



Federated learning for task offloading optimization: a systematic review in edge computing environments

Peng Kong^{1,2} · Kartinah Zen¹ · Sei Ping Lau³

Received: 6 June 2025 / Revised: 1 April 2026 / Accepted: 2 July 2026

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2026

Abstract

Federated learning (FL) has emerged as a promising method for reducing communication overhead and preserving data privacy in distributed systems. In edge computing, task offloading (TO) has garnered increasing attention to enhance efficiency, reduce communication costs, and protect user privacy. However, despite the growing adoption of FL, comprehensive reviews that systematically examine its variants for TO optimization remain limited. This paper addresses that gap by reviewing FL approaches applied to TO optimization from 2017 to 2025. We analyze major TO application scenarios, outline key optimization objectives, and discuss current challenges. Additionally, we examine FL methods and strategies designed explicitly for TO optimization and evaluate their effectiveness using performance metrics. The review also categorizes research methodologies based on model aggregation techniques, highlighting the strengths and limitations of different approaches. Finally, we identify emerging trends and propose directions for future research in this field.

Keywords Federated learning · Task offloading · Edge computing

Abbreviations

AC	Actor-Critic
AI	Artificial Intelligence
CS	Cloud Server
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
ED	Edge Device

ES	Edge Server
FL	Federated Learning
FRL	Federated Reinforcement Learning
HAB	High-altitude Balloon
HAP	High-Altitude platform
IoT	Internet of Things
IoV	Internet of Vehicles
ML	Machine Learning
PRISMA	The Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RL	Reinforcement Learning
RSU	Roadside Unit
TO	Task Offloading
UAV	Unmanned Aerial Vehicle
V2X	Vehicle-to-Everything
VEC	Vehicular Edge Computing
VFC	Vehicular Fog Computing

✉ Kartinah Zen
kartinah@unimas.my

✉ Sei Ping Lau
splau@unimas.my

Peng Kong
22010198@siswa.unimas.my

¹ Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak 94300, Kota Samarahan, Malaysia

² School of Information Engineering, Anhui Vocational College of Electronics and Information Technology, Bengbu 233000, China

³ Institute of Social Informatics and Technological Innovations, Universiti Malaysia Sarawak, Sarawak 94300, Kota Samarahan, Malaysia

1 Introduction

With the proliferation of smart devices and the rapid development of network technologies, the volume of data generated by network applications has grown exponentially. According to the International Data Corporation, by 2029,

global consumers and enterprises are projected to create 527.47 ZB of data annually, with nearly 57% of this data being generated in non-cloud environments [1]. This surge is accompanied by the widespread deployment of innovative applications leveraging edge devices (EDs) to deliver diverse network services. The massive processing demands of this data place substantial computational pressure on cloud servers (CSs). At the same time, rising user expectations in domains such as intelligent transportation, healthcare, and autonomous driving highlight the urgency of addressing task processing latency.

Edge computing has been proposed as a solution, bringing computation closer to end devices to reduce execution delays. Task offloading (TO) [2] is a critical component of this paradigm. Given the limited resources of EDs, offloading computation-intensive tasks to edge servers (ESs) with relatively greater resources can effectively reduce service latency, lower device energy consumption, and ensure service quality [3]. TO optimization primarily addresses three aspects: task allocation, resource allocation, and offloading strategies [4] to maximize computing resource utilization across EDs, ESs, and CSs, thereby improving both Quality of Experience and Quality of Service [5]. TO performance is

typically measured by minimizing latency, lowering energy consumption, or balancing both metrics [6].

In recent years, advances in Artificial Intelligence (AI), particularly Deep Learning (DL), have enabled the development of classic models such as convolutional neural networks, recurrent neural networks, long short-term memory networks, and deep neural networks. These models have been widely applied in vehicular networks [7–11], healthcare [12–16], smart cities [17–20], and intelligent logistics [21, 22]. Table 1 compiles the list of abbreviations used in this review.

FL was introduced in 2017 by McMahan et al. at Google [23] and is a distributed machine learning (ML) framework that enables multiple clients to collaboratively update a shared model without transmitting raw data. Optional techniques such as encryption or differential privacy are often used to enhance data confidentiality. FL trains a global model by aggregating locally trained parameters from participant devices, thereby preserving data locality while improving privacy and security. Initially designed for mobile devices, FL was later expanded by Yang et al. [24] into a broader privacy-preserving collaborative ML paradigm applicable across various domains.

Table 1 Contribution and Limitation of Existing Survey Papers

Reference	Contributions	Limitations
[33, 34]	These surveys review TO in edge–cloud environments and examines the application of control-theoretic optimization, mathematical methods, and artificial-intelligence approaches to TO.	The application of FL in TO has not yet been discussed.
[35]	This survey reviews the strategies, algorithms, and influencing factors of TO and resource allocation, with particular attention to two categories of algorithms: ML and non-ML approaches.	
[36]	This survey reviews various algorithms for TO and resource allocation from the perspective of energy optimization.	
[37]	This survey reviews the latest technologies for MEC TO in wireless power transfer scenarios, with particular emphasis on the classification of TO approaches that integrate MEC with wireless power transfer.	
[38]	This survey classifies vehicular TO from a communication perspective into three scenarios: vehicle-to-vehicle, vehicle-to-roadside infrastructure, and vehicle-to-everything.	
[39]	This survey reviews TO from the perspective of task types, focusing on its architectures, objectives, and strategies.	The application of FL in TO is not addressed. Only a few examples are provided to illustrate the use of FL for optimizing TO.
[40–42]	These surveys provide a comprehensive overview of FL in edge computing, covering communication, frameworks, and data privacy, and further discusses solution approaches for applying FL.	
[43, 46]	These surveys provide an overview of enhancing the practicality and efficiency of IoT systems through FL and introduces FL architectures and application potentials tailored for IoT.	
[45]	This survey reviews the enhancement of practicality and efficiency of Internet of Things (IoT) systems through FL and introduces FL architectures suitable for IoT as well as their application potential.	
[44]	This survey reviews the challenges of applying FL in edge networks, with a focus on communication, resource allocation, and data privacy.	
[47]	This survey reviews computational offloading and key resource issues of applying FL in Vehicular Edge Computing (VEC) and further discusses offloading frameworks, as well as security and privacy concerns related to VEC.	This survey is restricted to the scope of V2X, with an emphasis on the classification of computational offloading in VEC.

Researchers can modify parameters or strategies in the TO process and use performance metrics to evaluate optimization outcomes. However, TO optimization is typically formulated as a mixed integer non-linear programming problem, which is non-convex and Non-deterministic Polynomial-time hard (NP-hard) [25], prone to local optima, and requires complexity reduction. To date, three main approaches have been employed to address TO, namely classical algorithms, heuristic algorithms, and AI methods [3]. In edge environments, mobility and system dynamics introduce significant variability. The advantage of AI lies in its ability to learn these dynamic characteristics, build TO models, and predict new tasks, thereby providing adaptive solutions. Solving TO problems, therefore, requires careful consideration of application-specific requirements when selecting appropriate methods. Current research trends increasingly combine AI with conventional algorithms.

The distributed architecture and resource constraints of EDs make them highly compatible with FL's computational framework. FL further offers distinct advantages for optimizing TO: (1) Effective utilization of ED data. Training models under the FL framework enables collaborative learning across multiple EDs, thereby optimizing TO strategies. Examples include Federated Reinforcement Learning (FRL) [26] and Federated Deep Reinforcement Learning (DRL) [27]. (2) Reduced network transmission pressure. Instead of transmitting raw data, FL clients upload local model parameters to the server for global aggregation, reducing transmission delay and bandwidth overhead in edge environments [28, 29]. (3) Enhanced data privacy. By training models locally, FL avoids uploading raw data and reduces the risk of sensitive information leakage. Combined with encryption methods such as differential privacy and homomorphic encryption, privacy protection can be further strengthened [30–32].

Despite extensive studies on TO, there is a lack of systematic reviews focusing on the integration of FL within TO. To address this gap, the present study is guided by the following research questions:

RQ1 What are the main application scenarios for TO in edge computing environments?

RQ2 What problems are addressed by applying FL to TO?

RQ3 What methods and strategies have been proposed for FL-based TO optimization?

RQ4 What are the main performance metrics optimized by FL in TO?

RQ5 What unresolved issues and future research directions exist for applying FL to TO?

Building upon these research questions, the research objectives of this paper are structured to provide a comprehensive exploration of the key areas within the intersection of FL and TO in edge environments:

RO1 To analyze and categorize the main application scenarios of FL in TO in edge computing environments.

RO2 To investigate the key problems addressed by FL in TO.

RO3 To summarize and evaluate the methods and strategies of FL in TO.

RO4 To systematically identify and categorize the key performance metrics optimized by FL in TO.

RO5 To explore the challenges of applying FL in TO and propose future research directions.

1.1 Related works and survey comparison

TO has long been a central research focus in edge computing, with numerous surveys examining it from diverse perspectives. B. Wang et al. [33] and F. Saeik et al. [34] systematically analyzed taxonomies and solution approaches for TO in edge-cloud environments and established essential foundations for future research in edge contexts. A. Acheampong et al. [35] investigated TO and resource allocation algorithms, highlighting the role of ML. J. Yang et al. [36] classified TO strategies in mobile edge computing (MEC) from the perspective of energy optimization, while E. Mustafa et al. [37] reviewed TO techniques in MEC with a particular focus on wireless power transfer. Other studies explored TO from more specialized viewpoints. For example, M. Ahmed et al. [38] categorized TO in vehicular networks based on communication characteristics, and S. Zhang et al. [39] classified TO tasks by static versus

dynamic data requirements. Collectively, these works contribute valuable syntheses on taxonomies, energy efficiency, and application-specific scenarios. Nevertheless, they do not provide a systematic assessment of how FL can be applied to and adapted for TO optimization.

Parallel to these efforts, several surveys have investigated FL in edge computing more broadly. Q. Xia et al. [40] reviewed FL with attention to tool chains, communication paradigms, and implementation challenges. A. Brecko et al. [41] discussed FL architectures and their usage in edge settings, while M. Al-Quraan et al. [42] surveyed enabling technologies and FL-driven applications for sixth-generation networks. Similarly, M. Adam et al. [43] provided a comprehensive taxonomy of FL in IoT, including architectural designs and performance metrics. While these surveys delivered extensive insights into FL applications, they do not explicitly address FL for TO optimization.

A few studies explicitly mention FL in the context of TO. W. Y. B. Lim et al. [44] comprehensively reviewed FL applications in mobile edge networks; H. G. Abreha et al. [45] surveyed FL protocols and architectures for edge computing; and B. S. Diba et al. [46] examined FL in IoT with attention to secure data sharing, TO, and attack detection. Across these works, TO remains a secondary focus, discussed briefly in subsections without a structured synthesis of how FL supports TO. Among existing literature, only M. K. Hasan et al. [47] addressed this intersection comprehensively, classifying vehicular computational offloading research into problem-oriented and experimental categories. While this work emphasizes vehicular edge computing, it does not provide an in-depth analysis of the FL training process in offloading contexts.

Taken together, these findings point to a clear research gap. Existing surveys either examine TO without integrating FL or analyze FL in edge settings without addressing its role in TO optimization. Studies that consider FL for TO do so only partially, lacking systematic classification or critical synthesis. This fragmentation limits understanding of how FL advances TO and restricts its practical application in latency-sensitive edge environments. The contributions and limitations of existing surveys are shown in Table 1.

1.2 Contributions of this review

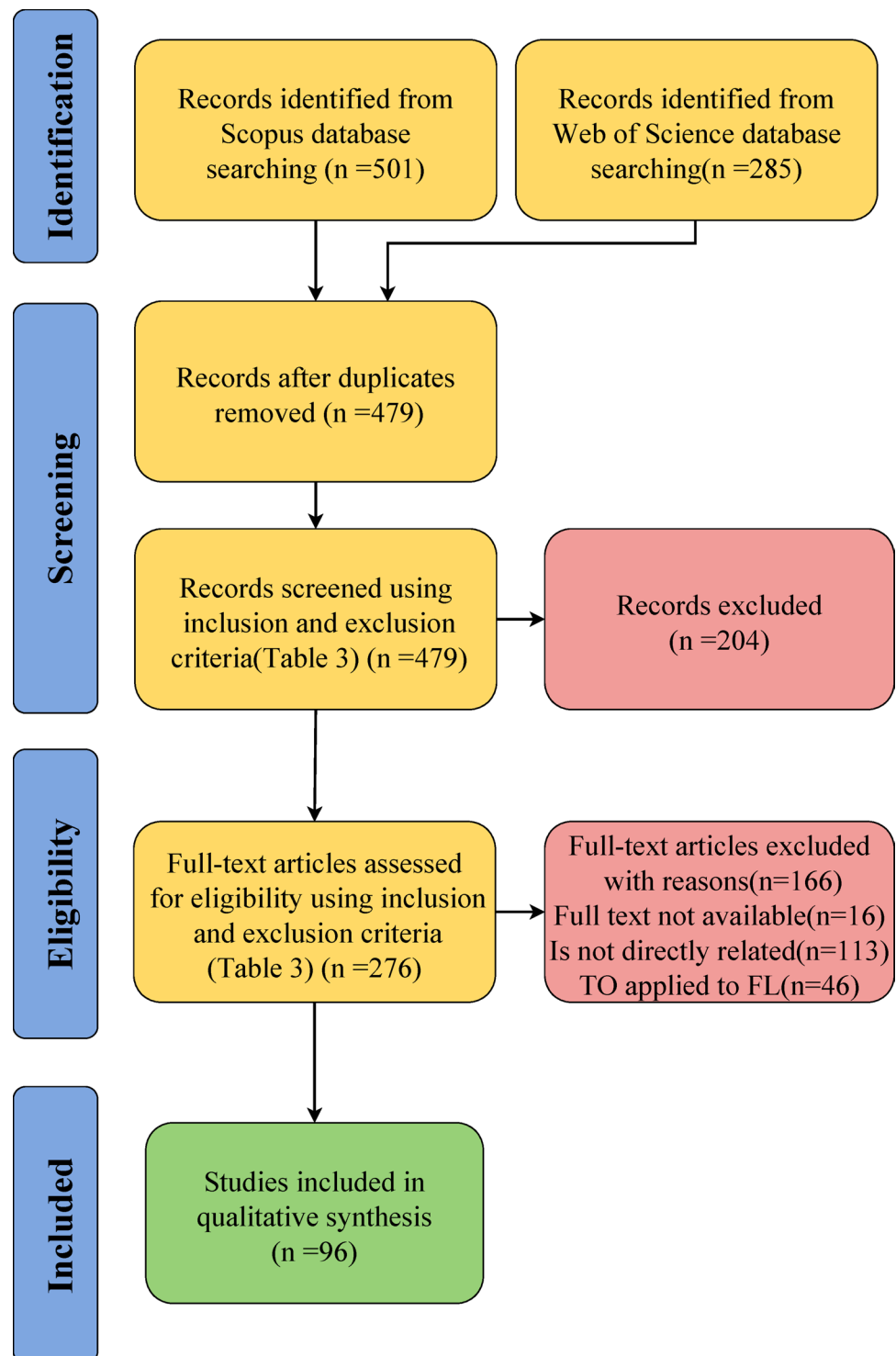
To address this gap, this paper provides a comprehensive review of recent advances in FL-based TO optimization in edge computing environments. It critically examines the convergence of FL and TO, identifies open problems, and highlights developing research directions. The key contributions of this work are as follows: (1) Comprehensive Categorization and Analysis of Application Scenarios. We

