www.nature.com/scientificreports

scientific reports

OPEN

Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering

Fahad Razaque Mughal[1], Jingsha He[1], Bhagwan Das[2]✉, Fayaz Ali Dharejo[3], Nafei Zhu[1]✉, Surbhi Bhatia Khan[4] & Saeed Alzahrani[5]

In the rapidly growing Internet of Things (IoT) landscape, federated learning (FL) plays a crucial role in enhancing the performance of heterogeneous edge computing environments due to its scalability, robustness, and low energy consumption. However, one of the major challenges in such environments is the efficient selection of edge nodes and the optimization of resource allocation, especially in dynamic and resource-constrained settings. To address this, we propose a novel architecture called Multi-Edge Clustered and Edge AI Heterogeneous Federated Learning (MEC-AI HetFL), which leverages multi-edge clustering and AI-driven node communication. This architecture enables edge AI nodes to collaborate, dynamically selecting significant nodes and optimizing global learning tasks with low complexity. Compared to existing solutions like EdgeFed, FedSA, FedMP, and H-DDPG, MEC-AI HetFL improves resource allocation, quality score, and learning accuracy, offering up to 5 times better performance in heterogeneous and distributed environments. The solution is validated through simulations and network traffic tests, demonstrating its ability to address the key challenges in IoT edge computing deployments.

Keywords Internet of things, Resource management, Edge artificial intelligence, Heterogeneous cluster networks, Edge computing, Federated learning

The rapid increase of Internet of Things (IoT) devices and transportable technology in recent years has led to an exponential increase in application data generation[1]. According to a Cisco report[2], around 2.32 zeta bytes of data are produced daily at the network edge of 2023. This enormous volume of edge data has driven the adoption of edge computing, which allows efficient local data processing by shifting computation closer to the edge[3]. Furthermore, edge computing has enabled the emergence of federated learning (FL), a distributed machine learning (ML) approach that collaboratively trains models across edge nodes[4]. The surging scale of edge devices and data has made edge computing and federated learning pivotal technologies for managing and extracting from massive decentralized edge data.

To analyze the massive amounts of edge data, machine learning approaches like Federated Learning (FL) can enable intelligent services[5,6]. Different regions in heterogeneous networks produce large amounts of sensitive data. Thus, preserving the privacy of this data is crucial. Traditionally, centralized model training is widely used, collecting vast amounts of raw data from devices to a centralized edge server[7,8]. Despite providing accuracy, centralized methods encounter privacy violation risks and limits on transfer capacity. In heterogeneous networks, devices generate imbalanced, non-independent, and nonidentical distributed data. Synchronous and asynchronous FL methods generate long training latency and require massive communication resources[9]. A semi-asynchronous FL mechanism combines both methods' advantages or forces local models' synchronization while performing aggregation asynchronous. This balances computing efficiency (training accuracy and latency) and communication efficiency (communication rate and transmission latency). However, parts of selected devices still need to wait for the slowest device to complete their training, wasting their computing resources[10].

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. [2]Centre for Artificial Intelligence Research and Optimization (AIRO), Design and Creative Technology Vertical, Torrens University, 46-52 Mountain Street, Ultimo 2007, NSW, Australia. [3]Computer Vision Lab, CAIDAS, IFI, University of Wurzburg, Würzburg, Germany. [4]School of Science, Engineering and Environment, University of Salford, Salford, UK. [5]Management Information System, King Saud University, Riyadh, Saudi Arabia. ✉email: bhagwan.das@torrens.edu.au; znf@bjut.edu.cn

In another research, the author investigates the complexities of workload distribution and task scheduling in edge computing environments, focusing on dynamic resource allocation strategies[11]. It provides a comprehensive overview of management techniques for optimized performance and efficient resource utilization. RoofSplit[12], an edge computing framework that splits CNN models for collaborative inference across heterogeneous nodes. It optimizes split layers based on node capabilities and network conditions, minimizing latency. Test results show it reduces inference latency by up to 63%. A heterogeneous graph neural network approach is used for mobile app recommendation at the edge, using a graph to model user, app, and context interrelations[13]. The approach provides 11% higher accuracy and 21x lower latency than centralized methods. It enables personalized app recommendations while preserving user privacy. The approach also offers enhanced performance, scalability, and agility through the orchestrated use of heterogeneous resources[14]. Security considerations are crucial in this environment, as computation-intensive tasks can be offloaded to the cloud while latency-sensitive operations thrive in edge environments[15].

The integration of edge computing and IoT technologies revolutionizes data processing, analytics, and services at the network's edge, reducing latency and enhancing real-time analytics[16,17]. Decentralized processing for IoT applications improves resource management, energy efficiency, and optimization strategies. The challenges of device heterogeneity and dynamic environments require efficient resource allocation. Federated deep learning algorithms, with gating mechanisms and optimized aggregation weights, improve accuracy and speed and enable collaborative deep learning on diverse edge computing hardware under non-IID data distributions[18]. Machine learning techniques are designed to optimize model deployment and execution on edge devices, ensuring efficient deployment and execution[19]. These techniques include model compression, quantization, and optimization for edge-specific hardware. They focus on adapting models to edge devices' varying data distributions and characteristics, using federated learning and collaborative model training strategies. This privacy-preserving approach addresses concerns in heterogeneous edge environments[20].

A flexible data fusion approach for object positioning using heterogeneous sensors in edge computing uses an adaptive Kalman filter and edge servers to coordinate calibration and sensor selection[21,22]. This strategy reduces positioning errors by up to 41% compared to static fusion. Acceleration techniques for decentralized, federated learning on heterogeneous edge devices include sharding, aggregating updates, and caching frequently used parameters. These techniques provide efficient, decentralized, federated learning on heterogeneous edge networks[23].

We introduce a novel Multi-Edge Clustering and Edge AI architecture for Federated Learning MEC-AI (HetFL) architecture for diverse networks. It uses asynchronous edge devices, federated learning, and synchronous Edge AI and FL mechanisms. This allows training across heterogeneous networks and IoT devices, ensuring collaborative learning, diversity quality scores, and accuracy improvements. Using MEC-AI (HetFL) across three layers:

- EDGE CLUSTERS-DEVICES layer: Every edge cluster selects multiple IoT/edge devices depending on the arrival sequence in each iteration. A node selection strategy ensures enhanced asynchronous selected devices are allowed repetitively in their quality score to handle system heterogeneity between edge clusters and end devices.
- MEC-IN-EDGE AI layer: Edge AI nodes select local models from edge clusters based on arrival order for each round. An edge AI update approach maintains leverage synchronously within each edge cluster for efficient model individuality, allowing chosen clusters to iteratively retrain local models during waiting to increase accuracy.
- MEC-AI-HetFL layer: performs synchronous heterogeneous federated learning aggregates client models from all edge AI nodes to share information. Edge AI maintains the model's individuality, allowing chosen clients to retrain global models to enhance accuracy further repetitively.We propose a state-of-the-art MEC-AI (HetFL) with an asynchronous mechanism, synchronous edge AI and FL mechanism architecture, and an optimization framework for heterogeneous federated learning. We propose an optimization objective for joint device node selection and resource allocation to maximize efficiency. We also propose an Edge Node Selection algorithm and low-complexity EDGE AI/FL approach for collaborative learning of selection and allocation strategies compared to existing work EdgeFed[24], FedSA[25], FedMP[26], and H-DDPG[27]. A detiled overview of these strategies is given in the related work section.
-
- We introduce a real-time asynchronous FL approach for edge AI-IoT device layers in a MEC-AI (HetFL) architecture. Edge clusters can locally select and train on devices with new/updated data in an asynchronous manner without needing to wait for other devices. This allows training to continue even if some devices are unavailable. Edge AI modules on each cluster can independently select optimal local models based on the sequence and types of data/tasks seen by available devices. This enables personalized localized models. Devices that contribute high quality data can be prioritized and retrained more regularly to further refine models for their usage patterns and environments. This iterative retraining leads to improved overall accuracy.
- We propose an EDGE AI algorithm that applies MEC-AI (HetFL) to solve the Edge cluster selection heterogeneous devices combinational problem. A joint approach is developed to allocate Edge node resources and select appropriate devices for each cluster based on their capabilities and available resources. Constraints like communication bandwidth and device/cluster computational power are considered to achieve efficient resource utilization over the heterogeneous network. Devices are selected based on volumes and types of data generated as well as whether they provide data in a continuous Stream or discrete batches. This matches device properties to cluster needs.
- The novelty of the proposed MEC-AI HetFL framework lies in its ability to efficiently address the challenges posed by heterogeneous edge computing environments and non-IID data distributions, which are not

adequately tackled by existing federated learning approaches. Unlike conventional frameworks, such as EdgeFed[24], FedSA[25], FedMP[26], and H-DDPG[27], which either suffer from high communication costs or inefficiencies in node selection, MEC-AI HetFL introduces a multi-edge clustering mechanism combined with an asynchronous edge AI update strategy. This innovation allows the framework to dynamically select edge nodes and optimize resource allocation in real-time, significantly improving training speed and model accuracy. Furthermore, our Edge Node Selection Algorithm incorporates a dual-factor quality score (combining network conditions and AI expertise), enabling more precise and efficient node selection, which reduces computational overhead and enhances the scalability of the system. The extensive performance evaluation demonstrates that our approach achieves up to 5 times better performance in communication efficiency and model accuracy compared to the state-of-the-art methods, proving the clear advantage and applicability of our framework in real-world IoT scenarios.

Related work

With the increasing popularity of the Internet of Things (IoT), significant amounts of data are generated from the physical world per second[28]. These vast amounts of data are traditionally forwarded to the remote cloud for processing and training. This may cause a significant delay due to long-distance transmission and potential privacy leakage. To this end, edge computing is proposed to shift more computation to the network edge, enabling efficient data processing locally. Besides, it also promotes the role of federated learning (FL), which performs distributed machine learning over edge nodes[29].

Researchers have presented three main federated learning (FL) paradigms: synchronous, asynchronous, and semi-asynchronous[30]. The first type, Synchronous FL[31], creates a global model by combining local models with devices. This improves synchronous FL performance through efficient resource allocation, incentives, and data aggregation. However, synchronous FL efficiency suffers from long waits for diverse edge devices. Current solutions address this by selecting a subset for edge aggregation, but idle devices waste computing resources. Balancing subset participation with full resource utilization remains a challenge[32,33].

The second type, Asynchronous FL[34], this method where a model is sent to an edge server, combining all models and sending the latest global model. This improves round duration by eliminating waiting time but has high communication costs and reduced accuracy due to uneven device participation. Achieving a balance between these two types remains a challenge, as other devices continue training during aggregation to avoid waiting for slow devices and shorten round duration[35]. The third type, Semi-asynchronous FL[36], combines synchronous and asynchronous mechanisms to improve round efficiency and convergence. It optimizes participant selection based on edge characteristics, enhancing accuracy and resource utilization. However, current solutions lack data-based, heterogeneity-based, and utilization-based participant selection, which can reduce accuracy. SAFA considers heterogeneity and data distribution but still underutilizes idle computing resources from non-selected devices[37].

