International Journal of Swarm Intelligence Research
Volume 15 • Issue 1 • January-December 2024

Efficient Scheduling of Energy-Constrained Tasks in Internet of Things Edge Computing Networks

Shaolei Chen
State Grid Sichuan Electric Power Company, China

Hanyuan Tang
State Grid Panzhihua Electric Power Supply Company, China

Min Zhao
https://orcid.org/0009-0003-7260-2502
State Grid Sichuan Information Communication Company, China

Yu Chen
State Grid Yibin Electric Power Supply Company, China

Xin Yang
https://orcid.org/0009-0006-5227-0207
State Grid Yibin Electric Power Supply Company, China

Kejue Hu
State Grid Tianfuxinqu Electric Power Supply Company, China

ABSTRACT

We offer task scheduling algorithms that are economical in terms of energy consumption for edge computing networks that are supported by the Internet of Things (IoT). The challenges of spectrum utilization and energy-efficient work scheduling that lead to novel design are not addressed in this study, despite the fact that it provides encouraging results for task offloading. There is a possibility that the larger homogeneous fog computing architecture will include all homogeneous nodes, in addition to additional spectrum for node-to-node and device-to-device communications and work scheduling. We create a fog computing architecture that is efficient in terms of energy consumption for edge computing networks that are supported by the Internet of Things. By utilizing this approach, user-device nodes are able to collaborate while simultaneously reaping the benefits of diverse computing and network resources. In addition to this, we provide a solution to the problem of task scheduling that maximizes energy efficiency across all of the help nodes.

KEYWORDS

Internet of Things, Fog Computing, Scheduling Algorithm, Energy Efficiency, CPU Power, Offloading, and Helper Modes

EFFICIENT SCHEDULING OF ENERGY-CONSTRAINED TASKS IN INTERNET OF THINGS EDGE COMPUTING NETWORKS

Edge computing, a novel paradigm, leverages network-edge servers to perform user services (Xue et al., 2023; Bhatia, & Sood, 2023). Time-critical cloud or Internet of Things (IoT) applications may avoid network latency and backbone network traffic by deploying an edge server instead of a data center server. 5G's lower latency and higher bandwidth will make edge computing more significant. Mobile networks link additional devices to the internet.

Tens of billions of resource-limited devices like smartphones can connect to networks due to rising IoT adoption (Yang et al., 2024). Thus, fog node mobile devices are growing more popular and we cannot live without them. Large-scale networks, multitasking applications, and faster networking

DOI: 10.4018/IJSIR.350221

progress together. Smartphone apps include movie streaming, augmented reality, online gaming, and intelligent driving (Hu et al., 2024). Popular applications. These revolutionary applications need plenty of computer and communication resources for real-time processing and high data exchange rates owing to decreased latency. Physical restrictions limit mobile device resources. Designing these new apps for mobile devices is complex (Zhang et al., 2023). Task scheduling and execution have not been addressed in energy-efficient communication situations (Li et al., 2023; Wang et al., 2023). Scheduling and execution are crucial to fog computing.

Academics and businesspeople have explored task offloading for years. Mobile cloud computing offloads work since faraway cloud servers have plenty of processing and storage capacity. Mobile cloud computing is promising. Mobile cloud computing research has spanned a decade (Zhang et al., 2024; Cheng et al., 2023; Guo et al., 2018). These tests wirelessly unload computation-intensive activities from many mobile users. Although many cloud services have been commercialized, poor wireless connections, such as deep fading, sometimes cause packet loss and unacceptable wide area network latency between mobile devices and clouds. Despite the widespread use of cloud services, this remains true. Fog computing, a network-based task computation approach, is extensively virtualized (Sahni et al., 2018; Reiss-Mirzaei et al., 2023; Asgarian et al., 2024). Fog computing may provide a lot of computational, storage, and networking services (Wen et al., 2023). Fog networks' flexible pooling of processing and communication resources may increase mobile devices' energy efficiency. Given the rapid growth of 5G wireless communication technologies, homogeneous fog networks may deploy many cooperating smart devices (Chen et al., 2023; Ghosh & De, 2023; Singh et al., 2024).

Data collection and transmission are handled by billions of devices in IoT (Dasari & Kaluri, 2024; Devarajan et al., 2024). Typically powered by batteries, these gadgets need to work well so they may enjoy their lives to the fullest. Devices have limited battery power and must perform energy-constrained tasks, such as gathering data from sensors or processing information, with the power they have (Begum et al., 2023). Thus, the goal is to find the best order of operations that minimizes energy consumption without sacrificing the completion of critical tasks. The “edge” of the network is where data is processed using edge computing, as opposed to sending it all the way to a central cloud server. That's different from the way cloud computing has always been done. Because of its potential efficiency, this method may be useful for tasks that need low latency or fast response times. It is possible that certain tasks are more critical or time-sensitive than others. Even if there is a little increase in energy use, the scheduler may still prioritize and complete certain operations first. Reducing the quantity of electricity supplied by batteries is possible if there are devices that can harvest energy from their surroundings, such solar panels. The scheduler may then prioritize tasks when this energy is available. Rather than completing work on the device itself, it may be more effective to transfer it to a more capable device that has access to mains power or a larger battery, such a nearby server or computer. The use of these scheduling algorithms is being pursued by researchers in the hopes of enhancing the network's overall efficiency and prolonging the battery life of IoT devices. To ensure these devices work well and can be used for a long time in many applications including smart homes, industrial automation, and environmental monitoring, this is crucial.

We develop an energy-efficient fog computing architecture for IoT backed edge computing networks. By using this method, nodes of user devices may operate together and make use of various processing and network resources at the same time. Furthermore, we provide a resolution to the issue of work scheduling that optimizes energy efficiency across all of the assistance nodes. This technique efficiently establishes unloading time windows for task scheduling by using main network spectrum sharing.

RELATED WORK

According to previous research, designs may be divided into two groups for offloading jobs to support nodes (Rani & Srivastava, 2024; Pallewatta et al., 2023). A heterogeneous fog computing architecture deploys tiny clouds in base stations within local area networks via network operators and cloud providers. Base stations may then provide computational resources and arrange task offloading for mobile users. This allows simultaneous scheduling of processing and offloading resources like time and bandwidth. Liu et al. (2023) and Wang et al. (2023) optimized all users' offloading selections and computer and communication resource allocation to reduce energy, computation, and latency for all users. Offloading choices for all users' tasks are optimized together. Despite the positives, we should emphasize the downsides. Local cellular access task scheduling may initially increase cellular communications demand. In addition, the local cloud has limited resources, hence this architecture will rapidly meet service scaling difficulties. The exponential growth of mobile devices makes this a challenge. This architecture also fails urgent remote clients due to poor channel conditions.

