Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 3, 2026

Research on Edge-Cloud Collaborative Resource Scheduling and Security Management Based on Intelligent Optimization and Privacy Protection

Tianyu Luo *

School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei, China
*** Corresponding author Email:** lty7301@outlook.com

Abstract: With the rapid development of edge computing, cloud computing, big data, artificial intelligence and the Internet of Things, traditional centralized cloud computing faces increasing challenges in latency, bandwidth pressure, resource utilization and data privacy protection. Edge-cloud collaboration provides a new computing paradigm by extending computing, storage and network resources from centralized cloud data centers to edge nodes closer to users and data sources. This architecture can improve service response efficiency and reduce data transmission pressure, but it also introduces heterogeneous resources, dynamic task requests, unclear security boundaries and privacy leakage risks. This paper reviews existing studies on edge-cloud collaboration, resource scheduling, intelligent optimization and privacy protection, and then conducts an analytical discussion of the research gaps rather than an experimental evaluation. The review shows that existing studies have achieved valuable progress in task offloading, resource allocation, deep reinforcement learning, access control, encryption and federated learning. However, research on integrated frameworks that combine intelligent resource scheduling with privacy-aware security management remains limited. To address this limitation, this paper proposes a formalized privacy-aware scheduling perspective that incorporates latency, energy consumption, cost, node trustworthiness, data sensitivity, privacy leakage risk, reliability and auditability into a unified decision model. The analysis indicates that future edge-cloud systems should evolve from efficiency-centric scheduling toward secure, trustworthy and sustainable collaborative governance.

Keywords: Edge-cloud Collaboration; Resource Scheduling; Intelligent Optimization; Privacy Protection; Security Management; Edge Computing.

1. Introduction

In recent years, the rapid growth of the Internet of Things, artificial intelligence, mobile applications and big data services has changed how data are generated, transmitted and processed. In traditional cloud computing architecture, most computing and storage tasks are concentrated in remote cloud data centers. This centralized model offers strong computing power and unified resource management, but it also creates limitations when facing real-time, large-scale and distributed applications. Smart transportation, industrial Internet, smart healthcare, autonomous vehicles and financial technology require low latency, high reliability and strong privacy protection. If all data are transmitted to remote cloud centers for processing, the system may face high communication latency, bandwidth congestion, excessive energy consumption and privacy exposure.

Edge computing emerged as an important response to these limitations. By deploying computing, storage and network resources close to users or data sources, edge computing can process part of the data locally or near the terminal side. This reduces the pressure on cloud data centers and improves service responsiveness. However, edge computing does not simply replace cloud computing. The current development trend is edge-cloud collaboration, in which cloud centers, edge nodes and terminal devices work together to complete complex tasks. In this collaborative architecture, the cloud is responsible for large-scale computing, global optimization and long-term storage, while edge nodes handle latency-sensitive tasks, local data processing and preliminary decision-making[1][2].

Although edge-cloud collaboration improves system efficiency, it also makes resource management more complex. Edge nodes are heterogeneous in computing capability, storage capacity, network condition and security level. Tasks generated by users and devices are dynamic, diverse and uncertain. Therefore, allocating computing, storage, bandwidth and energy resources among cloud centers, edge nodes and terminal devices has become a key research problem. Existing studies have proposed mathematical optimization, heuristic algorithms, reinforcement learning and deep reinforcement learning to improve resource scheduling and task offloading performance [3, 4, 5, 6].

At the same time, edge-cloud systems face serious security and privacy challenges. Data may pass through collection, transmission, processing, storage, sharing and deletion. Sensitive information, such as medical records, financial transactions, location traces and industrial control data, may be exposed during these processes. Compared with traditional cloud computing, edge-cloud environments contain more distributed nodes, more complex access relationships and weaker physical protection at the edge side[7][8][9]. Therefore, security and privacy protection should be considered when scheduling decisions are made, rather than being treated only as post-processing measures.

The purpose of this paper is to review and analyze studies on edge-cloud collaborative resource scheduling and security management based on intelligent optimization and privacy protection. Compared with a purely descriptive review, this paper emphasizes analytical comparison across research directions, identifies gaps in the integration of performance and privacy objectives, and proposes a formalized scheduling

perspective for future research.