The paper[38], presents an efficient microservices offloading strategy aimed at optimizing costs in diverse Mobile Edge Computing (MEC) cloud networks. It explores dynamic resource allocation and service migration to reduce operational expenses while maintaining performance. The approach enhances cost efficiency across heterogeneous MEC environments by effectively managing microservices distribution. In this[39] paper, proposes a secure workflow scheduling algorithm that leverages hybrid optimization techniques in mobile edge computing (MEC) environments. It focuses on enhancing both security and performance while managing workflow tasks across distributed edge nodes. The approach ensures efficient task scheduling with reduced latency and optimized resource usage in dynamic MEC settings.

EdgeFed[24], new federated learning framework for efficient machine learning in edge computing environments . It uses a smart training approach, focusing on a subset of edge devices based on available resources and data relevance. The framework also employs an optimization strategy to maximize global model performance and minimize resource consumption across the edge network. Asynchronous model updates reduce communication overhead. The author in[25] introduces FedSA, a semi-asynchronous federated learning technique for heterogeneous edge computing networks. FedSA allows edge devices to perform local model updates independently, allowing for flexibility in handling intermittent connectivity and statistical differences. It uses a weighted aggregation method to integrate local model updates, making federated learning more robust for environments with variable device participation and diverse data distributions.

The author in[40] presents an energy-efficient task-scheduling algorithm for mobile edge computing based on traffic mapping. The algorithm minimizes mobile energy consumption under latency constraints, mapping computational tasks to edge or cloud resources based on network traffic profiles and device locations. This approach reduces energy usage by up to 41%, promoting green operation in heterogeneous edge computing environments.The paper[41] introduces “Lasagna”, a novel layered design approach for integrating aerial and ground-based network infrastructure, explaining its architecture and benefits for improving network coverage and performance.

The author in[42] presents a heterogeneous edge computing cluster consisting of Raspberry Pis, Jetson Nanos, and servers for resource monitoring and performance benchmarking. Kubernetes manages container orchestration, and Prometheus allows system-level monitoring of resources. Benchmarking tools evaluate workloads for machine learning, streaming, and storage use cases. Results quantify hardware tradeoffs for cost, power, and performance, providing insights into real-world deployment and management.

The author in[43] introduces an intelligent scheduling method using ant colony optimization for heterogeneous edge computing systems. The method aims to minimize energy consumption while meeting latency constraints. Simulations show up to 62% reduced energy usage compared to round-robin and greedy scheduling, offering an adaptive online scheduling strategy. The paper[44] presents a Broad Learning System (BLS) combined with Takagi-

Sugeno fuzzy logic to identify tobacco origins using near-infrared spectroscopy data, offering an improved method for tobacco authentication and classification.

The author in[26], the FedMP framework is a federated learning approach that enhances communication efficiency and addresses device heterogeneity. It uses a personalized model pruning technique, allowing each device to train a pruned subset of the model tailored to its local dataset. This reduces communication overhead for model updates and accounts for differences in device capabilities. FedMP enables efficient collaborative learning across devices with heterogeneous data distributions and capabilities, introducing innovations in adaptive model pruning and aggregation.

The author in[27] proposed an improved hierarchical federated learning framework for heterogeneous edge computing environments. It uses multiple edge servers to orchestrate collaborative learning across tiered edge devices. The model splitting technique partitions the model into public and private sections based on parameter sensitivity, ensuring selective sharing for privacy. An adaptive communication protocol adjusts bandwidth usage between hierarchy tiers to reduce communication costs. These innovations improve the federated learning efficiency, privacy, and scalability across diverse IoT edge devices.

In[45], the author presents an open-source edge computing platform for smart grid data analytics using Raspberry Pis, Jetson Nanos, and servers. Kubernetes enables container orchestration across devices, and distributed streaming and machine learning processes analyze real-time meter data. The platform can process smart grid measurements under cost, power, and hardware constraints. This[46] paper also investigates secure task allocation for edge computing across heterogeneous devices, focusing on minimizing latency and balancing workloads. The joint optimization approach provides lower latency and up to 29% higher security compared to baseline schemes, enabling efficient and secure edge computing.

The author in[47] proposed energy-efficient resource allocation in heterogeneous networks with parked vehicle assistance. It develops an optimization framework to minimize transmission power across cellular and V2V links, reducing network power consumption by up to 37% compared to standard LTE-V2V. The author in[48] proposed a novel resource management framework for heterogeneous clustered networks, utilizing intra-cluster federated learning. The proposed methodology employs asynchronous federated averaging to concurrently optimize resource allocation across diverse clusters while maintaining data locality. This approach shows promise for various distributed computing applications and merits further investigation to fully assess its effectiveness and broader applicability.

The paper[49] proposed a Vehicle-to-Vehicle (V2V) routing protocol for Vehicular Ad-hoc Networks (VANETs) based on the Autoregressive Integrated Moving Average (ARIMA) model to predict vehicle mobility patterns and improve routing efficiency. The paper[50] investigates Ultra-Reliable Low-Latency Communication (URLLC) for UAVs during pylon turn maneuvers, focusing on optimizing Age of Information (AoI) to maintain fresh data transmission under challenging flight conditions.

The author in[51] explores the optimization of social-aware and mobility-aware computation offloading in mobile edge computing systems. It develops a decentralized algorithm that selects edge servers based on user mobility and social ties, resulting in up to 63% cost reduction. In[52], the author proposed S-Edge, an adaptive traffic signal control framework using lightweight CNN models for traffic flow prediction. It reduces delays by up to 29% and improves utilization by up to 17% compared to fixed timing policies, enabling responsive traffic signal control from the edge. The author in[53] investigated multi-layer computation offloading in mobile edge computing networks, proposing a decision framework for optimizing task offloading based on device mobility and capabilities. The online learning algorithm estimates layer latencies, reducing task latency by 41%.

Proposed work

In this section, our approach is discussed in detail. The proposed MEC-AI (HetFL) model is primarily introduced first, as shown in Fig. 1. We describe our proposed MEC-AI (HetFL) Algorithm 1 for the IoT network's edge node selection Algorithm 2 for distributed learning. Finally, our proposed MEC-AI (HetFL) algorithm will carry out the three primary steps for federated learning in the edge cluster network : (intra-cluster, inter-cluster), local model, and global model update.

The model operates through a three-layered structure to optimize federated learning in heterogeneous edge computing environments. The first layer, Edge Clusters-Devices Layer, dynamically selects edge devices based on their data quality and computational resources, enabling efficient local model updates. The second layer, MEC-in-Edge AI Layer, performs local model training and aggregation within clusters, leveraging asynchronous updates to minimize communication latency and enhance resource utilization. The final layer, MEC-AI-HetFL Layer, handles global model aggregation across clusters, ensuring that local models are synchronized and updated globally while maintaining model individuality. To facilitate this process, we introduce the Edge Node Selection Algorithm, which computes a quality score (Q) for each node based on network conditions and AI expertise, ensuring optimal node selection for workload distribution. The MEC-AI (HetFL) Algorithm details the step-by-step process of intra-cluster and inter-cluster model aggregation, defining how local models are iteratively refined and synchronized to improve accuracy. This hierarchical and adaptive approach enables efficient federated learning in IoT environments, significantly reducing communication costs and enhancing model accuracy under non-IID data distributions.

Federated learning is a potential solution for connecting devices using machine learning and the Internet of Things (IoT). This approach gathers device data samples and conducts training locally, saving resources on wireless connectivity and security risks. As illustrated in Fig. 1, We implement the MEC-AI (HetFL) federated learning framework on computing nodes interconnected by an IoT and heterogeneous network. The edge node is the central variable server, while several IoT devices train the same learning model. IoT devices can gather unprocessed information from their operational scenarios. Existing federated learning approaches demand homogeneous systems and synchronous updates, limiting effectiveness for heterogeneous IoT networks.

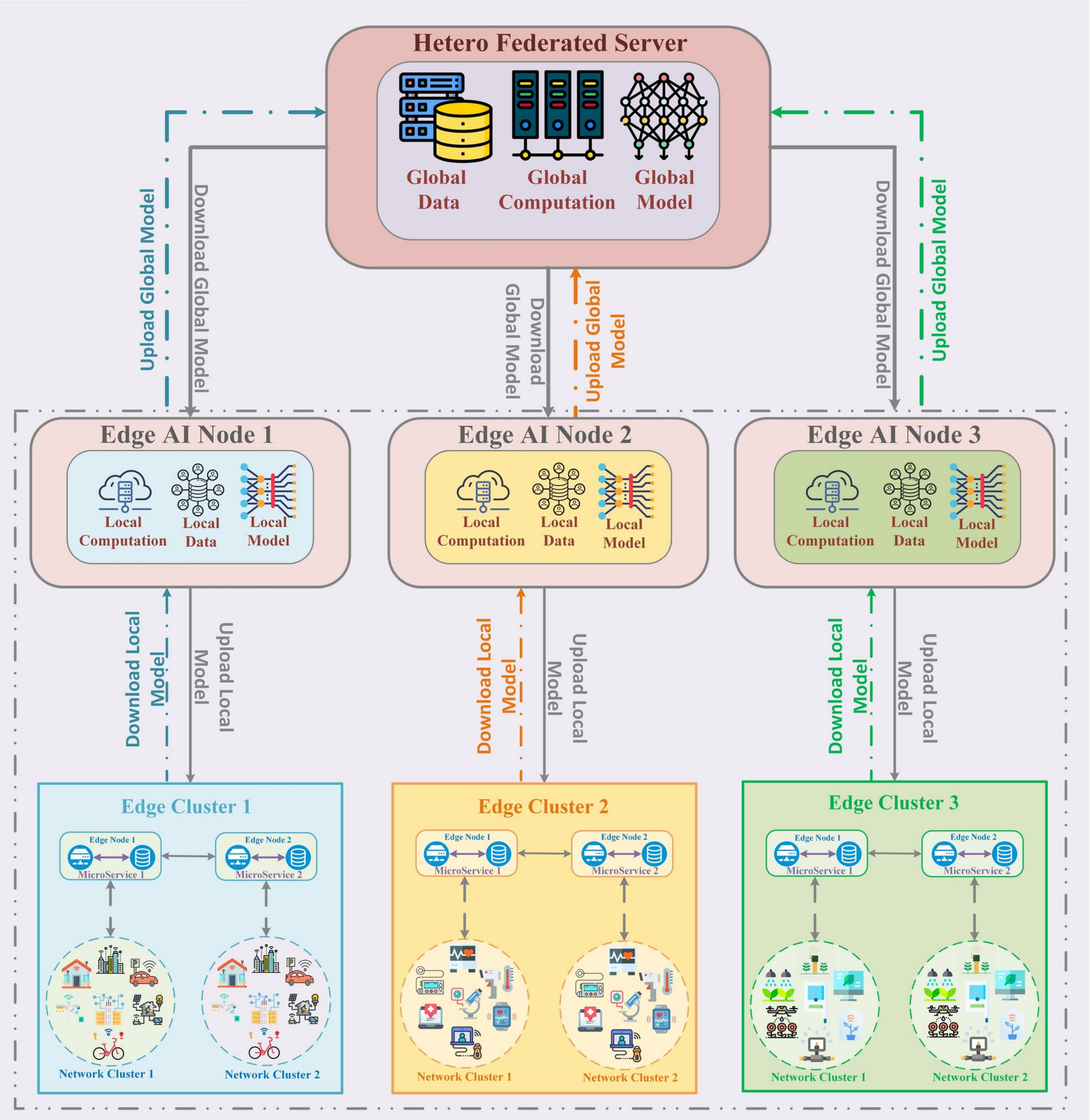

Figure 1. Our proposed MEC-in-edge AI with MEC-AI (HetFL).