provide a thorough taxonomy and systematic analysis of the primary application scenarios for integrating FL and TO in edge computing environments. This contribution methodically delineates the application of FL across six major domains: General IoT, Transportation/Vehicles, Smart Cities, Industrial IoT, Smart Healthcare & Medicine, and Aerial and Space Edge, thereby establishing a clear contextual foundation for the field. (2) In-depth Examination of Core Problems Addressed by FL. We conduct a profound investigation to elucidate the core problems solved by FL within TO. The analysis systematically synthesizes how FL optimizes TO objectives across four key dimensions: collaborative training, privacy preservation, communication cost reduction, and computational load reduction. (3) Systematic Summary and Critical Assessment of Methodologies. We present a systematic synthesis and critical evaluation of state-of-the-art methods and optimization strategies for FL-based TO. This contribution reviews various techniques, including TO system architecture analysis, FL architecture design, client and server selection strategies, and local model training algorithms, providing a comparative analysis of their advantages, limitations, and applicability. (4) Systematic Identification and Classification of Key Performance Metrics. We systematically identify and categorize the key performance metrics optimized by FL in TO. Through a comprehensive analysis, we propose a set of five pivotal metrics; latency, energy consumption, cost, reward, and convergence performance, to evaluate the performance of integrated FL and TO systems. (5) Critical Discussion on Limitations and Future Research Directions. We critically highlight the limitations inherent in current research and, prospectively, outline a series of promising future research directions for the field.

The remainder of this paper is organized as follows: Section II presents the literature search strategy and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)-based screening process. Section III presents the review results in response to the four research questions. Section IV provides a critical discussion of the reviewed literature. Section V concludes the paper and outlines future research directions.

2 Systematic review methodology

To provide a comprehensive overview of the current state of applying FL for TO optimization in edge environments, this study follows PRISMA guidelines [48] and adopts a systematic literature review methodology. Previous studies have demonstrated that the PRISMA framework supports the development of more rigorous and transparent systematic reviews. Figure 1 presents the PRISMA flow diagram

Fig. 1 Literature Selection Process

used in this study, outlining the review process and detailing the steps of searching and screening relevant literature.

This methodology builds on the systematic review framework described in [49]. It begins with a clear definition of research objectives and questions, followed by the use of Boolean search statements to identify relevant academic publications across selected databases. The retrieved studies

are then screened and assessed for quality according to pre-defined inclusion and exclusion criteria. Finally, data from the eligible studies are extracted, analyzed, and synthesized to derive the research findings.

2.1 Data collection and processing

This study employed a systematic search strategy to identify relevant literature in response to the research questions. A Boolean search query was applied in two scientific databases, Web of Science and Scopus, using the following string: (“distributed machine learning” OR “distributed learning” OR “federated learning” OR “federated”) AND (“task offloading” OR “computation offloading” OR “computational offloading” OR “cyber foraging”) AND (“edge” OR “MEC” OR “mobile edge computing” OR “fog” OR “IoT” OR “internet of things”). The Boolean query consisted of three components:

- (i) The keywords “distributed machine learning” and “distributed learning” were included since FL is a type of distributed ML. The term “federated” was also added to capture studies integrating FL with other AI models.
- (ii) Computation offloading refers to the offloading of computationally intensive tasks [50]. As it is considered either a synonym [34] or a subset [2] of TO, the terms “computation offloading,” “computational offloading,” and “cyber foraging” were treated as parallel conditions.
- (iii) Edge computing contexts. Keywords such as “edge,” “MEC,” and “IoT” were included, reflecting the environments in which FL is typically applied. To ensure comprehensiveness, the search timeframe was set to begin in 2010. No relevant publications were found between 2010 and 2018, nor in 2020.

Searches were conducted in the article title, abstract, and keywords fields of both databases. To ensure quality, Web of Science searches were limited to the Web of Science Core Collection. The initial search in December 2025 yielded 786 papers in total (285 from Web of Science and 501 from Scopus).

Table 2 Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Journal articles	Non-journal papers (e.g., conference abstracts, book chapters)
Published in or after 2010	Published before 2010
The title, abstract, and keywords match the Boolean search query.	Articles with unclear descriptions of the experimental process
Written in English	Studies in which FL and TO are not directly related
–	Studies where TO is applied to FL (reverse direction)

2.2 Inclusion and exclusion criteria

After applying the inclusion and exclusion criteria outlined in Tables 2 and 96 papers were retained for review. The retrieved documents were evaluated against predefined inclusion and exclusion criteria to determine their eligibility for analysis.

The selection process followed the PRISMA guidelines shown in Fig. 1, which involved identification, screening, and eligibility assessment. After applying these criteria, 96 papers were retained for analysis.

2.3 Data extraction and analysis approach

To ensure the rigor and operational feasibility of the study, we developed a structured protocol for data extraction and integrative analysis. During the extraction stage, we designed a standardized extraction form that covers bibliographic metadata, research context, details of FL, details of TO, technological integration, evaluation methods, and research conclusions and limitations. In the categorization and analysis stage, the extracted information was systematically mapped to five research questions, guiding targeted analysis and interpretation. For application scenarios, we classified the literature according to primary and secondary scenario tiers; for the problems addressed by FL, we identified its roles by extracting FL-related information from the studies; for methods and strategies, we conducted groupwise comparisons by the types of FL and integrated algorithms; for performance metrics, we clarified the optimization core by examining evaluation indicators and baseline algorithms; for open issues and future directions, we systematically connected documented limitations and proposed future work to potential solution pathways.

For the presentation of results, the study utilizes summary tables and visual charts as its primary analytical tools. The summary tables include a cross-study feature comparison table and a challenge synopsis that directly responds to future directions. The visualizations use bar charts to display the distribution of optimization objectives, pie charts to present the proportion of algorithms used, and an integrative conceptual framework diagram to depict the complex interactions among design choices, offloading strategies, and performance outcomes. This structured approach ensures transparency and rigor throughout the entire process from data extraction to categorization and presentation.

3 Results

This section presents the results of data extraction and synthesis from the reviewed literature, including descriptive statistics and findings related to the research questions.

3.1 Publication trends

The annual distribution of publications is shown in Fig. 2. However, FL was first applied by McMahan et al. [23] in 2017 for mobile input method training; its application to TO did not appear until 2019. Since then, the number of publications has increased steadily through 2023, reflecting growing research interest. As of December 31, 2025, a total of 35 studies were included for the full year, representing a significant increase from 2024 and continuing the growth trend observed in previous years.

As shown in Table 3, among the total of 96 papers, those published in IEEE journals amount to 66, accounting for 68.0%. This distribution indicates that the core academic positioning of research integrating FL and TO is essentially a system engineering problem at the intersection of communication networks and mobile computing. In other words, when addressing distributed intelligent collaboration, researchers take bandwidth, latency, connection

dynamics, and signaling overhead as the primary optimization objectives.

From the perspective of the cross-domain composition of journal types, the research topics broadly cover areas such as the IoT, vehicular networks, mobile computing, industrial informatics, and cloud computing, confirming that FL and TO have transitioned from pure algorithmic exploration to widespread application in diverse scenarios, including smart cities, autonomous driving, the industrial IoT, and space–air–ground integrated networks. Meanwhile, the fact that open-access journals such as IEEE Access account for 14.4% of the publications suggests that there is a substantial body of engineering-oriented, rapidly verifiable applied results in this field. Interestingly, 50% of the journals published only one paper, indicating that the research area remains in a frontier, exploratory stage.

3.2 Application scenarios of TO

TO has been studied across a wide range of computing paradigms, but this review focuses specifically on edge environments. With the rapid growth of edge computing, digital twins, drones, and related technologies, TO has been applied in a diverse range of scenarios. These application scenarios involve technologies such as EDs, ESs, vehicles, drones, high-altitude balloons, and satellites. The taxonomy

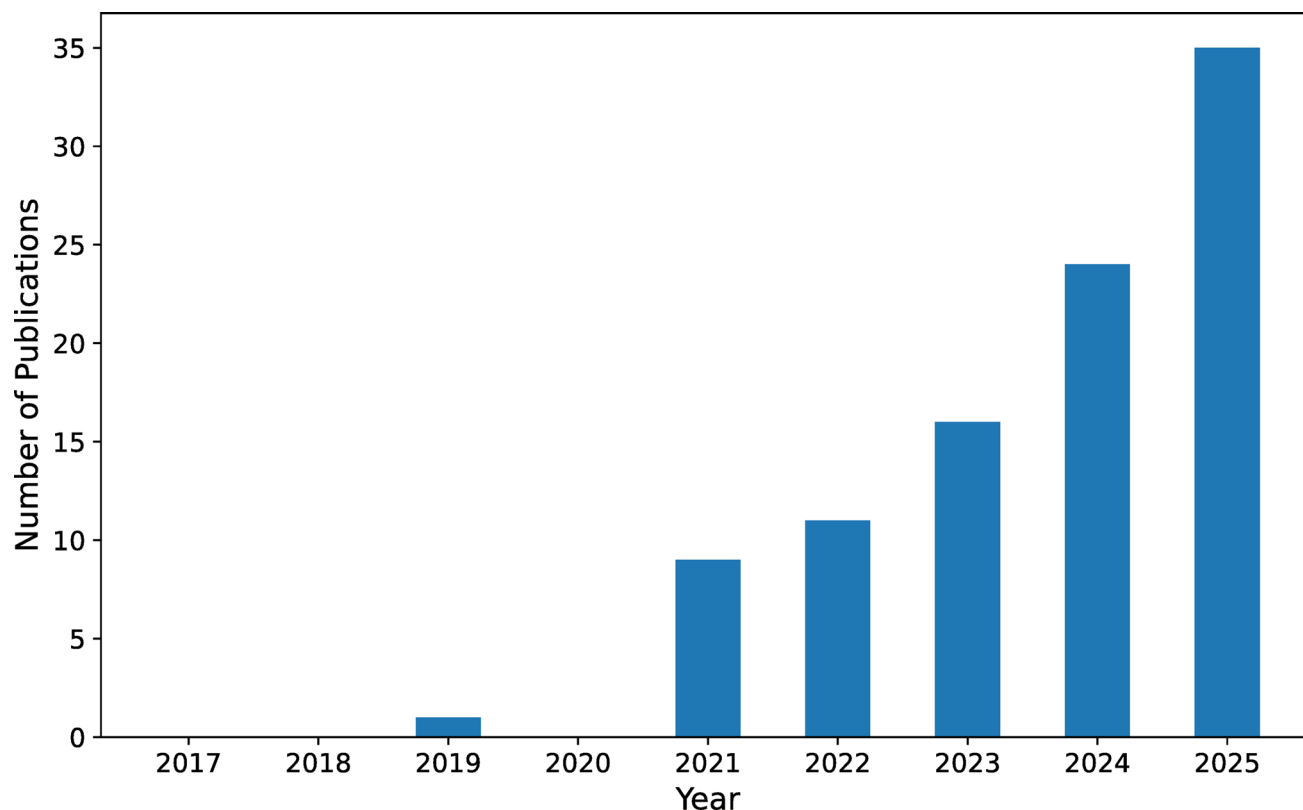


Fig. 2 Number of publications per year until December 2025

Table 3 Number of publications per journal until December 2025

Journal Title	Quantity
IEEE Internet of Things Journal	16
IEEE Transactions on Vehicular Technology	11
IEEE Transactions on Consumer Electronics	7
IEEE Access	6
IEEE Transactions on Intelligent Transportation Systems	4
IEEE Transactions on Mobile Computing	4
IEEE Transactions on Network and Service Management	3
IEEE Transactions on Network Science and Engineering	3
Sensors	3
ACM Transactions on Sensor Networks	2
Digital Communications and Networks	2
Electronics	2
Future Generation Computer Systems	2
IEEE Network	2
IEEE Open Journal of the Communications Society	2
IEEE Transactions on Green Communications and Networking	2
IEEE Transactions on Industrial Informatics	2
Journal of Grid Computing	2
Physical Communication	2
Alexandria Engineering Journal	1
Applied Intelligence	1
Computer Communications	1
Computer Networks	1
Discover Artificial Intelligence	1
Engineering	1
Frontiers of Information Technology and Electronic Engineering	1
IEEE Journal of Selected Topics in Signal Processing	1
IEEE Transactions on Communications	1
IEEE Transactions on Intelligent Vehicles	1
IEEE Transactions on Services Computing	1
Informatica (Slovenia)	1
International Journal of Web and Grid Services	1
Internet of Things (The Netherlands)	1
Journal of Network and Computer Applications	1
Journal of Supercomputing	1
KSII Transactions on Internet and Information Systems	1
PeerJ Computer Science	1
Vehicular Communications	1
Total:	96

