

Strategy Analysis of Mobile Edge Computing Based on EC-ANN in Task Vehicle Cooperative Unloading

Chenwei Wang, Xiating Li*

School of Information Engineering, Jiangxi V&T College of Communications, Nanchang 330013, China

E-mail: lixiating1976@126.com

*Corresponding author

Keywords: Edge computing, Internet of Things, Vehicle collaboration, DNN, RSU, V2V

Received: April 2, 2024

To improve the performance and reliability of task vehicle collaborative unloading, the study adopted Monte Carlo tree search and deep neural networks to optimize resource allocation of task vehicles in collaborative unloading. Secondly, through multi-mode collaboration, the relay unloading task of roadside units was carried out. Meanwhile, the service range of vehicle collaborative unloading was expanded based on the calculation results, achieving the full utilization of idle computing resources. These experiments confirmed that compared to random search and greedy search, the proposed network model scheme improved service latency performance by 58.3% and 47.1%, respectively. The proposed multi-mode joint unloading mechanism had significant performance improvement under the collaborative unloading mechanism from adjacent vehicles to vehicles. It offloaded tasks to service vehicles outside the communication range, reducing completion latency by approximately 33.6%. Therefore, this task vehicle collaboration unloading method improved the performance of mobile edge computing systems, reduced computing and storage costs, and lowered the energy consumption and maintenance costs of task vehicles. This research method can improve the efficiency and safety of task vehicle collaboration unloading, providing technical support for the optimization of intelligent transportation systems.

Povzetek: Raziskava uvaja strategijo mobilnega robnega računalništva z umetnimi nevronskimi mrežami (EC-ANN) za sodelovalno razbremenjevanje vozil, kar omogoča razbremenjevanje nalog z zmanjšanjem zakasnitve.

1 Introduction

In the context of accelerated global economic and urbanization development, the transportation industry is facing the needs of increasingly efficient, safe, and energy-saving development [1]. In transportation, unloading is an important process that not only affects the safety and efficiency of transportation, but also involves issues such as energy consumption and carbon emissions [2]. It should optimize and intelligentize the unloading process to improve the efficiency and safety of transportation unloading and reduce energy consuming and carbon emissions [3]. Edge Computing (EC) is a new type of computing mode that can distribute computing and data storage tasks at the network edge, achieving faster response speed and lower latency [4]. In the transportation and other related fields, EC can help achieve real-time monitoring, decision-making, and control, improving work efficiency and safety. In the unloading task, how to optimize the unloading strategy and improve unloading efficiency and safety through technologies such as EC is an urgent problem that needs to be solved in transportation [5]. Meanwhile, the traditional unloading method has high latency and low efficiency in the unloading of computing tasks. Therefore,

the study proposed a strategy for mobile EC based on Edge Computing-Artificial Neural Network (EC-ANN) in Task Vehicle Collaboration Unloading (TVCU), aiming to provide a new approach for the transportation industry. The study consists of four parts. Firstly, the research on EC and vehicle collaboration is summarized. Secondly, the TVCU method is designed. Then the proposed unloading method is validated. Finally, there is a summary of the entire study.

2 Related works

Mobile EC is a distributed computing model that moves data processing, storage, and analysis from centralized data centers to network edges. Xu et al. proposed an adaptive method for multi-user computing unloading by decoupling the long-term unloading problem into multiple single-time slot unloading problems. These experiments confirmed that this method demonstrated substantial performance advantages in a large number of experiments [6]. Gao et al. utilized ultra dense network scenarios and mobile EC servers to assist in transforming the problem into sub-problems of unloading strategy, channel allocating, and power allocating. Joint unloading and resource allocating algorithms were used to obtain an optimal joint strategy. These experiments confirmed that

the algorithm effectively reduced system energy consumption and improved overall system performance [7]. Laroui et al. proposed the optimal service unloading algorithm based on integer linear programming and proposed a service unloading protocol to support this use case. These experiments confirmed that the proposed algorithm significantly improved efficiency in service unloading, resource utilization, and network [8]. Aung et al. proposed a Fog Edge hybrid computing architecture for Metaverse applications to utilize the edge distributed computing paradigm to address the issues caused by long cloud access latency. The computing power of edge devices was utilized to perform heavy tasks in Metaverse applications. These experiments confirmed that the Fog Edge architecture reduced latency by 50% [9]. Pang et al. improved the existing EC system model and provided a computational model for the energy balance optimization of multiple devices and tasks. Meanwhile, a greedy algorithm was proposed, and the corresponding approximate ratio analysis was conducted. These experiments confirmed that compared to random algorithms, greedy algorithms further improved their average performance in energy balance by 66.59% [10].

Vehicle collaboration refers to achieving task division, path planning, safety control, and optimized driving through communication and computation between vehicles. Ma and Sun proposed a comprehensive solution framework for remote information processing with end-to-end collaboration. The experiment combined network resource deploying optimization for video transmission tasks in remote information processing application scenarios. These experiments confirmed that

the proposed scheme improved the performance of network systems [11]. Sun et al. built an interface server, which received information and applied hidden Markov models to predict and optimize future operating conditions. These experiments confirmed that the relative error of the proposed method in estimating the remaining driving distance remained within 5% [12]. Li et al. established a vehicle group model for the main road and ramp in the entrance ramp area using the acceptable gap theory in a connected vehicle environment. They proposed a simulation scheme based on time slices and virtual signal control strategies. These experiments confirmed that the joint implementation of mainline and ramp control strategies was more effective than using only mainline strategies [13]. Shi et al. put forward a multi-layer collaborative framework for logistics management in industrial parks. Effective logistics in industrial parks was achieved through device edge cloud collaboration by collaborating environmental perception, map construction, task allocation, path planning, and vehicle movement. These experiments confirmed that the logistics analysis of industrial parks validated this proposed cooperation framework's feasibility [14]. Li et al. proposed a sliding mode control design method for rail electric vehicles using global dynamic information. They used vehicle models with different accelerating parameters for verification under communication delay. These experiments confirmed that this method had fault tolerance and met rail electric vehicles' designing requirements [15]. The summary table for related works is shown in Table 1.