This paper presents MEC-AI HetFL to address such issues. The model dynamically selects nodes for model aggregation based on metrics like bandwidth and convergence, enabling asynchronous learning across diverse devices and improving resource-efficiency of collaborative machine learning tasks in IoT settings.

MEC-in-edge AI with federated learning

The MEC-AI (HetFL) technique aims to develop an autonomous edge AI node through the integration of edge artificial intelligence techniques, wireless communication, and a convergence system. It involves local computation, data collection, and model training at edge nodes as illustrated in Fig. 1. Edge AI is introduced in the context of the MEC-AI framework to address the challenges arising from the characteristics of the heterogeneous wireless edge environment. These include substantial training data volumes that require compression for efficient uplink transfer, privacy concerns with transmitting sensitive device data to edge servers, and maintaining consistency between local and global model updates while preserving user privacy. Edge AI and FL processing aims to handle these issues.

The proposed architecture includes a "MEC-AI" in Edge AI FL and Edge nodes, which can integrate the FL and significant Edge AI Nodes to create a robust artificial intelligence creature with high intelligence ability. Each Edge AI node provides system-level, dynamic assistance for AI tasks, internal optimization, and balancing, enhancing the overall balance and optimization of the MEC-AI (HetFL) systems.

This paper investigates a novel Multi Edge Clustered and Edge AI Heterogeneous Federated Learning (MEC-AI HetFL) architecture inspired by the effective communication process in MEC-AI (HetFL). The architecture involves multiple edge AI nodes collaborating to organize many edge nodes, updating the model locally, and uploading it to connected edge nodes for intra-cluster model aggregation. Edge AI nodes carry out the process of inter-cluster model aggregation following some intra-cluster model aggregations. This process involves frequent model interactions and collections of nearby edge nodes, allowing many client nodes to examine their data efficiently. The study also suggests that edge AI nodes communicate once, potentially increasing training accuracy due to model consistency across edge AI nodes.

The Edge AI in (MEC-AI HetFL) system here, $\theta_c(t)$ represents the local model weights of the Edge AI node *c* at iteration *t*, η represents the learning rate, ∇ represents the gradient operator, $\mathcal{L}$ means the loss function, D_e represents the data on Edge device *e*, D_c represents the data on the client node of Edge cluster *c*, *K* represents the number of iterations, *T* denotes the overall number of iterations. In contrast, *t* is the current state of iteration, and *Nbc* represents a number of inter-cluster model accretions. The algorithm performs iterative updates of the local models and collection of the global model across edge clusters to optimize the overall performance of the MEC-AI (HetFL) framework.

The algorithm 1 described above, Edge AI in MEC-AI (HetFL), aims to optimize the edge clusters and devices by utilizing edge AI nodes. The algorithm starts by initializing all edge devices' environments with the edge clusters and the same model (*C* converts to *E*). Additionally, the edge AI nodes of all edge nodes initialize. The algorithm iterates over the number of iterations $(k = 1, 2, ..., K)$. Within each iteration, the algorithm performs the following steps for each edge device *e* in parallel, and the algorithm determines the value of local model updates. the value of *t* is divisible by a specific factor $(modt = 0)$. In that case, the algorithm performs the following steps: intra-cluster model aggregation For each edge AI node *c* in parallel, the edge nodes receive data from the client nodes in *Ec* and update their local model $\theta_c(t)$ using the gradient descent update rule can be seen in (1):

$$\theta_c(t) = \theta_c(t-1) - \eta\nabla(L(\theta_c(t-1), D_e)) \tag{1}$$

Edge AI nodes on client nodes train their data using the updated local model $\theta_c(t)$ to optimize it through intra-cluster model aggregation can be seen in (2):

$$\theta_c(t) = \theta_c(t) - \eta\nabla(L(\theta_c(t), D_c)) \tag{2}$$

Global Model Aggregation: If the value of t is divisible by a specific factor $(modt = 0)$, the algorithm performs the following steps. Inter-Cluster Model Aggregation. The algorithm shares the models with *Nbc* inter-cluster model aggregations and computes the global model $\theta(t+1)$ as the average of the local models can be seen in (3):

$$\theta(t+1) = \frac{\sum\limits_{c} \theta_c(t)}{N_{bc}} \tag{3}$$

Updates the global model $(\theta(t+1))$ by incorporating edge AI nodes' information exchange to achieve consensus. The updated model is then broadcasted to client nodes in *Ec*. The final global model is then returned.

```
Input: Initialize all Edge devices with the Edge clusters in the environment.
Initialize the Edge clusters, and the same model $C$ converts to Edge nodes $E$.
Initialize the Edge AI nodes of all Edge nodes.
for each $k = 1, 2, \ldots, K$ do
    for each Edge device $e \in E$ in parallel do
        The Edge device, according to the value of $t = 1, 2, 3, 4, \ldots, t$.
        if mod $t = 0$ then
            for each edge AI node $c \in S$ in parallel do
                The Edge nodes from the client nodes in $E_c$ update the local model Edge AI Nodes as $\theta_c(t)$ according via (1)
                The Edge AI Nodes of the client node train its data to optimize $\theta_c(t)$ with the use of intra-cluster accretion via (2)
            end for
            if mod $t = 0$ then
                Set $\theta(t) = \theta_c(t)$.
            end if
            for $\tau = 1$ to $T$ do
                Share models with $N_{bc}$ inter-cluster aggregation and perform Global $\theta(t+1)$ according via (3)
                The Edge AI nodes Receive the Global Update $\theta_c(t+1)$ according to: $\theta_c(t+1) = \theta(t+1)$
                The Edge AI nodes Broadcast $\theta_c(t+1)$ to the client nodes in $E_c$.
            end for
        end if
        The consensus phase.
    end for
end for
return $\theta_c(t+1) = \theta(t+1)$.
```

Algorithm 1. : MEC-AI (HetFL) Algorithm based on Federated Learning

Optimizing edge clusters and devices by edge AI nodes

This section discusses optimizing edge clusters and devices using edge AI nodes, as shown in Fig. 2. It emphasizes the importance of careful placement, resource allocation, and lightweight models. The goal is to minimize latency, improve efficiency, and optimize computational resources, such as processing power, memory, and storage, according to workload requirements.

Optimization strategies involve collaboration between edge devices and the edge clusters, offloading computationally intensive tasks, dynamic workload management, and continuous performance monitoring to optimize resource utilization and address potential issues. These strategies ensure the cluster's adaptive allocation and distribution of edge AI workloads based on real-time demand and available resources.

Edge cluster and device optimization through edge AI nodes involve strategic placement, resource allocation, lightweight models, federated learning, data preprocessing, distributed computing, cloud collaboration, dynamic workload management, and continuous monitoring, enhancing performance, energy efficiency, and effectiveness.

The MEC-AI HetFL technique is proposed for selecting edge nodes for distributed machine learning in network clusters and IoT networks. Interconnected edge nodes are set up through these networks, serving as a centralized parameter controller. IoT devices collaborate to train the same training model, utilizing their computational power. This approach allows for training local data across dispersed nodes. However, current distributed learning initiatives require a homogeneous system with identical computing capacity, a reliable communication network, and synchronous learning. This technique is a more efficient and flexible approach to machine learning.

In contrast, Synchronous learning is ineffective in real-world situations due to IoT device heterogeneity. The MEC-AI HetFL model addresses this issue by obtaining computational resources opportunistically and dynamically selecting global integration nodes to improve learning outcomes, address the resource squandering problem, and improve the learning assignment outcome.

Edge node selection process algorithm

We developed the edge node selection algorithm in MEC-AI (HetFL), which is being developed to select the best edge nodes for offloading compute workloads based on network conditions, AI expertise, and computational requirements. The algorithm considers edge nodes' availability, resource capabilities, and proficiency in executing AI tasks. The description for this is in algorithm 2:

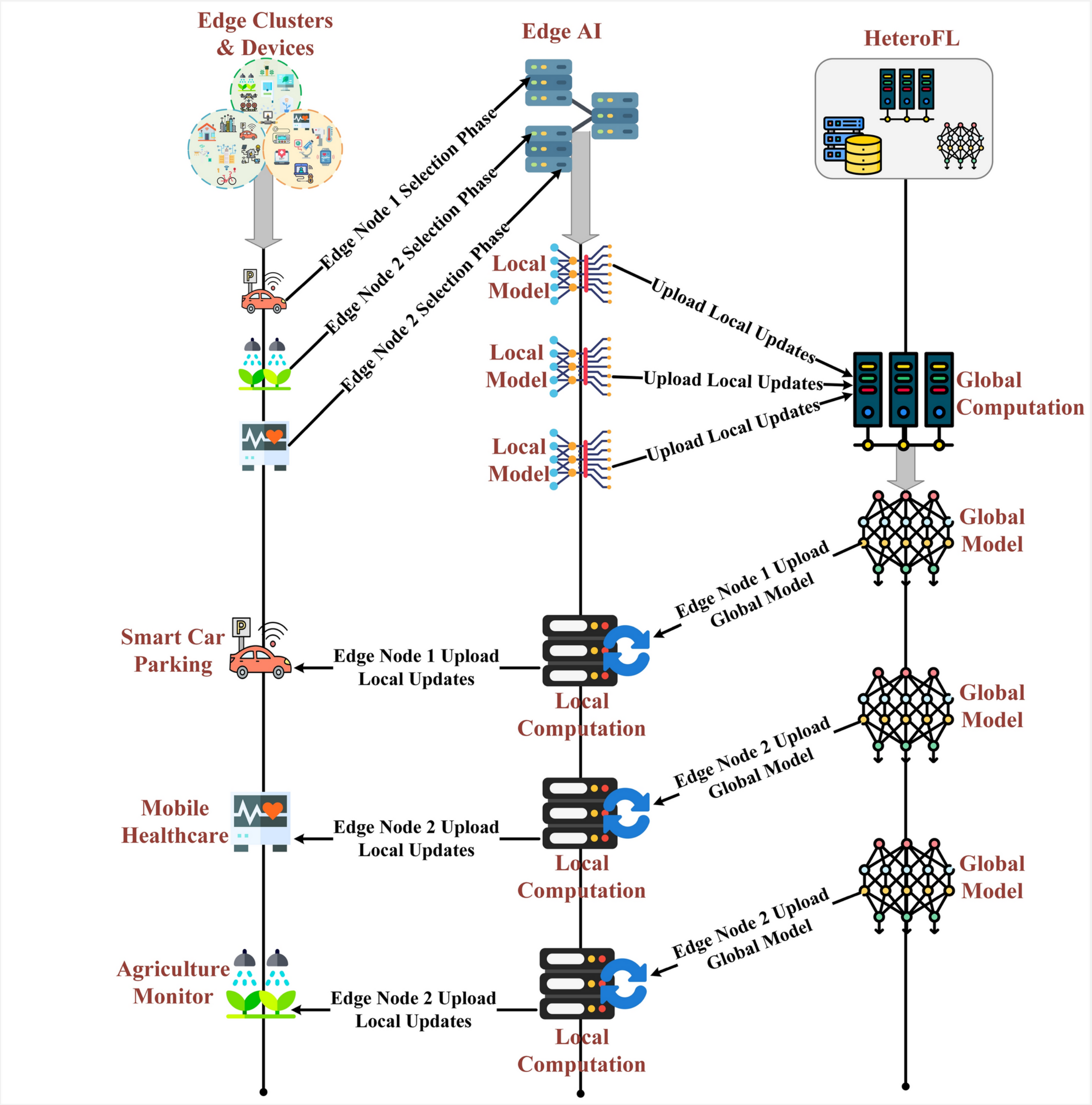

Figure 2. Optimizing edge clusters with edge devices and edge AI real-time communication process in MEC-AI (HetFL).

```
Input: Computation task $T$ with requirements, set of available edge AI nodes $E$, and network conditions.
Initialize an empty list of candidate edge nodes $C$.
Initialize the Edge AI nodes of all Edge nodes.
for each edge node $e$ in $E$ do
    Check if $e$ meets the resource requirements of task $T$
    Check if $e$ has the necessary AI expertise to execute the task.
    if $e$ meets requirements $T$ , AI expertise then
        add $e$ to $C$
        where $w_1$ and $w_2$ are weights assigned to each factor via (4).
    end if
    Based on network conditions and AI expertise, calculate a quality score $Q$ for each edge node in $C$.
    Sort the edge nodes in $C$ based on the quality score $Q$ in descending order.
    Select the top-ranked edge node from the sorted list as the optimal edge node for task offloading.
end for
Output: The selected optimal edge node for task execution.
```

Algorithm 2. : Edge Node Selection Algorithm in MEC-AI (HetFL)

The Edge Node Selection Algorithm 2 of MEC-AI (HetFL) calculates a quality score *Q* for each edge node in *C* based on network conditions and AI expertise scores.

Network Score *e* is the network condition score for edge node *e*. It represents the quality or performance of the network connection for that edge node. It also calculates the network contribution. Multiply the network condition score (Network Score *e*) by the weight assigned to the network condition factor *w*1. This step evaluates the contribution of the network condition to the overall quality score. The product $w1 * NetworkScore(e)$ represents the weighted network contribution. Expertise Score *e* the AI expertise score for edge node *e*. It represents the level of expertise or competence of that edge node in AI-related tasks. Calculate the expertise contribution. Multiply the AI expertise score (Expertise Score *e*) by the weight assigned to the expertise factor $w2$. This step evaluates the contribution of the AI expertise to the overall quality score. The product $w2 * ExpertiseScore(e)$ represents the weighted expertise contribution.

$w1$, $w2$, are the weights assigned to each factor. These weights determine the relative importance of the network condition score and the AI expertise score in the overall quality score calculation. It also combines the contribution scores. Add the weighted network contribution and the weighted expertise contribution together. Combine the two contributions to calculate the general quality score *Q* for edge node *e* can be seen in (4):

$$Q(e) = w_1, \text{NetworkScore}(e) + w_2, \text{ExpertiseScore}(e) \tag{4}$$

The quality score *Q* measures the suitability of an edge node *e* for a task based on its network condition and AI expertise. A higher score indicates better performance, making it more desirable. Weights *w*1 and *w*2 allow for each factor's importance adjustment, with increasing weight value amplifying its influence. This process calculates each edge node's quality score *Q*, assessing its suitability and performance for specific tasks and applications.

The paper presents an edge node selection algorithm that evaluates candidates based on their resource capabilities, network conditions, and expertise in AI-related tasks. Nodes with better connectivity and proficiency are prioritized through weighted scoring, ensuring efficient task offloading and heterogeneous capabilities utilization. Optimizing edge clusters involves considering proximity to data sources, workload-resource matching, and network topology design to improve performance metrics like latency and reliability. Edge computing is also utilized for IoT device training, leveraging edge nodes to minimize device processing loads while maximizing communication efficiency through techniques such as constant quality scoring and reduced complexity modeling. Strategic node placement near data sources further enhances response times.

Experimental results and discussions

Experimental environment

The MEC-AI HetFL framework illustrated in Fig. 1 provides a robust solution for privacy-preserving across heterogeneous edge environments based on federated learning through its integration of multi-edge nodes, multi-clusters and edge AI capabilities. This enables collaborative machine learning utilizing edge resources efficiently while maintaining user privacy. The proposed approach shows promise to significantly advance domains like IoT, smart cities and edge applications by endowing edge devices with localized intelligent functions. Example applications encompass healthcare through patient data analysis, smart transportation optimizing traffic, industrial anomaly detection for predictive maintenance aimed at reducing downtime and boosting operational efficiency.

Experimental setup

The model was designed using Python and Keras, with the TensorFlow backend utilized for its flexible distribution capabilities. Network weights were initialized using the random distribution functionality within NumPy. Experiments were run using a virtual machine hosted on Google Cloud, which provided varying compute capabilities up to 6vCPUs and 16GB of RAM. This hardware environment allowed for efficient distributed training of the model. The TensorFlow backend further enabled effective distribution of the model across the computational resources of the virtual machine. The random initialization of weights from NumPy helped ensure the network started from a random starting point for training.

We discovered that the MEC-AI (HetFL) proposed in this research reveals that the MEC-AI (HetFL) methodology outperforms other learning methods due to its superior performance in decentralized model updating and training data usage, despite node computational limitations that can impair model accuracy. The study examines evaluation metrics F1-Score, Precision, and Recall to evaluate classifier output per class Accuracy assessments are as follows (5,6,7,8) :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{7}$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{8}$$

Efficiency of edge clusters and in-edge AI for real-time communication system towards MEC-AI (HetFL)

Clusters and nodes allow a system to scale computing resources as needed, with each node contributing its processing power. Load balancing ensures even distribution of tasks across different nodes within a cluster, as shown in Fig. 1. It results in optimized resource utilization, efficient processing, and the ability to handle increasing workloads without compromising performance. This is particularly beneficial for computationally intensive tasks or large-scale data processing, leading to faster data processing and improved overall performance.

The system's performance and efficiency are enhanced by clusters and nodes, which ensure workloads are evenly distributed across nodes, avoiding hotspots and maximizing resource use. Parallel processing across multiple nodes expedites data analysis, resulting in lower latency and quicker response times. The system's flexibility to dynamically shift resources as demands change optimizes performance, with strategic workload scheduling, parallelization, and adaptive resource management enhancing efficiency. This versatile edge computing solution supports various applications and workloads.

Edge clusters data flow frequency, energy frequency

The data flow and energy frequency in each smart city cluster are influenced by its specific deployment, applications, devices, and operational characteristics. We can do so based on the specific requirements, data processing capabilities, and energy management strategies.

In cluster 1 and cluster 2 of the smart city deployment, The data flow frequency in clusters 1 and 2 varies based on the specific applications, services, and communication requirements within each cluster, representing Fig. 3 shows the results. This frequency is influenced by the types of sensors, devices, and urban infrastructure used in these clusters and their data processing needs. Additionally, these clusters' energy consumption and management are influenced by factors such as the power requirements of deployed sensors, IoT devices, infrastructure components, and the energy management strategies implemented within the clusters. The energy frequency in Cluster 2 differs from Cluster 1 based on the specific devices, applications, and energy optimization approaches employed in this cluster represents. Figure 3 shows the results.

The data flow and energy frequency in Mobile Healthcare Cluster 1 and Cluster 2 are designed to ensure efficient data flow, effective communication, and efficient energy management. This supports high-quality mobile healthcare services by considering factors like patient data volume, real-time monitoring, telemedicine requirements, and regional differences.

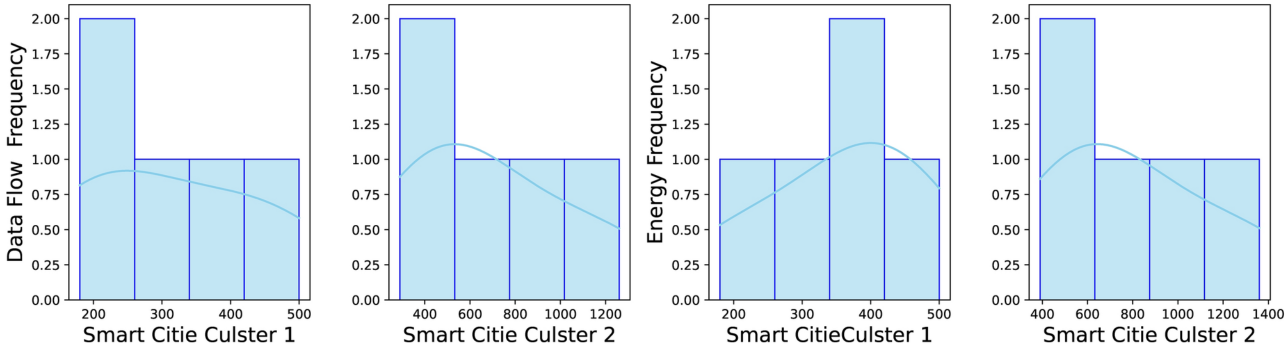

Figure 3. The data flow frequency and energy frequency for each cluster in smart cities.

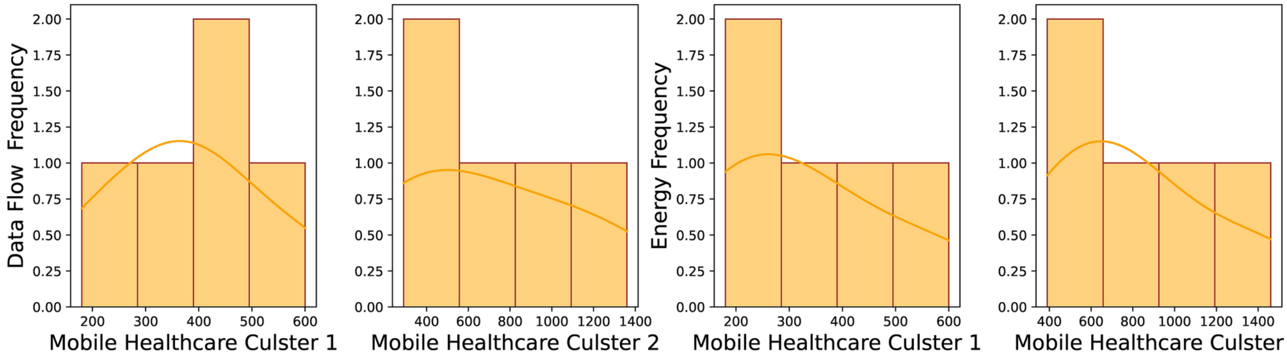

Figure 4. The data flow frequency and energy frequency for each cluster in mobile healthcare.

