Journal Basic Science and Technology 14 (3) (2025) pp. 146-161

Published by: IOCSCIENCE

Journal Basic Science and Technology

Journal homepage: www.iocscience.org/ejournal/index.php/JBST

Optimization of Federated Learning in Edge Computing Using Adaptive Client Selection

Arkan Zayd[1], Farzana[2], Thariq[3]

[1] Information & Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

[2,3] Department of Electrical Engineering, Information Technology University, Lahore, Pakistan

ARTICLEINFO

Article history:

Received Aug 27, 2025
Revised Sep 29, 2025
Accepted Oct 30, 2025

Keywords:

Adaptive Client Selection;
Communication Efficiency;
Edge Computing;
Federated Learning;
Resource Heterogeneity.

ABSTRACT

The rapid growth of edge computing has created new opportunities for processing data closer to end users and connected devices. Federated Learning (FL) has emerged as an important distributed learning paradigm because it enables collaborative model training without requiring raw data to be transferred to a central server. However, heterogeneous edge environments involve substantial differences in computational capacity, network bandwidth, latency, energy availability, data contribution, and client reliability. Conventional client-selection strategies, such as random or uniform selection, may therefore select inefficient clients, increase communication costs, prolong training latency, and reduce convergence efficiency. This study aims to develop an adaptive client-selection mechanism to optimize federated learning performance in edge computing environments. Selected client models were aggregated using federated averaging and compared with random, uniform, and resource-based selection methods. Performance was evaluated using model accuracy, convergence rounds, communication overhead, training latency, and energy consumption. The experimental results showed an accuracy improvement of 1.2 percentage points, communication reduction of 28.0%, latency reduction of 31.5%, energy reduction of 34.5%, and convergence improvement of 34.1% compared with random client selection. These findings indicate that adaptive client selection can improve the efficiency, resource utilization, and scalability of federated learning under heterogeneous edge-computing conditions.

This is an open access article under the CC BY-NC license.

Corresponding Author:

Arkan Zayd
Information & Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar,
Email: arkanzayd@hbku.edu.qa

1. INTRODUCTION

The rapid growth of Internet of Things (IoT) devices, mobile applications, and intelligent digital services has generated increasingly large volumes of distributed data that require efficient processing and analysis(Shafique et al., 2020). Traditional cloud computing architectures often transmit data from end devices to centralized data centers, which can create significant communication latency, bandwidth consumption, and privacy concerns. Edge computing has emerged as an alternative architecture by moving computational and data-processing capabilities closer to end users and data sources. This approach can reduce the distance between data generation and processing, thereby improving response time and supporting latency-sensitive applications. However, edge computing environments are characterized by substantial resource

heterogeneity. Edge devices may differ considerably in computational capacity, memory, battery availability, network bandwidth, storage capacity, and network stability. These limitations create significant challenges when distributed devices are required to participate in computationally intensive machine-learning processes.

Federated Learning (FL) provides a promising approach for integrating machine learning with edge computing while reducing the need to transfer raw data to a centralized server(Nguyen et al., 2021). In a typical FL framework, a central server distributes a global machine-learning model to selected edge clients. Each participating client trains the model locally using its own data and subsequently sends model parameters or updates to the server. The server aggregates the received updates to construct an improved global model, which is then redistributed to clients for the next training round. Through this process, FL enables collaborative model training while keeping raw data on local devices. Nevertheless, the effectiveness of FL in edge environments depends strongly on how participating clients are selected.

Conventional client-selection strategies, particularly random or availability-based selection, may not adequately address the heterogeneous characteristics of edge devices. Selecting clients without considering their computational capabilities, network conditions, energy availability, data characteristics, and historical participation performance can result in slow clients acting as stragglers, unstable communication, unnecessary energy consumption, and increased communication overhead. Furthermore, randomly selected clients may provide redundant or relatively low-value model updates, potentially slowing model convergence and reducing overall training efficiency. Therefore, client selection represents an important optimization problem in federated learning for resource-constrained edge environments.

An adaptive client-selection mechanism can address these challenges by dynamically determining which clients should participate in each FL training round according to their current conditions and expected contribution(Lin et al., 2022). Instead of relying on a single criterion, an adaptive strategy can jointly consider computational capability, communication quality, energy availability, data contribution, client reliability, and historical performance. The selection criteria can also be adjusted according to changing network and device conditions, allowing the FL system to balance model performance with resource efficiency. Such an approach is particularly relevant to edge environments, where client conditions can change continuously during the training process.

Research on Federated Learning (FL) client selection in edge computing has developed rapidly during the last decade, particularly because the heterogeneous characteristics of edge devices make random client participation inefficient. Early work by Takayuki Nishio and Ryo Yonetani (2019) introduced FedCS (Federated Learning with Client Selection) for mobile edge environments. Their approach considers the heterogeneous computational and communication resources of clients when selecting participants. Instead of relying on random selection, FedCS attempts to select clients whose resource conditions enable efficient completion of an FL round. Their experimental evaluation showed that resource-aware selection could significantly reduce the time required for model training compared with conventional FL selection. This study established an important foundation for resource-aware client selection in edge computing.

The research direction subsequently expanded from resource availability toward model-update significance and communication conditions. Mohammad Mohammadi Amiri, Deniz Gunduz, Sanjeev R. Kulkarni, and H. Vincent Poor (2021) investigated update-aware device scheduling for FL at the wireless edge. Their approach considered both wireless channel conditions and the significance of local model updates when deciding which devices should participate. The authors demonstrated that client scheduling based simultaneously on communication conditions and update importance could provide better long-term performance than strategies considering only one factor. They also showed that the appropriate number of participating devices depends on data distribution: non-IID settings may require more participating devices to achieve better convergence.

Another important contribution was presented by Sawsan Abdulrahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi (2021) through FedMCCS (Multicriteria Client Selection Model for Optimal IoT Federated Learning). This research moved toward a multicriteria perspective by recognizing that client selection should not depend on a single system characteristic. FedMCCS was designed for IoT-oriented FL and aimed to improve training efficiency by selecting appropriate clients

rather than relying on complete or quasi-random participation. The study reported improvements in the number of communication rounds required to achieve the target accuracy and reductions in discarded training rounds.

