Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.043684

ARTICLE

Flexible Global Aggregation and Dynamic Client Selection for Federated Learning in Internet of Vehicles

Tariq Qayyum[1], Zouheir Trabelsi[1,*], Asadullah Tariq[1], Muhammad Ali[2], Kadhim Hayawi[3] and Irfan Ud Din[4]

[1]College of Information Technology, United Arab Emirates University, Al Ain, 17551, UAE

[2]Department of Computing, School of EECS, National University of Sciences and Technology, Islamabad, 44000, Pakistan

[3]College of Technological Innovation, Zayed University, Abu Dhabi, 144534, UAE

[4]Department of Computer Science, The Superior University, Lahore, 54000, Pakistan

*Corresponding Author: Zouheir Trabelsi. Email: trabelsi@uaeu.ac.ae

Received: 10 July 2023 Accepted: 18 September 2023 Published: 29 November 2023

ABSTRACT

Federated Learning (FL) enables collaborative and privacy-preserving training of machine learning models within the Internet of Vehicles (IoV) realm. While FL effectively tackles privacy concerns, it also imposes significant resource requirements. In traditional FL, trained models are transmitted to a central server for global aggregation, typically in the cloud. This approach often leads to network congestion and bandwidth limitations when numerous devices communicate with the same server. The need for Flexible Global Aggregation and Dynamic Client Selection in FL for the IoV arises from the inherent characteristics of IoV environments. These include diverse and distributed data sources, varying data quality, and limited communication resources. By employing dynamic client selection, we can prioritize relevant and high-quality data sources, enhancing model accuracy. To address this issue, we propose an FL framework that selects global aggregation nodes dynamically rather than a single fixed aggregator. Flexible global aggregation ensures efficient utilization of limited network resources while accommodating the dynamic nature of IoV data sources. This approach optimizes both model performance and resource allocation, making FL in IoV more effective and adaptable. The selection of the global aggregation node is based on workload and communication speed considerations. Additionally, our framework overcomes the constraints associated with network, computational, and energy resources in the IoV environment by implementing a client selection algorithm that dynamically adjusts participants according to predefined parameters. Our approach surpasses Federated Averaging (FedAvg) and Hierarchical FL (HFL) regarding energy consumption, delay, and accuracy, yielding superior results.

KEYWORDS

IoT; Federated Learning; sensors; IoV; OMNeT++; edge computing

1 Introduction

The convergence of Internet of Things (IoT) devices, a myriad of sensors, and advanced Artificial Intelligence (AI) technologies have significantly contributed to the emergence and evolution of smart cities and forward-thinking businesses. Such developments are substantially owed to continuous breakthroughs in communication technologies, accompanied by the relentless expansion of computing capabilities. The treasure trove of data harnessed by these intelligent, connected devices can proffer critical insights, forming the basis for astute and informed decision-making processes. Nevertheless, the management of massive data volumes generated by an extensive array of devices remains a formidable challenge. This data overload necessitates immense computational and network resources, raising substantial concerns about privacy and security [1].

Traditionally, to leverage this data for machine learning model training, it was centralized by being transported to a primary server. However, such an approach can lead to privacy breaches and security risks, potentially discouraging users from sharing sensitive data with centralized servers [2]. Among the significant challenges in the field, ensuring the security and privacy of user data takes center stage. To tackle these challenges, FL has emerged as a powerful approach. Unlike traditional methods, in FL, the model parameters are sent to the clients instead of sending data to central servers, facilitating local-level model training. Upon completion of the training, the trained models are combined by a central server, which subsequently disseminates them to upper layers, thus effectively addressing the concerns of data privacy and resource constraints in machine learning practices [3].

As we venture into the IoV environment, smart vehicles with many sensors and high data generation capabilities burden network infrastructures significantly [4]. Cloud computing often exceeds high computational and latency-sensitive demands [5]. Moreover, edge devices, despite boasting superior computing capabilities compared to standard IoT devices, remain inadequate for high-density vehicular environments [6].

FL facilitates a solution by enabling simultaneous training of a model on multiple devices. The process begins with a central server distributing the model to edge devices. These edge devices then use their local data to train the model and send the updated version back to the central server for analysis [7]. This creates an ongoing cycle where the edge devices and the central server work together to enhance the model's performance. Integrating FL into the IoV environment can effectively address the challenges of communication overhead and privacy concerns. In vehicular communication, practical FL application involves selecting optimal nodes based on various factors and utilizing dynamic node aggregation selection [8]. This research stems from the challenges we face in today's rapidly changing technology landscape. With many devices like smart gadgets and sensors connected to the internet, we have access to a lot of data that can provide valuable insights. But handling all this data requires a lot of computer power and strong networks while ensuring people's data is safe and private. Traditional ways of training machine learning models by sending data to a central place have problems. That's why we're exploring a new approach called FL. We focus on making FL work even better in a particular setting called the IoV. In this setup, vehicles and sensors work together, creating a need for smart ways to handle data sharing, resource limits, and privacy. Our goal is to improve how FL works in the IoV context. We're creating a clever FL setup that uses flexible aggregation points and careful selection of participants, like picking the right players for a team. This way, we hope to make IoV technologies more efficient and secure.

At the start of the research, a deep dive was taken into FL within the context of the IoV. Through understanding past studies, a clear picture emerged of the opportunities and challenges of FL in various IoV scenarios. This knowledge paved the way for new and improved ideas. The proposed research introduces novel perspectives on enhancing FL's efficiency in the IoV. One standout idea is the use of flexible points for more effective data management. A smart method has been proposed to select the best clients for training the model. Local segments of the system have the capability to refine the model before sharing updates, accelerating its improvement. Moreover, an efficient approach has been identified to select the optimal node (edge server) for aggregation. These enhancements promise to make FL operate more seamlessly in the IoV and optimize the training process.