2. Literature Review

This section changes the literature review from a simple listing of studies to an analytical comparison. For each subdirection, the review summarizes research objectives, technical routes, application scenarios, evaluation indicators and limitations. This structure makes it clearer how existing work contributes to edge-cloud collaboration and where gaps remain.

2.1. Edge Computing and Edge-Cloud Collaboration

Edge computing has become one of the most important computing paradigms after cloud computing. Shi et al. [1] systematically explained the concept, vision and challenges of edge computing, arguing that computing resources should be moved closer to data sources to reduce latency, save bandwidth and improve privacy protection. Mao et al. [2] reviewed mobile edge computing from the communication perspective and showed that deploying computing resources at base stations or access points can support computing-intensive and latency-sensitive applications.

In edge-cloud collaboration, cloud centers and edge nodes are not isolated. They form a multi-layer architecture consisting of terminal devices, edge servers and cloud data centers. Terminal devices generate tasks and data; edge nodes provide nearby computing and temporary storage; cloud centers provide global computing power, centralized coordination and long-term storage. This structure is suitable for applications that require both real-time response and large-scale data processing.

However, this architecture also brings new challenges. Edge nodes have limited resources compared with cloud centers; terminal devices generate highly dynamic requests; different edge nodes may belong to different organizations or trust domains; and data may move across multiple nodes. These features require resource scheduling, trust management, access control and privacy-aware task allocation to be considered together rather than separately.

Table 1. Analytical Comparison of Edge Computing and Edge-Cloud Collaboration Studies

Study	Main objective	Technical route	Scenario or focus	Key indicators	Limitations
Shi et al. [1]	Define edge computing vision and challenges	Conceptual architecture and challenge analysis	IoT and distributed data processing	Latency, bandwidth, privacy, scalability	Provides a broad vision but limited formal scheduling model
Mao et al. [2]	Review mobile edge computing from communication perspective	Survey of computation offloading and radio-resource management	Mobile networks, base stations and access points	Latency, energy, communication cost	Security and privacy are not fully integrated with scheduling
Edge-cloud collaboration studies	Coordinate cloud, edge and terminal resources	Layered architecture and collaborative processing	Real-time applications with cloud-side analytics	Response time, throughput, resource utilization	Heterogeneous trust domains and data life-cycle risks require deeper analysis

2.2. Resource Scheduling and Task Offloading

Resource scheduling and task offloading are core issues in edge-cloud systems. Task offloading refers to deciding whether a task should be processed locally, at an edge node or in the cloud. Resource scheduling further involves allocating computing power, storage space, bandwidth and energy resources to different tasks. Feng et al. [3] reviewed computation offloading in mobile edge computing networks and summarized major optimization goals, including latency minimization, energy consumption reduction, cost control and service quality improvement. Dong et al. [4] also pointed out that task offloading is essential for improving the performance of mobile edge computing systems, but the problem is computationally challenging because edge resources are limited and user requests are dynamic.

Traditional scheduling methods include first-come-first-served, round-robin, shortest job first, genetic algorithms, particle swarm optimization, ant colony optimization and simulated annealing. These methods are easy to implement and may perform well in relatively stable environments. However, edge-cloud environments are dynamic and uncertain. Network conditions, server loads, user locations and task priorities may change frequently. Therefore, static algorithms may not achieve stable performance in real-world edge-cloud scenarios.

In recent years, reinforcement learning and deep reinforcement learning have been widely applied to resource scheduling. Zhou et al. [5] reviewed deep reinforcement learning-based methods for cloud resource scheduling and showed that deep reinforcement learning is suitable for high-dimensional, dynamic and complex scheduling environments. Ismail et al. [6] further discussed resource scheduling in multi-access edge computing from a deep reinforcement learning perspective. In such models, system states may include task queue length, server load, network bandwidth and energy consumption, while actions may include local execution, edge offloading, cloud offloading or resource migration. The reward function usually combines latency, energy consumption, cost and task completion rate.

Multi-agent reinforcement learning has also attracted attention. In large-scale edge-cloud systems, centralized scheduling may cause high communication overhead and single-point failure. Multi-agent methods allow edge nodes or servers to make distributed decisions based on local information. This approach is more suitable for distributed edge environments, but it also introduces problems such as agent coordination, communication cost and convergence stability [10].

