FEDRESOURCE: Federated Learning Based Resource Allocation in Modern Wireless Networks

Original Scientific Paper

P. G. Satheesh
Research Scholar,
Sathyabama Institute of Science and Technology,
Chennai, Tamil Nadu, India
reachpgs@gmail.com

T. Sasikala
Prof. & Dean, School of Computing,
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology,
Chennai, Tamil Nadu, India
dean.computing@sathyabama.ac.in / sasi.madhu2k2@yahoo.co.in

Abstract *– Deep reinforcement learning can effectively deal with resource allocation (RA) in wireless networks. However, more complex networks can have slower learning speeds, and a lack of network adaptability requires new policies to be learned for newly introduced systems. To address these issues, a novel federated learning-based resource allocation (FEDRESOURCE) has been proposed in this paper which efficiently performs RA in wireless networks. The proposed FEDRESOURCE technique uses federated learning (FL) which is a ML technique that shares the DRL-based RA model between distributed systems and a cloud server to describe a policy. The regularized local loss that occurs in the network will be reduced by using a butterfly optimization technique, which increases the convergence of the FL algorithm. The suggested FL framework speeds up policy learning and allows for adoption by employing deep learning and the optimization technique. Experiments were conducted using a Python-based simulator and detailed numerical results for the wireless RA sub-problems. The theoretical results of the novel FEDRESOURCE algorithm have been validated in terms of transmission power, convergence of algorithm, throughput, and cost. The proposed FEDRESOURCE technique achieves maximum transmit power up to 27%, 55%, and 68% energy efficiency compared to Scheduling policy, Asynchronous FL framework, and Heterogeneous computation schemes respectively. The proposed FEDRESOURCE technique can increase discrimination accuracy by 1.7%, 1.2%, and 0.78% compared to the scheduling policy framework, Asynchronous FL framework, and Heterogeneous computation schemes respectively.*

Keywords*: Deep reinforcement learning, federated learning, resource allocation, butterfly optimization technique*

1. INTRODUCTION

Modern wireless networks and mobile devices frequently come with sophisticated sensors and powerful computers, enabling them to acquire and interpret enormous amounts of data produced at the network edge [1]. The 5th generation (5G) wireless networks have strengthened the traditional connection service and supported many vertical industries [2]. A cloud and edge computing system [3, 4] that intelligently uploads user tasks to a cloud data center layer and an edge computing layer can provide computation and data storage services. Implementing energy-efficient node setup and RA in the course of cooperative operations is a major difficulty in wireless networks due to the high quality of service (QoS) requirements of IoT applications.

The efficient RA scheme can extend sensors' lifetime and play a major role in maximizing system performance along with better scheduling [3-5]. Utilizing machine learning (ML) techniques [6, 7], with a variety of RA strategies have recently been investigated which reduces the wireless networks becoming increasingly complex [8]. Particularly for difficult decision-making issues, deep reinforcement learning (DRL) has been applied extensively [9]. They can be used to train a DL model with a large representation capacity to develop a RA strategy for complicated networks. However, such DRL-based approaches still face significant obstacles in practice [10, 11].

An important problem is the policy's inability to respond to changes in network needs [12]. Wireless networks often introduce new systems with the same goals as current ones [13, 14]. It is therefore possible to apply policies to newly arrived systems with no additional learning if they are network adaptable [15, 16]. By using DRL-based wireless network approaches, we can deploy them more effectively than before. In this paper, a novel federated learning-based resource allocation (FEDRESOURCE) has been proposed, which efficiently performs RA in wireless networks. The main contributions of the FEDRESOURCE framework are as follows.

- The proposed FEDRESOURCE technique uses federated learning (FL) optimized using butterfly optimization for resource allocation in wireless networks, that share the DRL-based RA model between distributed systems and a cloud server to describe a policy.
- The policy for the RA in wireless networks can be learned collaboratively while taking use of the FL technique.
- The regularized local loss that occurs in the network will be reduced by using a butterfly optimization technique, which increases the convergence of the FL algorithm.
- The suggested FL framework speeds up policy learning and allows for adoption by employing deep learning and the optimization technique.

The remainder of the paper is organized as follows: Section 2 presents a thorough survey of current efforts on federated learning-based RA techniques. Section 3, presents the system model and problem formulation for resource allocation. Section 4 presents the design of the FEDRESOURCE framework in detail. Section 5 presents the experimental data and its analysis. Section 6 presents the conclusions and suggestions for future research.