This paper proposes an intelligent FL architecture for IoV, addressing network communication overhead and employing a client selection technique for each training round. Our proposed framework uses a flexible global aggregate node and multiple local updates before sharing local variables with corresponding edge nodes. Fig. 1 illustrates the architecture of edge-enabled FL. The main contributions of this paper include the proposal of an FL architecture for IoV, enabling vehicles and mobile nodes to participate in computation-intensive tasks and model training. We have developed the client selection algorithm and evaluated the simulation framework based on GUI delay, communication delay, and workload capacity. The key features of the proposed framework are:

- To tackle network communication overhead in dispersed IoV with edge nodes of limited capacity, we propose an FL system utilizing the concept of edge computing. To overcome the challenge of a single point of failure, our proposed approach introduces the idea of a dynamic aggregation node. This approach ensures that the aggregation process is not reliant on a single node, mitigating the risk of system failure or bottleneck at a specific point. Instead, the system dynamically selects the most suitable global aggregation node based on various factors, ensuring robustness and fault tolerance in the overall system architecture.
- Understanding that not all nodes can train, and some might hinder the learning progression, we propose a method for selecting clients for each training cycle. Our newly designed client selection algorithm assesses all possible nodes. It then selects the most appropriate nodes for training based on predefined standards and benchmarks.
- We recommend using a flexible global aggregate node, selecting one edge node from many Roadside Units (RSUs) for global aggregation. This selection considers factors such as workload and communication latency.
- In our proposed framework, local nodes perform multiple local updates before sharing their local variables with corresponding edge nodes.
- In conclusion, we measure the efficacy of our suggested model. Our proposed distributed machine learning strategy is evaluated against well-known techniques based on effectiveness and efficiency. We showcase the outcomes of time delay experienced, the necessity of global communication rounds, and energy usage.

The rest of the paper is organized as follows: Section 2 provides a brief literature review, Section 3 explains the system model followed by the proposed framework in Section 4, Section 5 describes the simulation setup and evaluation, and finally, Section 6 presents the conclusion.

2 Literature Review

In this Section, an extensive literature review of state-of-the-art FL frameworks has been conducted. The potentials and challenges of FL in IoV, including V2V, I2V, and autonomous vehicles, have been thoroughly examined. We discuss existing research on FL in various specialized fields. For instance, in IoV, FL is crucial in facilitating information sharing among nodes while safeguarding their privacy. However, methods for selecting clients and designing networking strategies for FL in a mobile setting, which can significantly affect communication delays, are still not well-addressed. Recent studies have focused on significant challenges in implementing FL in the edge/edge computing paradigm [9–12]. The authors in [13] introduced an adaptable privacy-preserving aggregation technique for FL-based navigation in vehicular edge networks. This innovative method balances computational complexity and privacy protection using a homomorphic threshold cryptosystem and the bounded Laplace mechanism.

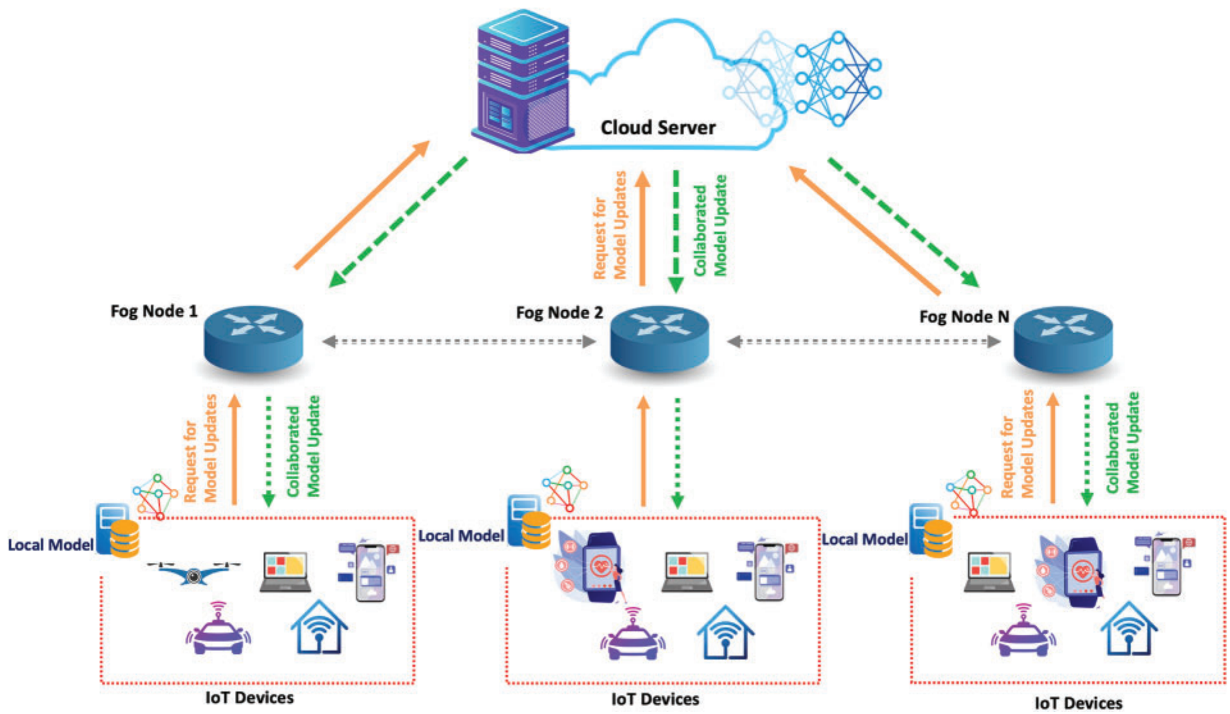

Figure 1: Illustration of edge-enabled FL architecture

Authors in [14] developed a framework where some automobiles act as edge nodes and are responsible for distributing the training model for FL. Their interactions with the environment improve the training models. A Deep FL scheme has been proposed in [15] to protect patient privacy and provide a framework for decentralizing sensitive healthcare data. They also offered an automated system for collecting training data. The authors in [16] proposed a novel FL approach to minimize uplink communication costs. Their method, known as the two-update method, was designed to improve data transmission efficiency in FL.

Additionally, a practical update method for FL was introduced in a separate study in [17]. The researchers conducted extensive empirical evaluations to assess the performance of different FL models. FMore [18] proposes to address the challenge of motivating edge nodes in Mobile Edge Computing (MEC) to engage in FL. Incentive methods used previously are inadequate for FL in MEC due to edge nodes' multi-faceted and ever-changing resources. The incentive mechanism presented, FMore, is a multi-dimensional procurement auction that selects K winners and is efficient and incentive compatible. The authors in [19] proposed an FL system for the IoT to combat attacks on IoT devices where current intrusion detection mechanisms fall short, as various manufacturers manufacture different devices. The proposed system uses FL to analyze the behavior patterns of different devices to detect anomalies and unidentified attacks.