2.3. Intelligent Optimization Methods

Intelligent optimization provides new possibilities for solving complex resource scheduling problems. Traditional mathematical optimization methods usually require clear objective functions and constraints. However, edge-cloud systems are highly dynamic, and many variables are difficult to model precisely. Intelligent optimization methods, especially deep reinforcement learning, can learn scheduling strategies through interaction with the environment.

Table 2. Analytical Comparison of Resource Scheduling and Task Offloading Methods

Method type	Objective function	Technical route	Scenario or dataset	Key metrics	Main limitations
Rule-based scheduling	Minimize waiting time or balance task queues	FCFS, round-robin, shortest job first	Stable and small-scale systems	Completion time, queue length	Weak adaptability to dynamic network and workload changes
Meta-heuristic algorithms	Minimize weighted latency, energy and cost	Genetic algorithm, PSO, ant colony, simulated annealing	Simulated edge-cloud task allocation [3]	Latency, energy, resource utilization	Search cost may be high; real-time response is difficult in large systems
Single-agent RL/DRL	Maximize long-term reward combining latency, energy and cost	DQN, DDQN, DDPG, PPO [5][6]	Cloud and MEC scheduling environments	Reward, task success rate, average latency	Training instability, sample cost and limited explainability
Multi-agent RL	Optimize distributed decisions under local information	Independent or cooperative agents [10]	Multi-user edge computing and distributed edge nodes	Convergence, communication overhead, fairness	Coordination and convergence become more complex as the system scales

Deep reinforcement learning combines deep neural networks with reinforcement learning. Algorithms such as DQN, DDQN, DDPG and PPO have been used in cloud computing and edge computing scheduling problems [5] [6] [11] [10]. These algorithms can optimize task offloading, virtual machine allocation, energy management and service migration. Their advantage is adaptability to high-dimensional states, but their performance depends heavily on reward design, training stability and the realism of the simulation environment.

Federated learning is another important intelligent method related to privacy protection. McMahan et al. [12] proposed a communication-efficient learning method from decentralized data, which became a foundation of federated learning. In federated learning, raw data remain on local devices, while only model updates are transmitted. In edge-cloud environments, federated learning can support distributed model training while reducing the need to upload sensitive raw data [13].

However, intelligent optimization methods also have limitations. Deep reinforcement learning usually requires many training samples and repeated interactions with the environment. The training process may be costly and unstable. The decision-making process of deep models is often difficult to explain, which may reduce trust in security-sensitive scenarios. Federated learning also faces communication overhead, non-independent and identically distributed data, model poisoning attacks and privacy inference attacks.

Table 3. Analytical Comparison of Intelligent Optimization Methods

Method	Optimization target	Technical route	Typical scenario	Evaluation indicators	Limitations
DQN/DDQN	Discrete offloading or allocation decisions	Value-function learning with deep neural networks [5]	Task offloading among local, edge and cloud nodes	Reward, latency, energy, convergence	Difficult to handle continuous actions without approximation
DDPG/PPO	Continuous resource allocation and policy optimization	Actor-critic or policy-gradient learning [6]	CPU frequency allocation, bandwidth allocation and service migration	Long-term reward, stability, resource utilization	Sensitive to hyperparameters and training environment
Federated learning	Collaborative model training with local data retention	Local training and model aggregation [12] [13]	Edge intelligence and privacy-preserving learning	Accuracy, communication cost, privacy risk	Non-IID data, model poisoning and inference attacks remain challenges
Hybrid intelligent optimization	Balance global search and adaptive learning	Combine heuristics, RL and domain rules [11][10]	Dynamic edge-cloud scheduling	Latency, energy, fairness, robustness	Model complexity and deployment cost may increase

2.4. Security and Privacy Protection

Security and privacy protection are essential in edge-cloud systems. Traditional cloud security focuses on data encryption, access control, identity authentication, key management, intrusion detection and audit logging. These mechanisms remain important, but they are not sufficient for edge-cloud environments. Edge nodes are widely distributed and may be deployed in open or semi-trusted environments. They may face physical attacks, unauthorized access, node impersonation and data leakage[8][9].

NIST-related studies show that as data processing moves from the cloud to the edge, new privacy problems appear in edge systems. Data generated by IoT devices may continuously reveal user behavior, location, identity and preferences[7]. Zero Trust Architecture emphasizes that no user, device or network location should be trusted by default. Access decisions should be based on continuous verification, least privilege and dynamic policy enforcement[14]. This idea is highly relevant to edge-cloud systems because cloud, edge and terminal devices often belong to different trust domains.

