Dynamic-PBFT: Enhanced Consensus in Decentralized Federated Averaging for V2V Networks

Zhishen Xia
Karlsruhe Institute of Technology
Karlsruhe, Germany
zhishen.xia@kit.edu

Jianxin Zhao
Karlsruhe Institute of Technology
Karlsruhe, Germany
jianxin.zhao@kit.edu

Alexey Vinel
Karlsruhe Institute of Technology
Karlsruhe, Germany
Halmstad University, Sweden
alexey.vinel@kit.edu

***Abstract*—Vehicle-to-everything (V2X) communication plays a crucial role in enabling collaborative intelligence among autonomous vehicles, by utilizing paradigms such as Federated Learning (FL). However, the dynamic and decentralized nature of vehicular networks poses challenges, particularly in maintaining model convergence and robustness without centralized coordination. In this paper, we propose a dynamic Practical Byzantine Fault Tolerant consensus mechanism tailored for decentralized FL in vehicular environments. Our method optimizes federated averaging by addressing the high mobility and intermittent connectivity of vehicles. Through simulations, we evaluate its performance against baseline method, demonstrating improved resilience, efficiency, and adaptability in the presence of adversarial conditions.**

***Index Terms*—Vehicle-to-Vehicle, VANET, V2X, Byzantine fault tolerance, decentralized federated learning**

I. Introduction

Recent advancements in vehicle-to-everything (V2X) communication have notably improved existing transport systems by enabling increased connectivity and driving autonomy levels. V2X allows on-board sensor data interactions with neighboring vehicles, roadside units (RSUs), and cloud applications over wireless local network or cellular connectivity [1], [2]. A typical V2X application scenario is enhancing driving safety via information exchange among multiple entities to avoid occlusion to the view of single vehicles [3].

However, when the communication pattern becomes complex, e.g., certain tasks require continuous exchange of information among various entities over a relatively long period of time, two features of vehicles raise challenges: (a) they are highly dynamic, meaning the information exchange may happen within a short time frame; (b) they are not centrally controllable, meaning that, it is hard to organize a formation of vehicles for a long time. Federated learning (FL) is such a prominent example of collaboratively utilizing complex and continuous V2X communication [4]. This paradigm utilizes the computing power and local data of a group of vehicles to keep training and improving machine learning models. But without central server's coordination, due to concerns about vehicle data privacy, the highly dynamic vehicle-to-vehicle (V2V) network poses both opportunities and challenge to vehicular FL [5]. This scenario can be illustrated in Fig. 1.

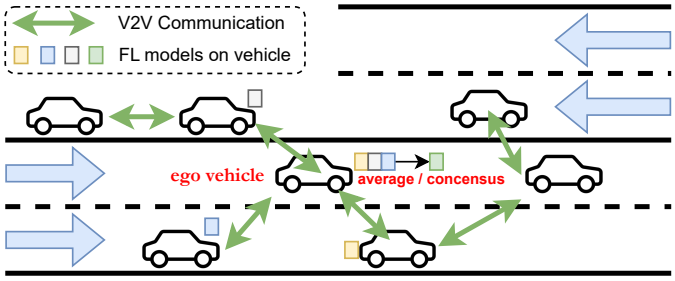

Fig. 1. A decentralized federated learning scenario based on V2V connection among vehicles. In one step of FL, the ego vehicle receives model from neighboring vehicles and get average/consensus to update its own model.

In this study, we explore the application of Byzantine Fault Tolerance (BFT) algorithms in decentralized federated learning (DFL) within dynamic vehicular ad-hoc network (VANET). Federated Averaging serves as a fundamental step in FL. In centralized FL, this process involves directly averaging local models from all participating nodes in a central server. But in DFL, model aggregation is achieved through consensus mechanisms, with BFT algorithms commonly employed to ensure consensus [6]. The high mobility and dynamic topology of V2V networks present both significant challenges also to conventional BFT algorithms.

Towards this end, in this work we make the following contributions. First, we propose a novel algorithm, dynamic-PBFT (Practical BFT), that optimizes the federated averaging process in decentralized learning scenarios in dynamic vehicular networks; its performance is then evaluated in simulation experiments. Compared to classic PBFT algorithm, the proposed method manages the join/leave of vehicular nodes, and provide mechanisms including malicious node detection and computing interruption recovery. Second, we analyze the impact of malicious nodes, dynamic node scale,

*Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them. Project grant no. 101069576.

and communication reliability on the algorithm's performance, providing insights into its robustness and scalability.

