorical fields follow each dataset's native schema; train/validation/test splits are disjoint by session/route or time window to prevent leakage, and labels are harmonized with benign versus attack/misbehavior for uniform metrics. The experiments were run on a hosted VM (Python 3.11.13, $\approx$13 GB RAM) using TensorFlow Federated and Matplotlib. Consensus is simulated in Google Colab via a lightweight, PoA^2-style round-level commit policy: one block is finalized per FL round containing the aggregator's model hash and round metadata (timestamp, validator set, modeled energy counters); no raw features or labels are persisted, allowing us to isolate round-level commits from per-update validation in terms of validator energy and wall-clock time.

B. Performance Evaluation

We evaluated the proposed co-scheduled consensus mechanism against conventional per update in four benchmark datasets (CICIoV2024, BurST–ADMA, CICIDS2018, and VeReMi) with client sizes ranging from 5 to 20 and training horizons of 10–30 rounds. Performance was measured in terms of energy consumption, throughput in rounds per second, and model update rates. Figure 2 shows that co-scheduling substantially reduces validator energy across all client scales. At 20 clients, the total energy drops from nearly 950 J under fixed consensus to below 170 J with coscheduling, yielding over 70% reduction. This trend holds across datasets.

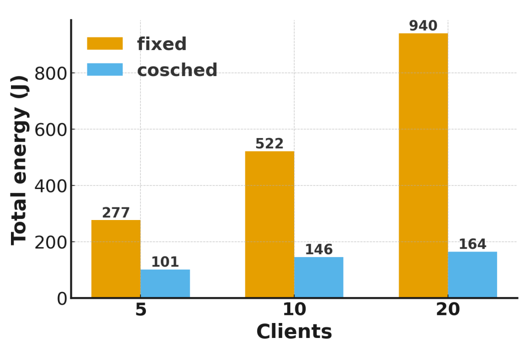

Fig. 2: Total energy vs. clients at α=0.3.

Figure 3 shows that co-scheduling not only reduces energy but also improves throughput. Updates per second scale more rapidly with coscheduling, achieving +23.3% improvement at 10 clients in CICIoV2024. This reflects the benefit of committing one block per round rather than per client, thus reducing consensus overheads. Figure 4 summarizes the percentage deltas in the datasets in the canonical setting of 10

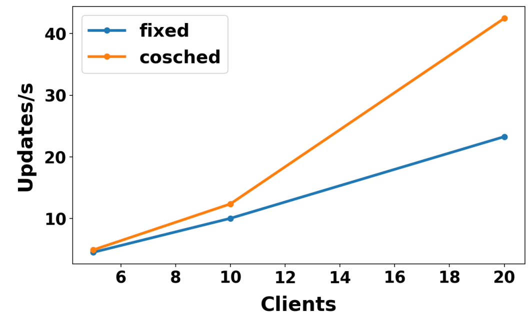

Fig. 3: Updates/s vs. clients at α=0.3.

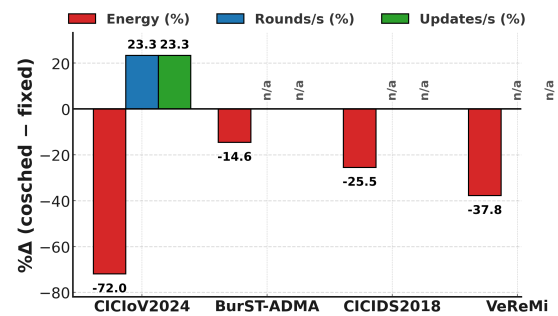

Fig. 4: Cross-dataset percentage deltas at clients = 10, rounds = 10 (%Δ = (cosched − fixed)/fixed $\times$ 100). Missing throughput values are shown as "n/a".

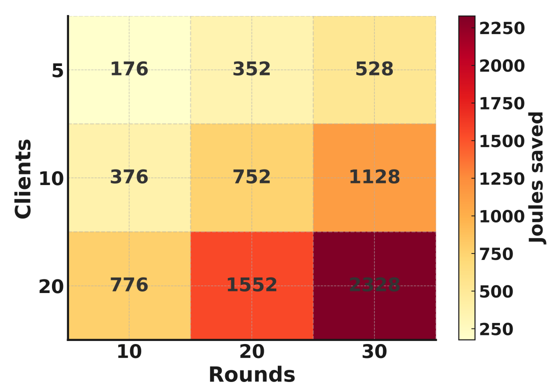

Fig. 5: Scalability heatmap (CICIoV2024, α=0.3): total energy saved (J) across clients$\times$rounds; darker indicates larger savings.

clients and 10 rounds. Although only CICIoV2024 provides complete throughput traces, energy savings remain substantial in all datasets, confirming that the benefits generalize beyond a single workload.

To assess scalability, Figure 5 provides a heatmap of the total energy saved in CICIoV2024. The energy savings grow monotonically with both the client population and the training horizon, reaching over 2200 J at 20 clients and 30 rounds. This confirms the advantages of the co-scheduling compound in larger and longer FL runs. Together, these results demonstrate that co-scheduled consensus simultaneously reduces validator

TABLE I: SDV BCFL design comparison

Ref.	Year	FL Type	Commit	Scope	Ledger	Energy
[10]	2020	AV BCFL (analysis)	✗	✗	✓	✗
[5]	2022	Edge FL	✗	✗	✗	✓
[9]	2022	Vehicular BCFL	✗	✗	✓	✓
[14]	2022	Fully-decentralized BCFL	✗	✗	✓	✗
This work	2025	Round-level BCFL	✓	✓	✓	✓

Legend — Commit: one block per FL round; Scope: RSU-aggregated round validation; Ledger: blockchain present; Energy: energy-aware scheduling/modeling.

energy and increases throughput in SDV settings, with gains strengthening as clients and rounds scale, underscoring suitability for resource-constrained SDV edge deployments.