The research work in [20] examined the problem of performance degradation when using FL and highlighted the benefits of augmenting data with an anomaly detection application for IoT datasets. The traditional healthcare system produces large amounts of data, which requires intensive processing and storage. The integration of IoT into healthcare systems has helped manage vital data, but security and data breaches remain a concern. Authors in [21] proposed an attribute-based Secure Access Control Mechanism (SACM) with FL, which addresses problems with secure access. The researchers found that SACM is particularly useful in the current environment for ensuring privacy by providing secure access control in the IoT systems. The research work in [22] presented a solution using FL and deep reinforcement learning (DRL) to enhance the management process. FL helps preserve the privacy and diversity of the data while reducing communication expenses and minimizing model training discrepancies. The authors in [23] presented resource-sharing methods in networks using edge computing. This paper introduces a design for blockchain-integrated fog clusters (B-FC), facilitating decentralized resource distribution among different Smart Machine (SM) within a fog computing setting, termed as Blockchain-Driven Resource Allocation in SM for Fog Networks (B-RSSF). Moreover, to harness the power of ubiquitous computing resources fully and efficiently, the wireless capabilities of fog computing and blockchain technology are seamlessly intertwined.

The earlier studies utilized a solitary, unchanging aggregate node in each instance. Furthermore, the selection of clients was not considered. Our analysis suggests that adopting consistent global aggregate nodes gives rise to various issues, including vulnerabilities in case of a single point of failure and potential network congestion. Additionally, involving all nodes in a training round could potentially degrade performance, as not every node is equally equipped to engage in the learning process. Limited data availability at specific nodes and energy and memory resource constraints contribute to this situation.

3 System Model

This section explains the system model of the proposed framework. We have a list edge node $R = R_1, R_2, R_3, \ldots, R_m$ that are considered as the RSU and are statically placed at the different locations of the roads. Each edge node is connected to several vehicles $V_i = v_1, v_2, \ldots, v_n$ that can volunteer for their resources for the training purpose.

Each vehicle has resources like CPU cycles CPU, energy E, memory, M. It is crucial that in an IoV, all the nodes cannot participate in the training process because of their limited resources and data. So, each edge node has a sub-module called client selection that selects some vehicles among the available lists. Along with client selection, each edge node has a resource provisioning module and performs the global and local aggregation. It is also essential to know that all the vehicles are candidate nodes, and upon selection by edge nodes, they become clients.

A dedicated communication channel in a specific region communicates between edge nodes and vehicles. Also, the number of vehicles exceeds the number of edge nodes ($V > R$). The system's edge nodes operate through a sequential process that spans multiple rounds, called r rounds. Within each round, several crucial steps are carried out. The nodes initiate a client discovery phase to identify and locate the available clients. A client selection procedure occurs, where specific clients are chosen based on predetermined criteria or algorithms. Once the clients are selected, the nodes configure them, ensuring optimal settings and compatibility for the subsequent operations. After configuration, local aggregation is performed by the nodes, where they gather and process data locally. Subsequently, a global aggregation step takes place, consolidating and analyzing data from multiple nodes to derive meaningful insights or outcomes. Finally, the processed information is transmitted to the cloud server, ensuring the cloud-based system remains updated with the most recent data and analysis from the edge nodes.

The initial stage involves edge nodes identifying clients and establishing connectivity. Subsequently, the client selection module performs the client selection using Algorithm 2. The chosen clients, denoted as C, also known as clients, proceed to update ϕ their local parameter.

The collection of local data sets, D, among the clients in C, showcases diverse and uneven properties. Within each data instance, denoted as $\gamma \in D_c$, there exist two components: X_γ, which represents a vector, and Y_γ, which represents the corresponding output. Table 1 gives a detailed reference to the symbols utilized in the paper.

Table 1: Summary of notations

Sr.	Symbol	Definition
1	R	List of edge nodes
2	ϕ_c	Local parameters at cth client
3	θ	Iteration number
4	Υ_c	Batch size at cth client
5	V	Vehicle list
6	ϱ	No. of local updates
7	Y_γ	Output corresponding to X_γ
8	e_c	Residual energy
9	P_c	CPU capability
10	CPU_c	CPU cycles for local aggregation
11	E_c	Training passes
12	ζ_F	No. of clients associated with F
13	γ_r	The workload at r
14	GA	Best edge node for global aggregation
15	$\aleph$	Accuracy of the model
16	$\tilde{\phi}$	Parameters at Global aggregation
17	$R_r(\theta)$	Network delay at R^{th} edge
18	$\varkappa$	Model size (Local)
19	ν_R	CPU at edge R
20	$R_{i,j}^{Tr}$	Transmission delay
21	$R_{i,j}^{Pr}$	Propagation delay
22	m_c	Memory
23	$R_{i,j}^{C}$	Processing delay
24	R_R^{Q}	Queuing delay
25	R_T	Transmission rate
26	R_D	Data rate
27	ϵ	No. of aggregation rounds (local)
28	Ψ	Learning rate

The loss function associated with a specific client c can be defined as follows:

$$L_c(\theta) = \frac{1}{N_c} \sum\nolimits_{\gamma \in D_c} f_\gamma(\theta) \tag{1}$$

where N_c represents the number of data samples in client c and $f_\gamma(\theta)$ is the loss function for data sample γ with parameter θ.

Each client node contributes its local dataset to the collaborative model training during the FL process. The local datasets across the clients, denoted as $D = D_1, D_2, D_3, \ldots, D_C$, exhibit varying characteristics and sizes. Each data sample γ in client c consists of a vector component X_γ an output component Y_γ. These components are used to calculate the loss function $f_\gamma(\theta)$, where θ represents the model parameters.

The FL algorithm proceeds through multiple iterations, where each client updates its local parameters based on the gradients of its loss function. The learning rate, denoted as α, determines the step size of parameter updates. The updated local parameters at client c at iteration $t + 1$ is given by Eq. (2).

$$\theta_c(t+1) = \theta_c(t) - \alpha \nabla L_c(\theta_c(t)) \tag{2}$$

where α is the learning rate. After the local updates, the parameters θ from all clients are sent to the server for aggregation.