II. Related Work

A. Vehicle-to-Everything

The evolution of V2X has seen significant advancements, particularly with the introduction of Cellular V2X (C-V2X) and the ongoing development of 5G-V2X. C-V2X, defined in 3GPP Release 14 and 15, leverages LTE networks to provide reliable low-latency communication, which is critical for safety applications [7].

Recent research has highlighted the enhanced capabilities of 5G-V2X, standardized in 3GPP Release 16, which promises higher data rates and reduced latency, thus broadening the scope of V2X applications to include more complex and data-intensive services [8]. The global deployment of these technologies has varied, with China and Europe leading in terms of large-scale trials and commercial implementations, while the United States has made regulatory strides to facilitate further deployments [9].

Among applications utilizing V2X, Collaborative Perception is a very important type. It is essential to address occlusion and sensor failure issues in autonomous driving. Two main challenges are which information should be exchanged over the V2X network and how the exchanged information is fused. [10] devises an effective collaboration method based on exchanging the outputs from each agent, achieving a better bandwidth-performance tradeoff. However, a large part of existing work focuses on utilizing V2X via instance or short-period collaboration, mostly at the intersection of a road, as shown in [11]. Long-term collaborations such as model training or collaborative optimization are not well discussed.

B. Decentralized Federated Learning

In recent years, FL has gained relevance in training collaborative models without sharing sensitive data. It has shown promise in being applied together with V2X in the field of collaborative driving [12]. Specifically, DFL emerged to address the concerns of latency and single-point failure by promoting decentralized model aggregation and minimizing reliance on centralized architectures [13].

In [14], a reputation-aware coordination mechanism is designed to coordinate a group of smart devices dynamically into a virtual cluster, in which the machine learning model aggregation is conducted in a decentralized P2P manner. In [15], the authors propose an FL framework that features a directed acyclic graph-based structure, where nodes represent uploaded models, and referencing relationships between models form the DAG that guides the aggregation process; this framework aims to provide both fairness and security in training.

DFL also faces challenges from the perspective of algorithms, such as system heterogeneity and statistical heterogeneity. To ensure fast convergence in the presence of slow edge devices, [16] presents an efficient DFL method that integrates adaptive control of both local updating frequency and network topology to better support heterogeneous participants. However, most current work on DFL still focuses on stable environments where the decentralized nodes can form a stable topology [13]. This assumption is not applicable in collaborative driving, where the vehicles cannot fall under the control of a single entity and form groups only temporarily.

C. Practical Byzantine Fault Tolerance

Consensus algorithms play a key role in decentralized distributed systems, and BFT algorithms are widely used to achieve consensus [17]. Traditional BFT algorithms however have several problems. For example, BFT assume synchronous networks, where message delivery is guaranteed within a known time. It also require an exponential number of message exchanges which makes it non-scalable. The PBFT is then proposed to address these issues [18]. PBFT can be used in areas such as Internet of Things due to its fault tolerance and efficiency in achieving consensus in distributed networks [19]. However, even with improvements, PBFT still faces challenges and limits in decentralized FL performed via V2V networks. It is still sensitive to the dynamic joining and departure of nodes and are ill equipped to handle environments characterized by large scale and pronounced dynamics [19]. The work [20], [21] aim to apply PBFT in decentralized FL, but they focus on addressing data privacy issues.

III. Algorithm Design

The main goal of this algorithm is to optimize a key step in FL process in dynamic vehicular networks: the federated averaging in decentralized learning scenarios. The proposed algorithm, *dynamic-PBFT*, is based on traditional PBFT and aims to generate good performance in highly dynamic networks where the vehicle nodes join and exit the decentralized learning process with high probability.

A. System Model

We adopt the network model of a dynamic vehicular network, where nodes are distributed according to a random movement model and interact via V2V communication. Each node moves in a two-dimensional coordinate plane, with its movement described by the following formulas:

$$X_{t+1} = X_t + \Delta x, \quad \Delta x \sim \mathcal{N}(0, \sigma_2) \tag{1}$$

$$Y_{t+1} = Y_t + \Delta y, \quad \Delta y \sim \mathcal{N}(0, \sigma_2) \tag{2}$$

where X_t and Y_t represent the coordinates of a node at time t, and Δx and Δy represent random change in position in the x- and y-axes, respectively.