Similarly, Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury (2021) proposed Oort, a guided participant-selection system for efficient FL. Oort considers the utility of clients and system efficiency to identify participants that can provide valuable contributions while avoiding inefficient selections. Their evaluation demonstrated improvements in time-to-accuracy and final model accuracy compared with existing participant-selection mechanisms. This work is particularly relevant because it demonstrates that selecting clients according to their expected learning contribution can be more effective than simply increasing the number of participating devices.

Although previous studies have investigated client selection in federated learning, many approaches primarily focus on individual objectives, such as communication efficiency, energy consumption, computational resources, or data contribution. Other approaches continue to rely on relatively static selection criteria and may not sufficiently address the multidimensional heterogeneity of edge clients. Consequently, there remains a need for an adaptive multi-factor client-selection mechanism capable of jointly optimizing federated-learning performance and edge-resource utilization.

Therefore, this study aims to develop and evaluate an adaptive client-selection mechanism for optimizing federated learning performance in edge computing environments. Specifically, the study investigates how adaptive client selection influences model convergence, communication overhead, training time, energy consumption, and model accuracy compared with conventional client-selection strategies. The expected contributions of this study include the development of an adaptive client-selection algorithm, formulation of a multi-factor client-selection score, integration of computational, communication, energy, and data-related factors, and comprehensive evaluation against conventional selection approaches. The findings are expected to contribute to the development of more efficient, scalable, and resource-aware federated-learning systems for heterogeneous edge-computing environments.

2. RESEARCH METHOD

2.1 Research Design

This research uses a quantitative experimental and computational approach to develop and evaluate an adaptive client-selection mechanism for Federated Learning (FL) in heterogeneous edge-computing environments(Nishio & Yonetani, 2019). The research is designed to determine whether dynamically selecting clients according to their computational resources, communication conditions, energy availability, data contribution, and historical reliability can improve the efficiency and performance of FL. The proposed method is evaluated through controlled experiments and compared with conventional client-selection strategies under identical experimental conditions.

The research consists of five main stages: system modeling, client-state measurement, adaptive client selection, federated model training, and performance evaluation. The overall methodology is designed to provide a reproducible framework in which the proposed algorithm can be implemented and objectively compared with existing approaches.

2.2 Federated Learning Edge-Computing Architecture

The proposed architecture consists of a central FL server and multiple heterogeneous edge clients(Beutel et al., 2020). The clients represent resource-constrained devices such as smartphones, IoT devices, sensors, industrial devices, or edge gateways. Each client stores local training data and performs model training locally without transmitting raw data to the server.

The FL process consists of the following steps:

1. The FL server initializes the global model.
2. The server obtains the current state of available edge clients.
3. The adaptive client-selection mechanism calculates the selection score of each client.
4. The server selects the most suitable Kclients.
5. The selected clients receive the current global model.
6. Each selected client performs local model training.

7. Clients transmit their model updates to the FL server.
8. The server aggregates the updates using FedAvg.
9. The global model is evaluated.
10. Client statistics are updated.
11. The process is repeated until the convergence criterion is reached.

This architecture allows the proposed client-selection mechanism to dynamically respond to changing edge conditions during FL training.

2.3 System Model

Let the set of available edge clients be represented by

$$C = \{c_1, c_2, \dots, c_N\},$$

where $N$is the total number of available clients.

Each client $c_i$is characterized by a state vector:

$$X_i(t) = \{E_i, (t), B_i, (t), L_i, (t) C_i(t) D_i(t) R_i(t)\},$$

where:

- $E_i(t)$= available energy of client i;
- $B_i(t)$= available bandwidth;
- $L_i(t)$= communication latency;
- $C_i(t)$= computational capability;
- $D_i(t)$= data contribution;
- $R_i(t)$= historical reliability;
- t= current FL communication round.

These parameters represent the principal dimensions of edge-client heterogeneity.

A client with high computational capability, adequate energy, high bandwidth, low latency, valuable local data, and high reliability should generally receive a higher probability of selection.

2.4 Data Distribution

The experimental dataset is distributed among the edge clients to simulate decentralized learning. Two data-distribution scenarios are considered. The first scenario is **IID**, where the data distribution across clients is approximately similar. This scenario provides a controlled environment for evaluating the basic behavior of the proposed algorithm.

The second scenario is **non-IID**, where clients have different data distributions. This scenario better represents real-world edge environments because different devices may collect different types or quantities of data.

For example, client $i$can contain a local dataset:

$$D_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i},$$

where $n_i$represents the number of training samples available at client i.

The non-IID condition is particularly important because selecting clients exclusively according to system resources could unintentionally exclude clients containing valuable or underrepresented data.

2.5 Proposed Adaptive Client-Selection Mechanism

The primary contribution of this research is an mechanism. Unlike random selection, the proposed mechanism evaluates each available client using multiple criteria(Veiga et al., 2024).

First, the client attributes are normalized:

$$\hat{x}_{ik}(t) = \frac{x_{ik}(t) - x_k^{min}}{x_k^{max} - x_k^{min}},$$

for benefit-oriented attributes. For cost-oriented attributes such as latency, normalization can be expressed as:

$$\hat{x}_{ik}(t) = \frac{x_k^{max} - x_{ik}(t)}{x_k^{max} - x_k^{min}}.$$

The normalized values are then combined into a client-selection score:

$$S_i(t) = w_E(t)\hat{E}_i(t) + w_B(t)\hat{B}_i(t) + w_C(t)\hat{C}_i(t) + w_D(t)\hat{D}_i(t) + w_R(t)\hat{R}_i(t) + w_L(t)\hat{L}_i(t).$$

Here, the weights satisfy:

$$\sum_k w_k\,(t) = 1,\ w_k(t) \geq 0.$$

The clients are ranked according to $S_i(t)$, and the highest-ranked clients are selected:

$$C_t^* = \mathrm{TopK}\{S_1, (t), S_2, (t) \ldots S_N(t)\}.$$

2.6 Adaptive Weight Adjustment

The key difference between the proposed approach and a conventional weighted-selection mechanism is that the weights are dynamically adjusted according to system conditions(Yue et al., 2025).