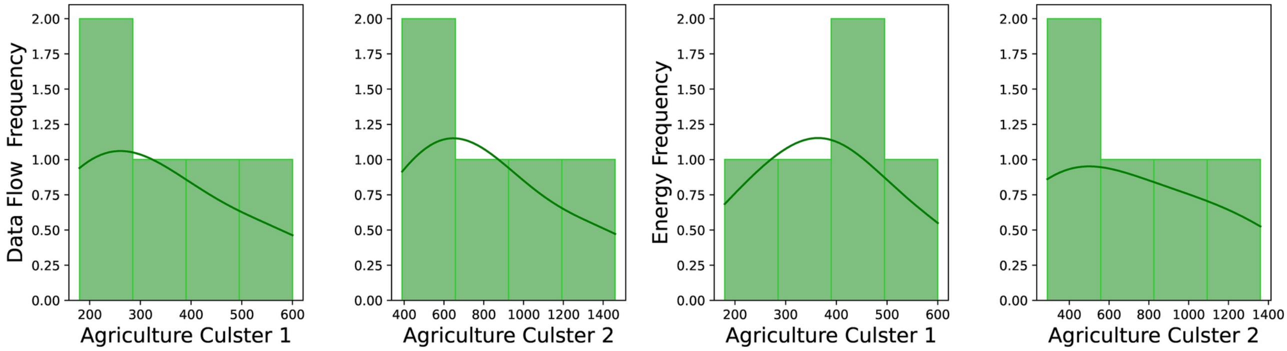

Figure 5. The data flow frequency and energy frequency for each cluster in agriculture.

Cluster 1 focuses on real-time monitoring and communication between healthcare providers and patients, while Cluster 2 may focus on specific services or patient groups. Data flow frequency is represented in Fig. 4, in these two clusters differs due to different applications, devices, or operational characteristics. Cluster 1 uses medical devices like monitors, diagnostic equipment, and wearable sensors, while Cluster 2 may implement energy efficiency measures like device optimization or renewable energy sources. The energy frequency rate is represented in Fig. 4, depending on factors like power requirements, mobile healthcare units, infrastructure, and energy management strategies. Cluster 2's energy frequency differs from Cluster 1 based on specific devices, applications, and energy optimization approaches.

The data flow and energy frequency in Agriculture clusters 1 and 2 are designed to ensure efficient data flow and effective management of energy resources, supporting sustainable and productive agricultural operations. These clusters consider sensor data collection, precision agriculture practices, crop health monitoring, power requirements, and scalability.

Cluster 1 and Cluster 2 are two clusters that collect data on environmental factors like soil moisture, temperature, humidity, and crop growth. Cluster 1 includes agricultural devices like irrigation systems and automated machinery, while Cluster 2 may implement energy-efficient practices like optimizing machine usage and smart irrigation systems. The data flow frequency is represented in Fig. 5 in each cluster and varies due to different applications, devices, and operational appearances. The energy frequency represented in Fig. 5 in Cluster 1 and Cluster 2 is determined by the power requirements of these devices. In contrast, Cluster 2 frequency is influenced by the specific devices, applications, and energy optimization approaches used in the clusters.

Edge AIs quality score, overall quality score, and accuracy, precision, recall, F1 score

Edge AI refers to utilizing artificial intelligence capabilities at the network edge, closer to data sources and end users. It plays a vital role in real-time systems using MEC-AI and HetFL frameworks. By enabling real-time localized processing and decision making in heterogeneous networks, Edge AI significantly reduces latency and network traffic through techniques such as local data filtering and preprocessing. Its distributed deployment across edge nodes addresses diverse workloads, dynamically scales resources with demand, and enhances privacy and security by limiting data transmission needs. Edge nodes can also perform ongoing intelligent decision making even during temporary network losses, improving communication systems, user experience, costs and overall performance in MEC-AI environments.

Edge AI nodes are being deployed in Smart Cities, Mobile Healthcare, and Agriculture Clusters for real-time communication, shown in Fig. 6. This reduces data transmission, ensuring sensitive information remains within the cluster and reducing potential attacks. Minimizing data transmission to centralized servers reduces

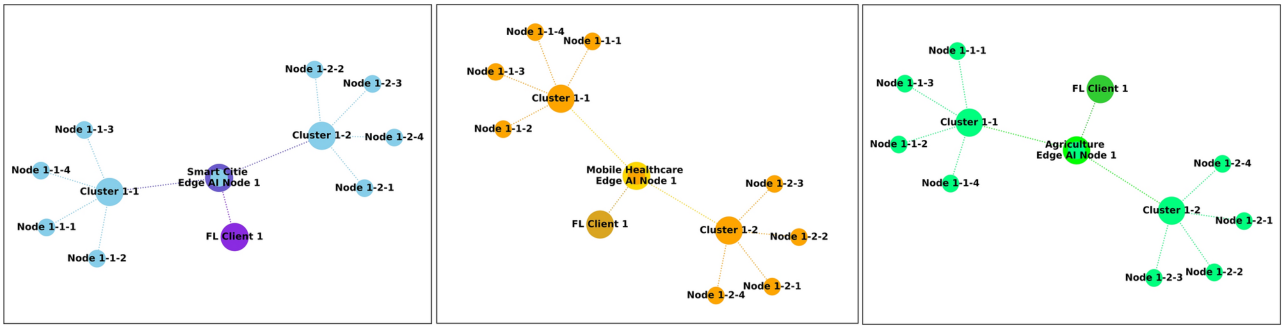

Figure 6. Edge AIs for real-time communication with edge clusters, nodes, and FL clients.

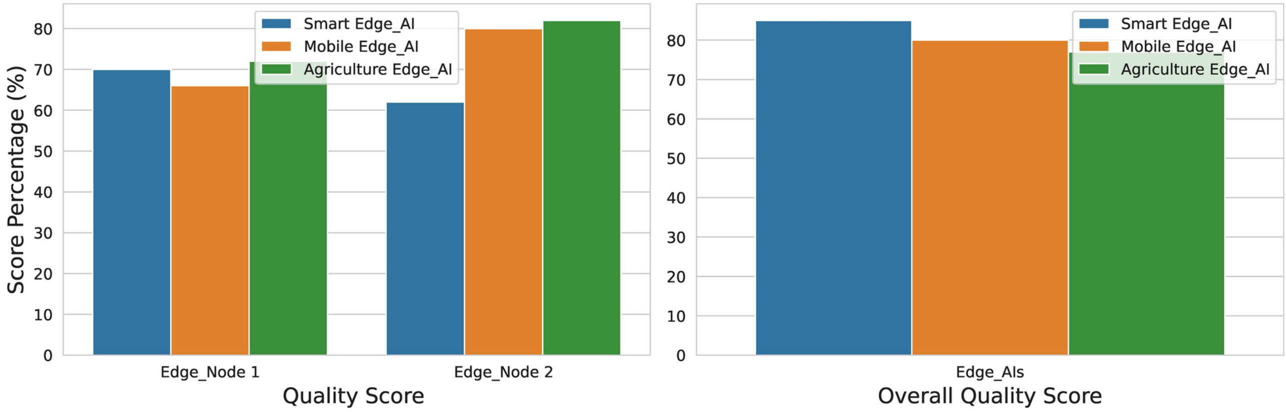

Figure 7. (**a**) Quality score of each edge AI (edge node 1, edge node 2), (**b**) overall quality score of edge AIs (smart, mobile, agriculture).

energy consumption, improving energy efficiency and sustainability efforts. Edge nodes can make intelligent decisions locally, reducing dependency on network availability and enabling faster response times for time-critical applications. This approach contributes to sustainability efforts in each cluster within Cluster 1 and 2.

Smart Cities Clusters, Mobile Healthcare Clusters, and Agriculture Clusters use Smart Edge AI infrastructure to distribute workload across edge nodes efficiently. These clusters optimize network traffic, balance resource utilization, and ensure load-balanced real-time communication shown in Fig. 6. As the cluster grows, the infrastructure can scale resources, add edge nodes, and adapt to changing workload requirements while maintaining high-performance real-time communication capabilities. This results in optimized communication, enhanced user experiences, reduced operational costs, and improved system performance in both Cluster 1 and Cluster 2.

Edge AI nodes enable real-time communication with clients shown in Fig. 6, minimizing latency and reducing data transmission to a central server. Federated Learning trains models on locally collected data, enabling real-time inference and decision-making without constant connectivity dependency. Local processing on clients reduces the energy required for communication, improving energy efficiency and battery life. Global models can be periodically updated using local data, allowing adaptive AI capabilities. Edge AI nodes enable clients to learn from real-time communication interactions, enhancing performance and adapting to changing user preferences. This results in optimized real-time communication, enhanced user experiences, improved privacy, reduced operational costs, and faster response times. Edge AI nodes also result in faster response times and enhanced real-time responsiveness for applications.

The Edge AI node quality scores for Edge Node 1 and 2 provide valuable insights into their performance and contributions to the system. The overall quality score evaluates the entire system holistically, considering accuracy, reliability, timeliness, resource utilization, and system-wide effectiveness. This evaluation aids decision-making, optimization, and improvement for high-quality AI capabilities at the edge.

The Edge AI node quality score for Edge Node 1 and Edge Node 2 represents a comprehensive assessment of the AI capabilities deployed on this specific node in Fig. 1. It considers factors such as accuracy, reliability, data flow, quality score, and resource utilization of the AI algorithms and models executed on Edge Node 1 and Edge Node 2. It captures the node's ability to fulfill its designated responsibilities, handle specific tasks, and contribute to the overall success of the Edge AI system. The score evaluates the performance and effectiveness of Edge Node 1 and Edge Node 2. As shown in Fig. 7a, the results in delivering high-quality AI capabilities at the edges quality score may consider additional factors like the edge node's computational power, data flow, and real-time connectivity for evaluating its performance.

The overall quality score of Edge AI nodes represents a holistic evaluation of the entire Edge AI system, considering the contributions of both Edge Node 1 and Edge Node 2. The overall quality score reflects. As shown in Fig. 7b, the results of the system-wide assessment of the Edge AI system, considering the accuracy, reliability, data flow, quality score, and resource utilization between Edge Node 1 and Edge Node 2. For a complete evaluation, the quality score may also consider the overall system architecture, data flow, real-time communication efficiency, and coordination between the edge nodes.

The Edge AI nodes (Smart, Mobile, and Agriculture) generate real-time communication predictions that improve system performance, as shown in Fig. 8 results within user experiences and resource utilization in real-time communication environments. These predictions enable proactive decision-making, efficient resource allocation, adaptive strategies, and personalized services. The AI models and algorithms are continuously monitored and refined to ensure accuracy, reliability, and relevance to real-time communication requirements. Edge AI analyzes incoming data, identifies patterns, and generates predictions about network performance, user behavior, service response, latency, data traffic, and service quality. These predictions aid in proactive decision-making, network optimization, dynamic resource allocation, load balancing, and predictive maintenance in real-time communication systems.

The Edge AI nodes' accuracy, precision, recall, and F1 score measure their effectiveness in making accurate predictions. These metrics are influenced by continuous monitoring, evaluation, and improvement of each node's AI models and algorithms, ensuring the reliability and accuracy of predictions in real-time communication scenarios.