Table 1: Summary table for related works

Field	Researchers	Research content	Research findings	Computation comple	Uninstall flexibility	Limitations
Mobile EC	Xu et al	self-adaptive method for multi-user computing offloading	Substantial performance advantages	Common	Common	It may be affected by factors
	Gao et al	Adopting joint offloading and resource allocation algorithms	Reduce system energy consumption	High	Common	It takes a long time to find the optimal solution.
	Laroui et al	Optimal service offloading algorithm	Improved service offloading, resource utilization	High	Common	The computational complexity is high
	Aung et al	Fog Edge hybrid computing architecture	Reduce latency by 50%.	Common	Common	Devices may have compatibility
	Pang et al	The existing edge	Average performance	Common	Common	The approximate

Vehicle collaboration	Ma and Sun	computing system model A comprehensive solution framework Application programming interface server	improvement by 66.59%. Improve network system performance	High	Common	ratio is not precise Network latency and bandwidth limitations
	Sun et al	Simulation scheme based on time slice and virtual control	The relative error kept within 5%. More effective than using only the mainline strategy	High	Common	Computational complexity
	Li et al	Multi level collaborative framework	The cooperation framework is feasible. This method has significant fault tolerance and meets the design requirements	Common	Common	Limited application scenarios
	Shi et al	Design method for sliding mode control of rail electric vehicles.		High	Common	System stability is easily affected
	Li et al			Common	Common	Communication delay may be affected by various factors

In summary, the above research has achieved certain results in the mobile EC, vehicle collaboration, and logistics management. However, their computational complexity is relatively high in specific situations, and the flexibility of unloading strategies is poor. Therefore, the study proposes a mobile EC based on EC-ANN, and the TVCU strategy is analyzed and optimized to improve system efficiency and performance.

3 Design of collaborative unloading method for task vehicles

The study optimizes resource allocation for task vehicles in collaborative unloading through Monte Carlo Tree Search (MCTS) and Deep Neural Network (DNN). On the basis of resource allocation, the service scope of

Vehicle Collaboration Unloading (VCU) is extended through multi-mode federation to relay unloading tasks or calculation results at Road Side Unit (RSU).

3.1 Resource allocation algorithm based on EC-ANN

EC refers to the execution of computing tasks at network edge nodes to reduce data transmission latency and improve network efficiency. For TVCU, a resource allocation algorithm based on EC-ANN is utilized in this study to optimize the resource allocation of task vehicles during collaborative unloading. Figure 1 is a resource allocation algorithm based on EC-ANN.

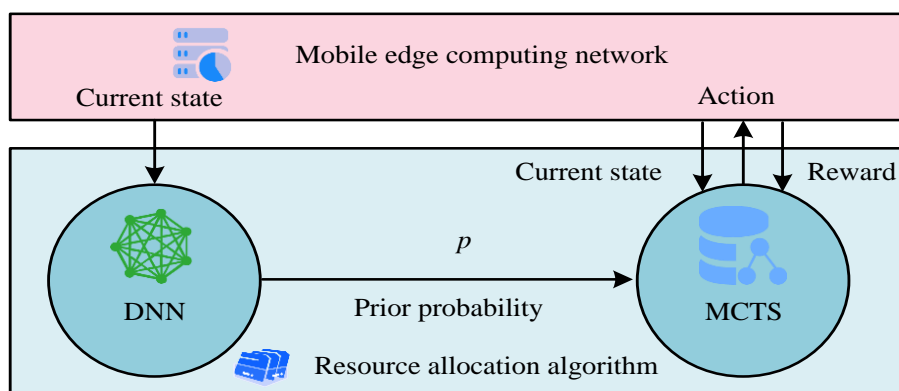


Figure 1: Structure of resource allocation algorithm based on EC-ANN

In Figure 1, the resource allocation algorithm based on EC-ANN mainly includes MCTS and DNN. MCTS is a random simulation and search algorithm that evaluates decisions through multiple random simulations and selects the best strategy in a tree structure. DNN is a deep learning model that learns the mapping relationship between input and output through a large number of training samples. DNN is used to predict and optimize the decision results of MCTS. MCTS uses a tree search process to find the optimal decision for each mobile device regarding unloading rate, computing resource ratio, and communication resource ratio. DNN is responsible

for generating a prior probability distribution to guide MCTS search to accelerate the convergence of MCTS [16]. To train DNN, the study collects training data and labels from the iterative results of MCTS. The input of training data is the same as the first layer state of MCTS. The labels are composed of the probability distribution of nodes in the corresponding layer after MCTS iteration. Then, the prior probabilities output by the trained DNN are used to guide the next MCTS search, thereby improving the quality of the MCTS output strategy. Figure 2 shows the search of MCTS.

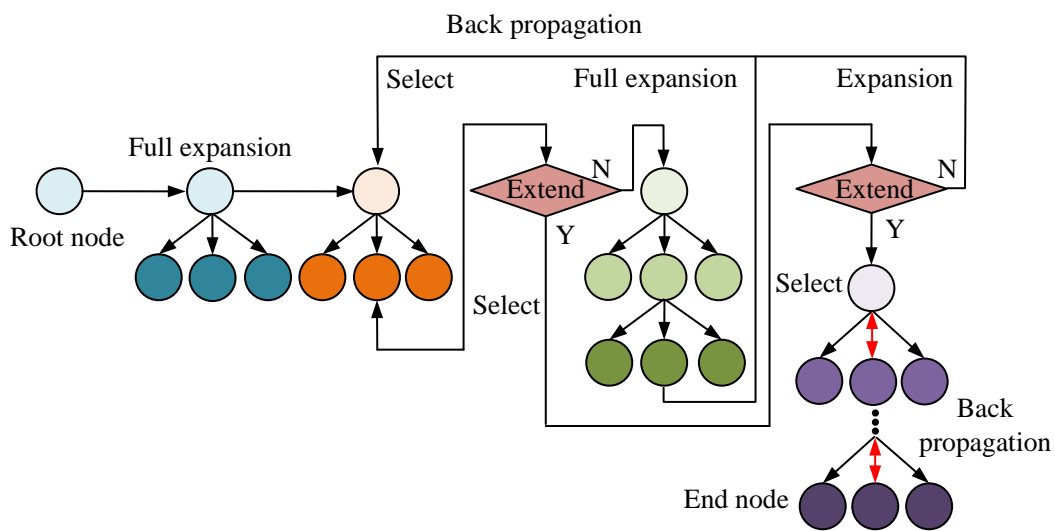


Figure 2: MCTS module search process

The search of MCTS mainly includes four steps: selection, expansion, evaluation, and backpropagation. Firstly, MCTS receives input parameters. Next, the expanding and selecting steps are repeated until reaching the leaf node. This path deriving from a root node to a leaf node means the actions of K mobile devices. At a leaf node, mobile devices offload tasks to edge servers,

which execute task T based on allocation decisions. After completing every mission, mission completion’s average waiting time will be fed back to MCTS as a rewarding signal to assess action performance [17]. MCTS iteration’s final step is backpropagation of rewards to update the search strategy. Figure 3 shows the DNN structure.