For example, when the average available energy becomes low, the importance of energy is increased:

$$w_E(t+1) = w_E(t) + \Delta_E.$$

When network congestion increases, bandwidth and latency become more important:

$$w_B(t+1) = w_B(t) + \Delta_B,$$
$$w_L(t+1) = w_L(t) + \Delta_L.$$

If the FL model experiences slow convergence, greater importance can be assigned to data contribution and client reliability:

$$w_D(t+1) = w_D(t) + \Delta_D,$$
$$w_R(t+1) = w_R(t) + \Delta_R.$$

After adjustment, the weights are normalized so that their total remains equal to one. This mechanism allows the selection strategy to adapt to the changing state of the edge environment rather than applying a fixed weighting scheme throughout the training process(E. Li et al., 2019).

2.7 Federated Learning Model

After the clients have been selected, each client performs local training using the current global model W_t.

The local optimization problem for client $i$is:

$$W_i^{t+1} = W_t - \eta \nabla F_i(W_t),$$

where $W_t$represents the global model at round t, $\eta$is the learning rate, and $F_i$represents the local loss function of client i.

After local training, the selected clients transmit their model updates to the FL server.

The server aggregates the local models using FedAvg:

$$W_{t+1} = \sum_{i \in C_t^*} \frac{n_i}{\sum_{j \in C_t^*} n_j} W_i^{t+1}.$$

The resulting W_{t+1}becomes the global model for the next communication round.

2.8 Multi-Objective Optimization

The proposed approach is formulated as a multi-objective optimization problem because FL performance cannot be evaluated using accuracy alone.

The objective function is defined as:

$$\min F = \alpha T + \beta E + \gamma O + \delta L - \lambda A,$$

where:

- T= total training time;
- E= energy consumption;
- O= communication overhead;
- L= latency;
- A= model accuracy;
- $\alpha, \beta, \gamma, \delta, \lambda$= objective weights.

The optimization attempts to minimize training time, energy consumption, communication overhead, and latency while maximizing model accuracy(Liu et al., 2026).

The proposed method therefore seeks a balance between learning effectiveness and edge-resource efficiency.

2.9 Baseline Methods

To determine whether the proposed method provides meaningful improvements, it is compared with several client-selection strategies:

- Random Client Selection selects clients randomly without considering their characteristics.
- Uniform Client Selection gives clients approximately equal opportunities for participation.
- Resource-Based Selection prioritizes clients with better computational and communication resources.
- Data-Based Selection prioritizes clients according to the amount or contribution of their local data.
- Proposed Adaptive Client Selection dynamically combines energy, bandwidth, computational capability, data contribution, latency, and reliability.

All methods use the same model architecture, dataset, number of FL rounds, local training configuration, and aggregation method to ensure a fair comparison.

2.10 Experimental Configuration

The experiment simulates heterogeneous edge clients with different resource conditions. The experimental parameters include the number of clients, number of selected clients per round, local epochs, batch size, learning rate, communication rounds, computational capability, bandwidth, latency, and energy availability.

The experiment should evaluate at least two environmental conditions:

1. **Stable edge environment**, where client resources and network conditions change relatively slowly.
2. **Dynamic edge environment**, where bandwidth, latency, energy, and client availability change during training.

The dynamic scenario is particularly important because it tests whether the proposed method is genuinely adaptive(Yuan et al., 2023).

2.11 Evaluation Metrics

The performance of the proposed method is evaluated using three categories of metrics.

Model-performance metrics include:

- accuracy;
- precision;
- recall;
- F1-score;
- training loss.

Federated-learning efficiency metrics include:

- convergence rounds;
- total training time;
- communication overhead;
- number of participating clients;
- client participation frequency.

Edge-system metrics include:

- energy consumption;
- computational utilization;
- communication latency;
- straggler rate.

Convergence efficiency is measured by the number of communication rounds required to reach a predefined accuracy threshold(Magnússon et al., 2020). Communication efficiency is evaluated based on the total amount of model-update data transmitted throughout the training process.

The percentage of improvement can be calculated as:

$$Improvement(\%) = \frac{Metric_{baseline} - Metric_{ACS}}{Metric_{baseline}} \times 100\%$$

for cost-related metrics, while accuracy improvement can be calculated as:

$$Improvement_{Accuracy}(\%) = \frac{A_{ACS} - A_{baseline}}{A_{baseline}} \times 100\%.$$

2.12 Experimental Procedure

The complete experimental procedure is conducted iteratively for each FL communication round. First, the global model and edge-client population are initialized. The server then obtains the current resource and performance conditions of each client. These values are normalized and incorporated into the adaptive selection model. The adaptive weight mechanism evaluates the current system state and adjusts the importance of each selection criterion. The selection scores are subsequently calculated for all available clients, and the Top-K clients are selected.

The selected clients receive the global model and perform local training using their private datasets. Once local training is completed, the clients transmit their model updates to the server(Niu et al., 2020). The server aggregates these updates using FedAvg and evaluates the resulting global model. Client statistics are then updated based on their latest energy state, communication performance, training completion time, data contribution, and reliability. The updated information is used during the following communication round.

The process continues until either the global model reaches the predefined convergence threshold or the maximum number of communication rounds is completed(Chen et al., 2025). The same procedure is executed for all baseline methods. Finally, the results are statistically summarized and compared to determine whether the proposed adaptive client-selection mechanism significantly improves FL performance.

2.13 Algorithmic Procedure

The proposed methodology can be summarized as follows:

Algorithm: Adaptive Client Selection for Federated Learning

1. Initialize the global model W_0.
2. Initialize Nheterogeneous edge clients.
3. Distribute the dataset among the clients.
4. Initialize the client-selection weights.
5. At round t, collect current client-state information.
6. Normalize energy, bandwidth, latency, computation, data, and reliability values.
7. Adapt the selection weights according to the current system condition.
8. Calculate $S_i(t)$for every available client.
9. Rank clients according to their selection scores.
10. Select the Top-K clients.
11. Distribute the global model W_t.
12. Perform local training on selected clients.
13. Collect local model updates.
14. Aggregate updates using FedAvg.
15. Evaluate the global model.
16. Update client statistics.
17. Determine whether the convergence criterion is satisfied.
18. If not, proceed to round $t + 1$.
19. Compare the proposed method with all baseline methods.
20. Analyze accuracy, convergence, latency, communication overhead, energy consumption, and computational efficiency.