2. LITERATURE REVIEW

Due to the high Quality-of-Service (QoS) requirements of IoT applications, resource scheduling wireless networks are becoming more important for better service. Several techniques have been developed by many experts for RA in wireless networks. Among these, we have discussed a few algorithms here. In [16] authors introduced a circumstance-independent policy that can successfully address the various network scenarios even with a single policy. Based on the outcomes of the simulation, a single suggested policy can be applied in a range of circumstances with results that are comparable to those of a situation-based policy, which chooses the appropriate course of action for each circumstance on its own.

In [17] authors presented a heterogeneous computation and RA approach based on heterogeneous mobile architectures. Using simulation data, the proposed scheme improves the energy efficiency of the wireless-powered FL system more than the baseline systems, according to the simulation data.

In [18] authors recommended using communication pipelining to enable FL in mobile edge computing applications to become more efficient at utilizing wireless spectrum and to become more concurrent. They also provide numerical findings that highlight the benefits of the suggested technique for various datasets and deep learning architectures.

In [19] authors proposed a new asynchronous FL framework that considers time-varying local training data, wireless link conditions, and computing capability. The framework also uses a dynamic scheduling algorithm to optimize learning performance under long-term energy constraints and per-round latency requirements. The proposed architecture has been demonstrated to improve learning performance and system efficiency over other approaches through numerical simulations.

In [20] authors proposed a scheduling policy that took user device training data representation and channel quality into consideration simultaneously. Based on simulations, the channel-aware data importance-based scheduling policy is shown to be more efficient than cutting-edge FL methods. In an asynchronous FL environment, an "age-aware" aggregation weighting approach can also improve learning performance.

In [21] authors established an efficient integration of common edge intelligence nodes based on research on energy-efficient bandwidth allocation, CPU frequency calculation, optimized transmission performance, and required level of learning accuracy. Based on the simulation results, the proposed Alternative Direction Algorithm (ADA) can reduce energy consumption while slightly increasing FL time in the central processing unit. There have been few studies that examine FL design in wireless networks for RA. However, FL structures used in the literature are not very effective, and model updates derived from old global models may have limited meaningful information about the current version, resulting in slow convergence. To the best of our knowledge, no work has specifically addressed how to use FL to resolve a wireless network RA issue. Instead, to efficiently run FL on wireless networks, current studies concentrate on finding a solution to the RA problem in wireless networks.

2.1. DIFFERENCES BETWEEN THE EXISTING AND PROPOSED WORK

The important findings from their research as well as the differences between the proposed study and the existing work are given below.

a. Unlike the proposed method many of the algorithms did not consider FL to resolve a wireless network RA issue

b. A new FEDRESOURCE technique that uses federated learning (FL) that shares the DRL-based RA model between distributed systems is presented in the proposed work that is not included in any other method so far, which makes it unique and significant from existing methods.

c. Proposed a federated learning architecture that incorporates policy chosen from DRL for resource allocation in wireless networks.

d. The existing techniques used in the literature are not very effective, and model updates derived from old global models may have limited meaningful information about the current version, resulting in slow convergence. However, it is discovered that the suggested method increases the convergence rate.

3. SYSTEM MODEL AND PROBLEM FORMULATION

We take into account a downlink of numerous TDMA networks in wireless networks, where each network has a local model and a global model, as shown in Fig. 1. R=1,... ,S, where S is the total number of systems, defines the set of systems. $\mho_s$=1,…,U_s, where U_s is the total number of sensing nodes (SN) in system s, defines the set of SN in system s. In the system $s \in R$, its AP provides services to U_s SN across discrete time intervals of $k \in \{1,2,\ldots\}$. We make the commonly recognized assumption that each system's wireless channels between the global model and local model are time-varying but constant during a timeslot and satisfy the Markov property. The global model schedules one user and transmission power in timeslot k of system s, where T is the set of potential transmission power levels. P_s is the definition of a state of the system s in timeslot k, and $p_s^{\,k}$ is the state space of the system s. Each user's feature information, including channel gain and QoS satisfaction levels, is represented by the state.

We use a tuple to represent SN m in system s in simple notation (s, m). We use $f^{\,k}_{\,s,m,i}$ to denote the ith feature information of the user (s, m) in timeslot k. $c_s^{\,k}$=($m_s^{\,k}$,$T_s^{\,k}$)∈ C_s is the definition of an action of system s in timeslot k, which denotes a scheduling choice. A system s policy is indicated by the notation π_s:P_s→C_s. Then, l($p_s^{\,k}$, π_s($p_s^{\,k}$)), where l(.,.) is a utility function typically utilized in the systems and is used to express the instantaneous utility of system s in timeslot k. In wireless networks with numerous systems, we can construct a general RA issue to maximize overall utility as follow state represents each individual's feature information.

$$maximize_{\pi} \sum_{s \epsilon \mathcal{R}} \mathbb{E}[\sum_{k=0}^{\infty} (\gamma)^k \, l(p_s^k, \pi_s, (p_s^k))] \qquad (1)$$

where the discount factor is 0 < γ < 1 and the policy for all systems is π:$\prod_{s \in R} P_s \rightarrow \prod_{s \in R} C_s$. For instance, one could use the formula l($p_s^{\,k}$, π_s, ($p_s^{\,k}$)) = d($p_s^{\,k}$, π_s,($p_s^{\,k}$)) to frame the problem of maximizing the total average data rate. where the function d(.,.) calculates the current data rate.