Homogeneous node-centric cooperation-based architecture is the second design. In this design, task nodes may help offload work to surrounding homogenous helper nodes with underutilized resources. There are enough homogenous nodes in daily life, such as cellphones and IoT devices, to implement this design. Offloading tasks to other nodes to facilitate collaborative task execution for various services may also use the multiplexing benefit (Cheng et al., 2023; Zhang et al., 2023). Mobile task nodes have several options to effectively unload their jobs. This architecture style has inspired many works. Xiao et al. (2023) and Song et al. (2023) introduced mobile users to task offloading frameworks. These frameworks exploit nearby devices' processing power to offload a mobile user's task. A study by Ogundoyin and Kamil (2023) proposed a hybrid task offloading system that used device-to-device (D2D) communication and cloud offloading. Wang et al. (2023) formulated an optimization problem to decrease the time-average energy usage of all D2D network users' job executions. Albaseer et al. (2024) had this formulation. The approach for offloading user data files improves cellular transmission (Ahmed et al., 2023). Local users would get traffic. 5G-enabled IoT services leverage this framework. The node-centric, time- and spectrum-based fog computing architecture must be energy-efficient to succeed.

Wireless D2D networks will always underlie the main cellular network. However, spectrum scarcity drove opportunistic spectrum access (Hazra et al., 2023). Under their worst channel conditions, though, cooperative nodes may have long offloading times, resulting in inefficient work offloading. Our energy-efficient job scheduling approach is for D2D fog networks, unlike other frameworks. This approach supports energy-efficient task transfer and computation. Table 1 presents a summary of our literature review.

Task Offloading

Numerous research has employed cloud computing to solve mobile device performance and resource issues (Sun et al., 2023). These studies offload computationally hard activities from mobile devices to the resource-rich cloud and then complete them in the cloud. These studies focus on cloud loading to improve mobile device performance rather than energy efficiency. Offloading improves mobile device energy efficiency (Chen et al., 2023). Hua et al. (2023) proposed a hybrid method that balances energy and performance. Edge computing (Wang et al., 2023) solves the issues. Offloading compute to edge servers avoids slow backbone networks. Edge computing research has focused on improving energy-constrained mobile and IoT device performance and energy usage.

Task Scheduling in Edge Computing

The relevance of scheduling algorithms in edge servers has expanded in tandem with the growing prevalence of the IoT and mobile devices in the edge computing community. IoT devices and edge servers, respectively, are responsible for performing job scheduling and energy allocation in order to maximize energy efficiency. A joint allocation mechanism for scheduling tasks among multiple available ones is proposed in Zhang et al. (2023), along with the imposition of heterogeneous latency criteria.

Table 1. Literature review

Study	Scheduling Approach	Focus	Advantages	Disadvantages
Tian et al., 2023	Decisive task scheduling for energy conservation (DTS-EC)	Prioritizes tasks based on energy level and data rate	- Reduces energy consumption- Improves data availability	- Limited to edge-assisted data dissemination scenarios
Zhang et al., 2020	Multiple Heuristics (METAR)	Considers energy harvesting, brown energy usage, and green energy transfer	- Minimizes brown energy consumption- Utilizes green energy sources	- Assumes a single VM in the system (may not reflect reality)
Llorens-Carrodeguas et al., 2021	Machine learning (Deep reinforcement learning)	Selects optimal processing location (cloud vs. edge)	- Reduces energy consumption- Improves resource utilization	- Requires significant training data- May be computationally expensive
Al-Masri et al., 2023	Coordinated task scheduling among edge devices	Improves energy efficiency through collaborative execution	- Reduces individual device workload- Enhances overall system performance	- Requires reliable communication and coordination protocols
Yu et al., 2023	Priority-based scheduling with deadline constraints	Ensures timely execution of critical tasks	- Minimizes energy consumption while meeting deadlines	- May not be suitable for non-real-time applications
Sun et al., 2020	Multi-objective optimization considering energy, execution time, and reliability	Balances competing objectives for optimal scheduling	- Achieves high task success rates while minimizing energy usage	- Increased computational complexity compared to simpler approaches
Xie et al, 2022	Dynamic voltage and frequency scaling (DVFS) with task migration	Optimizes energy consumption based on workload and device capabilities	- Reduces energy usage for resource-intensive tasks	- May require additional hardware support for DVFS
Raju & Mothku, 2023	Leverages harvested energy for task execution	Reduces reliance on grid power and improves sustainability	- Exploits renewable energy sources- Extends device lifetime	- Requires accurate prediction of energy harvesting rates
Hosseinioun et al., 2020	Evolutionary algorithms for task scheduling and resource allocation	Provides flexibility and adaptability to changing network conditions	- Efficiently handles complex optimization problems	- May require fine-tuning of algorithm parameters
Azizi et al., 2022	Considers both energy efficiency and load balancing for task scheduling	Optimizes resource utilization while minimizing energy consumption	- Improves overall system performance and fairness	- May require additional communication overhead for load balancing information exchange

In addition to these studies, there have been a number of research initiatives looking into ways to reduce the time it takes to carry out an action. The queuing theory is used by the scheduler (Li et al., 2023) in order to reduce the amount of time needed to complete bursty operations. Data partitioning methods are created for efficient distributed data processing with heterogeneous IoT devices. This results in a reduction in the amount of time required for the execution of data processing. When IoT devices are initialized and have local tasks loaded to the servers which offered a task scheduling approach with the goal of reducing the amount of time required for the total execution of the programmed. This was accomplished by taking into account cloud servers in addition to edge servers. In particular, it provides feedback mechanisms for determining the state of edge servers in order to address the issue of resource contention, which occurs in hotspot edge servers. When transmitting tasks from IoT devices to edge servers takes into account both the location of the data as well as the congestion in the networks. When networks become overloaded, a process that searches for alternative flow paths is activated based on the previous records of flow conditions.

Considerable research has developed innovative approaches to the scheduling and allocation of resources with the goals of reducing the amount of time needed for execution and reducing the amount of energy required (Khan et al., 2021). When processing data using IoT devices and edge servers, several studies attempted to minimize the amount of time required for the execution of applications while simultaneously maximizing the amount of compensation received by device or server suppliers. In order to accomplish this goal, game-theoretic techniques (Tsemogne et al., 2021) have been developed. This is because determining the optimal solution that satisfies both goals is an NP-hard task. An effective wholesale and buyback algorithm (Zhang et al., 2020) with the intention of maximizing the profit that edge server suppliers make.