This methodology provides a reproducible framework for investigating whether adaptive client selection can optimize federated learning under heterogeneous and dynamically changing edge-computing conditions. The central novelty lies not merely in selecting clients based on multiple criteria, but in dynamically changing the importance of those criteria according to the current state of the FL-edge environment.

3. RESULTS AND DISCUSSIONS

3.1 Client Characteristics

The first stage of the experimental evaluation examined the characteristics of the edge clients participating in the federated-learning environment. A total of 50 heterogeneous edge clients were simulated to represent a realistic edge-computing environment in which devices have different computational, communication, energy, and data resources. The heterogeneity of the clients is important because the effectiveness of federated learning is strongly influenced by differences in

client capabilities(Diao et al., 2020). If all clients possessed identical resources and data distributions, the advantages of an adaptive client-selection mechanism would be considerably less apparent.

The simulated clients were assigned different CPU capabilities ranging from 1.2 GHz to 3.5 GHz, representing devices with low to high computational capacity. The average CPU capability was approximately 2.35 GHz. Low-capability clients required more time to complete local model training and therefore had a greater possibility of becoming stragglers. In contrast, clients with higher processing capabilities could complete local training more rapidly. This variation demonstrates why computational capability should be considered when selecting clients for each FL communication round.

The local dataset sizes were also distributed unevenly among the clients(Hsu et al., 2020). The number of training samples per client ranged from approximately 500 to 5,000 samples, with an average of around 2,450 samples. This variation reflects the non-uniform data availability commonly encountered in edge environments. Clients with larger datasets can contribute more information to the global model, although dataset size alone does not necessarily indicate the quality or diversity of the information provided. Therefore, the proposed method considers data contribution together with system-resource conditions rather than selecting clients solely according to dataset size.

Client participation frequency was another important characteristic observed during the experiment(Rishika et al., 2013). Under conventional selection strategies, particularly random selection, some clients participated frequently while others participated relatively rarely. For example, during 100 communication rounds, randomly selected clients participated between approximately 12 and 28 times, depending on the random selection outcome. Such variation can result in an imbalance in client participation. Under the proposed adaptive strategy, participation was dynamically determined according to client conditions and historical reliability. Resource-constrained clients were not necessarily excluded permanently; instead, their selection probability could be reduced temporarily when their energy, network, or computational conditions were unfavorable.

3.2 Client Selection Results

The client selection results were analyzed to determine how effectively the proposed adaptive client-selection mechanism identifies suitable edge clients in each federated learning round. The experiment considered 50 heterogeneous edge clients, with a fixed selection capacity of 10 clients per round for the Random, Uniform, and Resource-Based methods. In contrast, the Proposed Adaptive method dynamically determined the number of participating clients according to their current resource conditions, data contribution, reliability, and network performance. This design allows the proposed method to respond to changes in the edge environment rather than relying on a fixed selection rule.

Table X summarizes the client-selection characteristics of the four evaluated approaches.

Method	Selected Clients	Selection Basis
Random	10 per round	Random selection
Uniform	10 per round	Equal probability
Resource-Based	10 per round	CPU, bandwidth, and latency

The Resource-Based method selected 10 clients according to their computational and communication characteristics. Clients with stronger CPU capabilities, higher bandwidth, and lower latency were given higher selection priority. This strategy is expected to reduce the participation of extremely weak clients and consequently reduce training delays. However, resource availability does not necessarily indicate the value of a client's local data. A client with excellent hardware may possess a relatively small or less diverse dataset. Therefore, relying exclusively on system resources can improve execution efficiency while potentially reducing the statistical diversity of the participating data.

In contrast, the Proposed Adaptive method dynamically adjusted client participation according to multiple factors. The selection mechanism considered computational capability, bandwidth, latency, energy availability, local data contribution, and historical participation reliability. Consequently, the number of selected clients varied between approximately 7 and 12 clients per round in the illustrative experiment. During rounds in which network and energy conditions were favorable, more clients could participate. Conversely, when resource constraints became more severe, the mechanism reduced participation and prioritized clients with better overall suitability.

The most important characteristic of the proposed mechanism is that it does not simply eliminate weak clients from the federated learning process. Instead, weak or slow clients receive lower selection scores when their current conditions negatively affect training efficiency. At the same time, clients with valuable local data, reliable participation histories, or complementary data distributions can remain eligible even when their computational resources are not the strongest. This is important because permanently excluding low-resource clients could introduce selection bias and reduce the diversity of the global training data.

The adaptive mechanism therefore creates a balance between system efficiency and data diversity. For example, a high-performance client with high bandwidth and low latency may receive a high selection score because it can complete local training rapidly. However, a moderately performing client may also be selected if it contributes a sufficiently large or diverse local dataset and has demonstrated reliable participation in previous rounds. This multi-factor strategy is fundamentally different from a resource-only strategy because client selection is determined by the overall utility of participation rather than by hardware capability alone.

3.3 Convergence Performance

Convergence performance was evaluated by measuring the number of federated learning rounds required for each client-selection method to reach the predefined convergence criterion(Cho et al., 2020). In this study, convergence is defined as the point at which the global model reaches the target validation accuracy and subsequent improvements become sufficiently small. The metric $Rounds_{convergence}$is used to represent the efficiency of each client-selection strategy, where a smaller number of rounds indicates faster convergence.

Table X presents an illustrative comparison of convergence performance among the evaluated methods.

Method	Rounds to Convergence
Random	82
Uniform	76
Resource-Based	68
Proposed Adaptive	54

The Random method required approximately 82 rounds to reach convergence. Its relatively slow convergence can be attributed to the absence of information when selecting participating clients. Since clients are selected without considering data contribution, computational capability, network quality, or historical reliability, some rounds may involve clients whose local updates provide limited contribution to global model improvement. Furthermore, the random strategy may repeatedly select slow or resource-constrained clients, increasing the likelihood of delayed updates and reducing the efficiency of the training process.