Various factors, such as network latency and workload, are considered to determine the most suitable global aggregation node. The selection of the global aggregate node, denoted as GA, aims to minimize workload and latency. The following equation calculates the workload as follows:

$$\Gamma_R(\theta) = \sum_{i=1}^{C} x_{R,i\varkappa} \tag{3}$$

In the context of the equation provided, $\varkappa$ corresponds to the dimension of the locally updated model, while $x_{R,c} \in 0, 1$ indicates the presence or absence of a connection between the c^{th} *client* and the R^{th} edge node. Conversely, network delay is characterized as follows:

$$R_i(\theta) = R_i^R(\theta) + R_i^C(\theta) + R_i^Q(\theta) \tag{4}$$

The transmission delay between i^{th} and j^{th} edge nodes is given by:

$$R_{i,j}^T(\theta) = \left(\frac{\varkappa}{R_D}\right) \tag{5}$$

Here, R_D represents data rate. Moreover, the propagation delay between i^{th} and j^{th} edge nodes is calculated as:

$$R_{i,j}^{Pr}(\theta) = \left[\frac{R_{i,j}^{distance}\varkappa}{c}\right], \quad \forall i, j \in R \tag{6}$$

In this context, $R_{i,j}^{distance}$ represents the spatial separation between the two edge nodes, while c signifies the speed of light. The processing delay at R^{th} edge node is calculated by:

$$T_R^C(\theta) = \sum_{c=1}^{C} x_{f,c}\left(\frac{\upsilon_2}{\nu_R}\right) \tag{7}$$

The parameter υ_2 corresponds to the number of CPU cycles necessary for local aggregation, while ν_R denotes the CPU capacity of the edge node. The workload arrival rate at each edge node adheres to a Poisson process with $M/M/1$ queuing model, as outlined in [24]. The queuing delay is determined by:

$$R_R^Q(\theta) = \frac{1}{s_R, \gamma_R} \tag{8}$$

Here, s_R indicates the specific service rate allocated to edge node R. The goal is to minimize the utility function while satisfying certain constraints. The optimization problem can be formulated as follows:

$$argmin_{R\in R}\cup_R \tag{9}$$

$$s.t. R_R(\theta) \leq T^{max} \forall R \in R$$

$$\Gamma_R(\theta) \leq \Gamma^{max} \quad \forall R \in R$$

Eq. (10) represents the objective function that aims to minimize the utility $\cup_R$ for edge node f. The utility function considers various factors such as latency, workload, and other relevant metrics to determine the overall performance of the edge node. The goal is to find the edge node that provides the best utility value. The constraints specify that the latency of edge node R should not exceed a predefined maximum latency threshold ($\Gamma^{\max}$), and the workload of edge node R should not exceed a maximum workload threshold ($\Gamma^{\max}$). These constraints ensure that the selected edge node satisfies the latency and workload requirements for efficient and effective processing.

The solution to this optimization problem provides the edge node GA that minimizes the overall utility function while satisfying the given constraints.

Once the global aggregate node GA is determined, the updated global parameters are calculated as:

$$\theta_{\text{global}}(t+1) = \frac{1}{C}\sum\nolimits_{c=1}^{C} \theta_c (t+1)\, c \tag{10}$$

where $\theta_c(t+1)$ is the updated parameter of client.

These updated global parameters are then communicated to the edge nodes and clients for the next iteration of the FL process. The iterative process continues until the desired model accuracy or convergence criteria are met.

The global parameters $\theta_{\text{global}}(t+1)$ are then broadcasted to all clients for the subsequent iteration. This iterative process continues until the convergence criteria are met, which can be based on the change in the loss function or the model accuracy.

4 Proposed Framework

The proposed framework introduces a novel approach to establish seamless connections between edge nodes or RSUs and many vehicles. This enables continuous data exchange and ensures that the edge nodes have access to real-time information about the connected nodes' available resources and data volume. The primary objective is facilitating efficient and distributed learning across the network, optimizing resource utilization, and minimizing delays.

The proposed framework performs many steps simultaneously. First, the cloud sends the global model to all the RSUs. At the same time, these RSUs run the client selection algorithm to choose the best clients for training. They then share the global model received from the cloud with all the clients. The clients train the model locally and share the parameters with the RSUs.

Meanwhile, the global aggregation selection algorithm is executed in parallel to determine the best aggregation node based on communication cost and workload. Once the optimal aggregation node is selected, it performs the global aggregation and shares the updated parameters with the cloud and all the RSUs. This process continues until the limit of the number of rounds is reached. The cloud deploys an initial model across all edge nodes to initiate learning. Each edge node leverages the client selection Algorithm 2 to identify suitable participants among all the candidate vehicles. The selection algorithm considers minimum residual energy, available memory, and data records/rows to determine the most appropriate candidates. These selected participants utilize their local data to train the model and transmit the updated model back to their respective edge nodes.

The system operation described in Algorithm 1, plays a central role in orchestrating the collaborative learning process. It takes as input a list of candidate vehicles V, a list of edge nodes R, and the number of training rounds K. The algorithm proceeds by iterating through each training round, leveraging algorithm (Algorithm 2) to select clients C, initiating local aggregation parameters, and performing parallel computations for each client to calculate local model updates.

Algorithm 1: System Operation Algorithm

INPUT $V = v_1, v_2, c_3, \ldots, v_n$ ➢ List of candidate vehicles
$C \leftarrow \varnothing$ ➢ Selected clients
$R = R^1, R^2, R^3, \ldots, R^n$ ➢ List of RSU/Edge
r ➢ No. of epochs
ϱ ➢ Local updates count
OUTPUT $\theta_{\text{global}}\ (\theta)$
FOR *each* $\theta \in r$ ➢ Call client selection algorithm
  $C \leftarrow ClientSelection()$ ➢ Local parameters initialization
  $Initiate\theta_c\ (\theta)\ \forall c, \in C$
  FOR *each* $R^x \in R$
    FOR *each client* $\in C$ *in parallel* ➢ Local parameter updates Eq. (2)
      $Evaluate\theta_c\ (\theta)$
    END FOR
    $Calculate\phi_{\text{F}}^{i}\ (\theta)$ ➢ Local aggregation
  END FOR
  IF $\theta / \% \epsilon == 0$ **THEN**
    Call Global Aggregation Algorithm ➢ Global aggregation RSU selection
      $Aggregation\ \theta_{Global}\ (\theta)$
  END IF
END IF

After receiving the updates from the participating clients, each edge node proceeds to carry out local aggregation using the obtained local model updates. As described by Eq. (2), this aggregation process allows edge nodes to consolidate and synthesize the knowledge from client devices. Consequently, each edge node possesses an updated and aggregated model reflecting the collective insights of its connected clients.

The cloud employs the global aggregation node selection algorithm to determine the optimal edge node for final aggregation (Algorithm 3). This algorithm evaluates the workload and delay associated with each edge node. It selects the one with the highest utility function, as defined by Eq. (10). The selected global aggregation node is the central hub for receiving the local models from all edge nodes.

Hence, the proposed framework enables a collaborative learning process across edge nodes and participating clients. It leverages Algorithms 1 and 2 to select clients, perform local updates, and aggregate models. Fig. 2 comprehensively illustrates the entire process, showcasing the data flow and interactions between the cloud, edge nodes, and client devices. By embracing this framework, the learning process achieves efficient resource utilization, minimizes delays, and fosters a collaborative ecosystem for distributed learning.