PROPOSED METHOD

System Model

Fog computing in a homogenous fog network. One single-antenna task node S offloads its computation duty to a collection of K matrix assistance nodes ($K = \{1,2,...,k\}$). These help nodes, which do not have computing tasks in their task queues, are needed to finish computation work from the same task node. Distance, severe fading, and spectrum resource limitations prevent idle or adjacent computing nodes from calculating the job. The offloading assistance nodes, offloaded work size, and time resources for each task scheduling pair are all determined by the central controller, who bases their decisions on maximal energy efficiency. Each time slot involves work offloading (local computing), task computing from surrounding assistance nodes, and uploading computation results to the task node. For most applications, the calculated output is substantially less than the computed input data. Next, we disregard the energy cost that neighbor assistance nodes would incur to send compute outputs to the task node.

We follow earlier studies like Rudenko et al. (1998) and Sakr et al. (2015). Downloading is faster than unloading since computation results are smaller. We will assume the second phase is short compared to the first and not consider it when scheduling resources.

Protocol for Medium-Scale Access

Opportunistic spectrum access is our spectrum paradigm. Task-scheduling broadcasts are planned to utilize unused spectrum from busy nodes. The task node can sense its environment and cognitively modify its spectrum properties to get opportunistic spectrum access. The node's spectrum occupancy may be described as a continuous-time Markov chain with idle and busy states. The chain's idle state allows this. This research assumes the task-scheduling system and busy nodes communicate data continuously. Therefore, the spectrum occupied periods from busy nodes are not reliant on one another and are spread in an exponential fashion with the aggregated parameter $s_a\left(i,j\right)$ for the available state and $s_u\left(i,j\right)$ due to the fact that the created task-scheduling connection is now in an inaccessible state i utilising j^{th} resource block (i $\{1,2,...,k\}$). These distributions are based on the independent and exponentially dispersed spectrum occupied times from busy nodes. This model gives the steady distribution of the spectrum's availability $\left(p_a\left(i,j\right)\right)$ and unavailable $\left(p_u\left(i,j\right)\right)$ probability, as in formulas (1) and (2), respectively (Zhao et al., 2008).

$$p_a(i,j) = \frac{s_a(i,j)}{s_a(i,j) + s_u(i,j)} \tag{1}$$

$$p_u\left(i,j\right) = \frac{s_u(i,j)}{s_a(i,j) + s_u\left(i,j\right)} \tag{2}$$

Computation of Tasks and Offloading of Work

If a mobile task node lacks processing power, surrounding homogenous assistance nodes may take over. These nodes create task-scheduling pairings. In this case, the task transmission time slot allotment for every task-scheduling combination$\tau_i \forall i \in K$. Mobile task node S must compute B_S bit

Figure 1. Block diagram for the computation of tasks and offloading of work

input data in time. From this data, l-bit is offloaded to the closest assistance nodes, and $\left(B_S - l\right)$ -bit may be calculated locally by its central processing unit (CPU).

A cloud server, IoT devices, and edge devices with increased computing capability are the three categories that the entities discriminate between. The production of tasks, local computation, offloading decisions, and distant execution are all highlighted by processes within this section. The data flow monitors the movement of tasks, outcomes, and data within the system. Considerations such as these highlight the importance of resource limits, communication costs, and latency requirements when making judgments about offloading. A detailed illustration of the elements that influence the offloading decision is provided by decision making. Figure 1 explains the block diagram for the computation of tasks and offloading of work.

Problem Formulation

In this section, we formalize task assignment and scheduling issues in IoT-assisted edge communications. The task assignment problem involves distributing tasks among devices to minimize work completion time. The task scheduling problem is to discover the best device-specific task execution schedule to optimize job completion. Find each device's optimal task execution schedule. When determining the most effective way to divide up the work, there are a few requirements that absolutely must be met. First, a job will only be given to a single device or server to complete. Formally,

$$\tau_i = \sum_{k \in K} \lambda_{i,k} \forall i \in K, \tag{3}$$

where $\lambda_{i,k}$ is a binary variable that shows whether or not a task τ_i is assigned to device k; more specifically, it is 1 when τ_i is assigned to k and 0 otherwise.

Second, each task has one or more input devices. A mathematical function that employs its parameters is a task that has its input data in every device. The job requires no input data. As a result, the following restriction applies:

$$\tau_i = \sum_{k \in K} \lambda_{i,k} \geq 1 \tag{4}$$

Third, each individual IoT device may have its own minimum energy level need, which is defined by the amount of energy that is used when it is carrying out the duties $\left(E_d\right)$ that are exclusive to it. As a result, the amount of energy that an IoT device has should never, even after it has finished all of the tasks that were loaded onto it, drop below the minimal amount of energy that the device has.

$$E_d^i - \sum_{\forall \tau_i} \lambda_{i,k} E_d^i \geq E_{min} \tag{5}$$

$$E_d^i = P_{CPU}^i \times \frac{C_i}{f_i} \tag{6}$$

$$P_{CPU} = \text{CPU power consumption in a device } i \tag{7}$$

$$C_i = CPU\ cycles\ consumed\ in\ a\ second\ task\ i \tag{8}$$

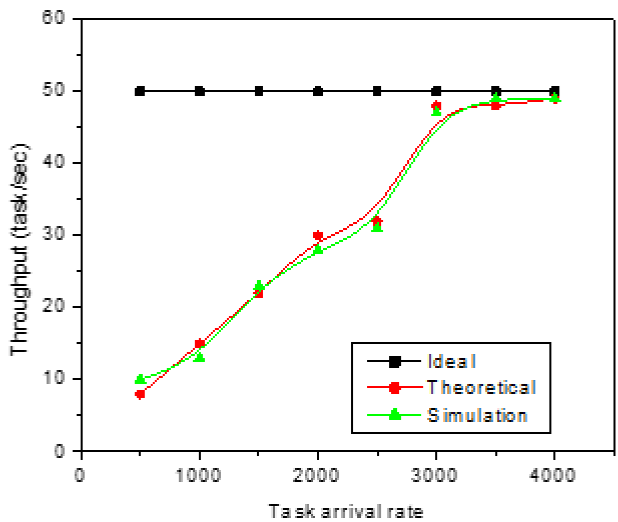
$$f_i = CPU\ frequency\ of\ i \tag{9}$$

The issue of improperly assigning tasks has now been ruled out.