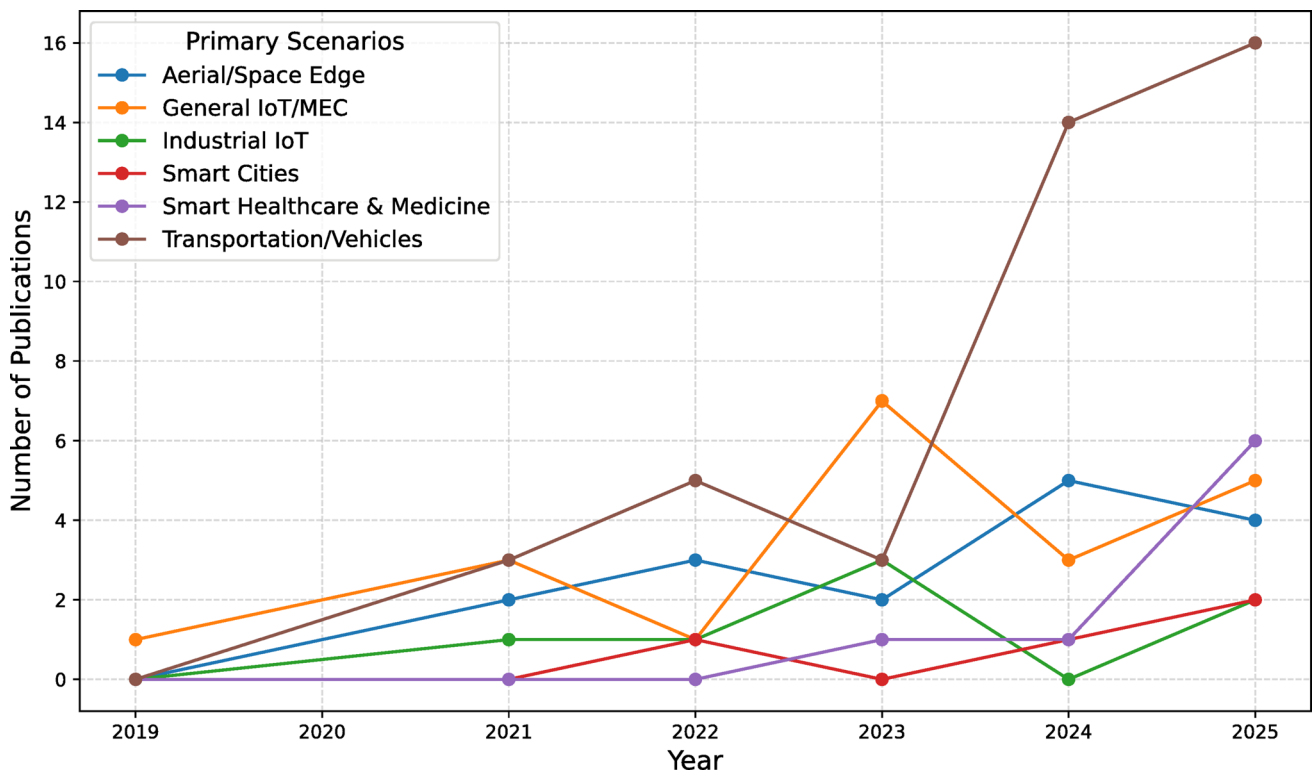
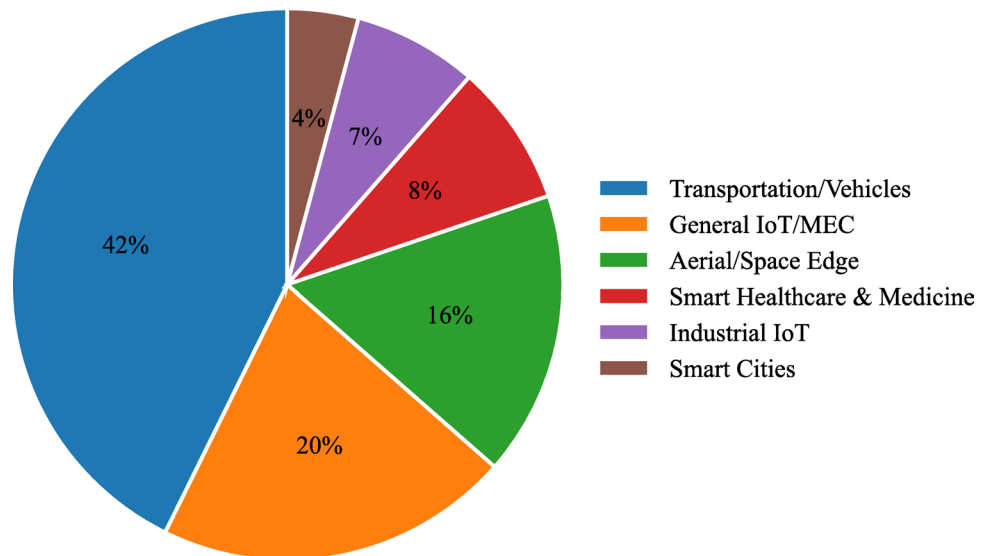
of TO scenarios in this work is based on the core application industry. For cross-industry research, classification is determined by the primary research focus. For example, in Unmanned Aerial Vehicle (UAV)-assisted vehicular network scenarios, if the UAV functions as an ED, it is categorized under the “vehicular” domain. In contrast, if it acts as an ES, it is classified within the “aerial computing” domain. Given the significant overlap between the “transportation” and “vehicular” fields, this study consolidates them into a single category. In addition, studies where aerial devices serve as edges, or CSs, are defined as belonging to the

emerging category of “aerial/space edge.” Finally, general research on IoT or MEC is uniformly categorized as “General IoT/MEC.”

As shown in Fig. 3, the transportation and vehicular scenario accounts for 42%, indicating that FL offers significant advantages in handling latency-sensitive decision-making tasks. The proportions of general IoT and mobile edge computing at 20% and aerospace and space edge at 16% are comparable, suggesting that fundamental research and the aerospace domain remain research hotspots. Smart healthcare & medicine, accounting for 8%, and the industrial IoT, at 7%, exhibit similar shares, reflecting that FL is gradually gaining attention in vertical industries characterized by high privacy protection requirements and strong device heterogeneity. Although smart cities account for the smallest proportion at 4%, this domain still holds considerable development potential due to its involvement in multi-system coordination and massive data integration, which can enhance urban efficiency and improve quality of life for residents.

As shown in Fig. 4, research across all scenarios exhibits an overall upward trend. However, general IoT and mobile edge computing experienced a decline in 2024 after reaching a peak in 2023. The transportation and vehicular scenario demonstrated the most significant increase in 2024, growing approximately fivefold compared with 2023, and continuing to grow in 2025. The aerospace and space edge sector experienced a slight decline in 2025 following rapid growth in 2024. The smart healthcare & medicine domain has shown a steady upward trend since 2023 and continued to experience modest growth in 2025. In general, research is moving in more specialized directions, with the transportation and vehicular scenario showing the most significant growth.

Early research on TO primarily focused on terrestrial scenarios, addressing issues such as wireless offloading, privacy protection, and mobility. For instance [51], integrates small-cell networks with MEC to mitigate inter-cell interference [52], employs privacy-preserving offloading to alleviate data scarcity on EDs, and [53] proposes latency reduction strategies in wireless-powered MEC. At the same time [54], considers user mobility in heterogeneous multi-base-station environments and develops privacy-preserving approaches that enhance user experience and data security while maintaining efficient communication. In recent years, the focus has shifted toward complex tasks and heterogeneous networks. For example [55], investigates multi-objective offloading and graph Reinforcement Learning (RL) for DAG-based tasks. More recent studies [56, 57] propose offloading strategies that integrate RL with multi-objective optimization, thereby improving decision efficiency and energy optimization in dynamic MEC environments.

Fig. 3 Primary Application Scenarios of Task Offloading**Fig. 4** Number of publications by primary TO scenarios until December 2025

In IoT environments [58] and [59], design customized offloading strategies for typical sensors and edge nodes, which aim to optimize resource allocation and improve overall system performance in real-time data processing [60]. integrates the principles of Industry 4.0 with MEC to develop privacy-preserving offloading methods [61]. leverages UAVs to transmit data to servers, while [62] combines low-earth-orbit satellites with UAVs to support blockchain-based communication, thereby enhancing security and

reducing latency [63, 64]. focus on intelligent operation and sustainable edge computing in the industrial IoT, using distributed auctions or hierarchical multi-agent learning to schedule resources, demonstrating the advanced application of TO in this domain. To address the resource-constrained nature of edge scenarios [65], employs lightweight digital twins to predict device performance, marking the first integration of explainable AI, digital twins, and FL to achieve predictive TO [66]. is the first to combine knowledge

distillation with FL to deploy lightweight models on edge devices [67], introduces split learning, wherein edge devices only compute the feature extraction portion of the model, leaving the remainder of the model to be computed by the edge server.

In vehicular TO, VEC primarily focuses on offloading between vehicles and roadside units (RSUs). Representative works include the vehicle-to-vehicle offloading strategy proposed in [68], the vehicular fog computing (VFC) challenges for autonomous driving discussed in [69], and the dew-enabled VFC network introduced in [29]. To supplement the high cost and limited availability of RSUs [28], introduces UAVs, while [70] and [71] adopt high-altitude balloons and LEO satellites, respectively, to provide vehicular network coverage. More advanced contributions include: [72], which proposes dependency-aware UAV-assisted offloading using asynchronous federated DRL to reduce latency and energy consumption; [73], which employs generative adversarial networks to augment task data and combines them with federated optimization to enhance recognition accuracy and energy efficiency; [74], which develops an adaptive federated deep deterministic policy gradient (DDPG) algorithm for heterogeneous air-ground integrated networks, mitigating the communication burden caused by mobility and heterogeneity through adaptive local updates and hybrid reward functions; and [75], which applies FL and DRL in LEO satellite edge computing systems to jointly optimize subchannel allocation and power control, thereby improving weighted sum rate while preserving user privacy [76]. is the first to propose the combination of FL and multi-agent RL. Building upon this [77], investigates a comprehensive framework that integrates digital twins, blockchain, and federated multi-agent DRL.

In air-ground integrated scenarios, research has gradually extended from terrestrial to aerial domains. For example [51] and [78], utilize UAVs as aerial base stations for ground terminals, while [79] and [80] employ ground ESs to support UAV computation. In emergency communication [81], introduces digital twin technology to design optimal offloading strategies. Recent works include [82], which optimizes user scheduling, TO, and UAV flight parameters through FedAvg in UAV-assisted MEC networks; [83], which introduces a digital twin environment and FRL to improve energy efficiency and balance workloads in artificial IoT systems constrained by battery life and computing resources; and [74, 82], and [84], which propose adaptive federated DDPG algorithms suitable for heterogeneous artificial general intelligence-MEC networks, reducing communication overhead and accelerating convergence [85] and [86]. integrate artificial IoT, digital twins, and UAV technologies to achieve environmental awareness, privacy protection, and distributed policy training, reflecting a

technological evolution from single UAVs to collaborative swarms [87]. addresses the issue of uneven contributions among agents in FL by introducing a UAV-assisted mechanism that assigns differentiated aggregation weights to each agent based on their contribution levels.

In pure aerial scenarios without ground servers [88], employs high-altitude balloons to replace ground base stations in supporting mobile devices, and [89] relies on low-altitude UAV swarms to train global models and execute TO without ES or CS. At the same time [90], develops a hybrid system comprising high-altitude airships, low-altitude UAVs, and mobile terminals, with UAVs and airships collaborating to train optimal offloading strategies [83]. further designs a fully ES/CS-free scenario in which UAV clusters perform global model training and TO. At the same time [55], explores cooperative training and task execution using high-altitude airships and UAVs. These studies highlight, that aerial scenarios are gradually moving away from reliance on ground servers toward autonomous and collaborative unmanned systems.