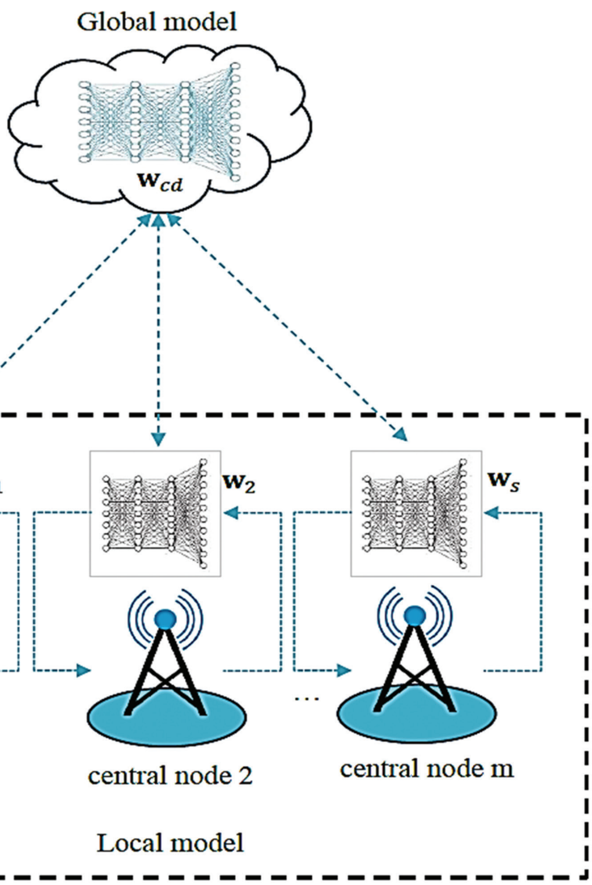

Fig. 1. System model

3.1. FEDERATED LEARNING FOR DYNAMIC RESOURCE ALLOCATION

When S, the number of systems, is high, the complexity of the problem is too great to tackle. To fix this problem, we break it down into its parts according to each system s as

$$\text{maximize}_{\pi_s}\ \mathbb{E}\left[\sum_{k=0}^{\infty} (\gamma)^k l\left(p_s^k, \pi_s(P_s^k)\right)\right] \qquad (2)$$

A (sub)optimal rule for the distributed system s deconstructed problem may then be identified, and the problem can be resolved. A Markov decision process is used to solve the decomposed problem, and the reward function and environment are the utility function $l(.,.)$ and transition probabilities over the state and action spaces, respectively. The well-known DRL can therefore be utilized to resolve the system-wise deconstructed problem, like the most recent research on RA in wireless networks. In particular, every system s participates in the DRL as an agent to learn the best policy $\pi{*}s$ for the problem's decomposition. Decomposed problems fall under the same class as other problems that use the same utility functions to accomplish the same objective (in this case, RA). FL can therefore be used to solve issues more effectively. Fig. 1 illustrates how FL can be used and provided if there is a common policy π^{-} that can be employed in any system.

Through her DRL technique, each system in FL learns common policies on its own. The cloud server then collects the common policies for the system and delivers the combined policies. As the experience of all systems is used, this accelerates the learning of common policies. By adding the new global model to the wireless network and using the cloud server's common policy, it is also possible to adapt to newly introduced systems.

4. DESIGN OF FEDERATED LEARNING-BASED RESOURCE ALLOCATION

In this section, a novel FEDRESOURCE has been proposed which efficiently performs RA in wireless networks. Federated learning is designed to minimize training loss while handling distributed neural network training across many devices with their local training data. The proposed FEDRESOURCE technique shares the DRL-based RA model between distributed systems and a center node to describe a policy for the FL framework. The weights of the DL models at the center node and the SN are indicated in the figure by the notations W^K_{s+1} and W_m, respectively. We refer to the deep learning models as policy models since they display the policies. A DRL approach is used by each distributed system to individually learn its local policy model. The overall block diagram for the proposed FEDRESOURCE model is given in Fig. 2.