It is important to do some research to determine whether or not the problems of job assignment and scheduling are orthogonal to one another before offering any heuristic solutions to these issues. To put it another way, if you have a collection of tasks, is it always feasible to find a distribution for those tasks that not only reduces the amount of time it takes to complete all of the tasks as a whole, but also ensures that a greater percentage of tasks will be able to reach their due dates compared to alternative distributions for those tasks. Generally, there is no solution that optimizes both the total amount of time needed to complete activities and the number of deadlines that are fulfilled, which compels us to devise a heuristic method for assigning tasks and developing a schedule that meets both objectives to a level that is acceptable.

Task scheduling also considers work duration. We assume each job has been performed and each edge server knows how many CPU cycles each task requires. Devices' ever-running non-user-requested processes' average CPU cycles per second. Operating system kernel, system administration, and IoT device local tasks are examples of these always-running processes. Unlike edge and cloud servers, larger IoT devices frequently run on batteries or environmental energy. In IoT-assisted edge computing, their power consumption from overloaded activities is crucial. We must appropriately

Figure 2. Throughput vs. task arrival rate for busy spectrum



analyse and integrate the "enhanced" energy consumption of IoT devices' loaded workloads in task assignment to regulate it.

RESULTS AND DISCUSSION

The findings of the simulation will make it possible for us to provide a guarantee that the edge computing architecture supported by the IoT is capable of offloading tasks in an effective manner while consuming a power consumption that is insignificant.

When edge computing and the IoT are brought together, the greatest throughput that is practically possible is achieved. To determine the task throughput per second, we first timed how long it took to finish all edge tasks (during which independently created local tasks were also carried out), and then we divided that amount by the number of tasks in total. This provided us with a general concept of the number of items that could be completed in a certain length of time. Because of this, we were able to calculate the number of jobs that could be completed in a certain length of time and, in turn, we were able to compute the number of distinct activities that could be completed in a single second. Figure 2 illustrates the throughput of each sample as a function of the growing rate of tasks that are being received. This is referred to as the load on the system or the stress on the system.

By summing up the clock rates of the CPUs located on each node, it is possible to get an approximation of the theoretical maximum processing capabilities that are possessed by our experimental system. This provides us with a ballpark figure for the amount of data that can be processed simultaneously by our system. A plot of throughput vs the task arrival rate is shown in Figure 3, which shows the spectrum to which we now have access.

It makes no difference what method is utilized to schedule work since the output of those jobs will always be maximized so long as there are jobs that are waiting to be finished. In the meantime, the method in which activities are delegated is likely to have an effect, at least in part, on the amount of time required to carry out the responsibilities associated with activities. This is due to the fact that an unequal job distribution may force resources that are not being fully used to wait for work to be done, which in turn increases the amount of time that is necessary to finish the task in its entirety.

A scatter plot depicting the relationship between the number of help nodes in a network and its overall energy efficiency is shown in Figure 4.

Figure 3. Throughput vs. task arrival rate for available spectrum

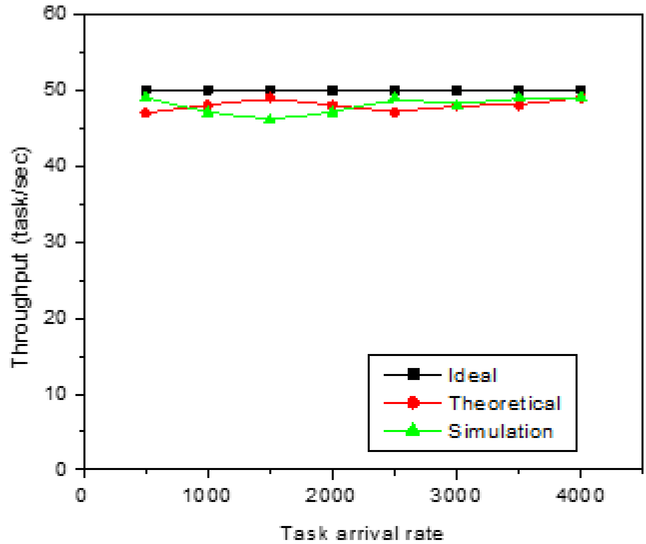
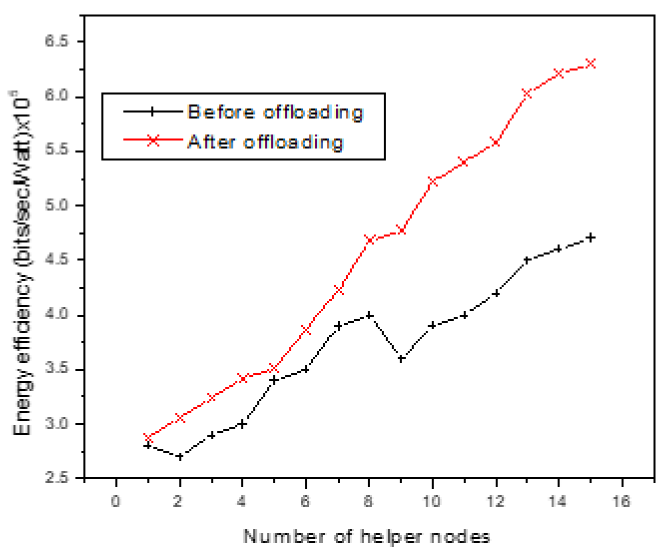
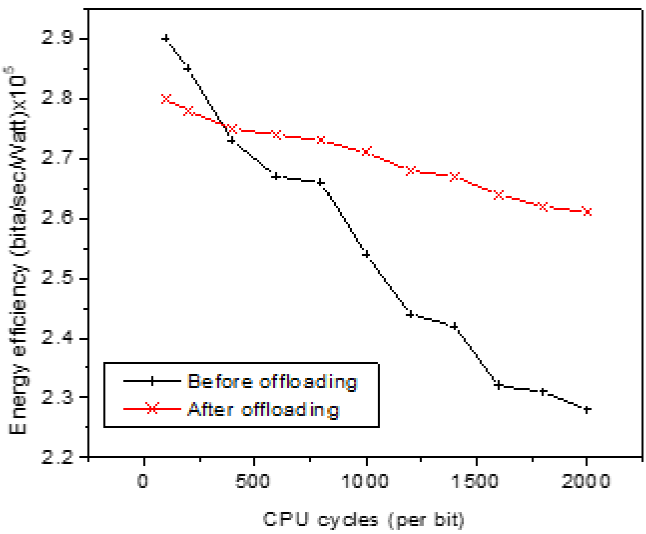