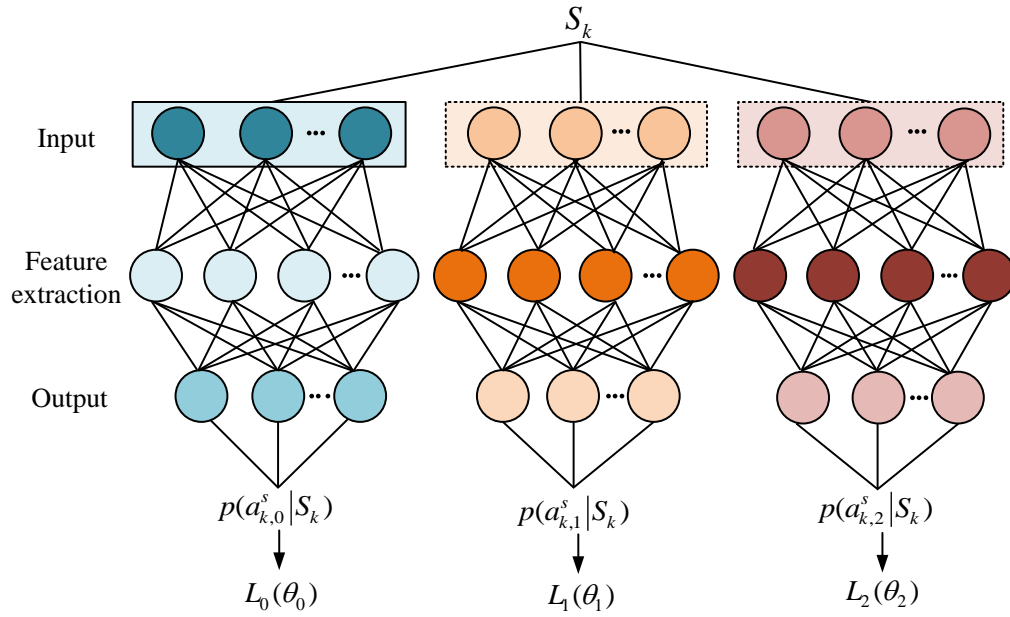


Figure 3: DNN structure

The DNN structure mainly includes an inputting layer, multiple hiding layers, and an outputting layer. An input layer is used to receive environmental information. An output layer is used to generate the probability distribution for each sub-action. A hidden layer is responsible for non-linear transformation and abstract representation of input features to extract more representative features [18]. During the training phase, DNN randomly extracts training data from the dataset output by MCTS and uses three optimizers with independent loss functions for training. DNN uses gradient descent to minimize the loss function and update network parameters to improve model prediction's accuracy. The action space of multiple sub-actions is represented by equation (1).

$$A_k = \{a_{k,l}\} = \{a_i(\eta), a_i(b), a_i(f^e)\} \quad (1)$$

In equation (1), the action space of the sub-action is A_k . The sub-action is represented as $a_{k,l}$. The output unloading probability is $a_i(\eta)$. The output wireless broadband resource is $a_i(b)$. The ratio of output edge servers to computing resource allocation is $a_i(f^e)$. The prior probability of the output sub-action unloading rate is represented by equation (2).

$$p(a_{k,0}^s | S_k) = f_{\theta_0}(S_k) \quad (2)$$

In equation (2), the prior probability of the output sub-action unloading rate is $p(a_{k,0}^s | S_k)$. The DNN parameter for outputting unloading rate is θ_0 . The input status parameter is S_k . The prior probability of wireless broadband output sub-actions is represented by equation (3).

$$p(a_{k,1}^s | S_k) = f_{\theta_1}(S_k) \quad (3)$$

In equation (3), the prior probability of the wireless broadband output sub-action is $p(a_{k,1}^s | S_k)$. The DNN parameter for outputting wireless bandwidth resources is θ_1 . The prior probability of calculating resource allocation ratio for output sub-actions is represented by equation (4).

$$p(a_{k,2}^s | S_k) = f_{\theta_2}(S_k) \quad (4)$$

In equation (4), the prior probability of calculating the resource allocation ratio for the output sub action is $p(a_{k,2}^s | S_k)$. The corresponding DNN parameter is θ_2 . The prior probability of the l th sub-action of task T_k is $p(a_{k,l}^s | S_k)$. Each layer's neurons number of DNN is represented by equation (5).

$$H^l = \{H_{in}, H_1, \dots, H_n, H_{out}\} \quad (5)$$

In equation (5), the total neurons in each layer of DNN are H^l . The neurons in the input layer are H_{in} . The intermediate layer neurons are H_1, \dots, H_n . The output layer's neurons are H_{out} . The loss function of DNN optimization is represented by equation (6).

$$L_l(\theta_l) = -\pi(a_{k,l}^s | S_k) \log p(a_{k,l}^s | S_k) + \zeta \|\theta_l\|^2 \quad (6)$$

In equation (6), the DNN optimization loss function is $L_l(\theta_l)$. The MCTS output dataset is $\{\pi(A_{k,l}^s | S_k)\}$.

The randomly extracted training data are $\{\pi(a_{k,l}^s | S_k)\}$.

The regular term is $\zeta \|\theta_t\|^2$.

3.2 Collaborative unloading strategy for task vehicles

The resource allocation algorithm based on EC-ANN can provide fair and efficient allocation of computing and storage resources. The TVCU strategy can maximize the

efficiency and effectiveness of collaborative unloading. A multi-mode joint VCU mechanism is proposed to address the issue of VCU. The service range of VCU is expanded by unloading tasks or computing results through RSU relay. Idle computing resources are fully utilized while ensuring unloading reliability. Figure 4 shows the scenario of a multi-mode joint VCU system.