As shown in Table 4, TO research scenarios can be grouped into three primary categories: (1) Fundamental MEC Scenarios, including heterogeneous, harsh, and wireless-powered MEC environments. (2) VEC, involving high-mobility vehicles integrated with roadside stations and aerial drones. (3) Air-Ground Integrated Environments, encompassing satellites, balloons, and drones that support terrestrial devices while reducing energy consumption. As shown in Table 4, the development of TO scenarios exhibits a chronological path, starting with terrestrial applications and basic MEC and ending with vehicular networks and air-ground integration. Since 2021, the rapid adoption of UAVs, high-altitude balloons, and satellites has accelerated the development of aerial edge computing. Recent research (e.g [28, 31, 71, 75, 86, 87, 91, 92]), has increasingly emphasized the use of UAV swarms and the integration of heterogeneous networks. As aerial edge computing technologies mature [88, 90, 93] research is shifting toward multi-platform integration, including Space-Air-Ground Integrated Networks. Looking ahead, FL-driven TO is expected to play an essential role in domains such as smart healthcare [30, 94–96], smart cities [97–99], and the industrial Internet [63, 64].

3.3 Key challenges addressed by FL in TO

TO leverages available computing resources across the end-edge-cloud continuum, thereby enhancing overall system performance in edge environments. The main optimization objectives can be summarized as follows [6]: (1) Minimizing device energy consumption. Optimized task allocation reduces the energy required for system-level task execution,

Table 4 Distribution of primary and secondary application scenarios identified in the reviewed literature

Primary Scenario	Secondary Scenario	Quantity
Aerial/Space Edge	Aerial Edge Computing Network	2
	Air-Ground Integrated	11
	Space-Air-Ground Integrated	2
	UAV-Assisted Vehicular Networks	1
	Unspecified	0
General IoT/MEC	Cyber twin	1
	Digital Twins	4
	Virtual reality	1
	Unspecified	13
	Air-Ground Integrated	2
Industrial IoT	Space-Air-Ground Integrated	1
	Vehicle-Assisted	1
	Unspecified	3
	UAV-Assisted Vehicular Networks	1
Smart Cities	Vehicle-Assisted	1
	Unspecified	2
	Air-Ground Integrated	2
Smart Healthcare & Medicine	Space-Air-Ground Integrated	1
	UAV-Assisted Vehicular Networks	1
	Unspecified	4
	Air-Ground Integrated	1
Transportation/Vehicles	High Altitude Platform	3
	Space-Assisted Vehicular Networks	1
	UAV-Assisted Vehicular Networks	11
	Vehicle-Assisted	26
	Unspecified	4

Of the 96 papers reviewed, only 73 reported both primary and secondary application scenarios

particularly in energy-constrained EDs, thereby extending their operational lifetime [56, 69, 80, 86, 92, 100]. (2) Minimizing time delay. By selecting optimal offloading destinations (e.g., nearby EDs, ESs, or CSs), task completion latency can be shortened [57, 68, 71, 75, 101–104]. (3) Balancing delay and energy consumption. In many application scenarios, optimization considers a weighted combination of latency and energy consumption to satisfy diverse service requirements [51, 52, 58, 63, 77, 91, 95, 105–109].

Based on the reviewed literature, four major problems characterize TO optimization: (1) Insufficient collaborative training among EDs. Dynamic variations in task data necessitate frequent adjustment of offloading schemes [88]. Limited ED resources often prevent efficient coordination with ESs [100]. In vehicular networks, ineffective task scheduling strategies fail to optimize the aggregation frequency across vehicles, resulting in delays that can hinder real-time data processing and negatively impact overall performance [73]. (2) Data privacy leakage. Researchers have increasingly focused on safeguarding user privacy without compromising maintaining system performance and user experience [52, 54, 72, 97, 110]. (3) High communication costs. Edge wireless networks face significant pressure due to task variability and bandwidth constraints [106]. For example, vehicular networks require low-latency

offloading to RSUs, yet current schemes incur high communication overhead [28]. Moreover, the instability of vehicular topologies introduces costly model re-training during dynamic updates [56], which can lead to increased latency and reduced overall system performance. (4) High computational costs on EDs. Algorithms such as Asynchronous Advantage Actor–Critic demand substantial resources, undermining energy efficiency in green IoT systems [111]. Similarly, DDPG-based approaches introduce significant computational and signaling overhead [81, 112], which can lead to increased energy consumption and reduced performance in green IoT systems.

As summarized in [3], existing methods for addressing TO fall into three categories: (1) Classical algorithms. Dynamic programming and branch-and-bound approaches achieve optimal solutions in small-scale MEC networks; however, they face exponential complexity in larger networks. (2) Heuristic algorithms, including convex relaxation and game-theoretic methods, reduce complexity through structured search but can be trapped in local optima. Meta-heuristic algorithms further enhance global exploration and have evolved with adaptive and self-tuning mechanisms that enable automatic parameter adjustment and improved performance in dynamic environments. (3) AI methods. RL identifies optimal policies through agent–environment

interaction, while DL is adequate for processing high-dimensional and nonlinear data. DRL combines these strengths [113].

Edge environments are inherently dynamic, with uncertainty introduced by device mobility and fluctuating resources. AI-based methods offer adaptability and real-time decision-making, enabling TO models to learn from dynamic conditions and adjust their strategies accordingly. However, training DRL agents is computationally expensive. Centralized training of single-agent DRL models requires extensive data transmission from EDs, increasing network load and privacy risks. Distributed multi-agent DRL approaches further strain ED resources and prolong training, while localized training often limits model generalization to specific environments [61, 79, 114]. This can lead to suboptimal performance in diverse scenarios, as the models may not adapt well to variations outside their training conditions. These limitations highlight the potential of FL for TO optimization. FL reduces reliance on centralized training by enabling local model updates and global aggregation, thereby aligning closely with the distributed nature of TO. From the reviewed literature, FL contributes to TO optimization in four ways: (1) Efficient collaborative training. FL aggregates local models trained on heterogeneous devices, enhancing coordination among EDs and improving adaptability to dynamic environments; (2) Privacy protection. Local training avoids transmitting raw data, while

techniques such as differential privacy and homomorphic encryption provide stronger protection against leakage. (3) Improved resource balance. The large number of EDs in edge environments collectively contributes to global model training, distributing computational loads more effectively; (4) FL requires only the exchange of model parameters rather than raw data, thereby reducing transmission costs in edge networks.

Among the 96 reviewed studies, FL applications in TO primarily target four key problems: collaborative training (87 mentions), privacy protection (79 mentions), communication cost reduction (69 mentions), and computational efficiency (54 mentions). As shown in Fig. 5, the distribution of papers addressing these problems is as follows: the largest group consists of papers addressing three problems (44 papers), followed by those addressing four problems (29 papers), those addressing two problems simultaneously (18 papers), and finally those addressing a single problem (6 papers). Although two-objective optimization experienced a rebound in 2025, it is no longer the primary driver of growth. Three-objective and four-objective optimization have increased year by year since 2022, reaching 15 and 12 papers in 2025, respectively, and have become the mainstream research focus.

Current research demonstrates that FL is primarily employed to enhance collaborative training while simultaneously addressing privacy, communication, and

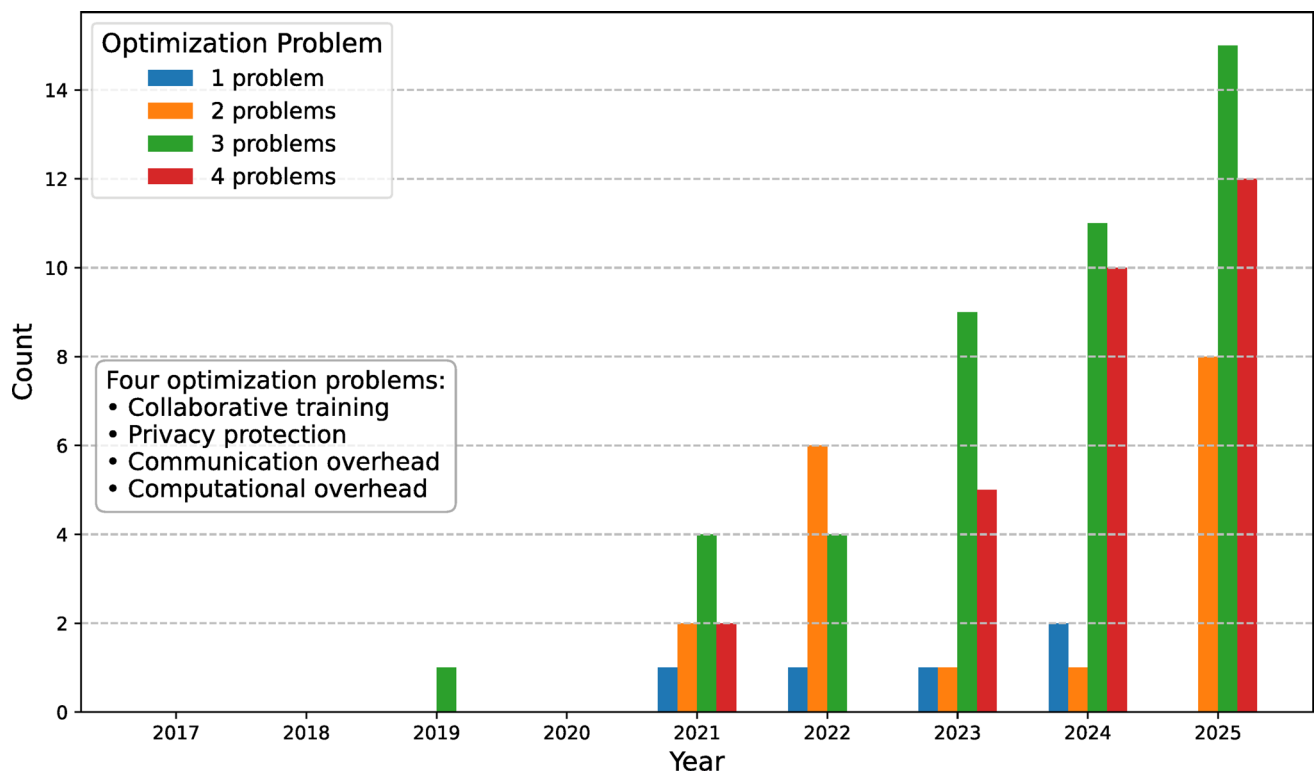


Fig. 5 Number of optimization problems per paper until December 2025

computational challenges. Recent literature reflects a clear progression toward more comprehensive solutions, with multi-problem optimization becoming a central trend since 2021. Future research should aim to extend this trajectory by developing holistic frameworks capable of jointly addressing multiple challenges in TO. The Table 5 presents the problems addressed by FL in TO, as identified from the papers reviewed in this study.

3.4 FL architectures for TO optimization

The edge computing environment can be structured into three layers: Edge Devices (EDs), Edge Servers (ESs), and Cloud Servers (CSs). EDs now include not only EDs and smartphones but also power EDs, vehicles, and drones. Edge layer components have grown from fixed ESs to include small cell base stations, vehicular servers, drones, RSUs, aerial base stations, and low-earth orbit satellites. This diversification enriches edge-layer capabilities while providing greater flexibility in TO strategies. Communication methods have similarly evolved from simple wireless links to multi-level, heterogeneous networks, enabling broader coverage and resilience in dynamic environments. A task can be offloaded either to peer devices within the same layer or to devices in other layers for execution. Latency-sensitive tasks generated at the terminal device layer may be executed locally or offloaded to the ES layer.

In contrast, computation-intensive tasks are generally offloaded to the edge or cloud layers. Figure 6 illustrates the four system architectures included in the literature. The proportion of three-layer systems exceeds that of two-layer systems by a factor of two, highlighting the significant role of cloud layers in TO.

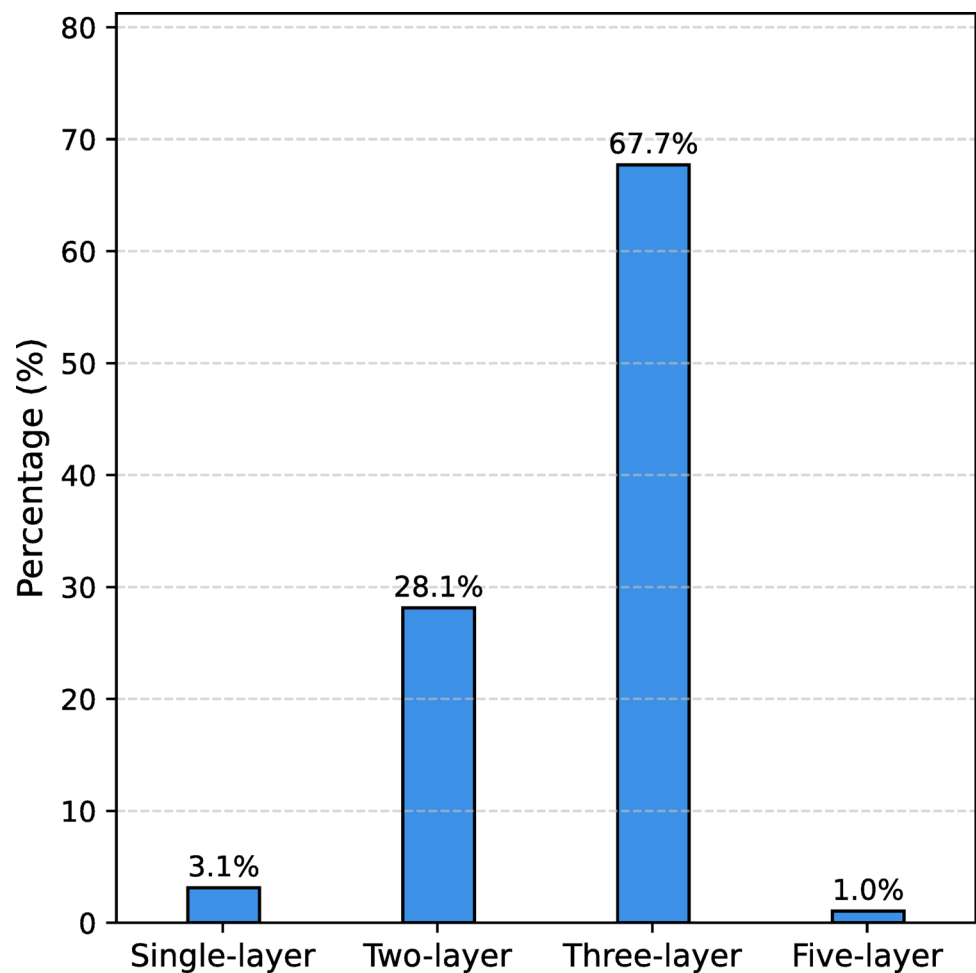
Single-layer system models focus on offloading within the terminal device layer. For example, Fast and Fair Computation Offloading Management in a Swarm of Drones Using a Rating-Based FL Approach explores drone swarms equipped with communication and computing capabilities that compete for resources based on task type and size. Similarly, Federated Double Deep Q-Learning-Based Computation Offloading in Mobility-Aware Vehicle Clusters examines resource sharing among vehicles via a directed acyclic graph. Such approaches are suitable for relatively small-scale tasks with low latency requirements. Two-layer system models extend TO from devices to ESs, which provide richer computational resources [32, 58, 116, 123]. This mechanism not only addresses resource limitations at the device layer but also achieves global resource optimization by allowing for more efficient distribution of tasks across the available computational resources in both the edge and cloud layers. Three-layer system models incorporate the cloud layer, enabling advanced global optimization in scheduling, load balancing, and resource allocation [68, 76, 78, 94, 112, 142]. In these architectures, tasks can be processed by ESs, offloaded directly to the cloud, or handled collaboratively by both layers. Among the reviewed studies, three-layer models account for 67.7%, while two-layer models represent 28.1%, implying that two-thirds of TO tasks involve substantial computational demands and still require support from the cloud layer.