This system faces two main types of threats. First is the *Byzantine nodes*. These malicious vehicle nodes may alter or forge their local models and send fraudulent information to the network. According to the PBFT algorithm, the system can tolerate up to $f \leq \frac{N-1}{3}$ Byzantine nodes, , but the presence of such nodes can severely affect the quality of model training. The second threats comes from non-malicious faulty nodes. These nodes can temporarily go offline due to increase distance, network disruptions, or insufficient computational

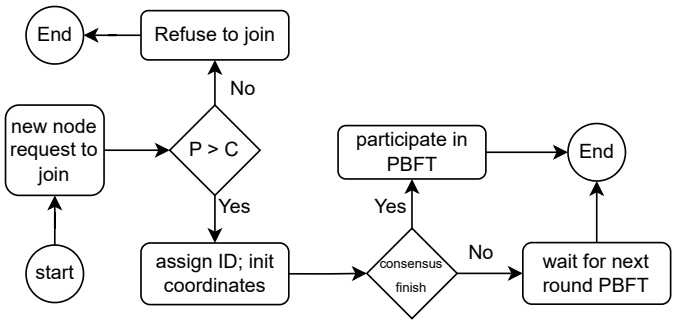

Fig. 2. Vehicle node join process in proposed dynamic PBFT algorithm.

resources. While they do not intentionally disrupt training, their disconnection can cause interruptions in model updates.

B. Dynamic PBFT Algorithm

Based on this system model, the proposed dynamic PBFT algorithm consists of the following components.

Dynamic Vehicle Node Management To ensure network stability, the joining and exiting of nodes must pass through a strict admission control process. New nodes are evaluated based on the overall reliability of network and current state of existing nodes. The admission process prevents performance degradation by selectively allowing nodes that meet predefined stability criteria. Specifically, the joining condition is based on the network's reliability indicator P and a dynamic threshold δ_C, and assign the node requesting to join to the next available round of consensus. It ensures new nodes are only added when the network's reliability is sufficient to support them. Fig. 2 shows the node joining process.

The condition for node joining can be calculated as follows.

$$P_{N+1} \leq \delta_C, T \leq (N+1)/2 \tag{3}$$

Here P_{N+1} represents the overall reliability of the network after adding a new vehicle node, and T is the minimum required number of valid responses. The dynamic threshold is calculated as in Eq. 4.

$$\delta_C = 1 - \frac{2\log(1-P)}{N} \tag{4}$$

By modeling factors such as path loss and signal-to-noise ratio (SNR), we can compute the communication success rate between each pair of nodes in real-time. The communication success rate is calculated using the following formula:

$$P_C(i,j) = \frac{1}{1+e^{-0.5(\text{SNR}-L)}}. \tag{5}$$

Here L is the path loss, and is defined as $L = L_0 + 10\gamma \log_{10} d$, where, $L_0 = 37$ dB is the reference loss, $\gamma = 2.7$ is the path loss exponent, and d is the distance between nodes. The overall network reliability indicator, P, is computed using Monte Carlo simulations as follows:

$$P = \sum_{k=T_{\min}}^{N} \binom{N}{k} P_{\text{success}}^k (1-P_{\text{success}})^{N-k} \tag{6}$$

where $T_{\min} = \frac{(N+1)}{2}$, $P_{\text{success}} = P_C(1-P_e)$ and P_e is the node failure rate.

Training Interruption Recovery Temporary disconnections in volatile vehicular environments can derail decentralized training progress. In the case of node disconnection, its training progress and model is saved in local snapshot, and when the node reconnects, it resumes training from where it left off. If the diconnection time is too long, which means the vehicle perhaps has already move in different direction than tha training group vehicles, the training data is then discarded. If the estimated training time is h and the progress in percentage is p, the timeout threashold is set to $(1-p)h$.

Malicious Node Detection To prevent malicious nodes from affecting the global model, the algorithm introduces a model validity verification mechanism in the PBFT consensus process. If a node's local model exhibits anomalies (e.g., values exceeding reasonable ranges), the node will be marked as malicious, and its model will be excluded from subsequent consensus rounds.

IV. Evaluation

A. Setup

To simulate a dynamic vehicular network environment, the evaluation is implemented based on the discrete event simulation framework SimPy on Python 3.9. A simulation time step is set to 1 unit, representing real-time operations in seconds. The dynamic behavior of the vehicle nodes is modeled following the following mechanisms:

- **Movement Model** The coordinates (x, y) of each node are updated in real-time, with a movement step limited to ± 1.0 meter to prevent excessive clustering. Total driving area is restricted to [-50, 50] meters in the simulation.
- **Communication Model** The maximum communication range of each node is 100 meters. The dynamic communication success rate is calculated using a logarithmic path loss mode.