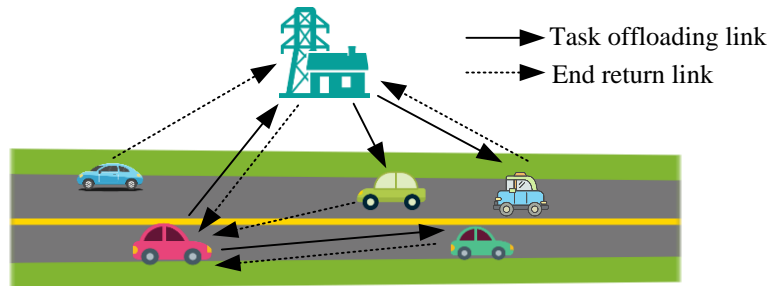


Figure 4: Multi-mode joint vehicle collaboration unloading system scenario

In this multimodal joint VCU system, a bidirectional road with unstable RSU coverage is considered. The coverage radius of RSU in the road is L , and the total coverage length is $2L$. The road is abstracted as a horizontal coordinate axis, and the entrance of RSU on the coverage area’s left side on this road is the coordinate origin [19]. Vehicles driving on the road are divided into task and service vehicles. Task vehicles are computing resource constrained and have a computationally intensive task to execute, while service vehicles have spare computing resources. In this scenario, RSU acts as the global control center to make unloading decisions, quickly making unloading decisions based on collected task information, computing power, speed, and location information of service vehicles traveling on the road [20]. The communication rate between vehicles is represented by equation (7).

$$n_{i,j}^v = B \log_2 \left(1 + \frac{P_v g_{i,j}}{n_0^2} \right) \quad (7)$$

In equation (7), the communication rate and channel gain between vehicles i and j are $n_{i,j}^v$ and $g_{i,j}$,

respectively. Each vehicle is connected to an orthogonal Vehicle-to-Vehicle (V2V) channel with a bandwidth of B . This vehicle’s transmission power is P_v . The additive Gaussian white noise on the communication link is n_0^2 . The communication rate from RSU to vehicle i is represented by equation (8).

$$n_i^d = B_d \log_2 \left(1 + \frac{P_v g_i}{n_0^2} \right) \quad (8)$$

In equation (8), the communication rate from RSU to vehicle i is n_i^d . The downlink channel bandwidth is B_d . This uplink vehicle i and RSU’s communication rate is represented by equation (9).

$$n_i^u = B_u \log_2 \left(1 + \frac{P_v g_i}{n_0^2} \right) \quad (9)$$

In equation (9), the communication rate between the uplink vehicle i and RSU is n_i^u . The uplink channel bandwidth is B_u . The study equates roads to a horizontal one-dimensional coordinate axis, treating vehicle movement as movement on the one-dimensional coordinate axis. Figure 5 shows the vehicle movement model.

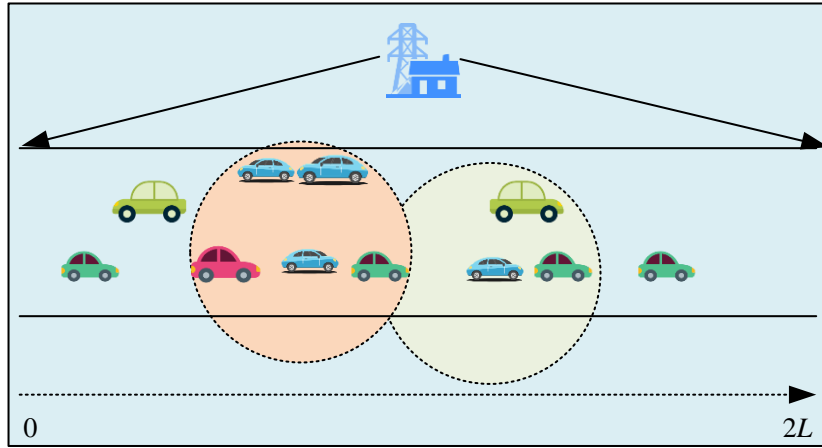


Figure 5: Vehicle mobility model

In the vehicle movement model, the motion state of the vehicle on the road can be described by defining the time when the task vehicle and service vehicle leave the RSU coverage area, as well as the time when the V2V link is established and disconnected. Meanwhile, by calculating the position coordinates of vehicles during the establishment of the V2V link and the moment of link disconnection, the interaction between vehicles can be further analyzed [21]. The moment of V2V link disconnection is calculated using equation (10).

$$\tau_{i,j} = f(d_i, d_j)\tau_{i,j}^+ + (1 - f(d_i, d_j))\tau_{i,j}^- \quad (10)$$

In equation (10), the moment of vehicles i and j at V2V link disconnection is $\tau_{i,j}$, corresponding to the road movement directions of d_i and d_j , respectively. This indicator function is f . The service time for positive movement is $\tau_{i,j}^+$. The service time for reverse motion is $\tau_{i,j}^-$. The completion delay of task vehicles includes local calculation delay and unloading task completion delay, and the local calculation delay is represented by equation (11).

$$t_{i,0} = \frac{b_{i,0}D_iC_i}{F_i} \quad (11)$$

In equation (11), the local calculation delay when vehicle i is 0 is $t_{i,0}$. The task proportion of unloading task vehicle i to service vehicle j is $b_{i,0}$. The input data size for the task is D_i . The required computational intensity for the task is C_i . The computing power of the task vehicle i itself is F_i . The task vehicle i has two unloading methods, namely unloading through V2V link or RSU relay. There are also two optional ways to return the results, either V2V or RSU relay return. Therefore, there are four values for setting pattern decision variables. For the service vehicles in the collection, the task vehicles can only pass through the fourth mode. According to the values of pattern decision variables, service vehicles are divided into four sets corresponding to four different patterns. In Mode 1, the task vehicle unloads the goods or tasks it carries onto the service

vehicle through V2V communication. At the same time, the service vehicle also returns the completed task results to the task vehicle through V2V. In Mode 1, the expression of the completion time is represented by equation (12).

$$t_{i,j}^1 = \delta_{i,j} + t_{i,j}^{1,com} + t_{i,j}^{1,cmp} + t_{i,j}^{1,return} \quad (12)$$

In equation (12), in Mode 1, the execution completion time is $t_{i,j}^1$. The delay waiting for the establishment of the V2V link is $\delta_{i,j}$. The task upload delay is $t_{i,j}^{1,com}$. The calculated delay is $t_{i,j}^{1,cmp}$. The delay in returning task results is $t_{i,j}^{1,return}$. At this point, the following constraints need to be met to ensure the reliability of unloading, represented by equation (13).

$$t_{i,j}^1 \leq \tau_{i,j} \quad (13)$$

In Mode 2, task vehicle i unloads task data to service vehicle j through V2V. Then, j returns the result to i through RSU relay. The task ratio, bandwidth ratio of task vehicle i , and resource ratio are determined. The expressions for the established delay, transmission delay, and calculation delay of the waiting V2V link are the same as those in Mode 1. However, the expression for the return delay of task results is different. It includes the proportion of uplink channel bandwidth used by service vehicles to return task results through RSU relays. This also includes the proportion of downlink channel bandwidth used when the task execution results offloaded to the service vehicle j are relayed back from the RSU to the task vehicle i . In Mode 2, the calculation of the completion time is represented by equation (14).