Furthermore [29], proposes a five-layer model. As shown in Fig. 7, it introduces hierarchical architecture for TO methods in vehicular fog computing. The bottom layer is the ED layer, primarily responsible for generating computational tasks. The introduced Dew layer enables local data backup and temporary storage, ensuring data buffering and task continuity for EDs under offline conditions. When network

Table 5 The problems addressed by FL in TO

Ref.	Advantages of Using FL				Total papers
	Collab. Training	Privacy Protection	Comm. Efficiency	Comp. Efficiency	
[82]	0	0	0	1	1
[28, 71, 88, 98]	1	0	0	0	4
[70, 91, 96], 115–[117]	0	1	1	0	6
[53, 54, 69, 76, 80, 85, 97, 99]	1	1	0	0	8
[61, 68]	1	0	1	0	2
[104]	1	0	0	1	1
[118]	0	1	1	1	1
[92]	0	0	1	1	1
[51, 62, 100, 102, 112], 119–[121]	1	0	1	1	8
[32, 52, 59, 60, 72, 93, 95, 105], 122–[124]	1	1	0	1	11
[30, 31], 55–[58, 66, 73, 77, 78, 81, 83, 87, 89, 90, 103, 108, 114], 125–[130]	1	1	1	0	24
[29], 63–[65, 67, 74, 75, 79, 84, 86, 94, 101, 106, 107], 109–[111], 131–[142]	1	1	1	1	29
Total:	86	79	71	51	96

Total indicates the overall number of problems found across the 96 papers reviewed. Collab.= Collaborative, Comm.= Communication, Comp.= Computational

Fig. 6 Layer Distribution of Task Offloading Models

connectivity is restored, the Dew layer forwards tasks and data further to the edge layer. The edge layer contains no servers and acts as a connection channel between the dew layer and the fog layer. Subsequently, the fog layer handles the collaborative processing of most offloaded tasks. In contrast, tasks with extremely high computational demands or excessive resource consumption are further offloaded to the cloud layer to ensure overall system stability and efficiency.

According to FL training methods, these FL architectures can be broadly categorized into three types: decentralized FL, centralized FL, and hierarchical FL [59]. First, decentralized FL does not rely on a central aggregator; instead, both local model training and global aggregation are performed within the terminal device layer [56, 80, 88, 89]. For example [89], employs a Q-model to score drones and select a cluster head as the aggregation node, while [56] designates a cluster-head vehicle to perform aggregation. Such architecture features flexible node connectivity, enabling low-latency processing within device clusters and thereby enhancing system autonomy. However, the absence of a central aggregator reduces global coordination, making the training process more complex. Second, centralized FL

follows the typical client–server paradigm, where multiple clients connect to a single server. This architecture dominates the included studies. In terminal-edge FL, local models are trained on EDs and then transmitted to ESs using synchronous, asynchronous, or semi-asynchronous aggregation methods to prioritize either latency reduction [103, 120] or energy consumption reduction [92, 96]. In cloud-edge-device FL, data are uploaded from EDs to the edge layer for local training, and the resulting models are aggregated in the cloud, which reduces device energy consumption and computational load [76, 77, 95, 102].

Nevertheless, this architecture requires frequent uploading of model updates, leading to considerable communication overhead. Finally, hierarchical FL integrates the advantages of both edge and cloud. A typical process involves aggregating local models at the edge first, followed by uploading the aggregated results to the cloud for global model updates [77, 93, 105, 139]. This approach balances computational efficiency across the edge-cloud continuum but also faces challenges related to communication latency.

In TO, within edge computing environments, the system architecture can typically be organized into one, two, or three

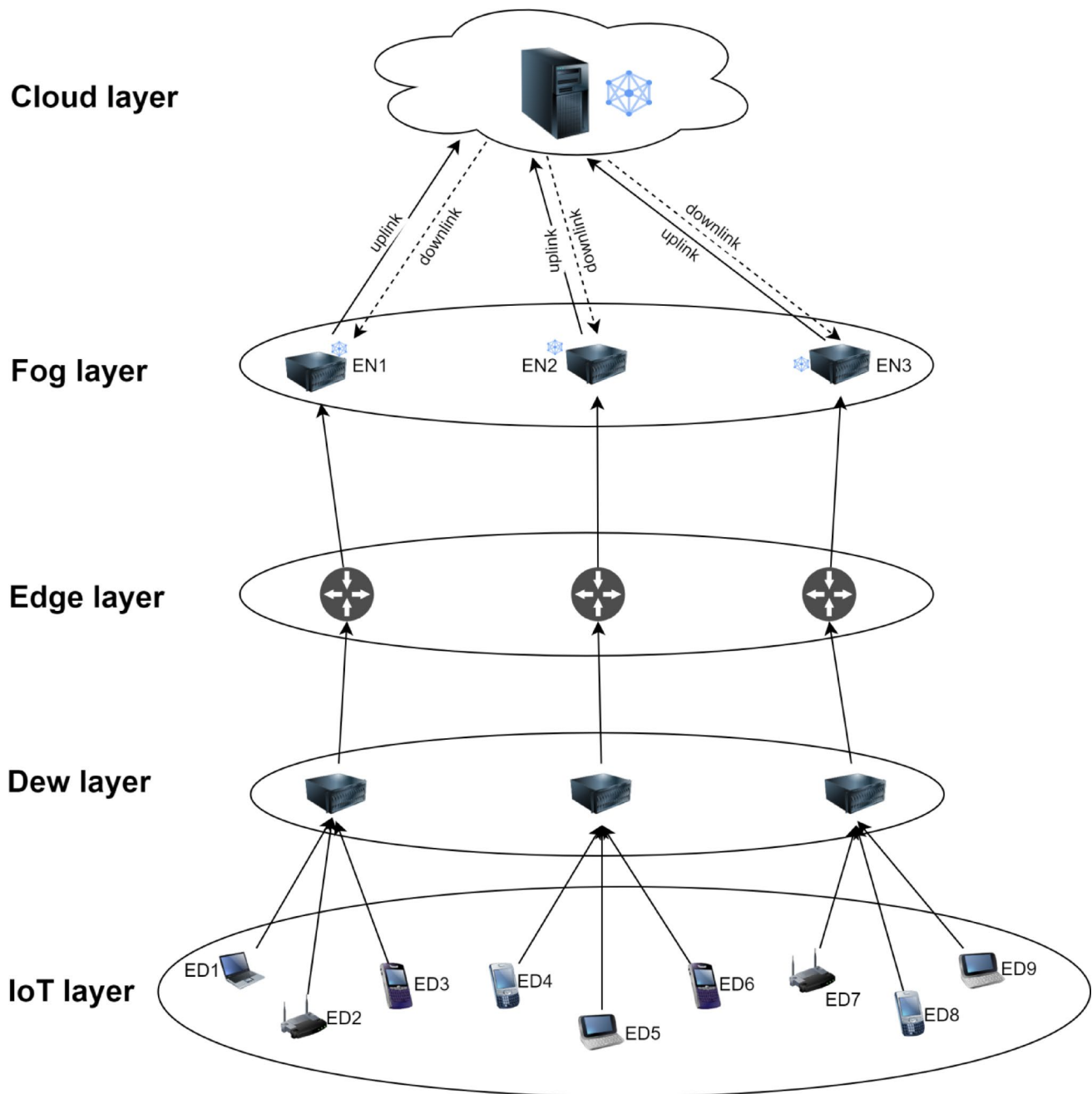


Fig. 7 Four-Layer Model in Task Offloading

layers, as shown in Fig. 8(a). A three-layer architecture can be further decomposed into two centralized forms, namely terminal-edge and edge-cloud, which exhibit a strong correspondence with the communication and training models of FL, as shown in Fig. 8(b). In TO, EDs are primarily responsible for task generation; in FL, these devices act as clients that train local models on their private data and upload model parameters to the aggregation server. The edge layer in TO is mainly responsible for handling latency-sensitive tasks and serving as a relay for connectivity, whereas in FL

it can function as an intermediate aggregator that performs partial model aggregation. The cloud layer in TO serves as the core computing and storage center, responsible for executing computationally intensive tasks and managing global resources. Correspondingly, in FL, the cloud layer acts as the worldwide server, responsible for aggregating global models and updates. This correspondence reveals the intrinsic alignment between the hierarchical structure of TO and the communication model of FL.

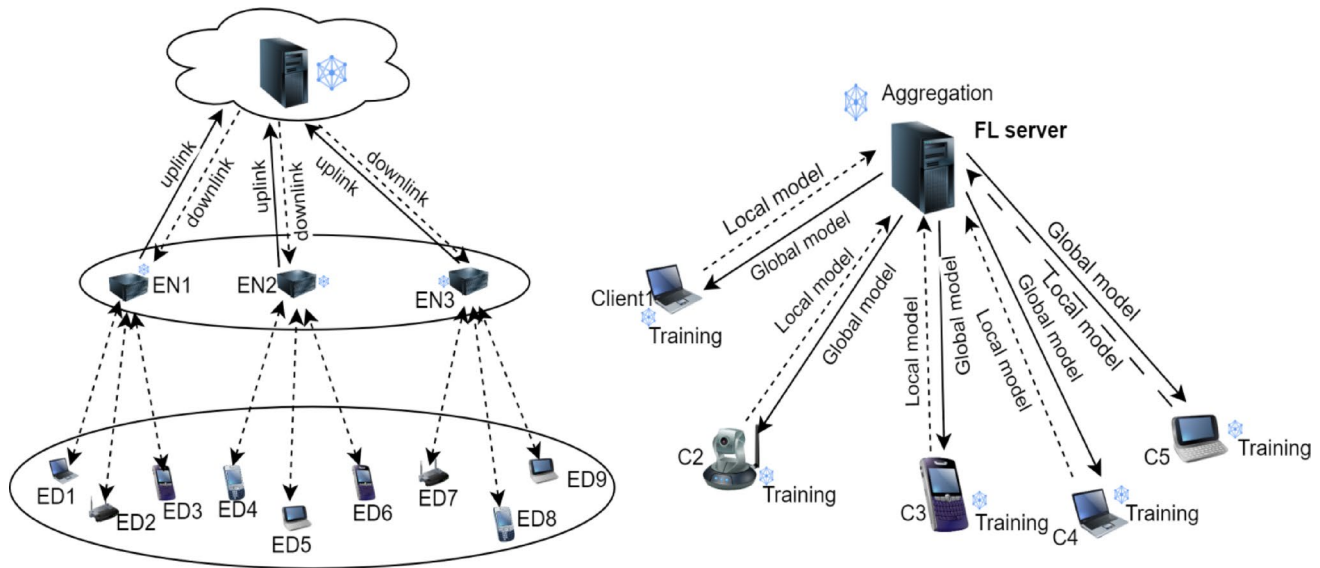


Fig. 8 The Structure of Task Offloading and Federated Learning

Table 6 The Mapping between System Architecture and Aggregation Models of FL

System Architecture	Aggregation Stage	Quantity	Ref.
Single-layer	Decentralized	2	[63, 89]
Single-layer	Single-stage	1	[127]
Two-layer	Decentralized	1	[88]
Two-layer	Single-stage	28	[31, 32], 52–[55, 58, 59, 74], 81–[83, 91, 92, 96, 100, 104, 108, 109, 111, 116, 120, 124, 125, 129], 134–[136]
Three-layer	Single-stage	52	[28, 51, 56, 57], 60–[62], 64–[73, 75, 76], 78–[80, 86, 87, 90, 95], 97–[99, 102, 103], 105–[107, 110, 112, 114, 115, 119], 121–[123, 126, 128, 130, 131, 133, 137, 138], 140–[142]
Three-layer	Two-stage	11	[30, 77, 84, 85, 93, 94, 101, 117, 118, 132, 139]
Five-layer	Single-stage	1	[29]
Total number of publications: 96			

As shown in Table 6, single-layer architectures are primarily mapped to decentralized aggregation or single-stage aggregation in client–server FL. In two-layer architecture, only one study adopted decentralized aggregation, while the others employed single-stage aggregation. In three-layer

architectures, 44 studies implemented single-stage aggregation, and seven studies adopted two-stage aggregation. In the exceptional five-layer architecture [29], single-stage aggregation was still applied. Overall, centralized FL dominates.