Edge AI (Smart, Mobile, and Agriculture) represents Fig. 8, evaluated for accuracy, precision, and recall. The accuracy is measured by the proportion of correctly predicted outcomes compared to the ground truth or reference data. Precision is the ratio of correctly predicted positive instances to all instances predicted as positive, indicating the precision and accuracy of the positive predictions. Recall measures the number of correctly predicted positive instances out of all actual positive instances in real-time communication, indicating Edge AI's ability to identify positive instances accurately. The F1 score, which combines precision and recall, provides a balanced performance measure, providing a comprehensive evaluation of Edge AI's predictive accuracy in Fig. 9 to show the results.

Overall deployment of MEC-AI(HetFL) and accuracy, precision, recall, F1 score

The study deployed three Edge AI nodes along with federated learning involving three client devices as illustrated in Fig. 10. This configuration demonstrated enhancements to performance, privacy, security, decentralized processing, personalized AI models, efficient resource use and continuous learning. As a result, the system realized optimized outcome metrics including system efficiency, data protection, workload management and tailored intelligence capabilities, while preserving robustness and fault tolerance under real-world communication scenarios. Federated learning in particular allowed clients to improve models without exposing raw data, ensuring privacy. Edge AI nodes featured distributed, specialized computational resources and

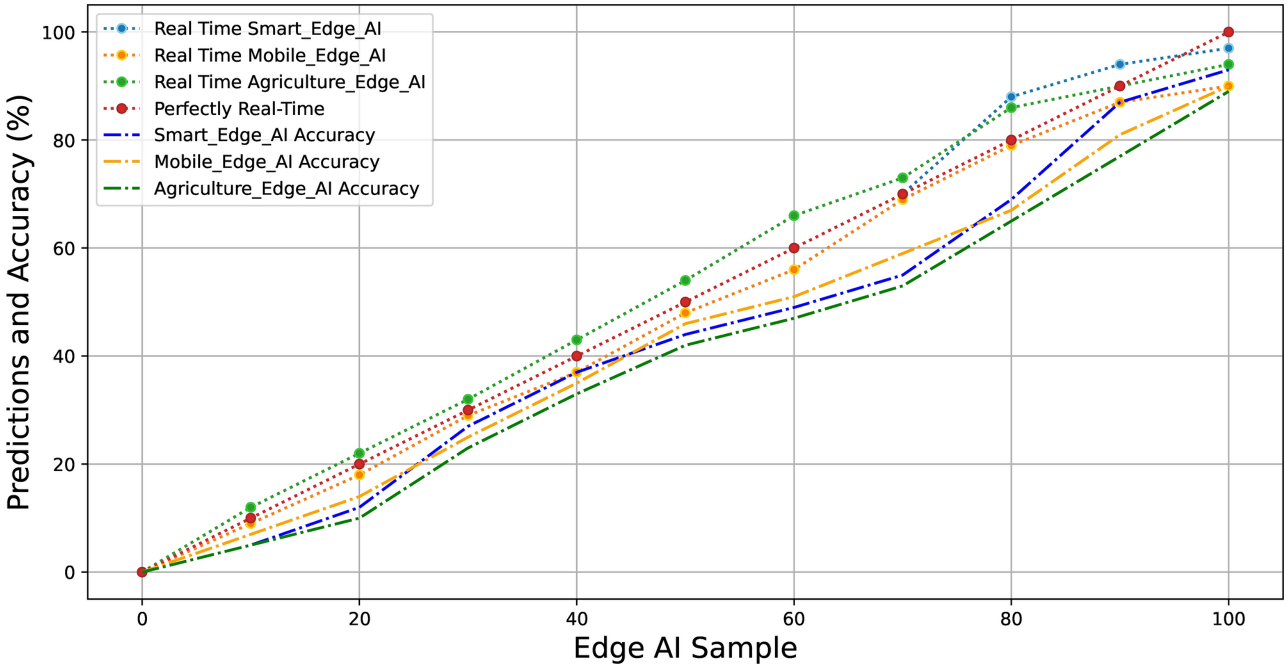

Figure 8. Edge AIs (smart, mobile, agriculture) real-time communication predictions and accuracy.

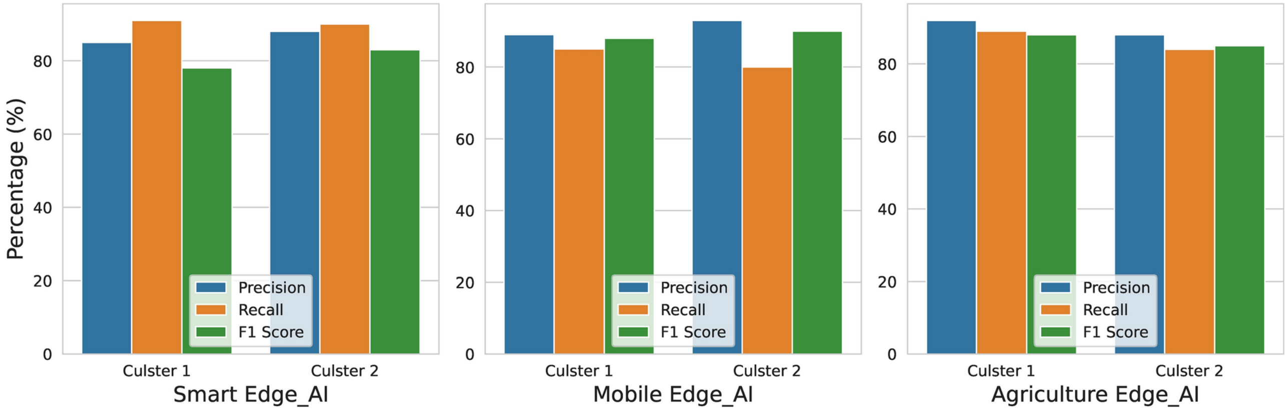

Figure 9. Edge AIs (Smart, mobile, agriculture) each cluster precision, recall, F1 score.

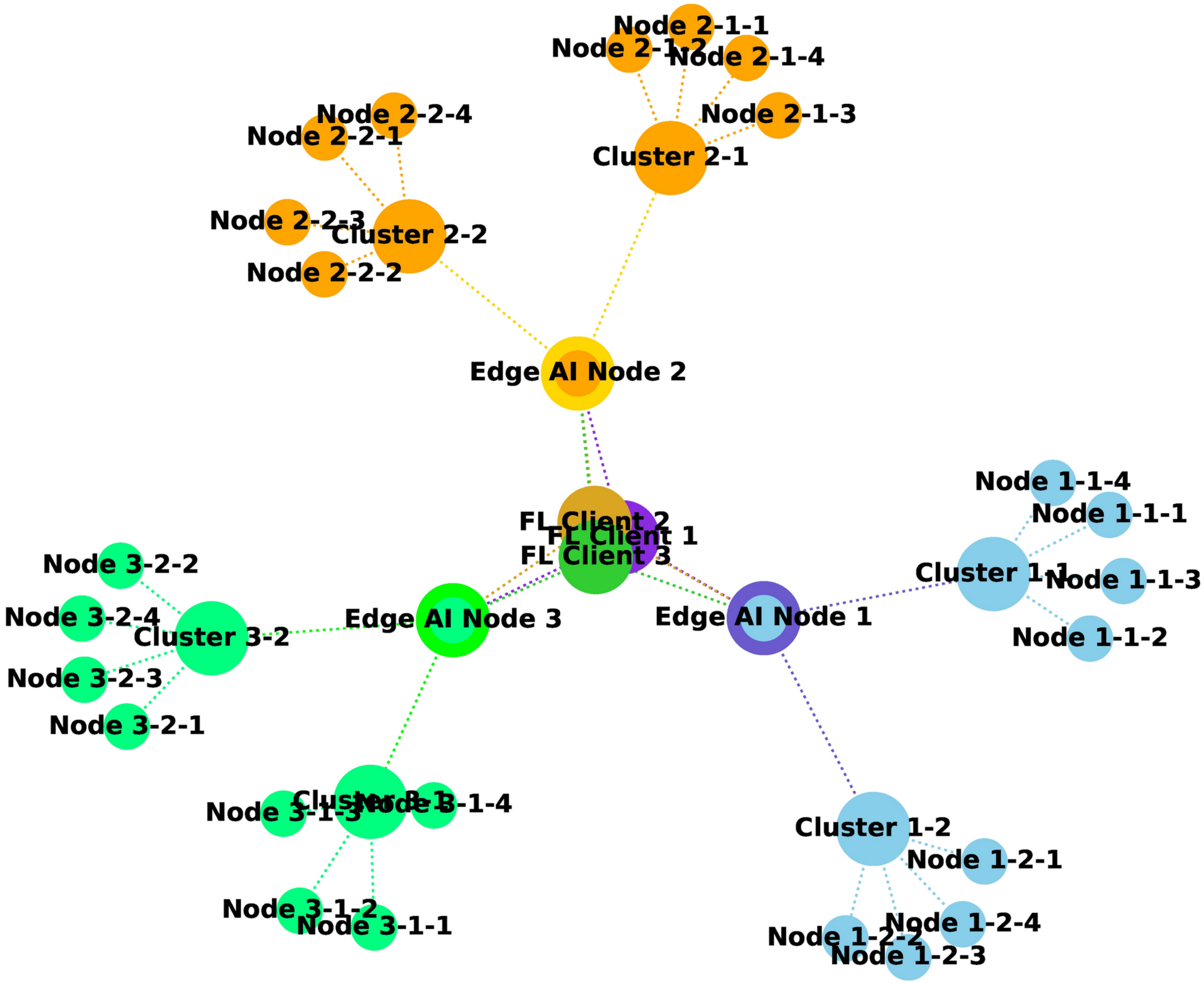

Figure 10. Overall real-time communication edge AIs for edge clusters with nodes and FL clients.

algorithms, supporting faster responses, lower latency and enhanced privacy through localized data processing at scale with load balancing.

FL system that allows clients to train personalized AI models without compromising data privacy. These models can adapt to individual preferences, providing customized recommendations or predictions. FL also

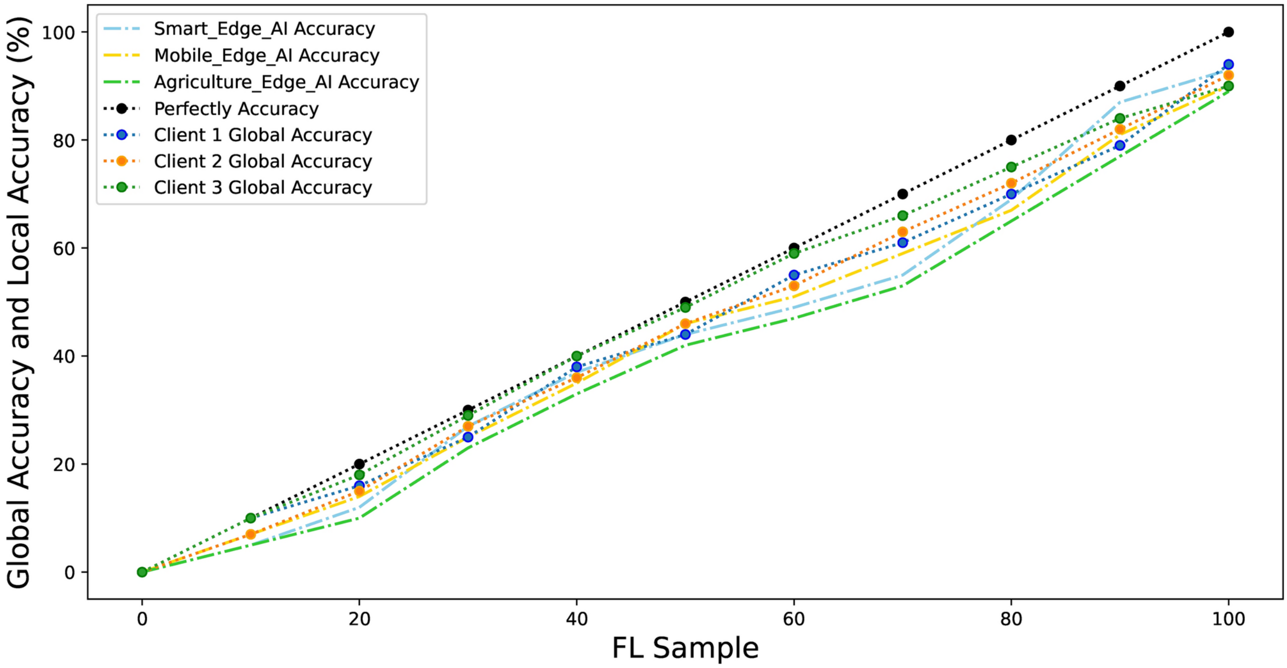

Figure 11. Local accuracy edge AIs (smart, mobile, agriculture) and global accuracy FL clients (client 1, client 2, client 3).