Algorithm 2: Node Selection

INPUT $V = v_1, v_2, c_3, \ldots, v_n$ ➢ List of candidate vehicles
RE^{min} ➢ Energy threshold
M^{min} ➢ Minimum threshold
DR^{min} ➢ Data records/ threshold
OUTPUT *Selected Vehicles*
Initialize C with empty
IF $\varrho == 1$ **THEN**
FOR *each* $v \in V$
IF $v_{RE} \geq RE^{min} \& v_M \geq M^{min} \& v_{DR} \geq DR^{min}$ **THEN**
$C \leftarrow v$
END IF
END FOR
ELSE IF ***N is odd*** **THEN**
FOR *each* $v \in V$
IF $v_{RE} \geq RE^{min} v_M \geq M^{min} v_{DR} \geq DR^{min} v_a \mathrm{c} \geq Ac$
THEN
$C \leftarrow v$
END IF
END FOR
END IF
Return C

Algorithm 3: GA Node/RSU Selection Algorithm (Global Aggregation)

INPUT $\boldsymbol{R} = \{R_1, R_2, R_3, \ldots, R_n\}$ ➢ List of RSUs/Edges
D^{max} ➢ Maximum Delay
Γ^{max} ➢ Workload
OUTPUT *RSU* ➢ Aggregation Node
FOR *each* $r_x \in R$ ***DO***
Calculate $\Gamma_r(\theta)$
FOR *each* $r_y \in R$ *and* $r_x \neq r_y$ ***DO***
Calculate $D_{r_x}(\theta)$
END FOR

(Continued)

Algorithm 3 (continued)

6: **IF** $\Gamma_{r_x}(\theta) \leq \Gamma^{max} D_{r_x}(\theta) \leq D^{max}$ ***THEN***
7: *Compute* $\cup_r$
8: **END IF**
9: **END FOR**
10: $GA = arg\min_{r \in R} \cup_r$

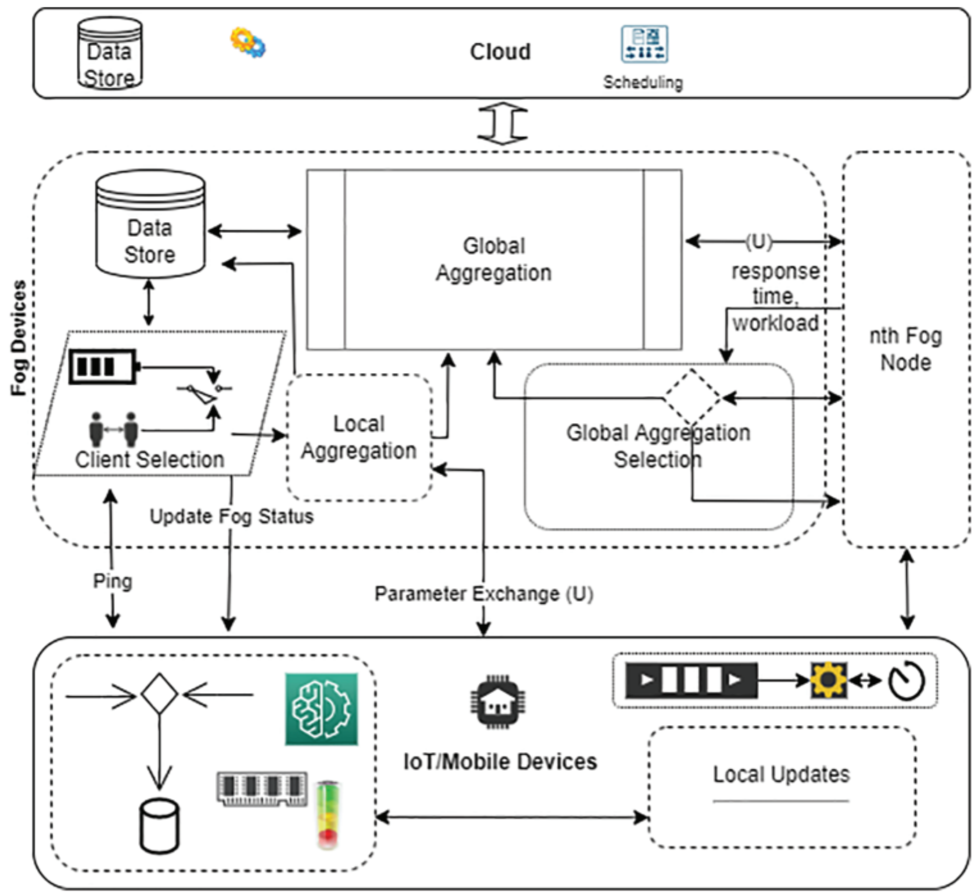

Figure 2: Operation of the system—The system establishes connections between distinct clients and their corresponding edge nodes. A comprehensive description all the steps including client selection, local training, GA selection, and global aggregation is provided

4.1 Client Selection Algorithm

The careful selection of clients for participation in the learning process is essential for its efficiency. Algorithm 2 is utilized to accomplish this, considering several vital client attributes. These selection criteria include the residual energy of clients, memory, and the minimum number of data records or rows.

This algorithm aims to identify suitable nodes from a given set of available nodes efficiently. It begins by taking an input list of available nodes, $V = v_1, v_2, v_3, \ldots, v_n$ each node possesses specific attributes, such as data records, memory, and residual energy. This algorithm utilizes minimum threshold values for these attributes, represented as RE^{min}, M^{min}, and DR^{min}. The resulting output of the algorithm is a list of selected nodes, $C = c_1, c_2, c_3, \ldots, c_m$. At the outset, the set of chosen nodes, denoted as C, is initialized as empty.

The algorithm employs two primary conditions for node selection: (1) when the parameter ϱ equals 1, and (2) when the count of nodes N is an odd number. In the first scenario, the algorithm sequentially examines each node v within the available node list V. If a node meets the minimum criteria in terms of its residual energy (v_RE), memory (v_M), and data records/rows (v_DR), it is included in the list of selected nodes C. In the second case, the algorithm again iterates through the available node list V. The selection criteria remain consistent with the first condition, but an additional stipulation is introduced: the node's acceleration (v_Ac) must surpass or equal a predetermined threshold (Ac). Nodes that fulfill all these conditions are appended to the list of selected nodes C.

4.2 Aggregation Node Selection (GA)

In the proposed framework, selecting the global aggregation node is a crucial aspect, which depends on the edge node's workload and communication delay. This selection process is detailed in Algorithm 3. The algorithm's inputs include all edge nodes, workload, and delay thresholds.