$$t_{i,j}^2 = \delta_{i,j} + t_{i,j}^{2,com} + t_{i,j}^{2,cmp} + t_{i,j}^{2,return} \quad (14)$$

In equation (14), in Mode 2, the execution completion time is $t_{i,j}^2$. The task upload delay is $t_{i,j}^{2,com}$. The calculation latency is $t_{i,j}^{2,cmp}$. The delay in returning task results is $t_{i,j}^{2,return}$. The constraint conditions that need to be met at this time are represented by equation

(15).

$$\delta_{i,j} + t_{i,j}^{2,com} \leq \tau_{i,j} \quad (15)$$

For Models 3 and 4, the delay calculation and constraint conditions for task unloading and returning in service vehicles and RSU relays are considered, respectively. This includes comprehensive considerations of communication delay, calculation delay, and return delay, and constraints on bandwidth allocation ratio and channel allocation. For Mode 3, it is also necessary to consider the reliable return time.

In Mode 3, when the task ratio, computing resource ratio, and service vehicle j are determined, the method for calculation latency and return latency is the same as in Mode 1. In Mode 4, unloading and returning are performed using RSU relay mode. For any service vehicle, the task vehicle can be divided into two sets,

representing the set of task vehicles that choose to return results in V2V mode and the set of task vehicles that choose to return results in RSU relay mode. The task completing time of the task vehicle is represented by equation (16).

$$t_i = \max(t_{i,0}, t_{i,j}^1, t_{i,j}^2, t_{i,j}^3, t_{i,j}^4) \quad (16)$$

In equation (16), the task completion time of task vehicle i is t_i . The execution completion time of Mode 3 is $t_{i,j}^3$. The execution completion time of Mode 4 is $t_{i,j}^4$. This optimization problem’s objective is to minimize all current task vehicles’ average completion delay. This can be achieved by optimizing pattern decision variables as well as task allocation ratio, bandwidth allocation ratio, and computational resource allocation ratio. The collaborative unloading process of task vehicles is shown in Figure 6.

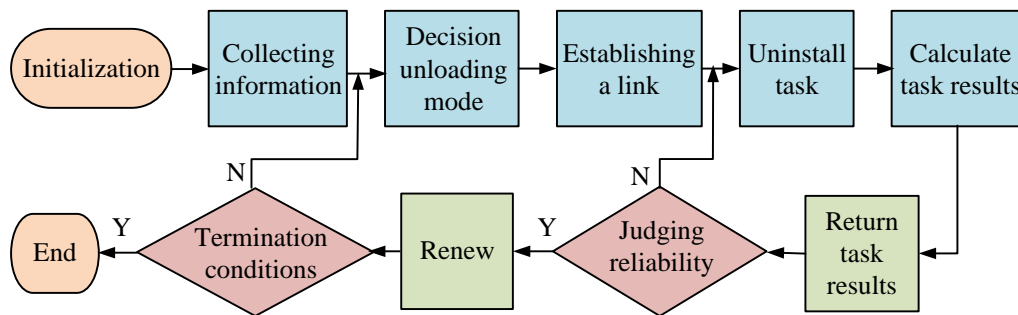


Figure 6: Task vehicle collaboration unloading process

The collaborative unloading process of task vehicles involves collecting real-time information, deciding on the unloading mode, establishing a V2V link, offloading computing tasks to service vehicles, and checking the unloading reliability. The mode decision variables are continuously optimized until all task vehicle tasks are completed.

4 Application analysis of collaborative unloading methods for task vehicles

The experiment compared algorithms such as EC-ANN, random search, greedy search, deep Q-network, and DNN. The advantages of EC-ANN in balancing network

performance and computational complexity were verified. Then, in the TVCU strategy of the EC-ANN scheme, the performance of different unloading mechanisms was compared.

4.1 Parameter settings and convergence performance of EC-ANN

The experiment adopted a large-scale connection and computing intensive Internet scenario. The base station is located in the center of a circular area of 1000 m2. An edge server serves multiple base station cells simultaneously. Many collaborative edging servers are randomly distributed around task request edging servers. Multiple mobile devices move randomly in the area at varying speeds or in random directions based on vehicle driving. Table 2 shows the EC-ANN parameter settings.

Table 2: EC-ANN model parameter settings

Serial number	Parameter	Value
1	Maximum search times for MCTS	200
2	Balance coefficient of MCTS	0.9
3	Search depth of MCTS	2K
4	Learning rate of DNN	0.05
5	The minibatch size of DNN	64
6	L2 norm constraint coefficient of DNN	10-5
7	The shared hidden layers in DNN	3

These edge servers were 4. These mobile devices were 110. Under both optimization objectives, the computing power of edge servers had two settings,

namely 10 GHz and 20 GHz. Figure 7 shows the convergence performance of EC-ANN.

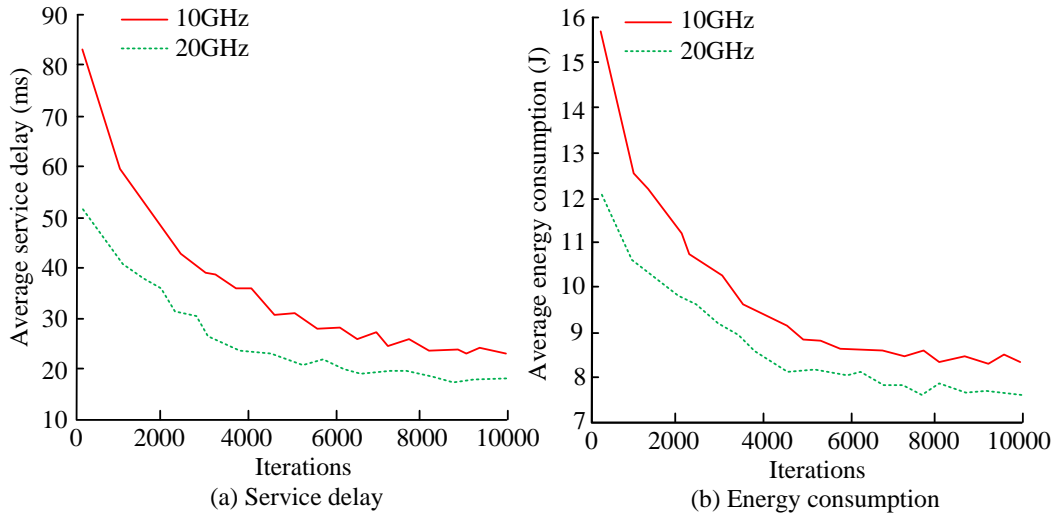


Figure 7: Convergent performance of the EC-ANN

Figure 7(a) shows the convergence of the average service delay. In the first 6200 iterations, the average service latency sharply decreased. After 6200 iterations, it gradually stabilized. Figure 7(b) presents average energy consumption’s convergence. In the first 5000 iterations, the average energy consumption sharply decreased. After 5000 iterations, it gradually stabilized. The computing power of edge servers at 10 GHz and 20 GHz had little impact on the convergence speed of EC-ANN. Iterations were set to 5000 in the subsequent experiments to balance

network performance and computational complexity.