Privacy-preserving task offloading has become an important research direction. Li et al. [15] summarized privacy-preserving offloading methods in mobile edge computing and divided them into categories such as offloading data management, secure transmission and target selection. This indicates that privacy protection should not only occur during data transmission but also during task partitioning, node selection and execution.

Other privacy protection technologies include differential

privacy, trusted execution environments, blockchain-based audit, anonymous authentication and secure multi-party computation. Differential privacy can reduce the risk of identifying individuals from released statistics or model outputs, although it may reduce model accuracy or query-result accuracy. Trusted execution environments can protect data during processing. Blockchain can provide traceable and tamper-resistant audit records. ENISA [16] also emphasizes cloud security governance, which is relevant when edge-cloud systems span multiple organizations. However, these mechanisms often introduce additional computing, communication and storage overhead. Therefore, privacy protection must be coordinated with resource scheduling.

Table 4. Analytical Comparison of Security and Privacy Protection Methods

Mechanism	Protection objective	Technical route	Applicable stage	Key indicators	Limitations
Encryption and secure transmission	Protect confidentiality during data transfer	TLS, symmetric/asymmetric encryption, key management	Transmission and storage [8] [9]	Encryption overhead, confidentiality, key security	Computation and latency overhead increase under resource-constrained edge nodes
Access control and zero trust	Prevent unauthorized access and privilege abuse	Continuous verification, least privilege and policy enforcement [14]	Collection, processing, sharing	Authentication strength, access denial rate, policy compliance	Policy design and real-time enforcement may be complex
Privacy-preserving offloading	Reduce privacy leakage during node selection and task execution	Sensitive-data classification, secure target selection and privacy-aware offloading [15]	Offloading and execution	Privacy risk, latency, node trust	Privacy metrics are often not formalized with scheduling objectives
Federated learning and differential privacy	Reduce exposure of raw data and inference risk	Local training, aggregation and noise addition [12][13]	Model training and analytics	Model accuracy, communication cost, privacy budget	Non-IID data, poisoning attacks and accuracy-privacy trade-offs
Audit and governance mechanisms	Improve accountability and compliance	Audit logging, blockchain records and compliance checks [16]	Storage, sharing and deletion	Traceability, log completeness, compliance records	Storage overhead and governance rules require standardization

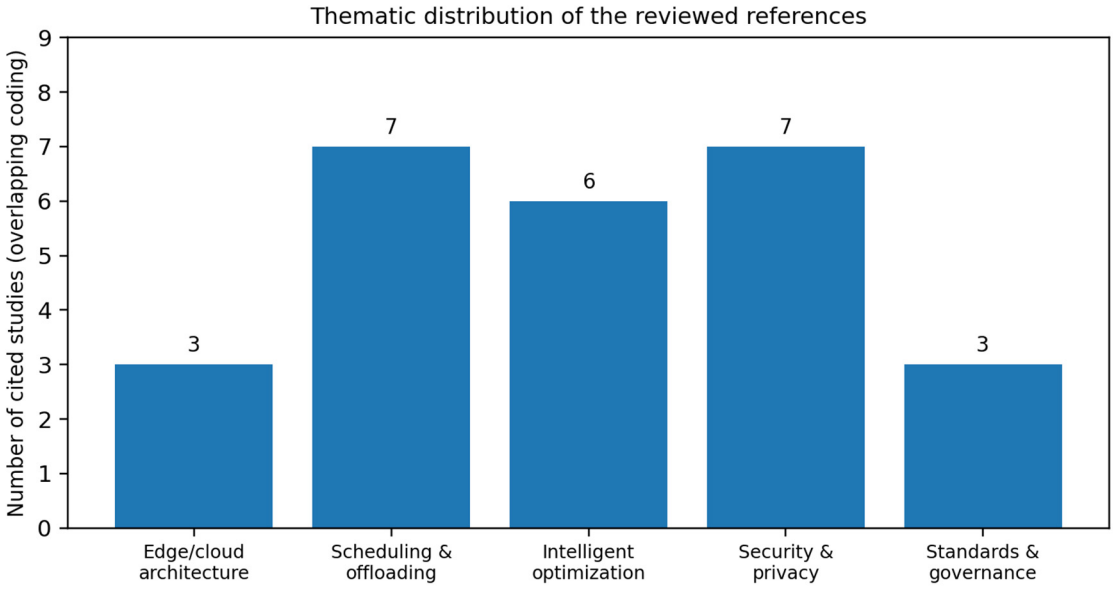

Figure 1. Thematic Distribution of the Reviewed References

Figure 1 is based on the thematic coding of the 16 cited references. Because several studies cover more than one theme, the counts are overlapping rather than mutually exclusive. The figure shows that the reviewed literature is concentrated on scheduling/offloading, intelligent optimization and security/privacy, which supports the need to analyze how these directions can be integrated.