Figure 4. Number of helper nodes vs. energy efficiency



In the absence of any further context, the relevant simulation parameters will be given in the following format. The offloading task node is located in close proximity to 10 help nodes, all of which are approximately at the same physical place in relation to one another. It is possible that these nodes take up a large chunk of the entire capacity of the network. This probe takes into account both the efficiency of unloading, as seen in Figure 4, as well as the efficiency of using energy. We can see a graphical depiction of the link between the total number of task scheduling support nodes and the amount of energy efficiency that is obtained in the following figure. The network's support nodes have grown, improving system efficiency. Task offloading benefits from the correlation between

Figure 5. CPU cycles (per bit) vs. energy efficiency



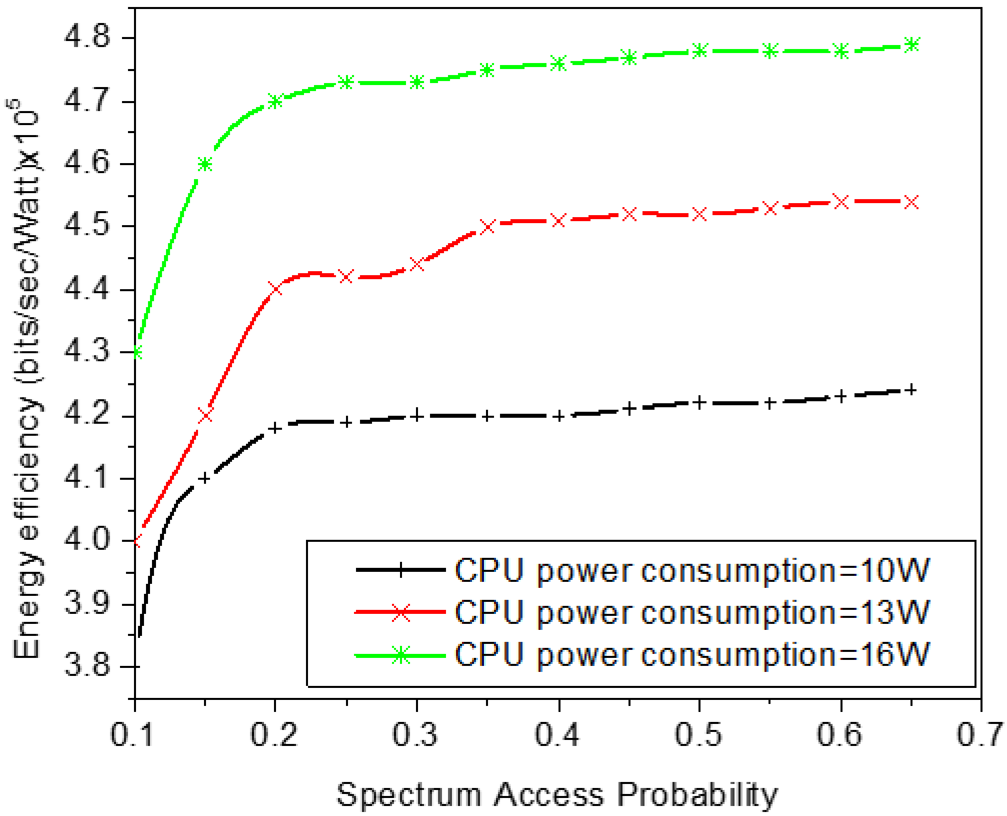
helper node availability and work that can be offloaded. The chart also illustrates that when there are less than five help nodes, energy-efficient time slot allocation is less successful than equal time slot distribution. The aim requires five assistance nodes, proving this. Even with few support nodes, offloading requires more power than computation. This is due to the fact that offloading consumes a far higher amount of energy than computation does and the fact that the two processes cannot operate in parallel at the same time. Despite this, the energy efficiency of the equal time slot allocation is more than that of the energy efficiency of the energy-efficient time slot allocation, and there are more than eight helper nodes involved. This means that the method described in Figure 4 for assigning time slots based on energy efficiency has the same impact as the strategy shown in Figure 4 for allocating time slots based on equal distribution over a large area of many different help nodes. Figure 4 is about allocation of time slots based on energy efficiency and equitable distribution across a large area of many distinct assistance nodes.

Figure 5 illustrates the energy efficiency of the system as measured by the number of CPU cycles needed to compute one bit for each assist node *i*. The value of this is shown as a percentage of the total number of assistance nodes in the system. This is shown in a manner that is proportional to the total number of computations that were performed.

In Figure 5, a comparison is made between the quantity of energy that is utilized and the number of CPU cycles that are used (per bit). Compared to the strategy of equitably distributing time slots, the approach of assigning time slots in an energy-efficient manner has the potential to achieve a higher level of efficiency with the use of time slots. This is something that we can observe for ourselves. Even if the higher compute energy demand does degrade the energy efficiency, the optimal offloading technique may guarantee that the energy efficiency of a homogenous fog computing system is maintained. This may be the case if the energy efficiency is maintained because, while using the offloading strategy, a lesser amount of work is necessary to maintain the same level of energy efficiency as before. This is because the most efficient method of offloading requires a much lower amount of power than the other choices that are available. This is the case regardless of whether or not an increase in the energy requirements of computations leads to a decline in the energy efficiency of the system.

Figure 6 compares the probability of gaining access to the spectrum with the amount of energy efficiently used. In addition to this, we complete an in-depth numerical analysis of the number of CPU

Figure 6. Spectrum access probability vs. energy efficiency



cycles that are necessary to calculate each bit. This is done in order to make the comparison of the different offloading mechanisms as easy and fundamental as is practically possible. The effectiveness of energy consumption in relation to the number of assistance nodes making use of various spectrum access probabilities on the premise that 10 watts, 13 watts, and 16 watts of CPU power are being used. For the purposes of this study, it was assumed that the power consumption of the CPU would be somewhere between 10 and 16 watts.

The results of our simulation demonstrate how critical it is to consider the use of compute power while searching for an offloading strategy that is efficient about energy use in the context of a standardized IoT-enhanced edge computing environment. The findings of our simulations indicate that it is essential to take into account the amount of computing energy that is needed. To put this into perspective, in order to discover an effective method of offloading in terms of energy consumption, it is necessary to consider the energy that is used by computers. As a business, we have arrived at this conclusion via group deliberation.