4.2 Comparison of EC-ANN model performance

To validate EC-ANN’s performance, random search, greedy search, and Deep Q-Network (DQN) and DNN were compared. Figure 8 shows the impact of edge server computing power on optimizing average service latency and average energy consumption.

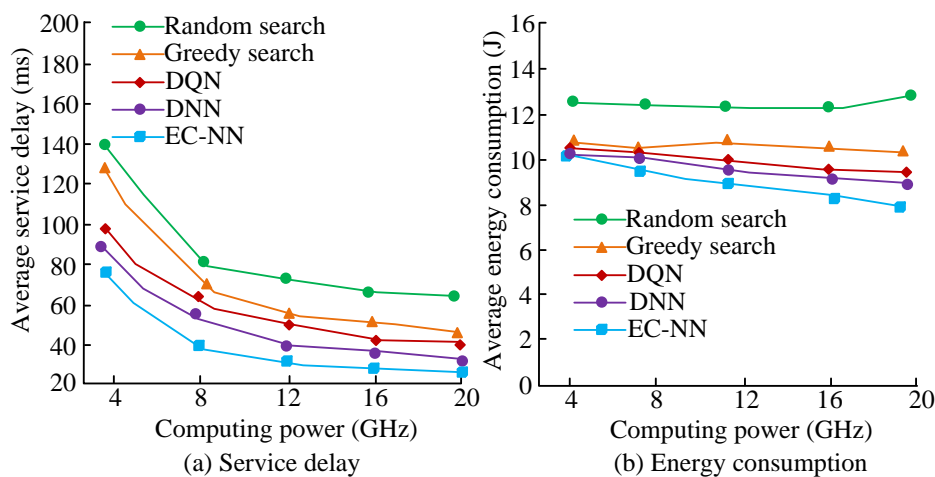


Figure 8: The effect of edge server computing power on optimizing average service delay and average energy expenditure

In Figure 8(a), the average service latency of these five algorithms showed a decreasing trend with the improvement of computing power. Compared with

random search, greedy search, DQN, and DNN, the EC-ANN-based solution improved service latency performance by 58.3%, 47.1%, 41.2%, and 39.8%,

respectively. In Figure 8(b), compared with random search, greedy search, DQN, and DNN, the EC-ANN-based approach reduced average energy consumption by 23.6%, 11.7%, 10.6%, and 9.5%,

respectively. Figure 9 shows the impact of edge servers and mobile devices on optimizing average service latency.

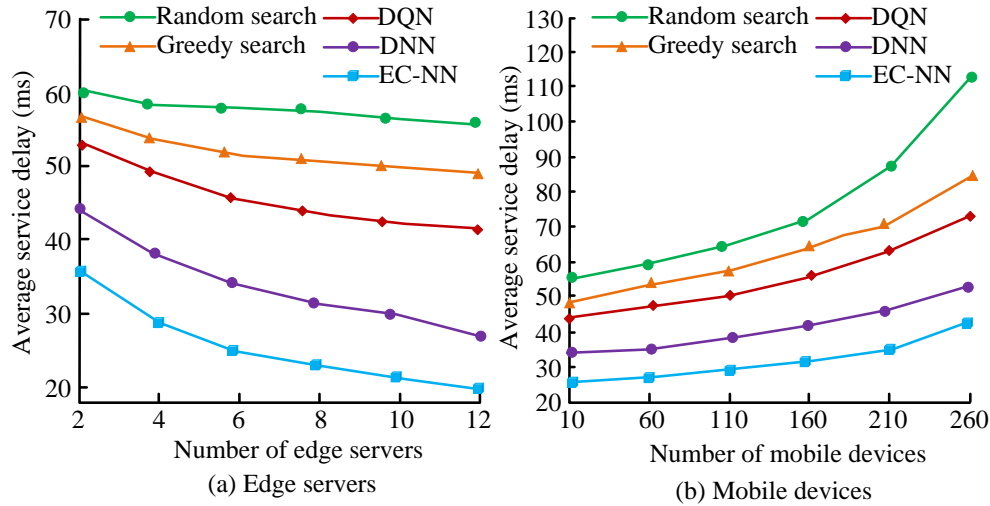


Figure 9: The effect of the number of edge servers and mobile devices on optimizing the average service delay

In Figure 9(a), compared to random search, greedy search, DQN, and DNN, EC-ANN reduced the average service latency by 64.1%, 59.8%, 51.8%, and 49.9%, respectively. In Figure 9(b), compared with random search, greedy search, DQN, and DNN, EC-ANN

significantly reduced average service latency by 62.1%, 47.8%, 42.2%, and 41.6%, respectively. Figure 10 shows the average service delay and energy consumption based on the EC-ANN scheme in the vehicle driving scenario.

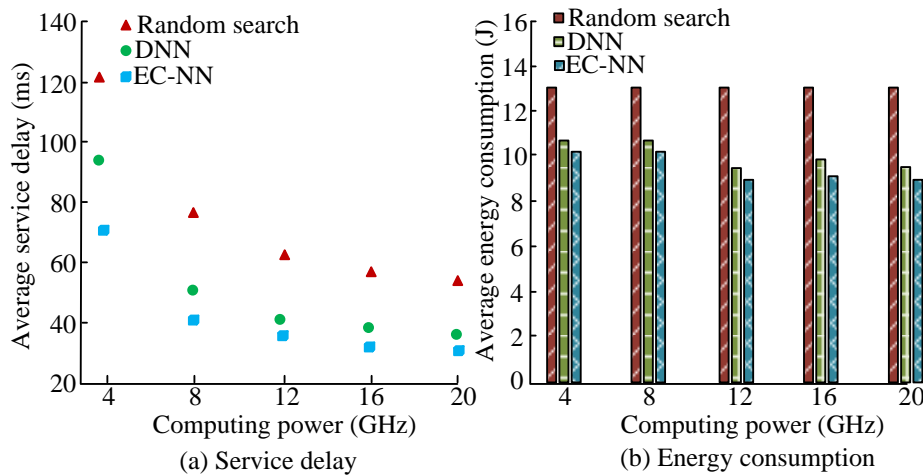


Figure 10: Average service delay and energy expenditure based on the EC-ANN scheme

Figure 10(a) shows the optimized average service delay of different algorithms in the vehicle driving scenario. Compared to random search and DNN, the average service latency based on EC-ANN scheme was reduced by 45.3% and 20.6%, respectively. Figure 10(b) shows the optimized average energy consumption of different algorithms in the vehicle driving scenario.