Table I contrasts the prior BCFL with our round-level design: earlier work omits a ledger [5] or lacks round-level commit and RSU-scope validation [10], [11], [15], with validator costs largely unquantified. We commit one PoA^2 style block per round, validate in the scope of the RSU, and use energy-aware scheduling, improving auditability, scalability, and efficiency.

V. Conclusions

We presented a round-level, blockchain-coordinated FL scheme for SD-ITS that finalizes one PoA^2 block after each RSU aggregation and couples this with energy-aware client selection and backlog-aware multi-RSU aggregation. Across BurST-ADMA, VeReMi, CICIDS2018, and CICIoV2024, detection quality is preserved while the validator energy drops substantially (CICIoV2024: 64–83%; others: 15–38%), updates/s increases (up to +82% at larger fleets), and $\mathrm{EDP/ED^2P}$ improves. Savings grow with clients and rounds and remain stable under non-IID skew and client churn, indicating good data/scale efficiency. By aligning consensus with FL rounds, the design reduces validator overhead and strengthens auditability without altering models or features, enabling practical and verifiable edge learning.

Future work will emphasize hardware-in-the-loop evaluation on vehicular edge platforms with detailed latency, communication, and energy profiling. This includes lightweight, contract-automated consensus with reputation-aware validator selection, adaptive sampling with compression/quantization, and validation across broader datasets and driving scenarios. Further analysis will assess the impact of networking overhead, block propagation, and smart-contract execution on system efficiency, as well as convergence behavior, robustness to adversarial updates, and model stability beyond aggregate detection performance.

Acknowledgment

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ITRC support program (IITP-2026-RS-2024-00438430, 34%).

References

[1] T. S. Madhuri and J. Bala Vishnu, "Software-defined vehicles: The future of automobile industry," in *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, 2025, pp. 192–197.

[2] Z. Liu, W. Zhang, and F. Zhao, "Impact, challenges and prospect of software-defined vehicles," *Automotive Innovation*, vol. 5, no. 2, pp. 180–194, 2022.

[3] A. Gouissem, Z. Chkirbene, and R. Hamila, "A comprehensive survey on energy efficiency in federated learning: Strategies and challenges," in *2024 IEEE 8th International Energy Conference (ENERGYCON)*, 2024, pp. 1–8.

[4] H. L. Nakayiza, L. A. Chijioke Ahakonye, D.-S. Kim, and J. M. Lee, "Resource-aware adaptive federated learning for enhanced ddos detection in vehicular ad hoc networks," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, 2024, pp. 1262–1267.

[5] M. Mendula and P. Bellavista, "Energy-aware edge federated learning for enhanced reliability and sustainability," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, 2022, pp. 349–360.

[6] H. L. Nakayiza, L. A. C. Ahakonye, D.-S. Kim, and J. M. Lee, "Homomorphic encryption for privacy-preserving misbehavior detection in the internet of vehicles," in *2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2025, pp. 0320–0324.

[7] M. Udurume, L. A. Chijioke Ahakonye, D.-S. Kim, and J. M. Lee, "Scalable and Efficient PureChain-Based Federated Learning for Intelligent Transportation Systems," in *2025 16th International Conference on Information and Communication Technology Convergence (ICTC)*, 2025.

[8] L. A. C. Ahakonye, C. I. Nwakanma, and D.-S. Kim, "Tides of Blockchain in IoT Cybersecurity," *Sensors*, vol. 24, p. 3111, 05 2024.

[9] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "Purechain-Enhanced Federated Learning for Dynamic Fault Tolerance and Attack Detection in Distributed Systems," *High-Confidence Computing*, p. 100354, 2025.

[10] M. Aloqaily, I. Al Ridhawi, and M. Guizani, "Energy-aware blockchain and federated learning-supported vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22 641–22 653, 2022.

[11] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[12] W. Ning, J. Zhang *et al.*, "Blockchain-based federated learning: A survey and new perspectives," *Applied Sciences*, vol. 14, no. 20, p. 9459, 2024.

[13] E. Goh, D.-Y. Kim, K. Lee, S. Oh, J.-E. Chae, and D.-Y. Kim, "Blockchain-enabled federated learning: A reference architecture design, implementation, and verification," *arXiv preprint arXiv:2306.10841*, 2023.

[14] Y. Liu, K.-Y. Lam, X. Ding, L. Chen, Q. Bai, and Y. Guo, "Blockchain-enabled asynchronous federated learning in edge computing," *Sensors*, vol. 21, no. 10, p. 3335, 2021.

[15] Z. Qin, X. Yan, M. Zhou, and S. Deng, "Blockdfl: A blockchain-based fully decentralized peer-to-peer federated learning framework," *arXiv preprint arXiv:2205.10568*, 2022.

[16] M. A. Amanullah, M. Baruwal Chhetri, S. W. Loke, and R. Doss, "BurST-ADMA: Towards an Australian Dataset for Misbehaviour Detection in the Internet of Vehicles," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 624–629.

[17] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[18] E. C. P. Neto, H. Taslimasa, S. Dadkhah, S. Iqbal, P. Xiong, T. Rahman, and A. A. Ghorbani, "CICIoV2024: Advancing Realistic IDS Approaches Against DoS and Spoofing Attack in IoV CAN Bus," *Internet of Things*, vol. 26, p. 101209, 2024.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Cicids2018: Intrusion detection evaluation dataset," Canadian Institute for Cybersecurity (CIC), 2018, available from CIC/UNB.