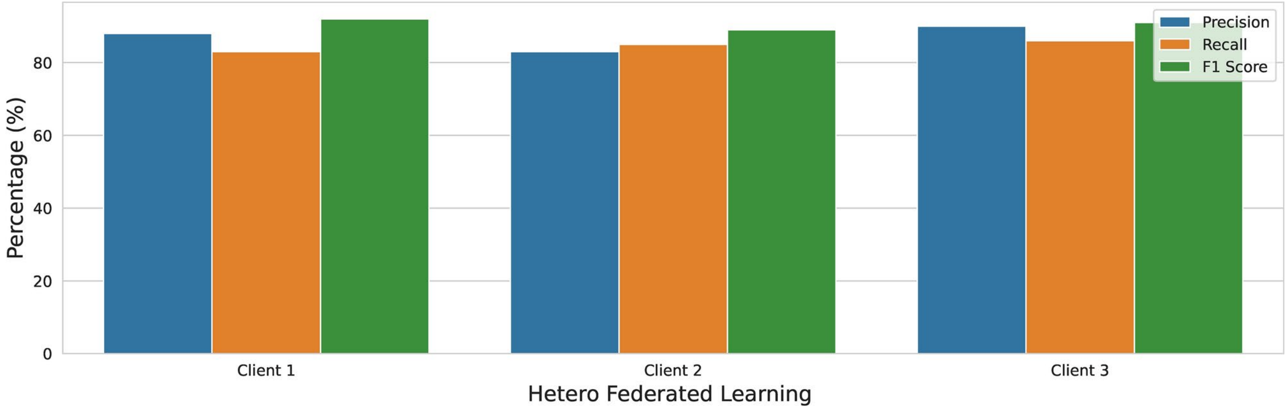

Figure 12. Edge AIs (Smart, mobile, agriculture) each cluster precision, recall, F1 score.

allows continuous learning and model improvement, with Edge AI nodes and clients updating and refining their models using local data. This adaptive approach reduces reliance on a central server, improves response times, and enhances scalability and fault tolerance. Computational tasks are distributed among Edge AI nodes, and clients contribute their global computing resources for collaborative model training. FL leads to more accurate and robust models.

The study evaluates the accuracy of three Edge AI nodes in predicting outcomes for three clients in shown Fig. 11. The results provide insights into the system's performance and effectiveness, evaluating the accuracy and reliability of the predictions. Continuous monitoring, validation, and refinement of the AI models and algorithms are used to optimize these accuracy metrics, ensuring reliable predictions in real-time communication scenarios. Global accuracy is computed by aggregating the correctly predicted outcomes across all predictions made by the three Edge AI nodes and clients. A higher global accuracy in Fig. 11 indicates that Edge AI nodes and clients are making accurate predictions and achieving reliable results. Continuous monitoring and validation contribute to optimizing these accuracy metrics and ensuring reliable predictions in real-time communication scenarios.

Continuous monitoring and refinement of client's FL models and algorithms optimize metrics, ensuring reliability and accuracy of predictions. This leads to higher precision, recall, and F1 scores, positively impacting decision-making and system performance. Precision measures the proportion of correctly predicted positive instances, reflecting the accuracy and precision of the positive predictions made by Clients 1, 2, and 3 in shown Fig. 12 results. Recall, or sensitivity, measures the proportion of correctly predicted positive instances out of

Method	Accuracy (100-th round)	Loss (100-th round)	Selected devices (per round)	Training time (per round/s)	Computation and communication time (per round/ms)	Data amount (per round)	Training epochs (per round)
EdgeFed[24]	**97.8%**	0.09	10(per edge server)	12s	8ms	50	5
FedSA[25]	**96.2%**	0.13	20% of total devices	15 s	10 ms	10	3
FedMP[26]	**95.5%**	0.15	30 devices	18 s	12 ms	30	4
H-DDPG[27]	**96.8%**	0.11	15 devices per cluster	14 s	9 ms	20	4
Our proposed MEC-AI HetFL	**98.6%**	0.07	Total devices quality scores per cluster	10 s	6 ms	65	5

Table 1. Comparison of different methods for efficient FL. Significant values are given in bold.

all actual positive instances in real-time communication, indicating the ability of Clients 1, 2, and 3 in the shown Fig. 12 results. To identify positive instances accurately. The F1 score, the harmonic mean of precision and recall, provides a balanced performance measure, combining the predictions' accuracy and completeness. Higher precision indicates more accurate positive predictions, while higher recall suggests we can correctly identify a greater proportion of positive instances. The F1 score provides a comprehensive evaluation of overall performance, balancing both precision and recall.

Table 1 provides a comparative analysis of 5 federated learning methods on performance metrics, including accuracy, loss, device selection, training time, communication overhead, data usage, and number of epochs. MEC-AI HetFL and EdgeFed represent two high-performing federated learning approaches from the table. MEC-AI HetFL is a federated learning method that achieves the highest accuracy of 98.6% among the approaches compared in the table. It reaches this through training on all available devices, filtered by quality scores assigned per cluster. MEC-AI HetFL has a longer training time of 10 seconds and a computation and communication overhead of 6ms due to its broad device participation. It uses a larger data amount of 65 per device and runs for five epochs. In contrast, EdgeFed reaches a competitive 97.8% accuracy but with a lower training time of 12s and communication of 8ms by selective training on ten devices per edge server. FedSA and H-DDPG represent federated learning approaches that balance accuracy and efficiency tradeoffs. FedSA reaches 96.2% accuracy by training on 20% of devices for 15s and three epochs. H-DDPG achieves slightly higher 96.8% accuracy with 15 devices per cluster, 14s training, and four epochs. Both have lower overheads of 9-10ms versus 12-18ms for other methods. FedMP reaches an accuracy of 95.5%, which is lower than other methods such as MEC-AI HetFL, EdgeFed, FedSA, and H-DDPG. However, FedMP trains on 30 selected devices in 18 s and uses model compression to reduce communication to 12 ms per round.

Table 1 outlines the tradeoffs between accuracy, speed, scalability, and other factors for the MEC-AI HetFL federated learning technique designed for efficient collaborative learning on the device in many heterogeneous edge devices in edge environments. While it doesn't match the 98.6% accuracy of MEC-AI HetFL, it is more selective in device participation and training time. Efficiency optimizations like selective device participation, hierarchical coordination, and bandwidth adaptation enable collaborative learning across edge nodes with reasonable accuracy and resource tradeoffs.

Specifically, we have now included detailed comparisons with EdgeFed, FedSA, FedMP, and H-DDPG frameworks, which are widely recognized in federated learning for edge computing environments. These methods were selected based on their performance in handling heterogeneous data and resource-constrained IoT networks, aligning closely with the problem space of our work. We compared the proposed MEC-AI HetFL architecture with these methods in terms of accuracy, communication costs, training time, and resource allocation efficiency. Our results show that MEC-AI HetFL achieves up to 5 times better performance in training speed and accuracy in non-IID data scenarios. This is validated through extensive simulations and network traffic tests, showing improved resource allocation efficiency and quality scores across heterogeneous networks. In addition to performance metrics, we have also discussed how MEC-AI HetFL addresses key limitations in existing frameworks such as inefficiencies in node selection, long training latency, and higher communication costs observed in synchronous FL methods like EdgeFed and asynchronous models like FedSA. Our method overcomes these challenges through dynamic multi-edge clustering and a low-complexity node selection strategy, ensuring more efficient model training without waiting for slower nodes to complete their iterations. The proposed MEC-AI HetFL framework introduces a novel solution through the implementation of dynamic multi-edge clustering coupled with a computationally efficient node selection algorithm. This approach effectively mitigates the straggler effect prevalent in traditional frameworks by eliminating the necessity to await completion from slower nodes during training iterations, thereby optimizing the overall model training process.

Conclusion

This paper proposed MEC-AI HetFL, a novel federated learning framework that utilizes multiple collaborating edge AI nodes organized in a multi-edge clustered topology. This architecture dynamically acquires compute resources and selects nodes for model aggregation. Experiments evaluated MEC-AI HetFL against methods like EdgeFed, FedSA, FedMP and H-DDPG. Results demonstrated MEC-AI HetFL achieved the highest test accuracy of 98.6%, outperforming competitors. However, its overhead was also higher than techniques aimed at local efficiency optimization like EdgeFed. Approaches such as FedSA and H-DDPG provided a balance between accuracy and performance. While MEC-AI HetFL prioritized accuracy maximization, EdgeFed and FedMP focused on speed and efficiency gains through model compression and selective participation. Tradeoffs between accuracy, speed and scalability underscore the design challenges in federated learning. Future work includes enhancing the multi-edge clustered architecture and resource allocation strategies to further improve computation and communication efficiency under diverse networking conditions. Integrating privacy-preserving methods compatible with MEC-AI HetFL could expand its capabilities to support more heterogeneous edge devices and datasets. Additionally, evaluating the approach across different application domains and data distributions would test its effectiveness in heterogeneous environments. Overall, MEC-AI HetFL presents an effective federated learning solution for edge intelligence but leaves opportunities for optimization.

Data availability

The datasets used and analyzed during the current study available from the corresponding author on reasonable request and with permission.