3. Analysis and Discussion: Research Gaps and Integrated Framework

Based on the comparative review above, this section analyzes the main research gaps in edge-cloud collaborative scheduling and security management. The discussion no longer treats the framework as a purely conceptual statement. Instead, it defines decision variables, indicators, constraints and information flows that can support executable scheduling decisions.

3.1. Separation Between Resource Scheduling and Privacy Protection

Although existing studies have made significant progress in edge-cloud resource scheduling and privacy protection, the two research areas are still not fully integrated. Most resource

scheduling studies focus on performance-oriented indicators such as latency, energy consumption, cost, load balancing and task completion rate. In contrast, privacy protection studies usually focus on encryption, access control, identity authentication, federated learning and differential privacy. These two directions are often discussed separately.

This separation may lead to practical problems. In real edge-cloud systems, the node with the best computing performance is not necessarily the safest node. A nearby edge server may reduce task latency, but it may have a lower security level or weaker access control. Similarly, cloud data centers may provide stronger computing power, but uploading sensitive data to the cloud may increase privacy exposure during transmission. Therefore, resource scheduling should determine not only where tasks can be completed most efficiently, but also where they can be processed most securely and reliably.

Table 5. Comparison Between Traditional Scheduling and Privacy-Aware Scheduling

Aspect	Traditional resource scheduling	Privacy-aware resource scheduling
Main objective	Reduce latency, energy consumption and cost	Balance efficiency, security and privacy
Main indicators	Task completion time, bandwidth, energy and resource utilization	Data sensitivity, node trust, privacy risk and access permission
Scheduling logic	Select the fastest or lowest-cost node	Select the most suitable node under both performance and security constraints
Security assumption	Nodes are often assumed to be trusted	Nodes are evaluated according to dynamic trust and risk level
Main limitation	Privacy protection is weak or ignored	Model complexity and security overhead may increase

As shown in Table 5, traditional resource scheduling is mainly efficiency-oriented, while privacy-aware scheduling introduces security and privacy factors into the decision-making process. This means that privacy protection should become an internal variable of the resource scheduling model rather than an additional module after task allocation.

3.2. Need for Data Life-Cycle Security Management

Another limitation of existing research is the lack of a full life-cycle view of data security. Many studies focus on a single stage, such as secure transmission, encrypted storage or privacy-preserving model training. However, in edge-cloud environments, data usually experience multiple stages, including collection, transmission, processing, storage, sharing and deletion.

Each stage has different security risks and protection requirements. During data collection, device identity and data source authenticity should be verified. During data transmission, encryption and secure communication channels are necessary. During task processing, the system should evaluate whether the selected node is trustworthy. During data storage, access control and audit logging are required. During data sharing, anonymization, differential privacy or blockchain-based audit may be used. During data deletion, secure erasure and compliance verification should be considered.

Life-cycle security management should therefore be linked with scheduling. Before a task is offloaded, the system should identify the sensitivity level of the data. During execution, the trust level of the selected edge node should be monitored. After execution, the system should decide whether the data should be stored, encrypted, shared or deleted. This transforms security management from a static defense mechanism into a continuous decision process.

3.3. Balance Between Efficiency and Security

The core contradiction in edge-cloud resource scheduling and security management is the balance between efficiency and security. Efficiency-oriented scheduling aims to reduce latency, energy consumption and cost. Security-oriented management aims to reduce privacy leakage, unauthorized access and system risks. However, security mechanisms usually introduce additional overhead. Encryption increases computing cost, differential privacy may reduce model accuracy or query-result accuracy, federated learning increases communication cost, and trusted execution environments may affect execution efficiency.

Therefore, the goal should not be to maximize security at any cost or maximize efficiency while ignoring privacy. The system should seek a dynamic balance between efficiency and security according to application scenarios. For example, smart healthcare may assign greater weight to privacy and auditability, while video analytics for traffic control may assign greater weight to latency and throughput under minimum security constraints.