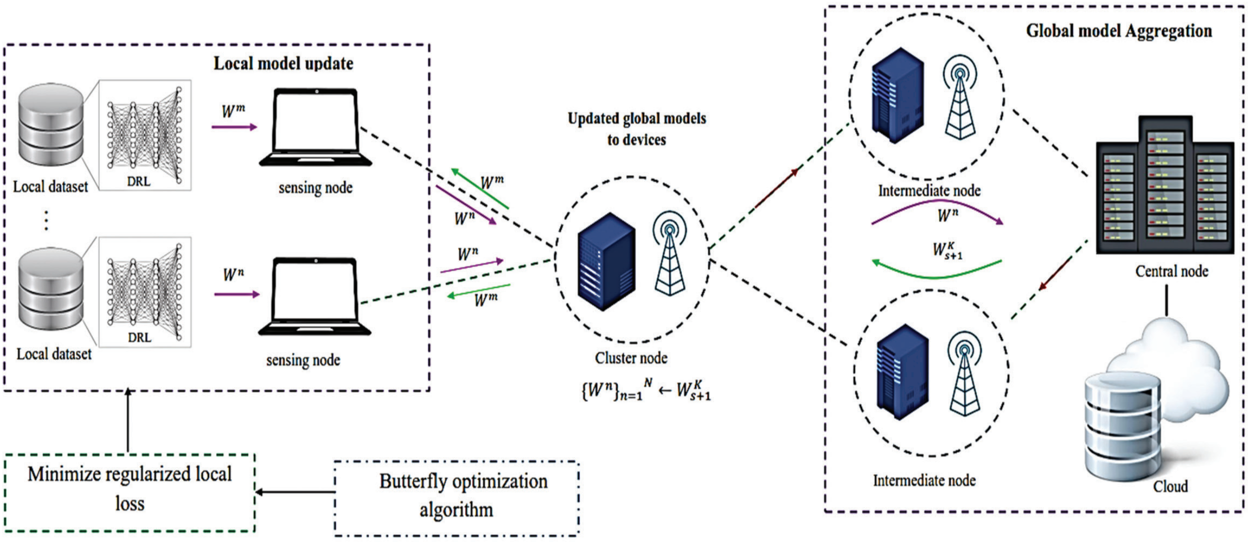

Fig. 2. Proposed FEDRESOURCE model

Similar to the traditional FL approach, the cloud server combines policy models learned from the system to update the central policy model. Then, as shown in Fig. 2, every system replaces its local policy model with the updated central policy model that was redistributed by the cloud server to the systems. To exploit all local experiences in the distributed system for learning, even if local experiences are not broadcast, FL can quicken the learning of the policy model by performing this iteratively. The central policy architecture at the center node also offers adaptation to recently arrived systems.

FEDRESOURCE employs an iterative method that calls for S_g global rounds for global model changes. The SN and cloud server interact in the following ways during each global round. SNs modify regional models: Each SN m first receives the feedback data from the server, to generate the local model w_m^r at a global round r. It then minimizes the surrogate function.

$$min_{w\in\mathbb{R}^k} R_m^r(w) := E_m(w) + \langle \eta\nabla\bar{E}^{r-1} -, w\rangle \qquad (3)$$

One of the fundamental principles of FEDRESOURCE is that a sensing device can roughly solve the problem to provide an approximation solution $w_m^{\ r}$ satisfying $\|\nabla R^r_{\ m}\ (w_m^{\ r})\|\leq\theta\|\nabla R^r_{\ m}\ (w^{r-1})\ )\|,\forall m$, which is parametrized by a local accuracy (0, 1) that is shared by all sensing devices. Here, $\theta = 0$ indicates that the local problem must be handled optimally, and $\theta = 1$ indicates that no progress has been made, for example, by setting $w_m^{\ r} = w^{\ r-1}$. FEDRESOURCE avoids employing proximal terms to restrict an extra controlling parameter (i.e.,), uses the global gradient estimate $\nabla E^{\ r-1}$ which the server can measure from the SN's data—instead of the exact but unrealistic $\nabla E(w^{r-1})$ and flexibly resolves local problems roughly by controlling will have $R_m^{\ r}(w)=E_m(w)+\eta\nabla E^{r-1}-\nabla E_m(w^{\ r-1})$, that contains both local gradient estimate E_m (w) and global gradient estimate weighted by a programmable parameter η. Later, we'll discover how influences FEDRESOURCE convergence. To attain the advantages of a) theoretical linear convergence and b) experimentally fast convergence which will be discussed in later sections, FEDRESOURCE needs more information than currently accepted standard approaches.