Current research shows that decentralized FL emphasizes low latency but suffers from limited coordination, and hierarchical FL demonstrates the potential to balance efficiency and performance in complex systems, making it a promising approach for applications requiring both speed and effective resource management.

As shown in Table 7, among the 96 studies, four adopted decentralized FL models focusing on low-latency task processing within the ED layer. The remaining studies employed centralized architectures, where FL occurs across device–edge–cloud layers. Models involving EDs and ESs primarily emphasize privacy preservation and latency reduction, while those between EDs and CSs focus on improving energy efficiency and balancing computational load. In device–cloud frameworks, data privacy and computational cost are prioritized, with local training performed on devices and global aggregation conducted in the cloud. Some studies also introduce two-stage aggregation mechanisms, enabling hierarchical model fusion from device to edge to cloud, thereby optimizing the utilization of distributed computing resources and mitigating communication latency. In particular, the agents in [130], including base stations, unmanned aerial vehicles, and vehicles, all participate in the global aggregation of FL as FL clients.

Table 7 Distribution of Federated Learning Architectures

Model	Ref.
Decentralized	[63, 80, 88, 89, 127]
Centralized	
ES, EDs	[28], [31], [32], [52], [53], [54], [55], [57], [58], [59], [64], [65], [67], [68], [74], [82], [83], [86], [92], [93], [97], [105], [109], [110], [112], [116], [117], [121], [125], [126], [127], [130], [132], [135], [136], [138], [142]
CS, EDs	[56], [61], [62], [70], [71], [76], [104], [108], [115], [120], [123], [137]
CS, Ess, and EDS	[131]
CS, ESs	[29], [51], [60], [66], [69], [72], [73], [75], [78], [79], [81], [87], [90], [96], [98], [99], [100], [101], [103], [106], [107], [111], [113], [122], [124], [129], [134], [139], [141], [143]
Hierarchical	
Secondary Model Aggregation	[30], [77], [84], [85], [94], [95], [102], [118], [119], [133], [140]

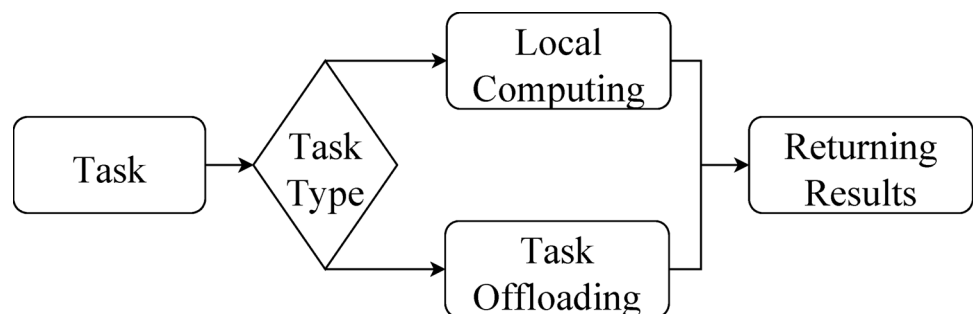
3.5 TO strategies enabled by FL

TO is generally categorized into two strategies: full offloading and partial offloading. The former, also referred to as binary or coarse-grained offloading [56, 105, 107], treats a task as an indivisible unit that is entirely offloaded to peer devices within the same layer, ESs, or remote computing centers, such as CSs, for execution. The selection of the offloading destination typically depends on factors such as data volume, network conditions, available resources, and the task's sensitivity to latency and computational performance. Tasks with stringent delay requirements are more likely to be offloaded to nearby ESs, whereas those requiring higher computational capacity are better suited for offloading to CSs. Although this strategy features a relatively low degree of flexibility, it is simpler in algorithmic design. In contrast, partial offloading [108, 119, 120], also known as fine-grained offloading, applies to divisible tasks with low inter-component coupling. By analyzing the dependency relationships among task components, it selectively offloads computationally intensive segments, which can be

executed in parallel or sequential scheduling modes. This strategy can effectively reduce energy consumption while satisfying latency constraints, achieving a balanced computational load between local and remote nodes, and thus significantly improving the overall system efficiency. However, the increased flexibility of partial offloading comes at the cost of higher algorithmic complexity, as it requires joint optimization of task partitioning, resource allocation, and the trade-offs between energy consumption and latency. In addition, several studies have proposed hybrid offloading strategies. For example [79], introduced a two-stage offloading scheme, in which binary offloading is applied from EDs to the edge, and partial offloading is adopted from the edge to the cloud. Considering computational delay [81], defined three offloading modes: local execution, full offloading, and partial offloading in a UAV-assisted vehicular network. Similarly [109], developed an offloading strategy that jointly optimizes delay, energy consumption, and cost and proposed an adaptive offloading mode capable of dynamically adjusting the offloading ratio.

The TO process can be divided into the following four steps: (1) Task Partitioning: Based on the types of tasks generated by the application, tasks can be classified into two categories: Indivisible Tasks, which must be executed locally or offloaded; and Divisible Tasks, which can be broken down into sub-tasks that can be offloaded if there are no dependencies among them. (2) TO Strategy: By analyzing the parameters of EDs and the types of tasks, it is determined whether a task needs to be offloaded, and a suitable offloading device is selected based on network conditions and computational resource availability. (3) Task Execution: The selected device executes the computational task. (4) Result Return: The execution results are transmitted back to the application by the executing device. As illustrated in Fig. 9 below:

TO optimization commonly seeks an optimal strategy under specified objectives, accounting for task characteristics, resource availability at both the source and destination, network bandwidth, and link stability. As the number of EDs scales, training data volumes grow substantially, rendering traditional optimization methods insufficient. Consequently, ML has been widely adopted for offloading decisions [143].

Fig. 9 Task Execution Steps in Task Offloading

Mathematical formulations that evaluate and optimize latency and energy consumption are referred to as computation models [144]. These models can be trained in centralized or distributed modes. In centralized training, data is aggregated at a server to learn global policies, whereas in distributed training, each ED trains a local model and makes independent decisions. Given the dynamic nature of edge environments, hybrid approaches that integrate centralized and distributed training are increasingly employed. Within the FL framework, EDs train local models while a federated server aggregates global updates, which reduces

communication overhead and enhances privacy protection [145]. This hybrid paradigm has been widely applied in MEC, vehicular networks, and air-ground integrated networks.

Table 8 reflects the current preferences in algorithm and scenario selection within FL research. Among the reviewed studies, the local model training algorithms in FL exhibit notable diversity. They can be broadly classified into three categories: (1) Traditional supervised learning algorithms, such as support vector machines, are primarily employed

Table 8 Federated Learning Variants and their roles in Task Offloading

Offloading Strategy	Ref.	Local Algorithm	Role
Binary Offloading	[61, 64, 99, 106, 118]	AC	J
	[57, 60, 62, 66, 125, 138]	AC	P
	[30, 32, 73, 74, 97, 112, 123, 133]	DDPG	J
	[85, 122]	DDPG	P
	[53, 132]	DL	J
	[63, 137]	DL	P
	[28, 58]	DQN	C
	[31, 56]	DQN	C
	[55, 75, 83, 115, 126, 136, 141]	DQN	J
	[52, 59, 71, 72, 80, 84, 86, 89, 90, 95, 101, 103, 105, 107, 116, 135, 139, 140]	DQN	P
	[102]	RL	J
	[128]	TD3	C
	[54, 98]	TD3	J
	[110]	TD3	P
	[87]	AC	J/C
	[76]	DDPG	J/C
	[65]	DQN	P/C
	[91]	DQN + TD3	J
Partial Offloading	[96]	DDPG	J/C
	[130]	DDPG + PPO	J
	[67]	DQN	C
	[129]	Knowledge Distillation	P
	[142]	Local SGD	C
	[92, 109, 124]	Local SGD	J
	[117]	Personalized SGD	J/C
	[104]	Split Learning	C
	[134]	AC	J
	[108]	AC	P
	[94, 119]	DDPG	C
	[51, 121]	DDPG	J
	[77, 82, 100, 111, 114]	DDPG	P
	[70]	DL	P
	[93, 127]	DQN	J
	[29, 69, 120]	DQN	P
	[131]	DRL	P
	[88]	ML	J
	[68]	QL	P
	[78]	TD3	J
Hybrid Offloading	[79]	DQN	J
	[81]	TD3	P

QL = Q-Learning; TD3 = Twin Delayed Deep Deterministic Policy Gradient; FL Role: J = Joint Policy Learning and Resource Optimization, P = Policy Learning, and C = Communication-Computation Optimization

for model optimization and classification tasks. (2) DL models such as deep neural networks and long short-term memory demonstrate superior performance in feature extraction and prediction tasks. (3) RL and DRL methods constitute the majority, including value-based approaches such as Deep Q-Network (DQN) and Double DQN, policy-based algorithms such as DDPG, TD3, and PPO, multi-agent variants such as MADDPG and MAD3QN, and asynchronous Actor–Critic methods such as Asynchronous Advantage Actor–Critic and Advantage Actor–Critic. These algorithms are commonly used for dynamic resource allocation and task scheduling. The analysis indicates that most studies still rely on DQN and DDPG as representative DRL algorithms, which perform effectively in discrete and continuous action spaces, respectively. However, more advanced algorithms such as PPO remain rarely adopted, limiting their applicability to complex continuous decision-making scenarios. Furthermore, as multi-user and multi-server settings become increasingly prevalent in MEC, the use of multi-agent versions of DDPG and Actor–Critic (AC) algorithms remains limited. Additionally, due to their simple model structures and suitability for discrete DRL training, approximately 70% of the reviewed studies adopt binary offloading strategies, which overlook task diversity and heterogeneity in local computing capabilities. In contrast, researchers have demonstrated that combining partial offloading with multi-agent algorithms can alleviate server congestion and enhance resource utilization efficiency in a collaborative manner [65]. integrates explainable artificial intelligence, digital twins, and multi-agent systems, using digital twins to predict device computational performance, thereby enabling

agents to avoid offloading tasks when edge devices are low on battery or have insufficient available resources. This mechanism allows multiple agents to dynamically share the global computational burden [87] and [76]. introduce a priority incentive mechanism based on resource contribution into the federated multi-agent model, where vehicles obtain corresponding TO priorities according to their contributions. This incentive-driven mechanism enables tasks to be dynamically offloaded to multiple edge devices, achieving load balancing in dynamic environments. The application of the above three multi-agent approaches can effectively alleviate server congestion and improve overall resource utilization.

3.6 Performance metrics for FL-enabled TO optimization

Performance metrics play a crucial role in evaluating the effectiveness of FL-TO schemes. They not only provide the core quantitative basis for assessing whether the integrated FL-TO design achieves its intended optimization objectives but also serve as key criteria for validating the overall performance of the framework under the communication, computation, and heterogeneity challenges inherent to FL. Therefore, the scientific definition and selection of performance metrics are crucial for accurately assessing both the efficiency of the FL process and its impact on optimizing offloading decisions. This section aims to systematically review and categorize the performance metrics adopted in existing studies, clarify their mapping to different offloading objectives, and thereby provide a theoretical reference for constructing a comprehensive evaluation framework.

As shown in Fig. 10, the correspondence between scenarios and TO optimization objectives reveals the differences in performance requirements across various contexts. In the general IoT/MEC scenario, approximately 75% of studies aim to minimize both latency and energy consumption [65–67], while 20% focus solely on latency minimization. This indicates that users generally seek a balance between system responsiveness and energy efficiency, though they tend to be more sensitive to latency when trade-offs are required [53, 101, 140]. In the Industrial IoT scenario, the trend is broadly consistent with that of General IoT/MEC, with 71.4% of studies minimizing both latency and energy consumption and none focusing exclusively on energy minimization. This highlights the dual requirement for real-time performance and energy efficiency in industrial production environments [95, 102]. By contrast, the Transportation/Vehicles scenario places greater emphasis on latency performance, with 45.5% of studies focusing solely on minimizing latency. In comparison, 48.5% optimize both latency and energy consumption [29, 102, 109] reflecting

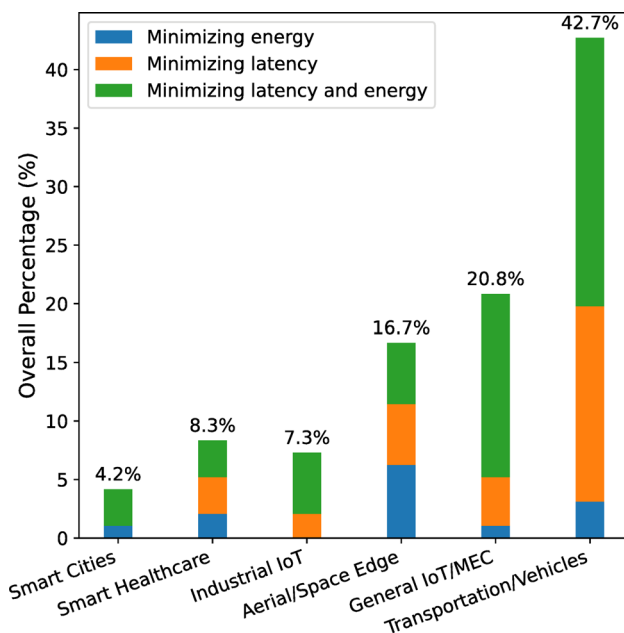


Fig. 10 Task Offloading Scenarios and Their Optimization Objectives

the importance of rapid response and low communication delay in vehicular network applications. Similarly, the Aerial/Space Edge scenario exhibits a trend comparable to Transportation/Vehicles, showing a high degree of latency sensitivity; however, due to limited communication links and energy resources, these studies also account for energy constraints while ensuring low latency [75, 78, 87]. In the smart healthcare & medicine scenario, studies focusing solely on minimizing latency [74, 103], together with those optimizing both latency and energy consumption [86, 139] account for a combined total of 75%, indicating that latency performance remains the core concern for the majority of research in this field.