We may say that effective energy and the spectrum bandwidth are connected because there is a correlation between the two, and this connection shows that energy-efficiency increases in tandem with an increase in the total amount of accessible spectrum bandwidth. We can say that effective energy and the spectrum bandwidth are linked because there is a correlation between the two. To restate, the proportion of the spectrum's bandwidth to the effective energy describes the relationship between the two. To be more explicit, the correlation demonstrates that an increase in the total amount of spectrum bandwidth that is accessible is followed by higher energy efficiency in the system. This may be explained by the fact that the connection functions as a positive feedback loop in the system. This is mostly because of the typically greater relevance that spectrum allocation has on offloading rate. This, in turn, may be related to the observation that the task node is more likely to offload more work when the spectrum bandwidth is enhanced. This is something that has been noticed. This sequence of causes and effects may be responsible for the fact that spectrum allocation has, in most cases, a more significant influence on the offloading rate. This is where the issue really started in the first place.

The deadline for the locally operating operations that are now taking place on the IoT device is set to one second, and the CPU cycle has been set to 500 Mcycles. The settings menu of the IoT device allows for adjustments to be made to both of these variables. In addition, the standard for the management of all global processes was established as a CPU cycle that lasts one second. The proposed method is used by the task distributor to determine where in the cloud, on the edge, or on IoT devices each individual job should be dispersed. Alternatively, the task distributor may employ both algorithms. A task executor container has its very own process for the communication module, in addition to the task executor process and the process for the communication module. In addition to that, the processes that are necessary for the task executor and the communication module are included here.

CONCLUSIONS

Our study is targeted at creating a method for scheduling jobs in edge computing networks that makes effective use of energy. Although this research demonstrates promising outcomes for task offloading, it has not yet dealt with the issue of combining spectrum utilization with energy-efficient job scheduling to create a revolutionary architecture. This is essential work that must be completed, even if there is a probability that the combined factors may provide a novel structure. This global homogeneous fog computing architecture manages all homogeneous nodes, so the excess spectrum may be used for scheduling and device communication. The IoT enables a new fog computing paradigm in edge computing networks. User-device-node interaction allows nodes to have various processing capacity and network quality. In this approach, user devices work together as nodes. We also propose a problem formulation that prioritizes support nodes' energy efficiency while fulfilling their duties. A solution that satisfied all criteria did this. Sharing spectrum with the core network may efficiently offload task scheduling time periods. The simulation results show that the suggested scheduling approach enhances energy efficiency over current scheduling methods. This is because the method considers the energy consumption of calculations under a wide range of system parameters. Therefore, the algorithm gets excellent results in terms of energy efficiency.

In many cases, existing models presume that workloads are constant. It is possible to enhance efficiency via research on dynamically adjusting schedules based on the arrival of tasks in real time and accurate predictions of workload. If one takes into account the unpredictability of energy harvesting rates, fluctuations in network bandwidth, and the possibility of device failures, one may develop scheduling algorithms that are more durable. It is possible that further investigation into deep reinforcement learning, genetic algorithms, and other artificial intelligence approaches might result in scheduling choices that are more intelligent and adaptable. When it comes to improving scalability and fault tolerance, investigating decentralized scheduling options that include edge devices working together to plan jobs might be beneficial.

CONFLICTS OF INTEREST

The authors of this publication declare that there are no competing interests.

FUNDING STATEMENT

This paper was supported by the Science and Technology Project of State Grid Sichuan Electric Power Company; the fund project name is Research on Communication, Protection, and Stability Control Coupling Simulation Analysis Technology Based on Regulation Cloud (521911230001).

PROCESS DATES

Received: April 30, 2024, Revision: July 2, 2024, Accepted: July 3, 2024

CORRESPONDING AUTHOR

Correspondence should be addressed to: Xin Yang, yangx.012972@111.com

REFERENCES

Ahmed, M., Mirza, M. A., Raza, S., Ahmad, H., Xu, F., Lin, Q., & Han, Z. (2023). Vehicular communication network enabled CAV data offloading: A review. *IEEE Transactions on Intelligent Transportation Systems*, *24*(8), 7869–7897. 10.1109/TITS.2023.3263643

Al-Masri, E., Souri, A., Mohammed, H., Yang, W., Olmstead, J., & Kotevska, O. (2023). Energy-efficient cooperative resource allocation and task scheduling for Internet of things environments. *Internet of Things : Engineering Cyber Physical Human Systems*, *23*, 100832. Advance online publication. 10.1016/j.iot.2023.100832

Albaseer, A., Abdi, N., Abdallah, M., Qaraqe, M., & Al-Kuwari, S. (2024). FedPot: A quality-aware collaborative and incentivized honeypot-based detector for smart grid networks. *IEEE Transactions on Network and Service Management* (Early Access). 10.1109/TNSM.2024.3387710

Asgarian, M., Jamshidi, K., & Bohlooli, A. (2024). An efficient approximation algorithm for service function chaining placement in edge-cloud computing industrial Internet of things. *IEEE Internet of Things Journal*, *11*(7), 12815–12822. 10.1109/JIOT.2023.3338516

Azizi, S., Shojafar, M., Abawajy, J., & Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. *Journal of Network and Computer Applications*, *201*, 103333. Advance online publication. 10.1016/j.jnca.2022.103333

Begum, M. B., Deepa, N., Uddin, M., Kaluri, R., Abdelhaq, M., & Alsaqour, R. (2023). An efficient and secure compression technique for data protection using burrows-wheeler transform algorithm. *Heliyon*, *9*(6), E17602. Advance online publication. 10.1016/j.heliyon.2023.e1760237457815

Bhatia, M., & Sood, S. (2024). Quantum computing-inspired optimal power allocation mechanism in edge computing environment. *IEEE Internet of Things Journal*, *11*(10), 17878–17885. 10.1109/JIOT.2024.3358900

Chen, J., Deng, Q., & Yang, X. (2023). Non-cooperative game algorithms for computation offloading in mobile edge computing environments. *Journal of Parallel and Distributed Computing*, *172*, 18–31. 10.1016/j.jpdc.2022.10.004

Chen, Y., Chang, Z., Min, G., Mao, S., & Hämäläinen, T. (2023). Joint optimization of sensing and computation for status update in mobile edge computing systems. *IEEE Transactions on Wireless Communications*, *22*(11), 8230–8243. 10.1109/TWC.2023.3261338

Cheng, K., Fang, X., & Wang, X. (2023). Energy efficient edge computing and data compression collaboration scheme for UAV-assisted network. *IEEE Transactions on Vehicular Technology*, *72*(12), 16395–16408. 10.1109/TVT.2023.3289962

Cheng, P., Chen, Y., Ding, M., Chen, Z., Liu, S., & Chen, Y. P. P. (2023). Deep reinforcement learning for online resource allocation in IoT networks: Technology, development, and future challenges. *IEEE Communications Magazine*, *61*(6), 111–117. 10.1109/MCOM.001.2200526