4.1. DYNAMIC LEARNING FOR RESOURCE ALLOCATION USING FL

In this section, the RA strategy for numerous systems with a center node has been done by maintaining the policy. For the DRL-based policy that has been frequently utilized, we here assume a typical DQN method, but any alternative DRL-based techniques can also be applied. The optimal action-value function A^* (st, ac), that denotes the maximum return which can be realized in state st with action ac, is approximated using a deep neural network (DNN) trained to perform the DQN method. DQN, as a result, is the name given to the DNN, and it is employed to construct policy by identifying the action that maximizes return for a given state. We use $\bar{\pi}(\bar{st}; w)$ to represent the common policy based on DNN.

We designate w_{cd} and w_s, respectively, as the weights of the DNN at the center node and system s. Each system s initializes its DNNs w_s and $w_s^{\ pr}$ as w_{cd} once the center node initializes its DNN w_{cd}. System s uses the DNN w_s and the translation functions $t_s^{\ st}$ and $t_s^{\ ac}$ to select the action $ac_s^{\ k}$ in timeslot k after observing its state $st_s^{\ k}$. The chosen action can be simply identified by the formula $ac_s^{\ k}=t_s^{\ ac}(\bar{\pi}(t_s^{\ st}\ (st_s^{\ k});\ w_s))$. The system provides services to the user following the selected action $(ac_s^{\ k})$, and it monitors the utility as $l_s^{\ k}=l(st_s^{\ k},\ ac_s^{\ k})$. When training the DNN, the experience of system s in timeslot k is described as $(st_s^{\ -k},\ ac_s^{\ -k},\ l_s^{\ k},\ st_s^{\ -k+1})$ and stored in the buffer. $st_s^{\ -k+1}= t_s^{\ st}\ (st_s^{\ k+1})$. The DNN w_s is trained to utilize the experiences using a variety of training methods, including experience replay and fixed target-Q. Each system s determines its local gradients for each FL interval by deducting its previous aggregated DNN, $w_s^{\ pr}$ from its present DNN. The cloud server then updates its DNN w_{cd} by combining the local gradients from all systems. The cloud server broadcasts w_{cd} to all systems after aggregation, and each system substitutes w_s and $w_s^{\ pr}$ with w_{cd}. Algorithm 1 provides a summary of the process.

Algorithm 1 FEDRESOURCE

1: The cloud server initializes w_{cd}

2: Each system s initializes w_s and $w_s^{\ pr}$ as w_{cd}

3: for $k \in\{0,1,\ldots\}$ $d0$

4: for each system s do ▹ DQN Algorithm

5: Observe $st_s^{\ k}$ and translate it as $\bar{st}_s^{\ k}\leftarrow t_s^{\ k}\ (st_s^{\ k})$

6: Choose $\bar{\bar{ac}}_s^{\ k}\leftarrow\bar{\pi}(4w_s)$ and translate it as $ac_s^{\ k}\leftarrow t_s^{\ st}\ (\bar{\bar{ac}}_s^{\ k})$

7: Do action $ac_s^{\ k}$ and observe $l_s^{\ k}$ and $st_s^{\ k+1}$

8: Translate $st_s^{\ k+1}$ as $\bar{st}_s^{\ k+1}\leftarrow t_s^{\ st}\ (st_s^{\ k+1})$

9: Store experience $(\bar{st}_s^{\ k},\ \bar{a}_s^{\ k},\ l_s^{\ k},\ \bar{st}_s^{\ k+1})$

10: Update w_s using its experiences by a DQN algorithm

11: end for

12: if mod$(t,\ T_{FL})$==0 then ▹ FL

13: All systems calculate their local gradients ∇E_s's from their previous DNN w_s To the current DNNs w_s's

14: The cloud server updates w_{Cd} by aggregating the local gradients from all system

15: All system replaces their DNNs w_s's and $W_s^{\ pr}$'s to w_{Cd}

16: endif

17: end for

4.2. MINIMIZE REGULARIZED LOCAL LOSS

Using the DRL algorithm may introduce regularized loss. To reduce the loss, the butterfly optimization algorithm (BOA) has been used in this paper. Butterfly reproductive behavior and its attraction to pheromones have been modeled by the BOA, a meta-heuristic algorithm with an emphasis on group and swarm behavior. To attract the opposite gender or to advertise where the best blooms are in the environment, butterflies release pheromones into their surroundings. Pheromones are not only employed by butterflies; other insects, such as ants, also release this chemical into the environment and use it to guide or lead other creatures. The more pheromones a butterfly produces, the more likely it is to attract additional butterflies, as butterflies like to travel to pheromone-rich environments. This algorithm makes the following assumptions:

- Every butterfly offers a different approach to the issue.
- The objective function decides which butterflies are eligible for pheromone release.
- The more pheromones present, the better the butterfly's ability to draw in additional butterflies and the better the problem's resolution.