Table 6. Trade-Off Between Efficiency and Security

Optimization dimension	Efficiency-oriented choice	Security-oriented choice	Possible conflict
Task execution location	Select the nearest or fastest node	Select the most trusted node	The nearest node may not be secure
Data transmission	Upload data directly for fast processing	Use encryption or anonymization	Security protection increases latency
Model training	Centralized training for higher efficiency	Federated learning for privacy protection	Federated learning increases communication cost
Data storage	Cache data at edge nodes for fast access	Encrypt or restrict storage	Secure storage may reduce access speed
Access management	Simplified access for convenience	Strict authentication and authorization	Strong access control may reduce flexibility

Table 6 shows that performance and security often have competing requirements. An effective edge-cloud scheduling model should include both performance indicators and security indicators, and the weights of these indicators should be adjusted according to task sensitivity, service-level agreements and compliance requirements.

3.4. Trustworthiness of Intelligent Optimization Algorithms

Intelligent optimization methods provide strong adaptability for edge-cloud resource scheduling. Deep reinforcement learning can learn scheduling strategies from dynamic environments, while multi-agent reinforcement learning can support distributed decision-making among multiple edge nodes. Federated learning can support collaborative model training without directly sharing raw data.

However, intelligent optimization methods also introduce new trustworthiness issues. First, deep reinforcement learning models are often difficult to explain. In security-sensitive scenarios, system administrators may need to know why a task is assigned to a specific node. Second, intelligent models may be affected by data poisoning, model inversion, adversarial attacks or malicious model updates. Third, if the reward function is poorly designed, the algorithm may learn strategies that improve efficiency but damage security.

Future research should therefore improve the trustworthiness of intelligent optimization algorithms. A privacy-aware reward function can penalize unsafe scheduling decisions. A dynamic trust evaluation mechanism can update the security level of edge nodes. Explainable artificial intelligence can help administrators understand the reasons behind scheduling decisions. Security verification can ensure that the learned strategy does not violate privacy or access-control requirements.

Table 7. Trustworthiness Issues and Improvement Directions of Intelligent Optimization

Issue	Specific problem	Improvement direction
Low explainability	The model cannot clearly explain why a task is assigned to a node	Introduce explainable AI and interpretable scheduling rules
Training instability	DRL may require many training samples and may converge slowly	Improve reward design and training stability
Security vulnerability	Models may suffer from poisoning, inversion or adversarial attacks	Add security verification and robust learning mechanisms
Privacy risk	Model updates may still reveal sensitive information	Combine federated learning with differential privacy and secure aggregation
Weak generalization	A model trained in simulation may perform poorly in real systems	Use real-world data, transfer learning and adaptive online learning

Table 7 indicates that intelligent optimization is not automatically trustworthy. To apply it in edge-cloud security management, algorithmic efficiency, model robustness, explainability and privacy protection need to be considered together.

3.5. Formal Privacy-Aware Scheduling Model and Integrated Framework

Based on the above analysis, future edge-cloud resource scheduling should move from a schedule-first-protect-later model to a secure-scheduling-by-design model. This subsection formalizes the key variables, objective function, constraints and module interfaces so that the framework can be translated into executable scheduling decisions.

Let T = {t1, t2, ..., tm} denote the task set and N = {n1, n2, ..., nk} denote candidate execution nodes, including terminal devices, edge nodes and cloud servers. For each task ti and node nj, define a binary scheduling variable xij. If xij = 1, task ti is assigned to node nj; otherwise xij = 0. The scheduling decision should minimize a weighted objective that combines performance cost, security risk and reliability.

$$\min J = w1\ L\_norm + w2\ E\_norm + w3\ C\_norm + w4\ P\_norm + w5(1 - T\_norm) + w6(1 - R\_norm)$$

In this objective function, L_norm denotes normalized latency, E_norm denotes normalized energy consumption, C_norm denotes normalized monetary or resource cost, P_norm denotes normalized privacy leakage risk, T_norm denotes normalized node trustworthiness, R_norm denotes normalized reliability, and w1 to w6 are scenario-dependent weights. For example, healthcare tasks may use a larger w4 and w5, while real-time video analytics may use a larger w1.