Compared to random search and DNN, the average energy consumption of the EC-ANN scheme was reduced by 36.7% and 11.3%, respectively. Therefore, EC-ANN performed excellently in collaborative mobile EC resource allocation, effectively reducing average energy consumption and service latency.

4.3 Application analysis of collaborative unloading strategy for task vehicles

This experiment compared the task vehicle performance under the V2V and V2V+V2X collaborative unloading mechanisms to verify the proposed multi-mode joint unloading mechanism’s effectiveness. In the V2V unloading mechanism, the task vehicle was only unloaded to the service vehicle within the current V2V

communication range. In the V2V+V2X collaborative unloading mechanism, task vehicles used V2V unloading for service vehicles within the current V2V communication range. The service vehicles outside the current V2V communication range were unloaded through RSU relay. Table 3 shows the settings of the vehicle networking environment parameters.

Table 3: The Internet of vehicles environment parameter setting

Serial number	Parameter	Value	Unit
1	RSU coverage length	600	m
2	Vehicle communication radius	100	m
3	Vehicle speed	[12, 30]	m/s
4	Task size	[25, 85]	Mbits
5	V2V communication Bandwidth	1.5	MHz
6	RSU downlink bandwidth	2.0	MHz
7	RSU uplink bandwidth	1.2	MHz

In a single-task vehicle scenario, the scope of service vehicles included two situations. The service vehicles were all outside the V2V communication range of these task vehicles, and these service vehicles were not limited

to the V2V communication range of the task vehicles. Figure 11 shows the variation of latency with the service vehicle.

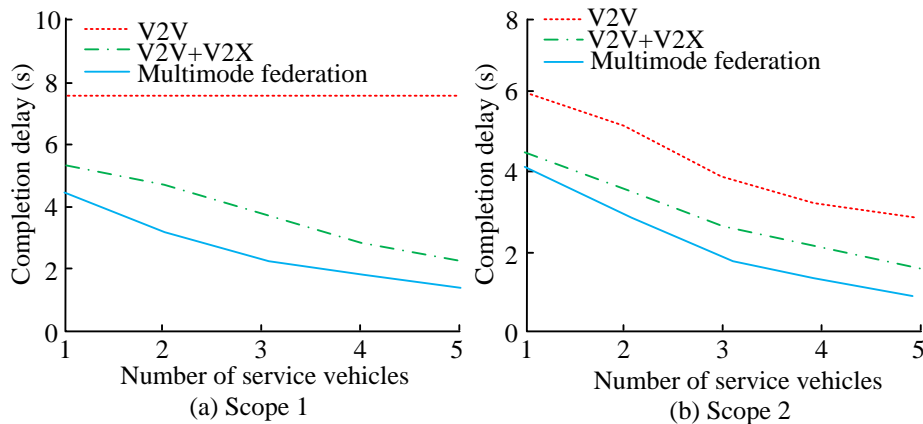


Figure 11: Time delay varies with the change of service vehicles

In Figure 11(a), the service vehicles were all outside the V2V communication range of the task vehicle. Compared to the V2V+V2X collaborative unloading mechanism, the multi-mode joint unloading mechanism offloaded tasks to service vehicles outside the V2V communication range, reducing completion latency by about 33.6%. This mechanism could be applied to situations where the service vehicle was outside the communication range of the task vehicle but was about to enter that range, improving unloading efficiency. In Figure 11(b), the service vehicle was not limited to task vehicle’s V2V communication range. As service vehicles increased, the V2V unloading mechanism’s latency was gradually decreasing. However, compared to the proposed multi-mode joint unloading mechanism, it still had the best latency performance. Compared to the

collaborative unloading mechanisms V2V and V2V+V2X, the multi-mode joint unloading mechanism reduced the completion latency by approximately 59.7% and 21.7%, respectively.

5 Discussion

Compared with other related works, the experimental results showed that EC-ANN had significant advantages in balancing network performance and computational complexity. Compared with other algorithms such as random search, greedy search, deep Q-network, and DNN, EC-ANN performed well in reducing average service latency and energy consumption. The EC-ANN algorithm had significant effect in resource allocation of cooperative mobile EC, which effectively reduced the average energy consumption and service delay. The study

applied the EC-ANN algorithm to collaborative offloading strategy, achieving optimized resource allocation of task vehicles in different scenarios. A multi-mode joint unloading strategy was designed for vehicle driving scenarios, further improving unloading efficiency. The reason why the EC-ANN algorithm exhibits high efficiency and effectiveness is that it can quickly find the optimal solution in complex environments. Meanwhile, the EC-ANN algorithm can adaptively adjust the offloading strategy to balance network performance and computational complexity by introducing the edge servers and mobile devices as optimization factors. The proposed EC-ANN algorithm and multi-mode joint unloading strategy achieve remarkable results in EC resource allocation in the Internet of Vehicles environment, which provides support for practical applications.

6 Conclusion

To improve the unloading efficiency and overall performance of task vehicles, a strategy analysis of mobile EC based on EC-ANN in TVCU was proposed. The resource allocation of task vehicles in collaborative unloading was optimized through MCTS and DNN modules. Secondly, the study proposed a multi-mode joint VCU mechanism, which expanded the service range of VCU by unloading tasks or computing results through RSU relay. Compared with random search, greedy search, DQN, and DNN, the EC-ANN-based solution improved service latency performance by 58.3%, 47.1%, 41.2%, and 39.8%, respectively. Compared to the V2V+V2X collaborative unloading mechanism, the multi-mode joint unloading mechanism offloaded tasks to service vehicles outside the V2V communication range, reducing completion latency by about 33.6%. Therefore, this strategy can effectively elevate task vehicles' unloading efficiency and overall performance, providing an effective solution for TVCU in vehicle networking environments. The limitation of this study is that it only used data from specific scenarios for performance analysis. Future research can expand the scenarios, explore the unloading mechanism of EC-ANN in different scenarios, and optimize it.