We propose an aggregation node selection algorithm to identify a global aggregation node from a given list of RSUs or edge nodes. The algorithm takes as input a list of RSUs or edge nodes, $R = R_1, R_2, R_3, \ldots, R_n$, and predefined threshold values for maximum delay D^{max} and maximum workload Γ^{max}. The algorithm's output is a single global aggregation node, GA.

The algorithm begins by iterating through each edge node R_x in the input list R. Each edge node computes the workload parameter $\Gamma_f(\theta)$ using a predefined Eq. (4). Next, the algorithm iterates through all other edge nodes R_y In the input list, ensuring that $R_x \neq R_y$, and computes the delay $D_{R_x}(\theta)$ for each pair of edge nodes using another predefined Eq. (5).

Upon calculating the workload and delay for a given edge node R_x, the algorithm checks if the computed values are within the maximum acceptable thresholds, i.e., $\Gamma_{R_x}(\theta) \leq \Gamma^{max}$ and $D_{R_x}(\theta) \leq D^{max}$. If the edge node meets these criteria, the algorithm computes a utility function U_R using another specified equation. Finally, the algorithm identifies the global aggregation node GA by selecting the edge node with the minimum utility value, as expressed by the Eq. (10) $GA = \arg\min R \in \mathcal{R} \,\&\, \cup_R$.

This approach ensures that the chosen aggregation node optimizes workload and delay parameters while adhering to the predefined maximum thresholds.

5 Evaluation

5.1 Experiment Setup

We have adopted a meticulous simulation-based methodology to ensure a thorough assessment of our proposed approach. These simulations are executed in a practical environment incorporating various tools and platforms. Specifically, we use OMNeT++\footnote {https://omnetpp.org/} along with a submodule known as Veins for simulating vehicular communication and mobility patterns. These platforms are coupled with SUMO\footnote {https://www.eclipse.org/sumo/}, which serves as a tool for simulating road networks and traffic dynamics. This comprehensive amalgamation empowers us to simulate and evaluate the performance of our proposed approach precisely. Through the synergy of these platforms, we can construct simulations that closely mirror real-world scenarios.

The experimental setup involves connecting diverse vehicles to each edge node while considering constraints such as computational capabilities, latency requirements, and energy limitations. Specifically, we simulate 500 vehicles and adjust the number of RSU nodes to determine the optimal configuration regarding performance, energy consumption, and delay. The simulation parameters are meticulously defined and documented in Table 2, providing a comprehensive analysis of the proposed technique, and showcasing its ability to meet the complex demands of the vehicular environment effectively.

Table 2: Simulation parameters

Sr.	Parameter	Value/description
1	Transmission range	300 m
2	Edge nodes	15
3	Data transmission rate	5 Mbps
4	Maximum acceleration	2.1 m/sec^2
5	Simulation runs	50 times
6	Simulation area	4.5 $\times$ 4.5 km
7	Road type	Two-way
8	Beacon size	192 B
9	Vehicle velocity	10–15 m/sec
10	MAC model	IEEE 802.11p WAVE
11	Vehicle length	2.5 m
12	Vehicle density	45–170/km

5.2 Dataset

The objective of this experiment is to assess the effectiveness of FL in the IoV domain, utilizing the widely recognized MNIST dataset as a benchmark [25]. The MNIST dataset comprises 60,000 training instances and 10,000 testing instances of handwritten patterns. Our experimental setup involves a central cloud server and multiple edge nodes connected to various client vehicles. Initially, the cloud server distributes the initial model parameters to all edge nodes. These parameters are then forwarded to respective clients for local training. Each client updates the model using its local data and sends the revised model back to the edge node for local aggregation. The RSU is responsible for global aggregation, which completes one round of the FL process, is selected based on specific criteria such as workload and communication delay, as outlined in Algorithm 3. Iterative FL continues until convergence criteria are met, and the aggregated results are transmitted back to the cloud for further utilization. To replicate real-world non-IID data scenarios in FL-based vehicular networks, the MNIST dataset is partitioned into 150 segments based on the label, with each segment randomly assigned to a client.

5.3 Discussion

The proposed distributed machine learning technique is compared to popular methods, such as HFL [26] and FedAvg [27], regarding effectiveness and efficiency. The results regarding delay incurred, global communication rounds required, and energy consumption are presented.

5.3.1 Performance Matrices

The performance evaluation of the proposed technique is meticulously examined against two prominent FL methods, namely HFL and FedAvg. The evaluation is conducted across multiple key metrics, including Accuracy, F1 Score, Precision, and Recall, over the course of the FL process. The results of this comprehensive analysis are illustrated in Fig. 3.

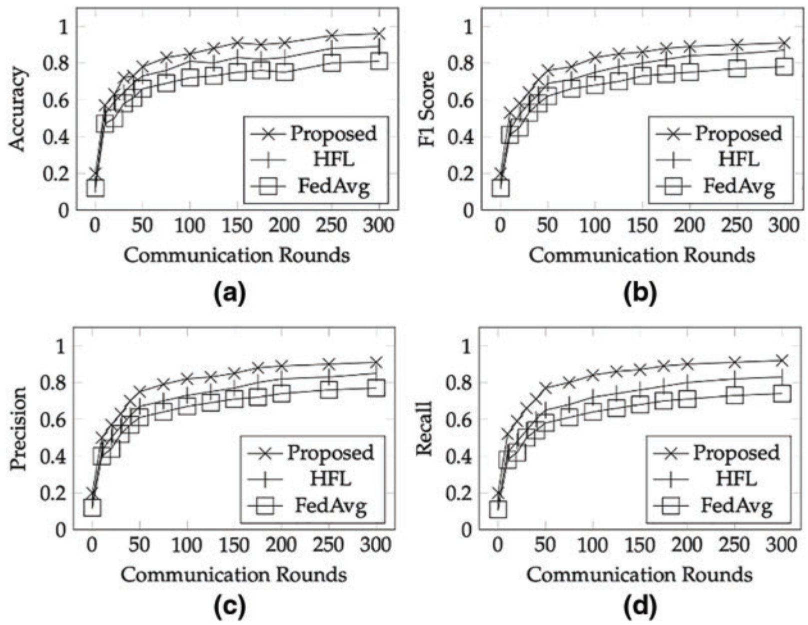

Figure 3: Comparative evaluation of the proposed technique, HFL [26], and FedAvg [27] about the accuracy, F1 Score, precision, and recall *vs*. communication rounds

In the context of Accuracy, the proposed technique consistently outperforms both HFL and FedAvg. Notably, the proposed approach demonstrates a rapid ascent to higher accuracy levels, showcasing its remarkable efficacy and proficiency in FL scenarios. This accelerated convergence to superior accuracy underscores the inherent advantage of the proposed technique. Further delving into the evaluation metrics, the F1 Score, Precision, and Recall also highlight the superiority of the proposed technique. These metrics reveal that the proposed approach attains higher and more stable values throughout the communication rounds compared to its counterparts, indicating its heightened precision in model updates and robustness in handling diverse data distributions.