The butterfly optimization process is considered to have a population member called a feature vector, each of whose components reflects the choice of the desired quality. Equation (4) demonstrates how the BOA algorithm produces a butterfly.

$$\mathrm{F} = \ll F_i^1, F_i^2 \dots \dots F_i^D \gg \quad (4)$$

F_i is a D-dimensional feature vector, and F_i^j denotes the j^{th} component of the I^{th} feature vector. These feature vectors F can be generated randomly and used as the initial BOA population, as shown by Eq (5).

$$\mathrm{F} = \ll F_1, F_2 \dots \dots F_n \gg \quad (5)$$

The initial population of feature vectors utilized for intrusion detection is denoted by the letter F, and the total number of feature vectors employed in the BOA is represented by the number n. The feature vector must be optimized by minimizing these two components of the objective function.

$$\text{fitness} = \alpha . mse + \beta \frac{||R||}{||N||} \quad (6)$$

In Eq. (6), Network intrusion detection uses a total of $||N||$ features, whereas $||R||$ is the number of features chosen to identify unauthorized traffic. The mean absolute error of approved network traffic is known as mse, and A number at random between 0 and 1 is "α", assuming β=1-α. In BOA, appropriate features can be selected to minimize the objective function.

$$F_i = F_i + (r^2 \times F^* - F_i) \times f_i \quad (7)$$

$$F_i = F_i + \left(r^2 \times F_j - F_i\right) \times f_i \quad (8)$$

In these equations, the feature vector, F_i can be updated by the vector F_j and F_k, as well as by optimized feature vectors like $F*$. F_i is the quantity of pheromone or attraction that a member of the BOA population produces, and r is a random number in the range of [0, 1].

$$F_i^j = \begin{cases} 0 & rand < |\frac{2}{\pi}\arctan\left(\frac{2}{\pi}F_j^i\right)| \\ 1 & rand < |\frac{2}{\pi}\arctan\left(\frac{2}{\pi}F_j^i\right)| \end{cases} \quad (9)$$

As a result, as indicated in Eq (9), extract the absolute or binary values using a transition function, such as a Gaussian or V-shaped function.

5. RESULT AND DISCUSSION

We create a special Python-based simulator just for the experiments, in which the following system is implemented. On a machine with 64 GB memory and an Intel Core i7-10700 processor, the simulation is run. We take into account various systems, each of which has a 5 MHz bandwidth. The noise spectral density is set to 106 dBm/Hz, the path loss exponent is set to 3.76, and a log-normal shadowing with a 6 dB standard deviation is taken into consideration. Each system's maximum transmission power will be 1 W. The Shannon capacity is used to determine the instantaneous data rate. we take into account a RA issue that seeks to satisfy average data requirements while minimizing average transmission power. The simulation setup for the proposed system is given in Table 1.

Table 1. Network simulation setup

Cellular network parameters	Values
Channel bandwidth	11.34
Noise power	2.76
The base station transmits power	2.03
Path loss between the base station and the user	3.76
Lognormal distribution shadow fading	6DB

5.1. PERFORMANCE ANALYSIS

The proposed method has been compared with existing techniques such as Heterogenous computation [18], Asynchronous FL framework [20], and scheduling policy [21] in terms of transmission power, convergence of algorithm, throughput, and cost.

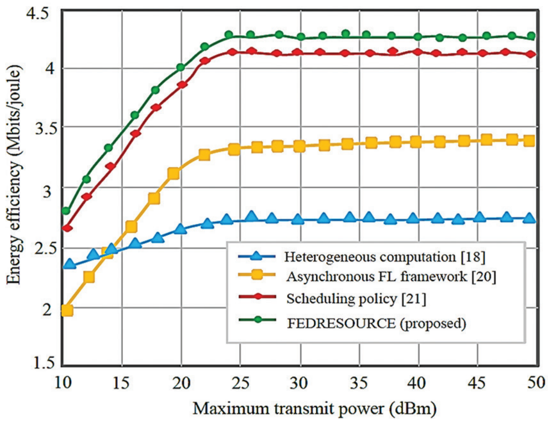

Fig. 3. Energy efficiency

When the maximum transmission power is altered, the energy efficiency is indicated in Fig. 3. This simulation demonstrates that as the maximum transmit power of the center node rises, all systems' energy efficiency initially rises and eventually stabilizes. This is because power efficiency does not increase monotonically with transmit power. The extra transmit power is not used since it is power-efficient if the maximum transmit power is 25 dBm or higher. Fig. 3 also indicates that the suggested FEDRESOURCE approach performs better than the Scheduling policy [21], Asynchronous FL framework [20], and Heterogeneous computation [18] schemes. For high maximum transmit power, FEDRESOURCE can increase up to 27%, 55%, and 68% energy efficiency when compared with the Scheduling policy [21], Asynchronous FL framework [20], and Heterogeneous computation [18] schemes respectively.