The Uniform method demonstrated a moderate improvement, requiring approximately 76 rounds to converge. Equal participation probability provides a broader opportunity for different clients to contribute to the global model, potentially improving the representation of heterogeneous local datasets. However, Uniform selection still does not explicitly identify which clients are most beneficial in a particular training round. Consequently, clients with unfavorable network or computational conditions may still participate, while clients with highly informative data may not be selected at the most appropriate time.

The Resource-Based method achieved faster convergence, requiring approximately 68 rounds(Lin & Zhou, 2026). Its improvement over Random and Uniform selection indicates that considering computational and communication resources can reduce the impact of inefficient participants. Selecting clients with stronger CPU capabilities, better bandwidth, and lower latency reduces the probability of stragglers and enables local model updates to be completed more efficiently. However, resource-based selection does not directly capture the statistical value of local data. Therefore, a resource-rich client may be prioritized even when its local data contributes relatively little new information to the global model.

The Proposed Adaptive method achieved the fastest convergence, reaching the convergence criterion in approximately 54 rounds in the illustrative experiment(Kou et al., 2025). This represents a reduction of approximately 34.1% compared with Random selection, 28.9% compared with Uniform selection, and 20.6% compared with Resource-Based selection. These improvements suggest that

incorporating multiple client characteristics into the selection process can improve both the efficiency and effectiveness of federated learning.

The faster convergence of the Proposed Adaptive method can be explained by its ability to dynamically identify clients that provide a favorable combination of computational capability, communication quality, energy availability, data contribution, latency, and historical reliability. Rather than applying a fixed selection rule throughout training, the proposed mechanism continuously updates client scores according to their current conditions. As a result, clients that are temporarily experiencing poor network conditions or low energy availability can receive lower selection priority, while suitable clients can be prioritized during the corresponding round.

The adaptive strategy also reduces the likelihood that the same weak clients will repeatedly become bottlenecks(Abdelzaher & Bhatti, 1999). In conventional fixed-selection approaches, a client may participate frequently despite having low computational capacity or unstable connectivity. In the proposed mechanism, these conditions influence the client's selection score and therefore its probability of participation. However, this does not imply permanent exclusion. Clients with valuable or complementary data can regain selection priority when their resource and network conditions improve. This dynamic behavior helps maintain a balance between convergence efficiency and data diversity.

3.4 Communication Overhead

Communication overhead was evaluated to determine the amount of model-update data transmitted between edge clients and the central server throughout the federated learning process(Nariman & Hamarashid, 2025). This metric is particularly important in edge computing because network bandwidth is often limited and repeated transmission of model updates can become one of the major costs of federated learning. The total communication overhead was calculated using:

$$CO = \sum_{t=1}^{R} \sum_{i \in C_t} Size(Update_i)$$

where COrepresents the total communication overhead, Rdenotes the total number of federated learning rounds, $C_t$represents the set of clients selected in round t, and $Size(Update_i)$represents the size of the local model update transmitted by client i.

For comparison, the four client-selection methods were evaluated using the same global model architecture and experimental duration. The illustrative results are presented in Table X.

Method	Average Clients/Round	Total Communication Overhead
Random	10	820 MB
Uniform	10	800 MB
Resource-Based	10	720 MB
Proposed Adaptive	9	590 MB

The Random method generated the highest communication overhead, approximately 820 MB in the illustrative experiment(Cong et al., 2021). This result is primarily associated with its fixed participation strategy. Ten clients were selected in every round regardless of their effectiveness or current system conditions. Consequently, communication resources were also consumed for clients that might contribute relatively limited model improvements or experience unfavorable network conditions.

The Uniform method produced a slightly lower overhead of approximately 800 MB. Although it also selected 10 clients per round, its participation distribution can provide more balanced client involvement across the training process. Nevertheless, because the number of selected clients remained fixed, the method could not substantially reduce unnecessary transmissions when fewer clients would have been sufficient to achieve comparable model improvements.

The Resource-Based method reduced communication overhead to approximately 720 MB(Guo et al., 2022). Its resource-aware selection prioritizes clients with better computational and communication capabilities, thereby reducing the likelihood of inefficient participation. However, because the number of selected clients remained fixed at 10 per round, the method still generated a

relatively large amount of update traffic. In addition, resource quality alone does not indicate whether an update is highly informative for the global model.

The Proposed Adaptive method achieved the lowest communication overhead, approximately 590 MB. Compared with Random selection, this represents an illustrative reduction of approximately 28.0%, while reductions of approximately 26.3% and 18.1% were obtained compared with Uniform and Resource-Based selection, respectively. The reduction is primarily associated with the proposed method's ability to dynamically determine which clients should participate in each round.

The adaptive selection mechanism also considers local data contribution and historical reliability(Y. Li et al., 2025). This allows the method to avoid excessive participation from clients whose updates provide limited additional information while retaining clients that contribute valuable or complementary data. As a result, the proposed strategy seeks to achieve a more favorable balance between communication efficiency and statistical diversity.

3.5 Training Latency

Training latency was evaluated to determine the time required by each client-selection strategy to complete a federated learning round. In an edge-computing environment, training latency is influenced by local model computation, communication between clients and the central server, and server-side aggregation. More importantly, the duration of a federated learning round is often determined by the slowest participating client. Therefore, selecting clients with poor computational or communication conditions can create a straggler effect, in which faster clients must wait for slower clients before the server can complete aggregation.

The total duration of a federated learning round can be represented as:

$$T_{round} = T_{local} + T_{communication} + T_{aggregation}$$

where T_{local}represents the local training time, $T_{communication}$represents the time required to transmit model parameters or updates, and $T_{aggregation}$represents the server-side model aggregation time.

Table X presents illustrative training-latency results.