The maximum number of vehicle nodes is 15. The initial proportion of malicious nodes is set to 0.1 (later extended to 0.3 for comparison). The probability of new node joining per time unit, which triggers dynamic management mechanism, is set to 50%. The probability of node disconnection, which validates the effectiveness of the recovery mechanism, is set to 30%. The consensus computation we use as example here is averaging of vectors from vehicle nodes, following the core step in Federated Learning. Network reliability threshold is calculated based on Eq. 4. For simplicity, we set the vector size to be 100 in this simulation, though in real world application this vector size can be millions.

The federated averaging process is simulated through periodic local training and secure global aggregation, synchronized with PBFT consensus protocols. Benign nodes generate parameters are random numbers between 0 and 1, while malicious nodes produce distorted values to simulate data poisoning. Training duration varies uniformly between 1 to 3 time units per round, emulating heterogeneous computational

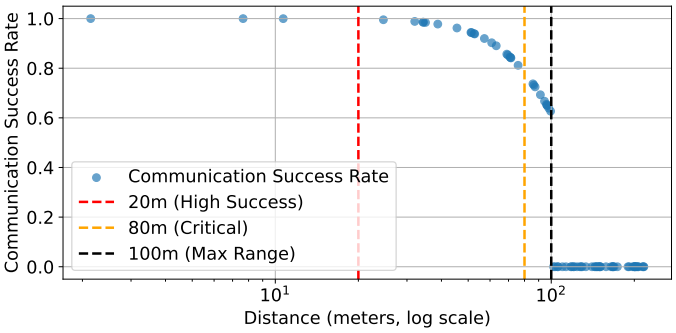

Fig. 3. Impact of vehicle-to-vehicle distance on communication success rate.

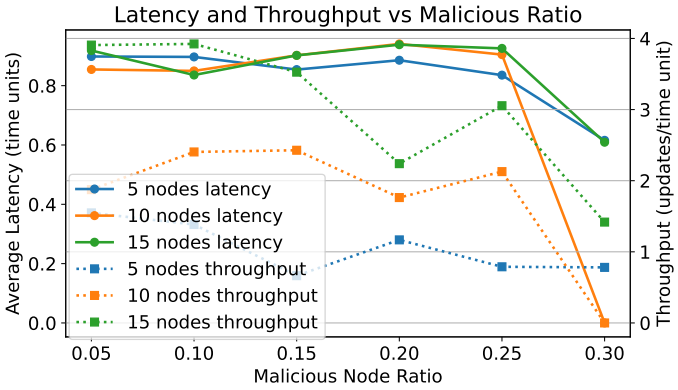

Fig. 4. Impact of malicious vehicle node ratio on latency and throughput.

capabilities across vehicles. For global model aggregation, after local training, nodes initiate a PBFT consensus round to validate and aggregate updates. A model is deemed valid if its parameters fall within [0,1), filtering out malicious outliers.

Keeping these parameters fixed, in the simulation we adjust key variables, e.g., the proportion of malicious nodes and participating vehicle nodes, to verify the algorithm's robustness. Each experiment is run for 3 iterations, and the average values are recorded to eliminate the impact of randomness.

B. Communication Reliability

The communication success rate between nodes is modeled using a logarithmic path loss model. The communication success rate decreases exponentially as the distance between nodes increases. The experiment results in Fig. 3 show the following patterns. At short distances, the communication success rate remains above 90%, primarily influenced by the SNR of 70 dB and a path loss exponent $\gamma = 1.8$. This stability drops then according to distance, leading to packet loss or delays in message delivery. To accurately reflect the network's communication state, the simulation updates the communication success rate matrix periodically, at each time unit. This real-time update reflects changes in node positions and communication conditions.

The experiment also compares the standard deviation of communication success rates under static and dynamic topologies. In static topology, the standard deviation is 0.12. As nodes remain stationary, some communication links gradually fail, resulting in higher variability. In dynamic topology, the standard deviation drops to 0.07 because node movement optimizes the link distribution. As a result, the average communication success rate improves by 15%.

C. Impact of Malicious Nodes

Fig. 4 presents a dual-axis chart comparing delay and throughput as the malicious node proportion changes. The left vertical axis represents delay (in time units), and the right vertical axis shows throughput (updates per unit time).