Overall, latency optimization serves as the foundational objective in edge computing TO and dominates across all scenarios. In contrast, the priority of energy optimization varies depending on the characteristics of the scenario, becoming particularly prominent in energy-constrained contexts such as aerial/space edge, where limited power resources necessitate a greater focus on energy efficiency to ensure operational viability. Joint optimization of latency and energy consumption has emerged as a mainstream research objective, reflecting the inherently coupled nature of these two factors in edge computing systems.

Temporal analysis in Fig. 11 indicates an increasing focus on latency from 2021 onward, often accompanied by

considerations of energy. This aligns with the heightened expectations for Quality of Service in real-time systems, such as MEC and vehicular networks, where low latency is crucial for responsiveness and user experience. For example, in [52], to address varying delays in parameter transmission and global iterations in FL caused by unstable network links, the authors proposed a univariate outlier detection algorithm to screen FL participants and exclude clients with inadequate contributions. Similarly, in [49] and [71], to mitigate TO delays caused by vehicles moving outside the coverage of RSUs, the authors introduced a high-altitude platform (HAP) as an FL server, thereby forming an air-ground integrated network that reduces both energy consumption and time delay.

Through the analysis process, we found that five performance metrics are most frequently used: Delay, Energy, Cost, Reward, and Convergence. Combined with modifiers such as average, total, and system, these metrics are employed a total of 270 times across the 96 reviewed studies, with an average of 2.78 times per study. Brief definitions follow: (1) Delay. End-to-end latency arising from strategy computation, transmission, execution, and result return; a primary TO objective. (2) Energy. Consumption for model training, device operation, data transmission, and computation; typically centered on EDs (ESs/CSs are grid-powered). (3) Cost. TO policies minimize composite resource

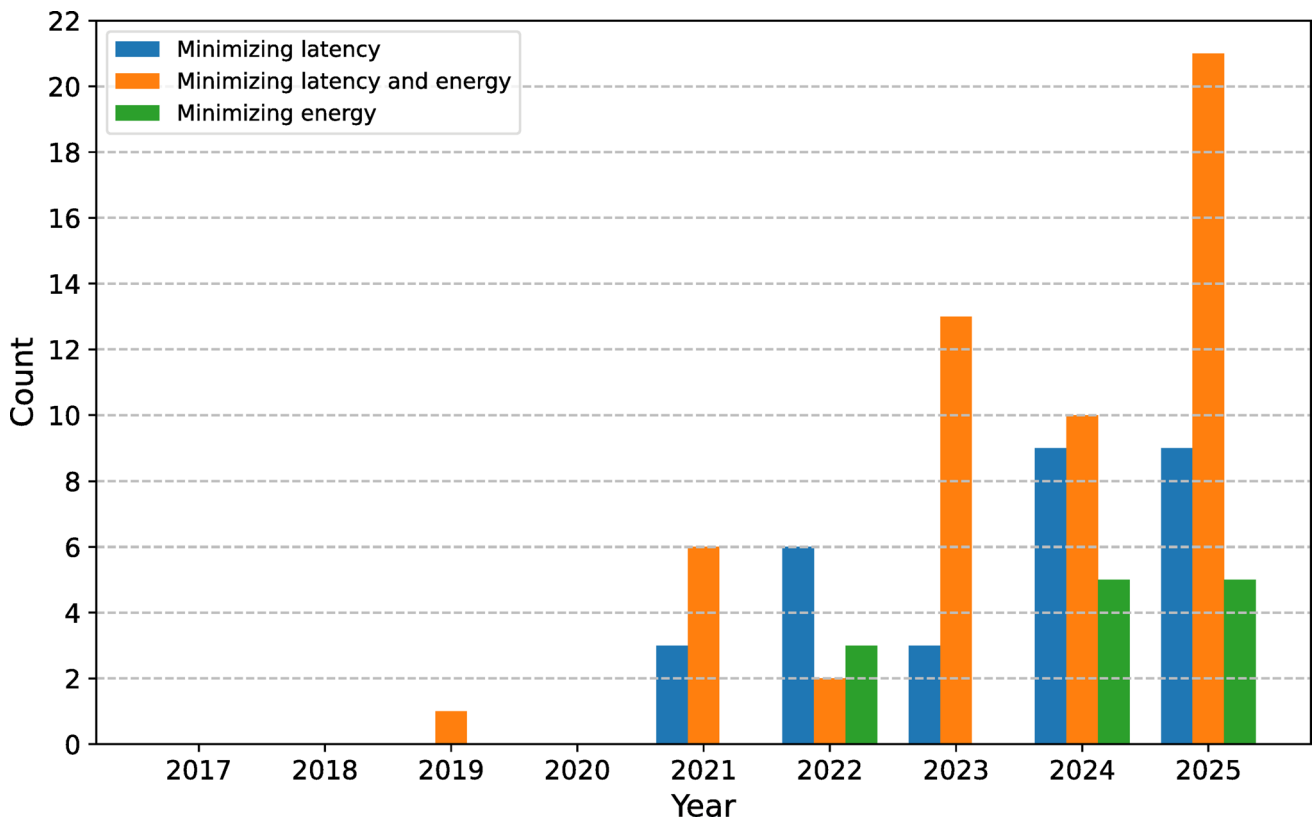


Fig. 11 Distribution of the Three Objectives in TO until December 2025

Table 9 Performance Metrics Adopted Across Papers for Task Offloading Optimization

Ref.	Performance Metrics				
	Delay	Energy	Cost	Reward	Convergence
[122]	0	0	0	1	0
[53, 75]	0	0	0	1	1
[111]	0	0	1	0	1
[100, 106]	0	0	1	1	0
[56]	0	1	0	0	0
[112]	0	1	0	0	1
[63, 137]	0	1	0	1	0
[92]	0	1	0	1	1
[32, 62, 68, 74, 81]	1	0	0	0	0
[103, 110, 120, 127]	1	0	0	0	1
[57, 71, 79, 84, 94, 102, 133, 136, 138]	1	0	0	1	0
[99, 101, 130]	1	0	0	1	1
[125, 126, 135]	1	0	1	0	0
[104]	1	0	1	0	1
[78, 115]	1	0	1	1	0
[28, 67, 116, 123]	1	0	1	1	1
[55, 65, 76, 82, 90, 107, 121, 129, 139]	1	1	0	0	0
[29, 52, 61, 64, 105, 142]	1	1	0	0	1
[54, 60], 86–[89, 98, 109, 119, 124, 128, 140]	1	1	0	1	0
[51, 59, 66, 69, 72, 73, 83, 93], 95–[97, 141]	1	1	0	1	1
[70, 114, 131, 132, 134]	1	1	1	0	0
[108, 118]	1	1	1	0	1
[30, 58, 80, 85, 117]	1	1	1	1	0
[31, 77, 91]	1	1	1	1	1
Total: 96	86	58	27	59	40

‘Total’ denotes the total number of performance metrics included in all reviewed papers

expenditure (compute, communication, storage, energy). (4) Reward. DRL feedback signals, reflecting delay, energy, success rate, or utilization, guide policy improvement. (5) Convergence. Stability and efficiency of learning concern both global and client-level convergence in FL. These metrics are evaluated under constraints such as average delay, total energy, average reward, and completion rate. Table 9 summarizes usage across the included studies.

3.7 The unresolved issues and future research directions

Despite notable advances in collaborative training, privacy protection, communication cost reduction, and computational load alleviation, several challenges persist in edge environments due to device heterogeneity, performance disparities, and complex communication patterns. These challenges limit the efficiency and robustness of FL-based TO strategies, pointing to important directions for future work. Accordingly, we synthesize the open problems as follows:

1) Model Convergence and Data Quality: Client drift frequently triggered by high correlations or non-independent and identically distributed (non-IID) characteristics

of training data, thereby degrading the performance of the global model and significantly slowing convergence [79, 84, 90, 94]. In addition, insufficient local training or a limited number of participating devices in aggregation can directly weaken the convergence quality and upper performance bound of the global model [69, 71, 73].

2) Communication and Resource Efficiency: Communication delay during model transmission and inefficient resource allocation jointly constrain the aggregation efficiency of FL [73, 89, 94, 106, 107, 112, 138]. Meanwhile, EDs acting as clients in FL experience significant increases in energy consumption, and the gradient update operations themselves impose additional computational and communication overhead. Consequently, EDs operating as clients in FL can experience significant increases in energy consumption [30, 56, 70, 138].

3) System Heterogeneity and Dynamic Adaptability: The performance of the global model has been compromised by large-scale, highly dynamic task contexts, as they cause a decrease in model training and aggregation [54, 56, 68, 69, 86, 100, 107]. Differences in hardware capabilities and energy conditions among devices, as well as data inconsistency and synchronization delays caused by their physical mobility, further degrade system aggregation performance

and overall adaptability [29, 31, 32, 71, 86, 89, 102, 107, 110]. It should be noted that device mobility and environmental dynamics may also impact link quality and scheduling, leading to communication delays and energy fluctuations; thus, there exists an overlap with “Communication and Resource Efficiency”. These two aspects are presented separately here for clarity.

4) Privacy and Security Protection: During the training and aggregation phases of FL, updated information is vulnerable to inference or reconstruction attacks, which potentially lead to privacy leakage [63, 81, 97]. Moreover, the system may be compromised by malicious nodes through distributed denial-of-service attacks or model poisoning, thereby threatening the security of both data and models [31, 56, 80, 94, 97, 111].

5) Algorithmic and Model Design Complexity: The inherently distributed and collaborative nature of FL frameworks can result in slow convergence and reduced training efficiency [100]. In FRL scenarios, substantial action or state spaces often require hierarchical or dimensionality-reduction strategies to control complexity, which in turn increases algorithm design difficulty [100]. Furthermore, advanced aggregation methods are usually challenging to implement due to algorithmic complexity, difficulty in convergence analysis, and high computational cost, leading most studies to adopt relatively suboptimal solutions [74].

In summary, heterogeneous client data impedes stable and efficient global aggregation, further complicated by mobility, variable client populations, and imbalanced computing/communication resources. Although FL mitigates privacy risks by sharing parameters rather than raw data, leakage vectors remain. Finally, DRL models can be resource-intensive, making local training on EDs challenging and energy-demanding. To enhance the implementation and evolution of FL in edge environments, subsequent research should concentrate on the following elements:

1. **Regional Integration of Application Scenarios:** Previous studies have examined ground, low-altitude, high-altitude, and space scenarios, and have initiated investigations into the utilization of integrated scenarios and digital twins in edge offloading. For example [77], combines blockchain, high-altitude platforms, and vehicular networks with digital twins to propose a resource allocation scheme for vehicular networks [92]. proposes a digital twin-driven collaborative offloading strategy for vehicular networks involving intelligent reflecting surfaces and unmanned aerial vehicles. However, existing research pays less attention to underwater scenarios. Future research on
- underwater scenarios and their integrated convergence will represent a promising direction.
2. **Dynamic Granularity Adjustment:** Existing research has begun exploring dynamic granularity adjustment from the perspective of task partitioning. For instance [104], proposes an in-vehicle deep neural network offloading method based on federated split hybrid learning to address device heterogeneity in vehicular networks [91]. designs a federated ensemble reinforcement learning offloading and resource allocation framework for directed acyclic graph tasks in mobile edge computing, reflecting exploration of fine-grained offloading and dynamic resource allocation. Future work can focus on hybrid local model training methods and adopt granularity-adaptive hierarchical model architectures or local model designs to accommodate the computational capabilities and data distribution characteristics of terminal devices.
3. **Efficient Communication Protocols:** To mitigate communication overhead in bandwidth-limited edge networks, future work should prioritize lightweight communication protocols through techniques such as gradient pruning, model quantization, and differential encoding. Additionally, leverage methods like model distillation or low-rank decomposition to simplify local models, thereby enhancing training efficiency and energy utilization on resource-constrained devices.
4. **Asynchronous Aggregation Strategies:** Future research should explore asynchronous and semi-asynchronous aggregation strategies combined with adaptive weighting mechanisms to ensure fairness and efficiency in model aggregation and, hence, address model drift issues and preserve stable convergence of the global model.
5. **Data Privacy and Security:** Due to merging security threats in FL, future research should explore integrating differential privacy or secure multi-party computation techniques to prevent data leakage. Such integrations ensure node credibility evaluation and distributed auditing to counter model poisoning and malicious gradient attacks, enhancing data security and training reliability.
6. **Dynamic Incentives and Resource Management:** Future research should investigate differentiated incentive mechanisms, such as dynamic reward allocation strategies based on real-time contributions (e.g., gradient utility and energy efficiency), and pair them with intelligent resource management policies. Dynamically tuning communication frequency, update intervals, and task granularity can help optimize system efficiency and robustness in highly

dynamic and resource-constrained edge networks. In large-scale and dynamic environments, dynamically adjusting parameters like communication frequency, local model update intervals, or training task distribution can make the system more stable and efficient.