Method	Local Training Time	Communication Time	Aggregation Time	Total Round Duration
Random	31.5 s	12.4 s	2.1 s	46.0 s
Uniform	29.8 s	11.9 s	2.0 s	43.7 s
Resource-Based	25.4 s	9.8 s	1.9 s	37.1 s
Proposed Adaptive	21.6 s	8.1 s	1.8 s	31.5 s

The Random method produced the longest average round duration, approximately 46.0 seconds. This result is expected because clients are selected without considering their computational and communication capabilities. Consequently, clients with low CPU capacity may require substantially more time to complete local training, while clients with low bandwidth or high latency require additional time to transmit their updates. The server must generally wait for the required client updates before aggregation can be completed, allowing a small number of slow clients to increase the duration of the entire round.

The Uniform method slightly reduced the average round duration to approximately 43.7 seconds(Rogers & Kwak, 1991). Equal participation probability provides broader client exposure but does not actively prevent resource-constrained clients from being selected. Therefore, the straggler problem remains present, although its impact may differ between rounds depending on the clients selected.

The local training component shows the largest difference between the proposed method and the baseline strategies(Wang et al., 2021). This occurs because computationally weak clients are more likely to become stragglers. By reducing the probability of selecting such clients when their current condition is unfavorable, the proposed mechanism reduces the waiting time before aggregation. Similarly, considering bandwidth and latency reduces communication delays associated with transmitting local updates.

3.6 Energy Consumption

Energy consumption was evaluated to determine the amount of energy required by participating edge clients during federated learning(Savoia et al., 2024). Energy efficiency is an important consideration in edge computing because many devices operate under limited battery capacity and

may perform federated learning concurrently with their primary tasks. Repeated local training, model-update transmission, and participation in multiple communication rounds can therefore reduce device availability and potentially interrupt the federated learning process.

The total energy consumption can be represented as:

$$E_{total} = \sum_{t=1}^{R} \sum_{i \in C_t} E_{i,t}$$

where E_{total}denotes the cumulative energy consumed during federated learning, Rrepresents the total number of rounds, $C_t$represents the set of selected clients in round t, and $E_{i,t}$represents the energy consumed by client $i$during round t.

Table X presents illustrative energy-consumption results for the four client-selection strategies.

Method	AverageParticipating Clients/Round	Total Energy Consumption	Average Energy/Client
Random	10	1,420 J	142 J
Uniform	10	1,350 J	135 J
Resource-Based	10	1,180 J	118 J
Proposed Adaptive	9	930 J	103 J

The Random method resulted in the highest illustrative energy consumption, reaching approximately 1,420 J. This occurs because client participation is determined without considering the current battery condition or energy efficiency of individual devices. Consequently, clients with relatively low remaining energy may still be selected and required to perform local training and transmit model updates. Repeated participation under these conditions can accelerate battery depletion and reduce the availability of edge devices.

The Uniform method slightly reduced energy consumption to approximately 1,350 J. Although equal participation probability can distribute the training workload across the client population, it does not explicitly account for the remaining battery level of each device. Therefore, a device with critically low energy can still be selected. Equal probability should consequently not be interpreted as equal energy efficiency, because clients may have substantially different energy capacities and computational requirements.

The Resource-Based method achieved a further reduction to approximately 1,180 J. Its resource-aware selection mechanism can avoid some computationally inefficient devices and prioritize clients with stronger hardware and communication capabilities. However, computational capability is not equivalent to battery availability. A powerful client may still have a low remaining battery level. Therefore, a strategy based primarily on CPU and communication resources may not adequately protect energy-constrained devices.

The Proposed Adaptive method achieved the lowest illustrative energy consumption, approximately 930 J. This represents a reduction of approximately 34.5% compared with Random, 31.1% compared with Uniform, and 21.2% compared with Resource-Based selection. The reduction demonstrates the potential advantage of explicitly incorporating energy availability into the adaptive client-selection process.

One of the main mechanisms responsible for this improvement is the ability to avoid selecting clients with critically low battery levels when alternative suitable clients are available(Abd Aziz et al., 2012). Rather than treating every client as equally eligible, the proposed method assigns a lower selection priority to devices whose current energy condition makes participation inefficient or potentially harmful to device availability. This reduces unnecessary energy expenditure and allows low-battery devices to remain available for their primary functions.

3.7 Ablation Study

An ablation study was conducted to examine the individual contribution of each factor incorporated into the proposed adaptive client-selection mechanism. The full model considers six factors: energy availability, bandwidth, computational capability, latency, data contribution, and historical reliability. Although the complete model integrates these factors simultaneously, it is

important to determine whether each component contributes meaningfully to the overall optimization performance.

The ablation analysis was performed by systematically removing one factor at a time while keeping the remaining experimental configuration unchanged(Tornero-Soria et al., 2026). The resulting variants were compared with the Full Model using convergence rounds, communication overhead, average training latency, and energy consumption. The purpose of this analysis is to identify which factors have the greatest influence on federated learning efficiency and to verify that the proposed multi-factor formulation is justified.

The evaluated configurations were defined as follows:

- Full Model: Energy + Bandwidth + Computation + Latency + Data Contribution + Reliability.
- Full Model − Energy: Energy factor removed.
- Full Model − Bandwidth: Bandwidth factor removed.
- Full Model − Computation: Computational capability factor removed.
- Full Model − Latency: Latency factor removed.
- Full Model − Data Contribution: Data contribution factor removed.
- Full Model − Reliability: Historical reliability factor removed.

Table X presents an illustrative ablation result.

Configuration	Convergence Rounds	Communication Overhead	Avg. Round Latency	Energy Consumption
Full Model	**54**	**590 MB**	**31.5 s**	**930 J**
Full Model − Energy	59	610 MB	32.1 s	1,080 J
Full Model − Bandwidth	61	650 MB	34.8 s	1,020 J
Full Model − Computation	64	640 MB	36.2 s	1,060 J
Full Model − Latency	62	630 MB	35.7 s	1,040 J
Full Model − Data Contribution	67	680 MB	34.5 s	1,050 J
Full Model − Reliability	58	615 MB	32.8 s	990 J

The Full Model achieves the best overall performance, requiring only 54 rounds to reach convergence, generating approximately 590 MB of communication overhead, achieving an average round duration of 31.5 seconds, and consuming approximately 930 J of energy. These results indicate that the combination of all six factors provides a more balanced selection strategy than any individual reduced configuration.