The curves for different network scales highlight the marginal benefits of increasing network size. As the proportion of malicious nodes increases, the consensus delay gradually increases. This is primarily due to repeated broadcasts and validation requests caused by node disconnections, which result in higher delays. With the increase in the proportion of malicious nodes, the number of invalid consensus rounds rises, leading to a decrease in throughput. For instance, in a 15-node network, when the malicious node ratio reaches 30%, the throughput drops from 3.8 updates per unit time to 2.1 updates per unit time. In terms of the reliability threshold effect, when the network reliability drops below a dynamically calculated threshold, the system refuses new nodes from joining the network. Concerning malicious node concealment, malicious nodes attempt to interfere with consensus by generating anomalous model parameters, but PBFT's three-phase voting mechanism effectively filters out these anomalies. When the number of malicious nodes is less than or equal to f, the verification phase can eliminate over 90% of invalid models.

D. Comparison with Baseline

The goal of this experiment is to compare the performance differences between the Dynamic PBFT protocol and the classic PBFT protocol in key metrics. The initial number of nodes is fixed at 10, with a maximum of 15 nodes, and a 20% malicious node ratio. Each round of simulation runs for a maximum of 10 time units.

First, we check the impact of nodes' join/leave on consensus success in a dynamic network environment. Specifically, in reach round of simulation we first calculate the dynamic load factor, which is `load factor * (N / max nodes number)`. The failure probability of each node then is the sum of base failure rate and the dynamic load factor. The communication success rate is then calculated as described in Sec. III-B. The maximum allowed failure rate threshold is set to 1. Evaluation results show that the average success rate is close to 100%. This indicates that the dynamic PBFT demonstrates the same performance as traditional PBFT, even in the presence of new nodes joining and nodes disconnecting.

We then compare the number of messages generated during each consensus round. The median communication overhead of Dynamic PBFT is approximately 180 messages per round, while Baseline PBFT is 300 messages per round, with the latter having a broader distribution range. Dynamic PBFT reduces redundant message passing by optimizing the message broadcasting strategy, such as selective broadcasting and dynamically updating communication links. In contrast, Baseline

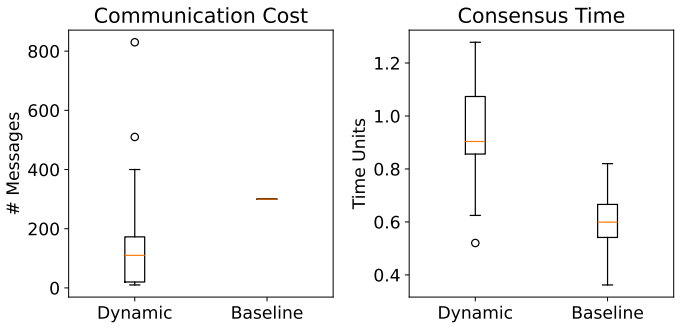

Fig. 5. Comparison of number of messages (left) and completion time of consensus (right) in different PBFT schemes.

PBFT uses a traditional full-node broadcasting mechanism, leading to communication overhead growing quadratically with the number of nodes ($O(N^2)$). As shown in the left part of Fig. 5, the Dynamic PBFT reduces communication overhead by 43.8% compared to Baseline PBFT.

Next we investigate the time required to complete a round of consensus using different PBFT scheme. The experimental results in the right part of Fig. 5 show that the median consensus time for Baseline PBFT is 0.6 time units, slightly lower than the 0.9 time units for Dynamic PBFT. This difference is mainly due to management overhead. Dynamic PBFT enhances network reliability by dynamically monitoring node status and adjusting communication links. But it also introduces additional negotiation steps, such as node recovery and link updates, resulting in increased consensus time per round. On the other hand, Baseline PBFT uses a static network configuration, which does not need to handle dynamic node changes, making it more efficient in stable scenarios.

Finally, we compare the number of global model updates completed per unit of time. Experimental results show that Baseline PBFT has a throughput of 6.6 operations per time unit, significantly higher than Dynamic PBFT's 2.7. This is also due to the dynamic management overhead and baseline's lightweight design. It is indeed one disadvantage of the proposed method which requires further optimization.

In general, although Dynamic PBFT has slightly higher consensus time due to added complexity, its advantages in dynamic network environments are still significant, including improved fault tolerance, optimized communication efficiency, and enhanced scalability.