The convergence of the suggested FL algorithm and the fundamental method is shown in Figure 4. This figure demonstrates that the suggested FL algorithm, when compared to the scheduling policy framework, Asynchronous FL framework, and Heterogeneous computation schemes can increase discrimination accuracy by roughly 1.7%, 1.2%, and 0.78%. This is because the proposed FL algorithm updates the policy in the local model and is monitored by the DRL technique.

The model loss is reduced by butterfly optimization which also helps to increase the identification accuracy and the convergence rate. From the figure, it is clear that the proposed method achieves higher identification accuracy than existing techniques.

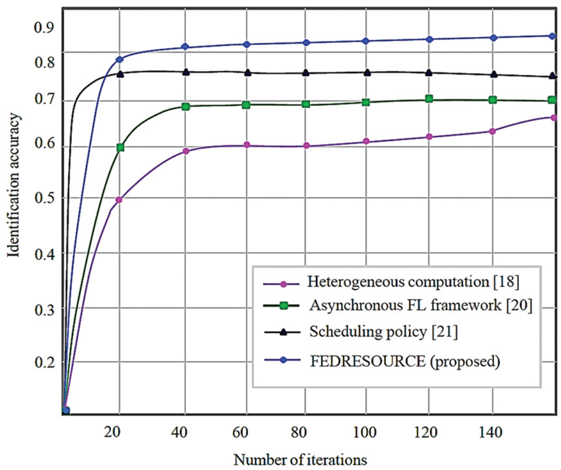

Fig.4. Convergence of FL algorithms

Fig. 5 displays the FL cost variation for the proposed approach with iteration for each number of devices. The term "iteration" refers to the execution of both the device allocation algorithm and the RA algorithm in a single process. The FL cost values show quick convergence (up to 6 iterations) for different device counts in the suggested design. With more SNs, the cost will display lower figures. The probability of being allocated to the closest SN grows as the number of sensor devices increases, which accounts for this pattern. Throughput is increased and FL costs are subsequently decreased by mapping the device to the nearest node.

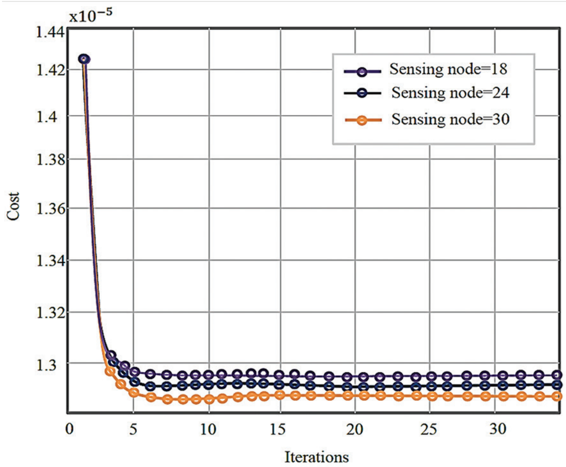

Fig. 5. Cost of FL

It can be seen from Fig. 6 that the suggested FEDRESOURCE framework can achieve almost the same throughput performance as the Scheduling policy algorithm. The reason is that the FEDRESOURCE algorithm considers the butterfly optimization algorithm, which can avoid loss when allocating resources for sensing devices. The throughput performance obtained by the Asynchronous FL algorithm is lower than that of the FEDRESOURCE algorithm. The uplink throughput performance of the Heterogenous computation algorithm is the lowest among the four algorithms.

6. CONCLUSIONS

In this paper, a novel FEDRESOURCE framework has been proposed which efficiently performs RA in modern wireless networks. We used experiments to show that the suggested FL framework may speed up RA policy learning and offer flexibility to new systems. Experiments were conducted using a Python-based simulator and detailed numerical results for the wireless RA sub-problems. The theoretical results of the novel FEDRESOURCE algorithm have been validated in terms of transmission power, convergence of algorithm, throughput, and cost. The proposed FEDRESOURCE technique achieves maximum transmit power up to 27%, 55%, and 68% energy efficiency when compared to Scheduling policy, Asynchronous FL framework, and Heterogeneous computation schemes respectively. Future research on this topic may include extending the suggested FL framework to address intercell interference.

Acknowledgment

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

7. REFERENCES

[1] V. D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless IoT networks", IEEE Internet of Things Journal, Vol. 8, No. 5, 2020, pp. 3394-3409.

[2] R. Liu, R.Y.N. Li, M. Di Renzo, L. Hanzo, "A Vision and An Evolutionary Framework for 6G: Scenarios", Capabilities and Enablers, arXiv:2305.13887, 2023.