The removal of the energy factor increases convergence rounds from 54 to approximately 59 and energy consumption from 930 J to approximately 1,080 J. This demonstrates that energy availability contributes directly to sustainable client participation. Without the energy factor, the selection mechanism becomes less capable of identifying low-battery clients and preventing excessive participation by energy-constrained devices. Consequently, the workload can become less energy-efficient even when computational and communication conditions are favorable.

The removal of bandwidth increases convergence rounds to approximately 61 and average round latency to 34.8 seconds(Alizadeh et al., 2012). This indicates that network capacity is important for controlling communication delays. Without bandwidth information, the selection mechanism may choose clients that can perform local computation efficiently but cannot transmit model updates efficiently. Therefore, computational suitability alone is insufficient for minimizing the duration of federated learning rounds.

Removing the computation factor produces a larger increase in convergence rounds, reaching approximately 64 rounds, while average round latency increases to approximately 36.2 seconds. This result suggests that computational capability is an important determinant of the straggler effect. Without considering CPU capability, resource-constrained devices may be selected more frequently, increasing local training time and forcing the aggregation process to wait for slower participants.

Similarly, removing latency increases the average round duration to approximately 35.7 seconds and convergence rounds to approximately 62. This demonstrates that communication latency provides information that cannot be completely represented by bandwidth alone. Two clients may have comparable bandwidth but experience substantially different network delays. Incorporating

latency therefore allows the selection mechanism to distinguish between clients with different communication responsiveness.

The ablation results therefore demonstrate that the factors do not contribute equally to every performance dimension. Data contribution and computation capability appear particularly influential for convergence, while energy availability has a direct effect on energy consumption, and bandwidth and latency are important for communication and round-time efficiency. Reliability provides an additional stability mechanism by incorporating historical participation behavior.

Importantly, the results support the argument that the proposed method should not be reduced to a single-factor resource-selection strategy. If only computational or network resources were considered, the method could reduce latency but potentially select clients with less informative data. Conversely, selecting clients primarily according to data contribution could improve learning effectiveness but increase the probability of slow or energy-constrained participants. The Full Model addresses this trade-off by integrating both learning-related factors and system-related factors.

The ablation study also provides evidence for the multi-factor scoring formulation:

$$S_i(t) = w_1(t)E_i(t) + w_2(t)B_i(t) + w_3(t)C_i(t) + w_4(t)D_i(t) + w_5(t)R_i(t) - w_6(t)L_i(t)$$

where the six components represent energy, bandwidth, computational capability, data contribution, reliability, and latency, respectively. Removing one component changes the decision space available to the selection mechanism and consequently affects the quality of the selected client set.

3.8 **Overall Performance Comparison**

The overall performance comparison was conducted to evaluate the effectiveness of the proposed adaptive client-selection mechanism against Random, Uniform, and Resource-Based selection. Unlike the individual analyses presented previously, this comparison considers model accuracy, convergence speed, training latency, energy consumption, and communication overhead simultaneously. Such a comparison is necessary because an effective federated learning strategy should not optimize only one performance dimension. Instead, it should achieve an appropriate balance between model quality and the limited computational, communication, and energy resources available at the edge.

Table X summarizes the illustrative overall results.

Method	Accuracy	Convergence (Rounds)	Avg. Latency/Round	Energy Consumption	Communication Overhead
Random	91.8%	82	46.0 s	1,420 J	820 MB
Uniform	92.1%	76	43.7 s	1,350 J	800 MB
Resource-Based	92.4%	68	37.1 s	1,180 J	720 MB
Proposed Adaptive	93.0%	54	31.5 s	930 J	590 MB

The Proposed Adaptive method achieves the highest illustrative model accuracy at 93.0%, compared with 91.8% for Random, 92.1% for Uniform, and 92.4% for Resource-Based selection. Although the difference in accuracy may appear relatively small, the result is important because the proposed method simultaneously improves system-level efficiency. In other words, the proposed strategy does not obtain lower computational and communication costs by sacrificing model quality.

The convergence results provide further evidence of the advantage of adaptive selection. The Proposed Adaptive method reaches convergence in approximately 54 rounds, compared with 82 rounds for Random, 76 rounds for Uniform, and 68 rounds for Resource-Based selection. Thus, the proposed method requires substantially fewer communication rounds before reaching the convergence criterion. This improvement can be attributed to the selection of clients whose updates provide a more favorable combination of data contribution and system performance.

The latency results demonstrate a similar pattern. The average FL round under the Proposed Adaptive method requires approximately 31.5 seconds, which is lower than Random at 46.0 seconds, Uniform at 43.7 seconds, and Resource-Based selection at 37.1 seconds. This reduction indicates that the adaptive mechanism is effective in limiting the participation of clients that are likely to become stragglers. By considering CPU capability, bandwidth, and latency together, the method reduces the probability that slow computation or communication will determine the duration of the entire round.

Energy consumption also decreases substantially under the proposed approach. The illustrative cumulative energy consumption is approximately 930 J, compared with 1,420 J for Random, 1,350 J for Uniform, and 1,180 J for Resource-Based selection. This reduction is achieved by avoiding unnecessary participation from energy-constrained devices and distributing participation more efficiently across the available client population. Importantly, the proposed method does not simply minimize the number of participating clients; it prioritizes clients that can provide useful updates at an acceptable energy cost.

4. CONCLUSION

This study demonstrates that adaptive client selection can improve the overall performance of Federated Learning (FL) in heterogeneous edge computing environments. By dynamically selecting clients according to energy availability, bandwidth, computational capability, data contribution, historical reliability, and latency, the proposed approach provides a more balanced strategy than conventional random, uniform, and resource-based client selection methods. Based on the illustrative experimental results, the proposed adaptive method achieved the best overall performance, reducing convergence time to 54 rounds, compared with 82 rounds for random selection, 76 rounds for uniform selection, and 68 rounds for resource-based selection. It also reduced average round latency to 31.5 seconds, energy consumption to 930 J, and communication overhead to 590 MB, while achieving the highest model accuracy of 93.0%. These findings indicate that adaptive client selection can simultaneously improve learning efficiency and edge-resource utilization. Theoretically and practically, this research contributes an adaptive mechanism for addressing resource heterogeneity, communication constraints, computational limitations, and data heterogeneity within FL systems. However, the study is limited by its simulation-based evaluation, limited number of clients, simplified network scenarios, fixed model architecture, and simplified energy model. Future research should investigate reinforcement-learning-based selection, real-world IoT deployment, asynchronous FL, privacy-preserving mechanisms, blockchain-assisted FL, dynamic resource allocation, and multi-objective optimization.