The accelerated convergence and heightened accuracy of the proposed technique can be attributed to several strategic features. Primarily, a meticulous client selection process ensures the active involvement of pertinent clients, resulting in more precise model updates. Additionally, the innovative hierarchical structure of the technique fosters efficient communication between client nodes and aggregation nodes, thereby expediting the aggregation of locally computed models. The technique's adeptness at balancing task distribution across multiple edge nodes alleviates individual node burdens, leading to accurate processing of respective model updates. Moreover, the proposed approach exhibits a remarkable adaptability to varying network topologies, ensuring consistent high accuracy even amidst fluctuations in the number of client nodes or edge nodes.

In summation, the proposed technique's proficiency in client selection, communication optimization, load distribution management, and adaptability substantiate its marked accuracy enhancement and swifter convergence in distributed learning scenarios. These findings underscore the substantial contributions of the proposed approach to the advancement of FL methodologies.

5.3.2 Network Delay

Fig. 4 presents the node density effect on network delay for the proposed technique, FedAvg, and HFL by varying the percentage of edge nodes in the system. Fig. 4a through Fig. 4d represent edge node percentages of 10%, 20%, 30%, and 40%, respectively. As the percentage of edge nodes in the system increases, the proposed technique demonstrates better performance in terms of network delay compared to FedAvg and HFL.

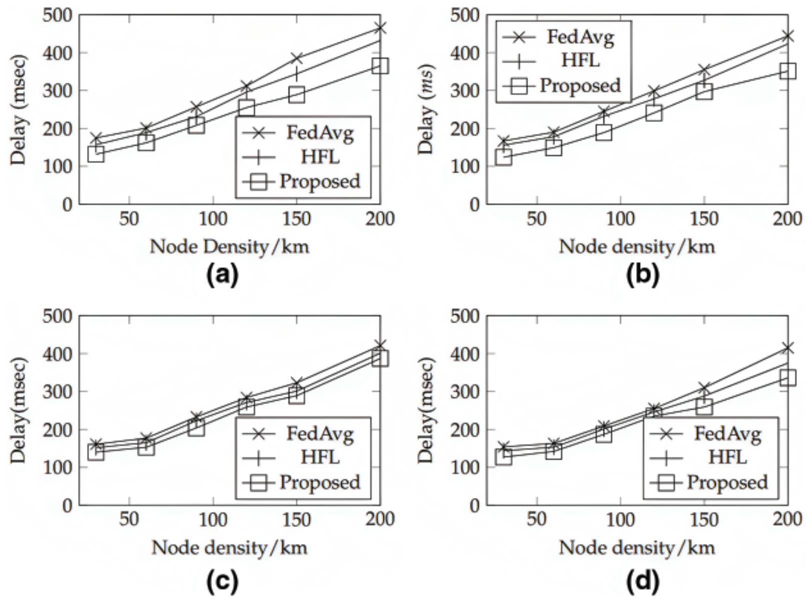

Figure 4: Comparison of energy consumption with HFL and FedAvg when the percentage of edge nodes varies from 10% to 40%. (a) edge nodes percentage = 10% (b) edge nodes percentage = 20% (c) edge nodes percentage = 30% (d) edge nodes percentage = 40%

Table 3 presents a more detailed comparison of the proposed technique, HFL, and FedAvg, with respect to the number of global communication rounds required for different numbers of computational nodes. It can be observed that the proposed technique consistently requires fewer rounds of communication, highlighting its efficiency in aggregating information from distributed nodes. This results in a faster convergence to an optimal global model while minimizing the communication overhead.

The superior performance of the proposed technique can be attributed to the efficient aggregation of locally computed models using a higher number of available nodes, reducing the load on individual computational nodes, and improving overall efficiency. Additionally, the hierarchical structure enables efficient communication between client and aggregation nodes, minimizing communication rounds and network delay. The technique's adaptability to changes in network topology further optimizes performance, even with varying numbers of client and edge nodes. Consequently, the proposed approach consistently outperforms FedAvg and HFL regarding network delay, especially with higher edge nodes. These improvements result from efficient model aggregation and balanced load distribution across nodes.

Table 3: Number of global communication rounds

Parameters	FedAvg	HFL	Proposed framework
C = 0.1	39	11	7
C = 0.2	21	9	6
C = 0.3	16	6	4
C = 0.4	13	5	3

5.3.3 *Energy Consumption*

Fig. 5 showcases the energy consumption of the suggested method, HFL, and FedAvg across varying node densities. Specifically, Figs. 5a to 5d depict various edge node percentages. The proposed approach consistently outperforms HFL and FedAvg in terms of energy consumption. This superiority is attributed to its efficient client selection process, improved communication, and balanced load distribution. The technique selects only relevant clients, reducing overall energy consumption. Its hierarchical structure minimizes communication rounds, conserving energy. Distributing the model aggregation load across multiple edge nodes also prevents excessive computational demands on individual nodes. The proposed technique maintains its energy consumption advantage as edge vehicle density increases. These benefits make it a sustainable and efficient solution for FL in distributed environments.

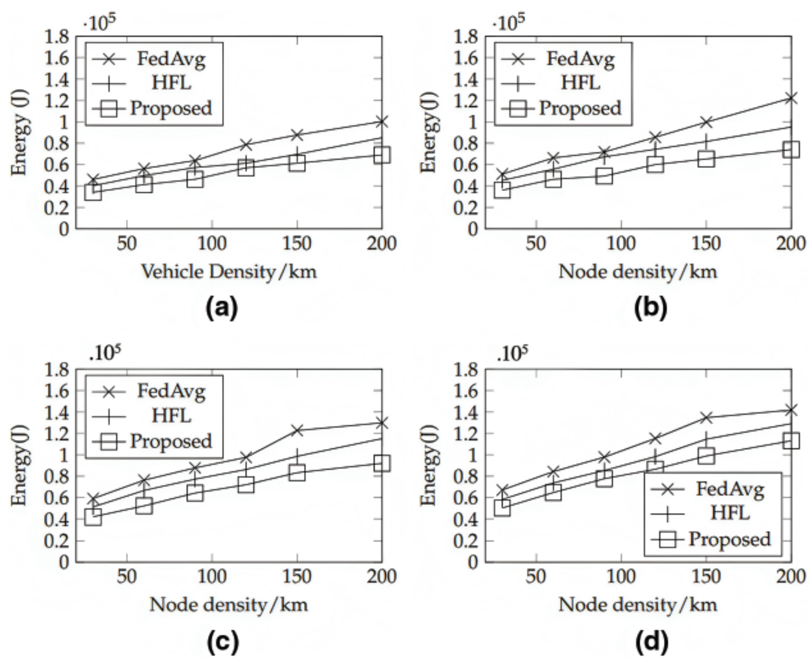

Figure 5: Comparison of energy consumption with HFL and FedAvg when the percentage of edge nodes (a) 10% (b) 20% (c) 30% (d) 40%