Dasari, S., & Kaluri, R. (2024). An effective classification of DDoS attacks in a distributed network by adopting hierarchical machine learning and hyperparameters optimization techniques. *IEEE Access : Practical Innovations, Open Solutions*, *12*, 10834–10845. 10.1109/ACCESS.2024.3352281

Devarajan, G. G., Nagarajan, S. M., Daniel, A., Vignesh, T., & Kaluri, R. (2024). Consumer product recommendation system using adapted PSO with federated learning method. *IEEE Transactions on Consumer Electronics*, *70*(1), 2708–2715. 10.1109/TCE.2023.3319374

Ghosh, S., & De, D. (2023). TARA: Weighted majority cooperative game theory-based task assignment and resource allocation in 5G heterogeneous fog network for IoT. *The Journal of Supercomputing*, *79*(13), 14633–14683. 10.1007/s11227-023-05228-w

Guo, H., Zhang, J., Liu, J., & Zhang, H. (2018). Energy-aware computation offloading and transmit power allocation in ultradense IoT networks. *IEEE Internet of Things Journal*, *6*(3), 4317–4329. 10.1109/JIOT.2018.2875535

Hazra, A., Kalita, A., & Gurusamy, M. (2023). Meeting the requirements of Internet of things: The promise of edge computing. *IEEE Internet of Things Journal*, *11*(5), 7474–7498. 10.1109/JIOT.2023.3339492

Hosseinioun, P., Kheirabadi, M., Tabbakh, S. R. K., & Ghaemi, R. (2020). A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm. *Journal of Parallel and Distributed Computing*, *143*, 88–96. 10.1016/j.jpdc.2020.04.008

Hu, B., Yan, Z., & Zhao, M. (2023). Workload-aware scheduling of multiple-criticality real-time applications in vehicular edge computing system. *IEEE Transactions on Industrial Informatics*, *19*(10), 10091–10101. 10.1109/TII.2022.3231420

Hua, W., Liu, P., & Huang, L. (2023). Energy-efficient resource allocation for heterogeneous edge-cloud computing. *IEEE Internet of Things Journal*, *11*(2), 2808–2818. 10.1109/JIOT.2023.3293164

Khan, W. U., Javed, M. A., Nguyen, T. N., Khan, S., & Elhalawany, B. M. (2021). Energy-efficient resource allocation for 6G backscatter-enabled NOMA IoV networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 9775–9785. 10.1109/TITS.2021.3110942

Li, B., Hou, F., Yang, G., Zhao, H., & Chen, S. (2023). Data analysis-oriented stochastic scheduling for cost efficient resource allocation in NFV based MEC network. *IEEE Transactions on Vehicular Technology*, *72*(5), 6695–6708. 10.1109/TVT.2023.3234285

Li, P., Xiao, Z., Wang, X., Huang, K., Huang, Y., & Gao, H. (2023). EPtask: Deep reinforcement learning based energy-efficient and priority-aware task scheduling for dynamic vehicular edge computing. *IEEE Transactions on Intelligent Vehicles*, *9*(1), 1830–1846. 10.1109/TIV.2023.3321679

Liu, F., Huang, J., & Wang, X. (2023). Joint task offloading and resource allocation for device-edge-cloud collaboration with subtask dependencies. *IEEE Transactions on Cloud Computing*, *11*(3), 3027–3039. 10.1109/TCC.2023.3251561

Llorens-Carrodeguas, A., Sagkriotis, S. G., Cervelló-Pastor, C., & Pezaros, D. P. (2021). An energy-friendly scheduler for edge computing systems. *Sensors (Basel)*, *21*(21), 7151. Advance online publication. 10.3390/s2121715134770455

Ogundoyin, S. O., & Kamil, I. A. (2023). Optimal fog node selection based on hybrid particle swarm optimization and firefly algorithm in dynamic fog computing services. *Engineering Applications of Artificial Intelligence*, *121*(C), 105998. Advance online publication. 10.1016/j.engappai.2023.105998

Pallewatta, S., Kostakos, V., & Buyya, R. (2023). Placement of microservices-based IoT applications in fog computing: A taxonomy and future directions. *ACM Computing Surveys*, *55*(14s), 321. Advance online publication. 10.1145/3592598

Raju, M. R., & Mothku, S. K. (2023). Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach. *Computer Networks*, *224*, 109603. Advance online publication. 10.1016/j.comnet.2023.109603

Reiss-Mirzaei, M., Ghobaei-Arani, M., & Esmaeili, L. (2023). A review on the edge caching mechanisms in the mobile edge computing: A social-aware perspective. *Internet of Things : Engineering Cyber Physical Human Systems*, *22*, 100690. Advance online publication. 10.1016/j.iot.2023.100690

Rudenko, A., Reiher, P., Popek, G. J., & Kuenning, G. H. (1998). Saving portable computer battery power through remote process execution. *Mobile Computing and Communications Review*, *2*(1), 19–26. 10.1145/584007.584008

Sahni, Y., Cao, J., & Yang, L. (2018). Data-aware task allocation for achieving low latency in collaborative edge computing. *IEEE Internet of Things Journal*, *6*(2), 3512–3524. 10.1109/JIOT.2018.2886757

Sakr, A. H., Tabassum, H., Hossain, E., & Kim, D. I. (2015). Cognitive spectrum access in device-to-device-enabled cellular networks. *IEEE Communications Magazine*, *53*(7), 126–133. 10.1109/MCOM.2015.7158275

Shruti, S., Rani, S., & Srivastava, G. (2024). Secure hierarchical fog computing-based architecture for industry 5.0 using an attribute-based encryption scheme. *Expert Systems with Applications*, *235*, 121180. Advance online publication. 10.1016/j.eswa.2023.121180

Singh, S., Sham, E. E., & Vidyarthi, D. P. (2024). Optimizing fog device deployment for maximal network connectivity and edge coverage using metaheuristic algorithm. *Future Generation Computer Systems*, *157*, 529–542. 10.1016/j.future.2024.04.010

Song, M., Shan, H., Fu, Y., Yang, H. H., Hou, F., Wang, W., & Quek, T. Q. (2023). Joint user-side recommendation and D2D-assisted offloading for cache-enabled cellular networks with mobility consideration. *IEEE Transactions on Wireless Communications*, *22*(11), 8080–8095. 10.1109/TWC.2023.3258525

Sun, J., Li, H., Zhang, Y., Xu, Y., Zhu, Y., Zang, Q., Wu, Z., & Wei, Z. (2020). Multi-objective task scheduling for energy-efficient cloud implementation of hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 587–600. 10.1109/JSTARS.2020.3036896