V. Conclusion

This paper proposed dynamic-PBFT to enhance federated averaging in decentralized federated learning for dynamic vehicular networks. Our approach improves robustness and efficiency compared to the baseline PBFT, maintaining model convergence despite adversarial conditions. Simulation results demonstrate its scalability and resilience under varying node dynamics. There are many parts that can be investigate further in future work, including extending scale of simulation, optimizing communication overhead, incorporating real-world mobility data, and exploring advanced consensus mechanisms to further improve reliability and efficiency.

References

[1] J. Clancy, D. Mullins, B. Deegan, J. Horgan, E. Ward, C. Eising, P. Denny, E. Jones, and M. Glavin, "Wireless access for v2x communications: Research, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, 2024.

[2] M. A. Naeem, S. Chaudhary, and Y. Meng, "Road to efficiency: V2v enabled intelligent transportation system," *Electronics*, vol. 13, no. 13, p. 2673, 2024.

[3] K. Yang, D. Yang, J. Zhang, M. Li, Y. Liu, J. Liu, H. Wang, P. Sun, and L. Song, "Spatio-temporal domain awareness for multi-agent collaborative perception," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 383–23 392.

[4] M. K. Hasan, N. Jahan, M. Z. A. Nazri, S. Islam, M. A. Khan, A. I. Alzahrani, N. Alalwan, and Y. Nam, "Federated learning for computational offloading and resource management of vehicular edge computing in 6g-v2x network," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3827–3847, 2024.

[5] J. Yan, T. Chen, Y. Sun, Z. Nan, S. Zhou, and Z. Niu, "Dynamic scheduling for vehicle-to-vehicle communications enhanced federated learning," *arXiv preprint arXiv:2406.17470*, 2024.

[6] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2022.

[7] G. Bansal and J. B. Kenney, "C-v2x: Cellular vehicle-to-everything technology," *IEEE Vehicular Technology Magazine*, vol. 12, no. 1, pp. 20–28, 2017.

[8] M. N. Avcil, M. Soyturk, and B. Kantarci, "Fair and efficient resource allocation via vehicle-edge cooperation in 5g-v2x networks," *Vehicular Communications*, vol. 48, p. 100773, 2024.

[9] 5G Automotive Association, "5gaa 2023 annual report charts global path for c-v2x deployment," *5GAA*, 2024. [Online]. Available: https://5gaa.org/5gaa-annual-report-charts-path-for-c-v2x-deployment/

[10] M.-Q. Dao, J. S. Berrio, V. Frémont, M. Shan, E. Héry, and S. Worrall, "Practical collaborative perception: A framework for asynchronous and multi-agent 3d object detection," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[11] T. Huang, J. Liu, X. Zhou, D. C. Nguyen, M. R. Azghadi, Y. Xia, Q.-L. Han, and S. Sun, "V2x cooperative perception for autonomous driving: Recent advances and challenges," *arXiv preprint arXiv:2310.03525*, 2023.

[12] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Vehicular Communications*, vol. 33, p. 100396, 2022.

[13] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Comm. Surveys & Tutorials*, 2023.

[14] X. Zhou, W. Liang, I. Kevin, K. Wang, Z. Yan, L. T. Yang, W. Wei, J. Ma, and Q. Jin, "Decentralized p2p federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 82–89, 2023.

[15] G. Yu, X. Wang, C. Sun, Q. Wang, P. Yu, W. Ni, and R. P. Liu, "Ironforge: an open, secure, fair, decentralized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[16] Y. Liao, Y. Xu, H. Xu, L. Wang, and C. Qian, "Adaptive configuration for heterogeneous participants in decentralized federated learning," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[17] W. Zhong, C. Yang, W. Liang, J. Cai, L. Chen, J. Liao, and N. Xiong, "Byzantine fault-tolerant consensus algorithms: A survey," *Electronics*, vol. 12, no. 18, p. 3801, 2023.

[18] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[19] C. Li, W. Qiu, X. Li, C. Liu, and Z. Zheng, "A dynamic adaptive framework for practical byzantine fault tolerance consensus protocol in the internet of things," *IEEE Transactions on Computers*, 2024.

[20] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J.-S. Weng, "Bdfl: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8639–8652, 2021.

[21] Y. Liu, Z. Jia, Z. Jiang, X. Lin, J. Liu, Q. Wu, and W. Susilo, "Bfl-sa: Blockchain-based federated learning via enhanced secure aggregation," *Journal of Systems Architecture*, vol. 152, p. 103163, 2024.