References

- [1] J. Guo, W. Luo, B. Song, F. Yu, and X. Du, "Intelligence-sharing vehicular networks with mobile edge computing and spatiotemporal knowledge transfer," *IEEE Network*, vol. 34, no. 4, pp. 256-262, 2020. <https://doi.org/10.1109/MNET.001.1900512>
- [2] G. Cui, Q. He, F. Chen, Y. Zhang, H. Jin and Y. Yang, "Interference-aware game-theoretic device allocation for mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4001-4012, 2021. <https://doi.org/10.1109/TMC.2021.3064063>
- [3] Y. Luo, W. Ding, B. Zhang, W. Huang, C. Liu, "Optimization of bits allocation and path planning with trajectory constraint in UAV-enabled mobile edge computing system," *Chinese Journal of Aeronautics*, vol. 33, no. 10, pp. 2716-2727, 2020. <https://doi.org/10.1016/j.cja.2020.04.014>
- [4] J. Fang, Z. Zhang, and R. V. Cowlagi, "Decentralized route-planning for multi-vehicle teams to satisfy a subclass of linear temporal logic specifications," *Automatica*, vol. 140, no. 1, pp. 110228-110238, 2022. <https://doi.org/10.1016/j.automatica.2022.110228>
- [5] M. Hamdani, N. Sahli, N. Jabeur, and N. Khezami, "Agent-based approach for connected vehicles and smart road signs collaboration," *Computing and Informatics*, vol. 41, no. 1, pp. 376-396, 2022. https://doi.org/10.31577/cai_2022_1_376
- [6] H. Xu, J. Zhou, W. Wei, and B. Cheng, "Multiuser computation offloading for long-term sequential tasks in mobile edge computing environments," *Tsinghua Science and Technology*, vol. 28, no. 1, pp. 93-104, 2023. <https://doi.org/10.26599/TST.2021.9010087>
- [7] Y. Gao, H. Zhang, F. Yu, Y. Xia, and N. Shi, "Joint computation offloading and resource allocation for mobile-edge computing assisted ultra-dense networks," *Journal of Communications and Information Networks*, vol. 7, no. 1, pp. 96-106, 2022. <https://doi.org/10.23919/JCIN.2022.9745485>
- [8] M. Laroui, H. Khedher, A. C. Moussa, M. Hassine, A. Hossam, and E. K. Ahmed, "SO-MEC: Service offloading in virtual mobile edge computing using deep reinforcement learning," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 10, pp. 4211-4236, 2021. <https://doi.org/10.1002/ett.4211>
- [9] N. Aung, S. Dhelim, L. Chen, H. Ning, and L. Atzori, "Edge-Enabled Metaverse: The convergence of metaverse and mobile edge computing," *Tsinghua Science and Technology*, vol. 29, no. 3, pp. 795-805, 2024. <https://doi.org/10.26599/TST.2023.9010052>
- [10] Y. Pang, J. Wu, L. Chen, and M. Yao, "Energy balancing for multiple devices with multiple tasks in mobile edge computing," *Journal of Frontiers of Computer Science and Technology*, vol. 16, no. 2, pp. 480-488, 2022. <https://doi.org/10.3778/j.issn.1673-9418.2009072>
- [11] Z. Maand and S. Sun, "Research on vehicle-to-road collaboration and end-to-end collaboration for multimedia services in the Internet of Vehicles," *IEEE Access*, vol. 10, no. 1, pp. 18146-18155, 2022. <https://doi.org/10.1109/ACCESS.2021.3112963>
- [12] T. Sun, Y. Xu, L. Feng, B. Xu, D. Chen, F. Zhang, X. Han, L. Zhao, and Y. Zheng, "A vehicle-cloud collaboration strategy for remaining driving range estimation based on online traffic route information and future operation condition prediction," *Energy*, vol. 248, no. 1, pp. 123608-123618, 2022.

- <https://doi.org/10.1016/j.energy.2022.123608>
- [13] H. Li, J. Zhang, Y. Li, Z. Huang, and H. Cao, "Modeling and simulation of vehicle group collaboration behaviors in an on-ramp area with a connected vehicle environment," *Simulation Modelling Practice and Theory*, vol. 110, no. 1, pp. 102332-102351, 2021. <https://doi.org/10.1016/j.simpat.2021.102332>
- [14] Y. Shi, Q. Han, W. Shen, and X. Wang, "A multi-layer collaboration framework for industrial parks with 5G vehicle-to-everything networks," *Engineering*, vol. 7, no. 3, pp. 818-831, 2021. <https://doi.org/10.1016/j.eng.2020.12.021>
- [15] H. Li, H. Wu, I. Gulati, S. Ali, V. Pickert, and S. Dlay, "An improved sliding mode control (SMC) approach for enhancement of communication delay in vehicle platoon system," *IET Intelligent Transport Systems*, vol. 16, no. 7, pp. 958-970, 2022. <https://doi.org/10.1049/itr2.12189>
- [16] C. Zhao, Y. Cai, A. Liu, M. Zhao, and L. Hanzo, "Mobile edge computing meets mm Wave communications: Joint beam forming and resource allocation for system delay minimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2382-2396, 2020. <https://doi.org/10.1109/TWC.2020.2964543>
- [17] L. Huang, L. Zhang, S. Yang, L. Qian, and Y. Wu, "Meta-learning based dynamic computation task offloading for mobile edge computing networks," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1568-1572, 2020. <https://doi.org/10.1109/LCOMM.2020.3048075>
- [18] U. Saleem, Y. Liu, S. Jangsher, Y. Li, and T. Jiang, "Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 360-374, 2020. <https://doi.org/10.1109/TWC.2020.3024538>
- [19] J. Feng and H. Zhao, "Dynamic nodes collaboration for target tracking in wireless sensor networks," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 21069-21079, 2021. <https://doi.org/10.1109/JSEN.2021.3093473>
- [20] A. S. Hashim, W. A. Awadh, and M. S. Hashim, "Non-dominated sorting genetic optimization-based fog cloudlet computing for wireless metropolitan area networks," *Informatica*, vol. 47, no. 10, pp. 1-8, 2023. <https://doi.org/10.31449/inf.v47i10.5118>
- [21] A. R. Mahlous, "Threat model and risk management for a smart home IoT system," *Informatica*, vol. 47, no. 1, pp. 51-64, 2023. <https://doi.org/10.31449/inf.v47i1.4526>