Sun, W. B., Xie, J., Yang, X., Wang, L., & Meng, W. X. (2023). Efficient computation offloading and resource allocation scheme for opportunistic access fog-cloud computing networks. *IEEE Transactions on Cognitive Communications and Networking*, *9*(2), 521–533. 10.1109/TCCN.2023.3234290

Tian, M.-W., Yan, S.-R., Guo, W., Mohammadzadeh, A., & Ghaderpour, E. (2023). A new task scheduling approach for energy conservation in Internet of things. *Energies*, *16*(5), 2394. Advance online publication. 10.3390/en16052394

Tsemogne, O., Hayel, Y., Kamhoua, C., & Deugoué, G. (2021). Game-theoretic modeling of cyber deception against epidemic botnets in internet of things. *IEEE Internet of Things Journal*, *9*(4), 2678–2687. 10.1109/JIOT.2021.3081751

Wang, H., Xie, J., & Muslam, M. M. A. (2023). Fair: Towards impartial resource allocation for intelligent vehicles with automotive edge computing. *IEEE Transactions on Intelligent Vehicles*, *8*(2), 1971–1982. 10.1109/TIV.2023.3234888

Wang, S., Song, X., Xu, H., Song, T., Zhang, G., & Yang, Y. (2023). Joint offloading decision and resource allocation in vehicular edge computing networks. *Digital Communications and Networks*. Advance online publication. 10.1016/j.dcan.2023.03.006

Wang, X., Li, J., Ning, Z., Song, Q., Guo, L., Guo, S., & Obaidat, M. S. (2023). Wireless powered mobile edge computing networks: A survey. *ACM Computing Surveys*, 55(13s). *ACM Computing Surveys*, *263*(13s), 1–37. 10.1145/3579992

Wang, X., Ye, J., & Lui, J. C. (2023). Mean field graph based D2D collaboration and offloading pricing in mobile edge computing. *IEEE/ACM Transactions on Networking*, *32*(1), 491–505. 10.1109/TNET.2023.3288558

Wen, J., Yang, J., Wang, T., Li, Y., & Lv, Z. (2023). Energy-efficient task allocation for reliable parallel computation of cluster-based wireless sensor network in edge computing. *Digital Communications and Networks*, *9*(2), 473–482. 10.1016/j.dcan.2022.06.014

Xiao, W., Fang, X., Liu, B., Wang, J., & Zhu, X. (2023). UNION: Fault-tolerant cooperative computing in opportunistic mobile edge cloud. *ACM Transactions on Internet Technology*, *23*(4), 1–27. 10.1145/3617994

Xie, Z., Song, X., Cao, J., & Xu, S. (2022). Energy efficiency task scheduling for battery level-aware mobile edge computing in heterogeneous networks. *ETRI Journal*, *44*(5), 746–758. 10.4218/etrij.2021-0312

Xue, H., Chen, D., Zhang, N., Dai, H. N., & Yu, K. (2023). Integration of blockchain and edge computing in Internet of things: A survey. *Future Generation Computer Systems*, *144*(C), 307–326. 10.1016/j.future.2022.10.029

Yang, C., Wang, Y., Lan, S., & Zhu, L. (2024). Multi-agent reinforcement learning based distributed channel access for industrial edge-cloud web 3.0. *IEEE Transactions on Network Science and Engineering*, 1–12. Advance online publication. 10.1109/TNSE.2024.3377441

Yu, L., Zhong, T., Bi, P., Wang, L., & Teng, F. (2023). Segment based power-efficient scheduling for real-time DAG tasks on edge devices. *Parallel Computing*, *116*, 103022. Advance online publication. 10.1016/j.parco.2023.103022

Zhang, J., Zhang, B., & Han, Z. (2023). Coalition formation game based information-energy collaboration in vehicle edge computing networks. *IEEE Transactions on Vehicular Technology*, *72*(6), 7717–7727. 10.1109/TVT.2023.3237369

Zhang, Q., Chen, D., Mahajan, Y., Chen, I.-R., Ha, D. S., & Cho, J. H. (2023). Attack-resistant, energy-adaptive monitoring for smart farms: Uncertainty-aware deep reinforcement learning approach. *IEEE Internet of Things Journal*, *10*(16), 14254–14268. 10.1109/JIOT.2023.3287069

Zhang, Q., Lin, X., Hao, Y., & Cao, J. (2020). Energy-aware scheduling in edge computing based on energy Internet. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 229052–229065. 10.1109/ACCESS.2020.3044932

Zhang, S., Liu, Q., Wang, T., Liang, W., Li, K. C., & Wang, G. (2024). FSAIR: Fine-grained secure approximate image retrieval for mobile cloud computing. *IEEE Internet of Things Journal*, *11*(13), 23297–23308. 10.1109/JIOT.2024.3384458

Zhang, S., Zhang, S., Yeung, L. K., & Yu, J. J. Q. (2023). Urban Internet of electric vehicle parking system for vehicle-to-grid scheduling: Formulation and distributed algorithm. *IEEE Transactions on Vehicular Technology*, *73*(1), 67–79. 10.1109/TVT.2023.3304718

Zhang, Y., Lan, X., Ren, J., & Cai, L. (2020). Efficient computing resource sharing for mobile edge-cloud computing networks. *IEEE/ACM Transactions on Networking*, *28*(3), 1227–1240. 10.1109/TNET.2020.2979807

Zhao, Q., Geirhofer, S., Tong, L., & Sadler, B. M. (2008). Opportunistic spectrum access via periodic channel sensing. *IEEE Transactions on Signal Processing*, *56*(2), 785–796. 10.1109/TSP.2007.907867

Shaolei Chen obtained a doctoral degree in Information and Communication Systems from Zhejiang University and currently works at State Grid Sichuan Electric Power Company, with a research focus on power communication technology.

Hanyuan Tang obtained a bachelor's degree in Communication Engineering from Southwest University of Science and Technology, and currently works at State Grid Panzhihua Power Supply Company, with a research focus on power communication

Min Zhao holds a master's degree in Signal and Information Processing and currently works at the Information and Communication Company of State Grid Sichuan Electric Power Company, with a research focus on power communication technology.

Yu Chen holds a bachelor's degree in dispatch and operation, currently working at State Grid Yibin Power Supply Company with a research focus on power communication technology.

Xin Yang holds a master's degree in Communication and Information Systems and currently works at State Grid Yibin Power Supply Company, with a research focus on power communication technology.

Kejue Hu holds a master's degree in communication engineering and currently works at State Grid Tianfu New Area Power Supply Company, with a research focus on power communication technology.