Received: 20 August 2024; Accepted: 29 October 2024
Published online: 20 November 2024

References

1. Hussain, F., Hassan, S. A., Hussain, R. & Hossain, E. Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. *IEEE Commun. Surveys Tutorials* **22**(2), 1251–1275 (2020).
2. Cisco global cloud index. *Forecast Methodol.* (Cisco Global Cloud, Cisco, California, CA, USA, 2018).
3. Sun, G., Wang, Z., Su, H., Yu, H., Lei, B., & Guizani, M. Profit maximization of independent task offloading in MEC-enabled 5G internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* (2024).
4. Trindade, S., Bittencourt, L. F., & da Fonseca, N. L. Resource management at the network edge for federated learning. *Digit. Commun. Netw.* (2022).
5. Liu, Y. et al. BFL-SA: Blockchain-based federated learning via enhanced secure aggregation. *J. Syst. Architect.* **152**, 103163 (2024).
6. Wang, P., Song, W., Qi, H., Zhou, C., Li, F., Wang, Y., & Zhang, Q. Server-initiated federated unlearning to eliminate impacts of low-quality data. *IEEE Trans. Serv. Comput.* (2024).
7. Zhang, F., Wang, M. M., Deng, R. & You, X. QoS optimization for Mobile ad hoc cloud: A multi-agent independent learning approach. *IEEE Trans. Veh. Technol.* **71**(1), 1077–1082 (2021).
8. Sun, G., Zhang, Y., Yu, H., Du, X. & Guizani, M. Intersection fog-based distributed routing for V2V communication in urban vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **21**(6), 2409–2426 (2019).
9. Zhao, Z. et al. Federated learning with non-IID data in wireless networks. *IEEE Trans. Wireless Commun.* **21**(3), 1927–1942 (2021).
10. Huang, W., Li, T., Cao, Y., Lyu, Z., Liang, Y., Yu, L., & Li, Y. Safe-NORA: Safe reinforcement learning-based mobile network resource allocation for diverse user demands. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 885–894 (2023).
11. Sellami, B., Hakiri, A., Yahia, S. B. & Berthou, P. Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network. *Comput. Netw.* **210**, 108957 (2022).
12. Huang, Y., Zhang, H., Shao, X., Li, X. & Ji, H. RoofSplit: an edge computing framework with heterogeneous nodes collaboration considering optimal CNN model splitting. *Futur. Gener. Comput. Syst.* **140**, 79–90 (2023).
13. Liang, T. et al. Mobile app recommendation via heterogeneous graph neural network in edge computing. *Appl. Soft Comput.* **103**, 107162 (2021).
14. Li, T., Hui, S., Zhang, S., Wang, H., Zhang, Y., Hui, P., & Li, Y. Mobile user traffic generation via multi-scale hierarchical GAN. *ACM Trans. Knowl. Discov. Data* (2024).
15. Zhou, H., Jiang, K., Liu, X., Li, X. & Leung, V. C. Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. *IEEE Internet Things J.* **9**(2), 1517–1530 (2021).
16. Aminizadeh, S., Heidari, A., Toumaj, S., Darbandi, M., Navimipour, N. J., Rezaei, M., & Unal, M. The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things. Computer methods and programs in biomedicine, 107745 (2023).
17. Dai, M., Luo, L., Ren, J., Yu, H. & Sun, G. PSACCF: Prioritized online slice admission control considering fairness in 5G/B5G networks. *IEEE Trans. Netw. Sci. Eng.* **9**(6), 4101–4114 (2022).
18. Ahmed, K. M., Imteaj, A., & Amini, M. H. Federated deep learning for heterogeneous edge computing. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1146–1152. IEEE (2021).
19. Heidari, A., Jafari Navimipour, N., Unal, M. & Zhang, G. Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues. *ACM Comput. Surv.* **55**(12), 1–45 (2023).
20. Mughal, F. R., He, J., Zhu, N., Almutiq, M., Dharejo, F. A., Jain, D. K., & Zardari, Z. A. An intelligent Hybrid-Q Learning clustering approach and resource management within heterogeneous cluster networks based on reinforcement learning. *Trans. Emerg. Telecommun. Technol.* **35**(4), e4852 (2024).
21. Li, C., Zhang, Y. & Luo, Y. Flexible heterogeneous data fusion strategy for object positioning applications in edge computing environment. *Comput. Netw.* **212**, 109083 (2022).
22. Zhang, Z. et al. Planet craters detection based on unsupervised domain adaptation. *IEEE Trans. Aerosp. Electron. Syst.* **59**(5), 7140–7152 (2023).
23. Wang, L., Xu, Y., Xu, H., Chen, M. & Huang, L. Accelerating decentralized federated learning in heterogeneous edge computing. *IEEE Trans. Mob. Comput.* **22**(9), 5001–5016 (2022).
24. Ye, Y., Li, S., Liu, F., Tang, Y. & Hu, W. EdgeFed: Optimized federated learning based on edge computing. *IEEE Access* **8**, 209191–209198 (2020).
25. Ma, Q. et al. FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE J. Sel. Areas Commun.* **39**(12), 3654–3672 (2021).
26. Jiang, Z., Xu, Y., Xu, H., Wang, Z., Qiao, C., & Zhao, Y. Fedmp: Federated learning through adaptive model pruning in heterogeneous edge computing. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (pp. 767-779). IEEE (2022).
27. Chen, Q., You, Z., Wen, D. & Zhang, Z. Enhanced hybrid hierarchical federated edge learning over heterogeneous networks. *IEEE Trans. Veh. Technol.* **72**(11), 14601–14614 (2023).
28. Luo, J., Zhao, C., Chen, Q. & Li, G. Using deep belief network to construct the agricultural information system based on Internet of Things. *J. Supercomput.* **78**(1), 379–405 (2022).
29. Liu, S., Yu, G., Chen, X. & Bennis, M. Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data. *IEEE Trans. Wireless Commun.* **21**(10), 7852–7866 (2022).
30. Kang, J., Li, X., Nie, J., Liu, Y., Xu, M., Xiong, Z., & Yan, Q. Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things. *IEEE Trans. Netw. Sci. Eng.* **9**(5), 2966–2977 (2022).
31. Xu, J. & Wang, H. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Trans. Wireless Commun.* **20**(2), 1188–1200 (2020).

32. Gong, Y., Yu, D., Cheng, X., Yuen, C., Bennis, M., & Debbah, M. Computation offloading and quantization schemes for federated satellite-ground graph networks. *IEEE Trans. Wirel. Commun.* (2024).
33. Shen, X., Jiang, H., Liu, D., Yang, K., Deng, F., Lui, J. C., & Luo, J. PupilRec: leveraging pupil morphology for recommending on smartphones. *IEEE Internet Things J.* **9**(17), 15538–15553 (2022).
34. Chen, Y., Sun, X. & Jin, Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(10), 4229–4238 (2019).
35. Wang, Z. et al. Asynchronous federated learning over wireless communication networks. *IEEE Trans. Wireless Commun.* **21**(9), 6961–6978 (2022).
36. Liu, J. et al. Adaptive asynchronous federated learning in resource-constrained edge computing. *IEEE Trans. Mob. Comput.* **22**(2), 674–690 (2021).
37. Wu, W. et al. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans. Comput.* **70**(5), 655–668 (2020).
38. Mahesar, A. R., Li, X. & Sajnani, D. K. Efficient microservices offloading for cost optimization in diverse MEC cloud networks. *J. Big Data* **11**(1), 123 (2024).
39. Sajnani, D. K., Li, X. & Mahesar, A. R. Secure workflow scheduling algorithm utilizing hybrid optimization in mobile edge computing environments. *Comput. Commun.* **226**, 107929 (2024).
40. Wang, C., Yu, X., Xu, L. & Wang, W. Energy-efficient task scheduling based on traffic mapping in heterogeneous mobile-edge computing: A green IoT perspective. *IEEE Trans. Green Commun. Netw.* **7**(2), 972–982 (2022).
41. Chen, P., Luo, L., Guo, D., Tang, G., Zhao, B., Li, Y., & Luo, X. Why and how lasagna works: a new design of air-ground integrated infrastructure. *IEEE Netw.* (2024).
42. Chan, Y. W., Fathoni, H., Yen, H. Y. & Yang, C. T. Implementation of a cluster-based heterogeneous edge computing system for resource monitoring and performance evaluation. *Ieee Access* **10**, 38458–38471 (2022).
43. Liu, J., Yang, P. & Chen, C. Intelligent energy-efficient scheduling with ant colony techniques for heterogeneous edge computing. *J. Parallel Distrib. Comput.* **172**, 84–96 (2023).
44. Wang, D. & Yang, S. X. Broad learning system with Takagi-Sugeno fuzzy subsystem for tobacco origin identification based on near infrared spectroscopy. *Appl. Soft Comput.* **134**, 109970 (2023).
45. Joglekar, A. et al. Open-source heterogeneous constrained edge-computing platform for smart grid measurements. *IEEE Trans. Instrum. Meas.* **70**, 1–12 (2021).
46. Wang, J. et al. Optimal task allocation and coding design for secure edge computing with heterogeneous edge devices. *IEEE Trans. Cloud Comput.* **10**(4), 2817–2833 (2021).
47. Sun, G., Sheng, L., Luo, L. & Yu, H. Game theoretic approach for multipriority data transmission in 5G vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **23**(12), 24672–24685 (2022).
48. Mughal, F. R., He, J., Zhu, N., Hussain, S., Zardari, Z. A., Mallah, G. A., & Dharejo, F. A. Resource management in multi-heterogeneous cluster networks using intelligent intra-clustered federated learning. *Comput. Commun.* **213**, 236–245 (2024).
49. Sun, G. et al. V2V routing in a VANET based on the autoregressive integrated moving average model. *IEEE Trans. Veh. Technol.* **68**(1), 908–922 (2018).
50. Wang, J., Bai, L., Fang, Z., Han, R., Wang, J., & Choi, J. Age of Information Based URLLC Transmission for UAVs on Pylon Turn. *IEEE Trans. Vehic. Technol.* (2024).
51. Xu, C., Xu, C., Li, B., Li, S. & Li, T. Joint social-aware and mobility-aware computation offloading in heterogeneous mobile edge computing. *IEEE Access* **10**, 28600–28613 (2022).
52. Sachan, A. & Kumar, N. S-Edge: heterogeneity-aware, light-weighted, and edge computing integrated adaptive traffic light control framework. *J. Supercomput.* **79**(13), 14923–14953 (2023).
53. Wang, P., Di, B., Song, L. & Jennings, N. R. Multi-layer computation offloading in distributed heterogeneous mobile edge computing networks. *IEEE Trans. Cogn. Commun. Netw.* **8**(2), 1301–1315 (2022).

Acknowledgements

The work supported by the Centre for Artificial Intelligence Research and Optimization (AIRO), Design and Creative Technology, Torrens University, New South Wales, Australia, and The authors extend their appreciation to the Researchers Supporting Program at King Saud University. Researchers Supporting Project number (RSP-D2024R867), King Saud University, Riyadh, Saudi Arabi.

Author contributions

Conceptualization, Fahad Razaque Mughal; Data curation, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu, Bhagwan Das; Formal analysis, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu, Bhagwan Das; Funding acquisition, Bhagwan Das, Saeed Alzahrani; Investigation, Fahad Razaque Mughal, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu; Methodology, Fahad Razaque Mughal, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu; Resources, Saeed Alzahrani, Surbhi Bhatia Khan; Software, Fahad Razaque Mughal, Bhagwan Das; Validation, Fahad Razaque Mughal, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu; Visualization, Fahad Razaque Mughal; Writing-original draft, Fahad Razaque Mughal; Writing-review and editing, Jingsha He, Fayaz Ali Dharejo, Nafei Zhu, Bhagwan Das, Saeed Alzahrani, and Surbhi Bhatia Khan.

Declarations

Competing Interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to B.D. or N.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.