The model should satisfy the following core constraints:

- Assignment constraint: each task is assigned to exactly one feasible execution location, expressed as sum_j xij = 1 for every task ti.
- Resource constraint: the total CPU, storage and bandwidth allocated to tasks on node nj cannot exceed the available resources of that node.
- Latency constraint: for latency-sensitive tasks, the estimated latency Lij must not exceed the maximum tolerable latency Li_max.
- Security constraint: a task containing sensitive data can be assigned only to nodes whose trust level and access-control capability satisfy the required threshold.
- Privacy-risk constraint: the estimated privacy risk Pij must not exceed the maximum acceptable privacy risk Pi_max for task ti.
- Reliability constraint: the predicted task success probability or service availability must satisfy the minimum reliability requirement.

The information flow of the framework is as follows. The task classification module receives task type, data size, computing demand, latency requirement and data sensitivity as input, and outputs task priority and security requirement. The resource perception module receives node load, CPU availability, storage, bandwidth, energy status and failure history, and outputs a dynamic resource-state vector. The trust and privacy evaluation module receives node identity, access-control status, audit history, vulnerability score and data sensitivity, and outputs node trust and privacy-risk scores. The intelligent scheduling module receives performance, resource, trust and privacy vectors, and outputs the selected execution node and resource allocation plan. The life-cycle security management module receives the scheduling result and outputs encryption, access-control, logging, storage, sharing and deletion policies.

Table 8. Definitions and Calculation Methods of Key Indicators

Indicator type	Indicator	Possible calculation method	Function in scheduling
Performance	Latency Lij	Lij = transmission delay + queueing delay + execution time	Measure service response efficiency
Resource	Energy Eij	Eij = local computing energy + transmission energy + edge/cloud execution energy	Evaluate resource and battery cost
Cost	Resource cost Cij	Weighted cost of CPU, storage, bandwidth and cloud service price	Control operational expenditure
Security	Node trust Tj	Tj = a*authentication score + b*historical success rate + c*(1 - vulnerability score) + d*compliance score	Measure whether a node is suitable for sensitive tasks
Privacy	Privacy leakage risk Pij	Pij = data sensitivity Si * exposure level Epi * (1 - Tj) * sharing scope Gij	Quantify privacy risk during offloading and processing
Reliability	Reliability Rij	Rij = predicted success probability based on failure rate, availability and load stability	Reduce task failure and service interruption
Auditability	Audit completeness Aij	Aij = ratio of complete logs, traceable operations and compliance records	Support accountability and governance

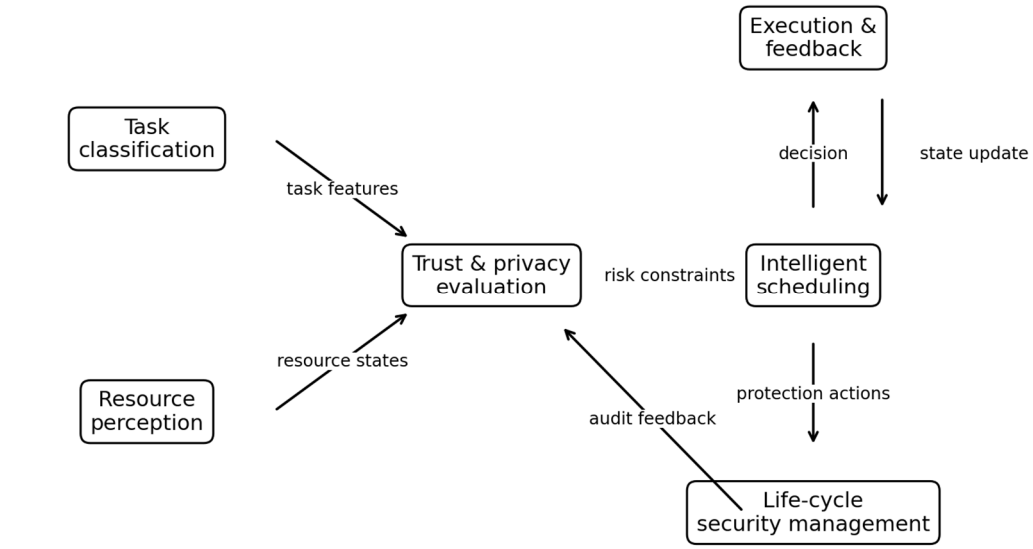

Figure 2. Integrated Framework for Privacy-Aware Edge-Cloud Resource Scheduling

Figure 2 is retained as the single core conceptual framework figure. Unlike the earlier multiple schematic figures, this framework is connected to the formal model above. Its input variables, output decisions and feedback paths can be implemented through rule-based optimization, reinforcement learning or multi-agent reinforcement learning. In an RL-based implementation, the state can include task features, resource states and trust scores; the action can be the selected node and resource allocation; and the reward can be the negative value of the objective function J. In a rule-based or mixed-integer optimization implementation, the same objective and constraints can be solved directly or approximated by heuristic search.