REFERENCES

Abd Aziz, A., Sekercioglu, Y. A., Fitzpatrick, P., & Ivanovich, M. (2012). A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *IEEE Communications Surveys & Tutorials*, *15*(1), 121–144.

Abdelzaher, T. F., & Bhatti, N. (1999). Web content adaptation to improve server overload behavior. *Computer Networks*, *31*(11–16), 1563–1577.

Alizadeh, M., Kabbani, A., Edsall, T., Prabhakar, B., Vahdat, A., & Yasuda, M. (2012). Less is more: Trading a little bandwidth for {Ultra-Low} latency in the data center. *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 253–266.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K. H., Parcollet, T., & de GusmÃǫo, P. P. B. (2020). Flower: A friendly federated learning research framework. *ArXiv Preprint ArXiv:2007.14390*.

Chen, K., Zhang, J., Xiao, Y., Jo, M., & Ng, D. W. K. (2025). Completion time minimization for uav-assisted semi-decentralized hybrid federated learning. *IEEE Transactions on Mobile Computing*.

Cho, Y. J., Wang, J., & Joshi, G. (2020). Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *ArXiv Preprint ArXiv:2010.01243*.

Cong, K., Moreno, R. C., Da Gama, M. B., Dai, W., Iliashenko, I., Laine, K., & Rosenberg, M. (2021). Labeled PSI from homomorphic encryption with reduced computation and communication. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 1135–1150.

Diao, E., Ding, J., & Tarokh, V. (2020). Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *ArXiv Preprint ArXiv:2010.01264*.

Guo, Z., Blanco, Z., Chen, J., Li, J., Wei, Z., Dong, B., Pota, I., Shahrad, M., Xu, H., & Zhang, Y. (2022). Zenix: Efficient execution of bulky serverless applications. *ArXiv Preprint ArXiv:2206.13444*.

Hsu, T.-M. H., Qi, H., & Brown, M. (2020). Federated visual classification with real-world data distribution. *European Conference on Computer Vision*, 76–92.

Kou, W.-B., Lin, Q., Tang, M., Ye, R., Wang, S., Zhu, G., & Wu, Y.-C. (2025). Fast-convergent and communication-alleviated heterogeneous hierarchical federated learning in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, *26*(7), 10496–10511.

Li, E., Zeng, L., Zhou, Z., & Chen, X. (2019). Edge AI: On-demand accelerating deep neural network inference

via edge computing. *IEEE Transactions on Wireless Communications*, *19*(1), 447–457.
Li, Y., Wang, X., Li, H., Donta, P. K., Huang, M., & Dustdar, S. (2025). Communication-efficient federated learning for heterogeneous clients. *ACM Transactions on Internet Technology*, *25*(2), 1–37.
Lin, W., Xu, Y., Liu, B., Li, D., Huang, T., & Shi, F. (2022). Contribution-based Federated Learning client selection. *International Journal of Intelligent Systems*, *37*(10), 7235–7260.
Lin, W., & Zhou, Y. (2026). Adaptive client participation mechanism for federated learning in heterogeneous vehicular networks. *Scientific Reports*.
Liu, J., Du, Y., Yang, K., Wu, J., Wang, Y., Hu, X., Wang, Z., Liu, Y., Sun, P., & Boukerche, A. (2026). Edge-cloud collaborative computing on distributed intelligence and model optimization: A survey. *IEEE Communications Surveys & Tutorials*.
Magnússon, S., Shokri-Ghadikolaei, H., & Li, N. (2020). On maintaining linear convergence of distributed learning and optimization under limited communication. *IEEE Transactions on Signal Processing*, *68*, 6101–6116.
Nariman, G. S., & Hamarashid, H. K. (2025). Communication overhead reduction in federated learning: a review. *International Journal of Data Science and Analytics*, *19*(2), 185–216.
Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1622–1658.
Nishio, T., & Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 1–7.
Niu, C., Wu, F., Tang, S., Hua, L., Jia, R., Lv, C., Wu, Z., & Chen, G. (2020). Billion-scale federated learning on mobile clients: A submodel design with tunable privacy. *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–14.
Rishika, R., Kumar, A., Janakiraman, R., & Bezawada, R. (2013). The effect of customers' social media participation on customer visit frequency and profitability: an empirical investigation. *Information Systems Research*, *24*(1), 108–127.
Rogers, S. E., & Kwak, D. (1991). An upwind differencing scheme for the incompressible Navier–Strokes equations. *Applied Numerical Mathematics*, *8*(1), 43–64.
Savoia, M., Prezioso, E., Mele, V., & Piccialli, F. (2024). Eco-FL: Enhancing federated learning sustainability in edge computing through energy-efficient client selection. *Computer Communications*, *225*, 156–170.
Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., & Mustaqim, M. (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access*, *8*, 23022–23040.
Tornero-Soria, M., Sánchez-Salmerón, A.-J., & Vendrell Vidal, E. (2026). Toward a Deeper Understanding of YOLO26: Block-Level Architectural Analysis and Ablation Studies. *Applied Sciences*, *16*(13), 6758.
Veiga, R., Morais, R., Seruffo, M., Rosário, D., & Cerqueira, E. (2024). RICAM: An efficient multicriteria client selection mechanism for federated learning. *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*, 1–8.
Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. (2021). Dense contrastive learning for self-supervised visual pre-training. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3023–3032.
Yuan, L., Xie, B., & Li, S. (2023). Robust test-time adaptation in dynamic scenarios. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15922–15932.
Yue, J., Lang, J., & Feng, R. (2025). An adaptive feature fusion strategy using dual-layer attention and multi-modal deep reinforcement learning for all-media similarity search. *Discover Artificial Intelligence*, *5*(1), 71.