[3] H. Yuan, M. Zhou, "Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems", IEEE Transactions on Automation Science and Engineering, Vol. 18, No. 3, 2020, pp. 1277-1287.

[4] H. Yuan, J. Bi, W. Tan, M. Zhou, B.H. Li, J. Li, "TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds", IEEE transactions on cybernetics, Vol. 47, No. 11, 2016, pp. 3658-3668.

[5] J. Huang, C. C. Xing, C. Wang, "Simultaneous wireless information and power transfer: Technologies, applications, and research challenges", IEEE

Communications Magazine, Vol. 55, No. 11, 2017, pp. 26-32.

[6] X. Kang, C. K. Ho, S. Sun, “Full-duplex wireless-powered communication network with energy causality”, IEEE Transactions on Wireless Communications, Vol. 14, No. 10, 2015, pp. 5539-5551.

[7] Z. Chu, F. Zhou, Z. Zhu, R.Q. Hu, P. Xiao, ”Wireless powered sensor networks for Internet of Things: Maximum throughput and optimal power allocation”, IEEE Internet of Things Journal, Vol. 5, No. 1, 2017, pp. 310-321.

[8] A. Mukherjee, P. Goswami, Z. Yan, L. Yang, J. J. Rodrigues, “ADAI and adaptive PSO-based resource allocation for wireless sensor networks”, IEEE Access, Vol. 7, 2019, pp. 131163-131171.

[9] S. M. Rajagopal, M. Supriya, R. Buyya, “FedSDM: Federated learning based smart decision-making module for ECG data in IoT integrated Edge-Fog-Cloud computing environments”, Internet of Things, Vol. 22, 2023, p. 100784.

[10] Z. Wang, M. Eisen, A. Ribeiro, “Learning decentralized wireless resource allocations with graph neural networks”, IEEE Transactions on Signal Processing, Vol. 70, 2022, pp. 1850-1863.

[11] X. Han, K. Xiao, R. Liu, X. Liu, G. C. Alexandropoulos, S. Jin, “Dynamic resource allocation schemes for eMBB and URLLC services in 5G wireless networks”, Intelligent and Converged Networks, Vol. 3, No. 2, 2022, pp. 145-160.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, S. Cui, “Performance optimization of federated learning over wireless networks”, Proceedings of the IEEE Global Communications Conference, Waikoloa, HI, USA, 9-13 December 2019, pp. 1-6.

[13] X. Liu, C. Xu, H. Yu, P. Zeng, “Multi-agent deep reinforcement learning for end—edge orchestrated resource allocation in industrial wireless networks”, Frontiers of Information Technology & Electronic Engineering, Vol. 23, No. 1, 2022, pp. 47-60.

[14] S. Sreethar, N. Nandhagopal, S. A. Karuppusamy, M. Dharmalingam, “A group teaching optimization algorithm for priority-based resource allocation in wireless networks”, Wireless Personal Communications, Vol. 123, 2022, pp. 1-24.

[15] J. Ren, X. Li, “Wireless network virtualization resource sharing based on dynamic resource allocation algorithm”, Wireless Communications and Mobile Computing, Vol. 2022, 2022.

[16] H. S. Lee, J. Y. Kim, J. W. Lee, “Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach”, IEEE Systems Journal, Vol. 14, No. 2, 2019, pp. 2589-2592.

[17] J. Feng, W. Zhang, Q. Pei, J. Wu, X. Lin, “Heterogeneous computation and resource allocation for wireless powered federated edge learning systems”, IEEE Transactions on Communications, Vol. 70, No. 5, 2022, pp. 3220-3233.

[18] C. Keçeci, M. Shaqfeh, F. Al-Qahtani, M. Ismail, E. Serpedin, “Clustered scheduling and communication pipelining for efficient resource management of wireless federated learning”, IEEE Internet of Things Journal, Vol. 10, No. 15, 2023.

[19] C. H. Hu, “Communication-Efficient Resource Allocation for Wireless Federated Learning Systems”, Linköping University Electronic Press 2023, PhD thesis.

[20] C. H. Hu, Z. Chen, E. G. Larsson, “Scheduling and aggregation design for asynchronous federated learning over wireless networks”, IEEE Journal on Selected Areas in Communications, Vol. 41, No. 4, 2023, pp. 874-886.

[21] A. Salh, R. Ngah, L. Audah, K. S. Kim, Q. Abdullah, Y. M. Al-Moliki, K. A. Aljaloud, H. N. Talib, “Energy-Efficient Federated Learning with Resource Allocation for Green IoT Edge Intelligence in B5G”, IEEE Access, Vol. 11, 2023, pp. 16353-16367.