4. Conclusion

Edge-cloud collaboration is an important computing paradigm for supporting modern intelligent applications. It can reduce latency, improve resource utilization and support real-time services by combining the strengths of cloud centers and edge nodes. However, the distributed and heterogeneous nature of edge-cloud systems also creates challenges in resource scheduling, data security and privacy protection.

This paper reviewed existing studies on edge computing, resource scheduling, intelligent optimization and privacy protection. The literature shows that current research has achieved valuable results in task offloading, deep reinforcement learning-based scheduling, federated learning, access control, encryption and zero trust security. The revised review further compares the objective functions, technical routes, scenarios, indicators and limitations of different research directions. The analysis shows that most studies still focus on separate problems. Resource scheduling research is often performance-oriented, while privacy protection research is often treated as an independent security mechanism. The integration between intelligent optimization and privacy-aware security management remains insufficient.

Future research should develop unified scheduling models that combine resource scheduling, intelligent optimization, privacy protection and data life-cycle security management. Such models should consider latency, energy consumption, cost, resource utilization, node trustworthiness, data sensitivity, privacy leakage risk, reliability and auditability at the same time. They should also improve the explainability, robustness and trustworthiness of intelligent algorithms. Through this integration, edge-cloud systems can evolve from efficiency-centric architectures toward secure, trustworthy and sustainable collaborative systems.

References

[1] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646. https://doi.org/10. 1109/JIOT. 2016. 2579198.

[2] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322-2358. https://doi.org/10.1109/COMST.2017.2723187.

[3] Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y., & Guo, L. (2022). Computation offloading in mobile edge computing networks: A survey. Journal of Network and Computer Applications, 202, 103366. https://doi.org/10. 1016/j. jnca. 2022. 103366.

[4] Dong, S., Xia, Y., & Peng, T. (2024). Task offloading strategies for mobile edge computing: A survey. Computer Networks.

[5] Zhou, G., Tian, W., Buyya, R., Xue, R., & Song, L. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. Artificial Intelligence Review.

[6] Ismail, A. A., Abdel-Qader, B., Al-Akaidi, M., & Aljohani, A. J. (2025). A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study. Cluster Computing.

[7] Kotevska, O., Johnson, R., Chandersekaran, C., & Simpson, S. (2022). Analyzing data privacy for edge systems. National Institute of Standards and Technology.

[8] Zhang, J., Zhao, Y., Chen, B., Hu, F., & Zhu, K. (2018). A survey of data security and privacy protection in edge computing. Journal on Communications, 39(3).

[9] Li, X., Chen, B., Yang, D., & Wu, G. (2022). A survey of security protocols in edge computing environments. Journal of Computer Research and Development, 59(4), 765-780.

[10] Kuang, Z., Chen, Q., Li, L., Deng, X., & Chen, Z. (2022). A multi-user edge computing task offloading scheduling and resource allocation algorithm based on deep reinforcement learning. Chinese Journal of Computers, 45(4), 812-824.

[11] Li, Q., & Peng, X. (2022). Cloud job scheduling and simulation based on deep reinforcement learning. Journal of System Simulation, 34(2), 258-268.

[12] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of AISTATS.

[13] Zhang, X., Liu, Y., Liu, J., & Han, Y. (2023). A survey of federated learning for edge intelligence. Journal of Computer Research and Development, 60(6), 1276-1295.

[14] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology.

[15] Li, T., He, X., Jiang, S., Li, Q., & Zhang, Y. (2022). A survey of privacy-preserving offloading methods in mobile-edge computing. Journal of Network and Computer Applications, 203, 103395. https://doi.org/10.1016/j.jnca.2022.103395.

[16] European Union Agency for Cybersecurity. (2023). Cloud cybersecurity market analysis. ENISA.