6 Conclusion

This study introduces a novel FL framework explicitly tailored for the IoV domain. The proposed strategy presents an advanced approach to selecting a global aggregation node, meticulously factoring in essential elements like workload and communication latency. Additionally, a client selection technique is seamlessly integrated, accounting for participating nodes' computational capabilities and energy reserves. The experimental outcomes effectively affirm the prowess of our system, showcasing its supremacy in energy consumption, latency reduction, and accuracy enhancement when contrasted with established methodologies like FedAvg and HFL. This research marks a notable stride in integrating FL into IoV, effectively tackling pivotal hurdles related to network constraints, computational resources, and energy optimization. In summation, our results firmly establish the viability and efficacy of FL for decentralized machine learning within the IoV realm. The presented framework yields promising outcomes and usher's prospects for further advancements in this arena. Future endeavors will encompass meticulous experimentation and simulation studies spanning diverse system configurations and network scenarios. This comprehensive approach will thoroughly validate and assess the performance of the proposed technique, thus enhancing our understanding and contributing to the progression of this field.

Acknowledgement: We would like to express our sincere gratitude to all the authors who have contributed to completing this research paper.

Funding Statement: This work was supported by the UAE University UPAR Research Grant Program under Grant 31T122.

Author Contributions: Study conception and design: T. Qayyum, A. Tariq, Z. Trabelsi; data collection: T. Qayyum, K. Hayawi, A. Tariq; analysis and interpretation of results: T. Qayyum, A. Tariq, I. U. Din; draft manuscript preparation: T. Qayyum, A. Tariq, M. Ali. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We used MNIST dataset, and the references are added in the bibliography section under [26].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Hu, M. Qiu, J. Li, T. Grant and D. Taylor, "A review on cloud computing: Design challenges in architecture and security," *Journal of Computing and Information Technology*, vol. 9, no. 1, pp. 25–55, 2011.

[2] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang and B. Liu, "Applications of federated learning in smart cities: Recent advances, taxonomy, and open challenges," *Connection Science*, vol. 34, no. 1, pp. 1–28, 2022.

[3] R. S. Antunes, C. A. Costa, A. Küderle, I. A. Yari and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 4, pp. 1–23, 2022.

[4] J. Contreras-Castillo, S. Zeadally and J. A. I. Guerrero, "Internet of vehicles: Architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[5] M. Aazam, S. Zeadally and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, 2018.

[6] O. A. Wahab, A. Mourad, H. Otrok and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 21342–21397, 2021.
[7] J. Pei, K. Zhong, M. A. Jan and J. Li, "Personalized federated learning framework for network traffic anomaly detection," *Computer Networks*, vol. 209, pp. 108906, 2022.
[8] Z. Du, C. Wu, T. Yoshinaga, K. L. A. Yau and Y. Ji, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2021.
[9] J. Posner, L. Tseng, M. Aloqail and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Network*, vol. 35, no. 2, pp. 152–159, 2021.
[10] 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.
[11] F. Majidi, M. R. Khayyambashi and B. Barekatain, "HFDRL: An intelligent dynamic cooperate cashing method based on hierarchical federated deep reinforcement learning in edge-enabled IoT," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1402–1413, 2022.
[12] A. Tariq, R. A. Rehman and B. S. Kim, "EPF—An efficient forwarding mechanism in SDN controller enabled named data IoTs," *Applied Sciences*, vol. 10, no. 21, pp. 7675, 2020.
[13] Q. Kong, F. Yin, R. Lu, B. Li and X. Wang, "Privacy-preserving aggregation for federated learning-based navigation in vehicular fog," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8453–8463, 2021.
[14] W. Bao, C. Wu, S. Guleng, J. Zhang and K. L. A. Yau, "Edge computing-based joint client selection and networking scheme for federated learning in vehicular IoT," *China Communications*, vol. 18, no. 6, pp. 39–52, 2021.
[15] H. Elayan, M. Aloqaily and M. Guizani, "Deep federated learning for IoT based decentralized healthcare systems," in *Int. Wireless Communications and Mobile Computing (IWCMC)*, Harbin, China, pp. 105–109, 2021.
[16] M. Chen, O. Semiari, W. Saad, X. Liu and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 177–191, 2020.
[17] N. I. Mowla, N. H. Tran, I. Doh and K. Chae, "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in fanet," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 244–258, 2020.
[18] R. Zeng, S. Zhang, J. Wang and X. Chu, "Fmore: An incentive scheme of a multi-dimensional auction for federated learning in MEC," in *Int. Conf. on Distributed Computing Systems (ICDCS)*, Taipei, Taiwan, pp. 278–288, 2020.
[19] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni and N. Asokan, "DIoT: A federated self-learning anomaly detection system for IoT," in *Int. Conf. on Distributed Computing Systems (ICDCS)*, Dallas, Texas, USA, pp. 756–767, 2019.
[20] B. Weinger, J. Kim, A. Sim, M. Nakashima and N. Moustafa, "Enhancing IoT anomaly detection performance for federated learning," in *Int. Conf. on Mobility, Sensing, and Networking (MSN)*, Tokyo, Japan, pp. 206–213, 2020.
[21] H. Lin, K. Kaur, X. Wang, G. Kaddoum, J. Hu *et al.,* "Privacy-aware access control in IoT-enabled healthcare: A federated deep learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2893–2902, 2021.
[22] P. Zhang, C. Wang, C. Jiang and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8475–8484, 2021.
[23] S. Rani, D. Gupta, N. Herencsar and G. Srivastava, "Blockchain-enabled cooperative computing strategy for resource sharing in fog networks," *Internet of Things*, vol. 21, pp. 100672, 2023.

[24] A. W. Malik, T. Qayyum, A. U. Rahman, M. A. Khan and O. Khalid, "xFogSim: A distributed fog resource management framework for sustainable IoT services," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 4, pp. 691–702, 2021.

[25] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] H. S. Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, Florida, pp. 1273–1282, 2017.