4 Discussion and critical evaluation

4.1 Key insights and synthesis

Optimization of TO is the core of edge computing. Since around 2019, FL has been introduced to optimize offloading decisions, providing a new solution for TO by enabling collaborative training across devices without sharing raw data. This review, based on a systematic analysis of 96 papers, demonstrates that FL-based TO has been applied to diverse scenarios, including terrestrial IoT, UAV networks, and even space satellite systems. FL exhibits unique advantages in multi-objective optimization involving latency, energy consumption, and resource utilization. It effectively reduces communication overhead, lowers server load, enhances device cooperation, and protects data privacy. These advantages have proven effective in complex applications, such as smart cities, vehicular networks, and industrial IoT. The research domain can be divided into six major categories: general IoT, transportation/vehicles, smart cities, industrial IoT, innovative healthcare, and aerospace edge. Each domain has different requirements, and FL-TO strategies adapt by balancing various metrics. FL mainly addresses four core problems: collaborative training, privacy protection, reduction of communication costs, and reduction of computational load. Methodologically, the research encompasses FL architectures tailored to the capabilities of EDs and TO objectives, often employing DRL for local training, and evaluates them using metrics such as latency, energy consumption, cost, reward, and convergence. Current research relies heavily on simulations, and several issues remain to be explored, including the compatibility between FL and partial offloading strategies, the effectiveness of local training algorithms in specific scenarios, and the lack of unified benchmarks for evaluating algorithm performance.

Recent emerging studies have begun to introduce federated multi-agent reinforcement learning and digital twin technologies [79, 82] to address device heterogeneity and the complexity of integrated air-ground networks. In addition, in terms of handling device heterogeneity in vehicular networks [84], explores federated split hybrid learning for offloading deep neural network parameters among vehicles. The integration of these technologies offers new insights

into the application of FL-based TO in complex scenarios such as vehicular networks and unmanned aerial vehicle-assisted networks.

4.2 Methodological limitations

This review has limitations in its research methods. Literature retrieval primarily relies on the Web of Science and Scopus databases, as well as specific keywords, which may overlook research from other sources and affect representativeness. Strict inclusion/exclusion criteria ensure quality but also limit the scope of study and subjective judgments in the screening process that may introduce bias. In terms of evaluation, most studies use simulated performance metrics (such as latency and energy consumption) and rarely consider real-world constraints like user experience or device heterogeneity, which casts doubt on the applicability of conclusions in practical deployment.

4.3 Analytical limitations and biases

There are inherent analytical limitations and biases in the current field. Many validations are based on simulation rather than real-world experiments; simplified environmental settings may make reported performance improvements difficult to replicate in practice. At the same time, the lack of benchmarking for different FL algorithms under the same settings, coupled with the absence of standardized datasets and evaluation protocols, makes it difficult to determine the optimal algorithm. Furthermore, binary and partial offloading decisions are not completely incorporated into the FL framework, as analytical models frequently presume full offloading, neglecting hybrid scenarios where tasks are performed in segments, resulting in research deficiencies and skewed results.

4.4 Practical implications

While simulation-based studies have given researchers valuable information about the potential of FL-enabled TO, the absence of real-world deployment and validation remains a significant limitation. Owing to these factors, to ensure the successful deployment of FL-TO systems, future research should prioritize field experiments that validate simulation results under real-world conditions.

5 Conclusion

This review synthesizes the progress of FL applications in TO from 2017 to 2025. By systematically classifying application scenarios, elucidating core problems, evaluating

methods, and summarizing performance metrics, it lays a foundation for subsequent research. Evidence suggests that FL can achieve multi-objective optimization through collaborative training, enhancing resource utilization while reducing communication overhead and protecting privacy. However, current research is dominated by simulation and lacks standardized comparison. Future work should focus on real-world validation, robust security protocols, dynamic adaptation mechanisms, and energy-efficiency optimization. It should also explore partial offloading, advanced local training algorithms, and aerospace edge computing within the FL framework, thereby jointly promoting the practicality and scalability of FL-TO systems through technological advancements.

Acknowledgements The authors would like to express their sincere gratitude to Universiti Malaysia Sarawak (UNIMAS) for their invaluable support in facilitating the publication of this study. The institutional resources, research environment, and encouragement provided by UNIMAS have been instrumental in the successful completion of this study.

Author contributions All authors (Kong Peng, Kartinah Zen, and Lau Sei Ping) contributed to the conception and design of the study. Kong Peng performed material preparation, data collection, and analysis under the supervision of Kartinah Zen and Lau Sei Ping. Kong Peng wrote the first draft of the manuscript, and Kartinah Zen and Lau Sei Ping provided feedback and revisions. All authors read and approved the final manuscript.

Funding This work was supported by the Major Natural Science Research Project for Higher Education Institutions in Anhui Province (2024AH040016).

Data availability No datasets were generated or analysed during the current study.

Declarations

Ethics approval Not applicable.

Competing interests The authors declare no competing interests.

References

1. International Data Corporation: Worldwide IDC Global DataSphere Forecast Update, 2025–2029: Regional Analysis. (2025)
2. Islam, A., Debnath, A., Ghose, M., Chakraborty, S.: A survey on task offloading in multi-access edge computing. *J. Syst. Architect.* **118**, 102225 (2021)
3. Akhlaqi, M.Y., Hanapi, Z.B.M.: Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions. *J. Netw. Comput. Appl.* **212**, 103568 (2023)
4. Shahhosseini, S., Anzanpour, A., Azimi, I., Labbaf, S., Seo, D., Lim, S.-S., et al.: Exploring computation offloading in IoT systems. *Inf. Syst.* **107**, 101860 (2022)
5. Cong, P., Zhou, J., Li, L., Cao, K., Wei, T., Li, K.: A survey of hierarchical energy optimization for mobile edge computing: A perspective from end devices to the cloud. *ACM Comput. Surv. (CSUR)*. **53**, 1–44 (2020)
6. Chen, Y., Gu, W., Li, K.: Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning. *Int. J. Commun. Syst.* **38**, e5154 (2022)
7. Wu, Q., Wang, W., Fan, P., Fan, Q., Wang, J., Letaief, K.B.: URLLC-aware resource allocation for heterogeneous vehicular edge computing. *IEEE Trans. Veh. Technol.* **73**, 11789–11805 (2024)
8. Wang, H., Liu, C., Cai, Y., Chen, L., Li, Y.: YOLOv8-QSD: An improved small object detection algorithm for autonomous vehicles based on YOLOv8. *IEEE Trans. Instrum. Meas.* **73**, 1–16 (2024)
9. Zhou, Z., Abawajy, J.: Reinforcement learning-based edge server placement in the intelligent internet of vehicles environment. *IEEE Trans. Intell. Transp. Syst.* **PP**(99), 1–11 (2025)
10. Wang, N., Yang, W., Wang, X., Wu, L., Guan, Z., Du, X., et al.: A blockchain based privacy-preserving federated learning scheme for Internet of Vehicles. *Digit. Commun. Networks*. **10**, 126–134 (2024)
11. Yan, M., Xiong, R., Wang, Y., Li, C.: Edge computing task offloading optimization for a UAV-assisted internet of vehicles via deep reinforcement learning. *IEEE Trans. Veh. Technol.* **73**, 5647–5658 (2023)
12. Sadybekov, A.V., Katritch, V.: Computational approaches streamlining drug discovery. *Nature*. **616**, 673–685 (2023)
13. Chen, J., Mei, J., Li, X., Lu, Y., Yu, Q., Wei, Q., et al.: TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers. *Med. Image Anal.* **97**, 103280 (2024)
14. Li, X.-Q., Wang, R.-Q., Wu, L.-Q., Chen, D.-M.: Transfer learning-enabled outcome prediction for guiding CRRT treatment of the pediatric patients with sepsis. *BMC Med. Inf. Decis. Mak.* **24**, 266 (2024)
15. Mataraso, S.J., Espinosa, C.A., Seong, D., Reincke, S.M., Berenson, E., Reiss, J.D., et al.: A machine learning approach to leveraging electronic health records for enhanced omics analysis. *Nat. Mach. Intell.* **7**, 293–306 (2025)
16. Pacal, I., MaxCerVixT: A novel lightweight vision transformer-based Approach for precise cervical cancer detection. *Knowl. Based Syst.* **289**, 111482 (2024)
17. Xi, M., Dai, H., He, J., Li, W., Wen, J., Xiao, S., et al.: A lightweight reinforcement-learning-based real-time path-planning method for unmanned aerial vehicles. *IEEE Internet Things J.* **11**, 21061–21071 (2024)
18. Sezer, B.B., Turkmen, H., Nuriyev, U., PPFchain: A novel framework privacy-preserving blockchain-based federated learning method for sensor networks. *Internet Things*. **22**, 100781 (2023)
19. Hijazi, N.M., Aloqaily, M., Guizani, M., Ouni, B., Karray, F.: Secure federated learning with fully homomorphic encryption for iot communications. *IEEE Internet Things J.* **11**, 4289–4300 (2023)
20. Tahir N.U.A., Long, Z., Zhang, Z., Asim, M., ELAffendi, M.: PVswin-YOLOv8s: UAV-based pedestrian and vehicle detection for traffic management in smart cities using improved YOLOv8. *Drones*. **8**, 84 (2024)
21. Luo, H., Wei, J., Zhao, S., Liang, A., Xu, Z., Jiang, R.: Intelligent logistics management robot path planning algorithm integrating transformer and gcnn network. **2**(4), 95–112 (2025). [arXiv:25010](https://arxiv.org/abs/25010)
22. Zhu, D., Han, Z., Du, X., Zuo, D., Cai, L., Xue, C.: A hybrid model integrating fuzzy systems and convolutional factorization machine for delivery time prediction in intelligent logistics. *IEEE Trans. Fuzzy Syst.* **PP**(99), 1–13 (2024)
23. McMahan, B., Moore, E., Ramage, D., Hampson, S., y, Arcas, B.A.: Communication-efficient learning of deep networks from

- decentralized data. Artificial intelligence and statistics, PMLR; pp. 1273–82. (2017)
24. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)*. **10**, 1–19 (2019)
 25. Wang, Y., Zhang, W., Deng, H., Li, X.: Efficient resource allocation for security-aware task offloading in MEC system using DVS. *Electron. (Basel)*. **11**, 3032 (2022)
 26. Fu, Y., Li, C., Yu, F.R., Luan, T.H., Zhang, Y.: A selective federated reinforcement learning strategy for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **24**, 1655–1668 (2022)
 27. Xu, Y., Bhuiyan, M.Z.A., Wang, T., Zhou, X.K., Singh, A.K.: C-FDRL: Context-Aware Privacy-Preserving Offloading Through Federated Deep Reinforcement Learning in Cloud-Enabled IoT. *IEEE Trans. Industr. Inf.* **19**, 1155–1164 (2023). <https://doi.org/10.1109/TII.2022.3149335>
 28. Song, X.Q., Xu, B.W., Zhang, X.T., Wang, S.M., Song, T.C., Xing, G.L., et al.: Everyone-Centric Heterogeneous Multi-Server Computation Offloading in ITS with Pervasive AI. *IEEE Netw.* **37**, 62–68 (2023). <https://doi.org/10.1109/MNET.131.2200394>
 29. Mishra, K., Rajareddy, G.N.V., Ghugar, U., Chhabra, G.S., Gandomi, A.H.: A Collaborative Computation and Offloading for Compute-Intensive and Latency-Sensitive Dependency-Aware Tasks in Dew-Enabled Vehicular Fog Computing: A Federated Deep Q-Learning Approach. *IEEE Trans. Netw. Serv. Manage.* **20**, 4600–4614 (2023). <https://doi.org/10.1109/TNSM.2023.3282795>
 30. Seid, A.M., Erbad, A., Abishu, H.N., Albaser, A., Abdallah, M., Guizani, M.: Multiagent Federated Reinforcement Learning for Resource Allocation in UAV-Enabled Internet of Medical Things Networks. *IEEE Internet Things J.* **10**, 19695–19711 (2023). <https://doi.org/10.1109/JIOT.2023.3283353>
 31. Consul, P., Budhiraja, I., Garg, D., Garg, S., Kaddoum, G., Hassan, M.M.: SFL-TUM: Energy efficient SFRL method for large scale AI model's task offloading in UAV-assisted MEC networks. *Veh. Commun.* **48** (2024). <https://doi.org/10.1016/j.vehcom.2024.100790>
 32. Zhou, Z.Y., Wang, Q., Li, J.Z., Li, Z.Y.: Resource Allocation Using Deep Deterministic Policy Gradient-Based Federated Learning for Multi-Access Edge Computing. *J. Grid Comput.* **22** (2024). <https://doi.org/10.1007/s10723-024-09774-2>
 33. Wang, B., Wang, C., Huang, W., Song, Y., Qin, X.: A survey and taxonomy on task offloading for edge-cloud computing. *IEEE Access.* **8**, 186080–186101 (2020)
 34. Saeik, F., Avgeris, M., Spatharakis, D., Santi, N., Dechouniotis, D., Violos, J., et al.: Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. *Comput. Netw.* **195**, 108177 (2021)
 35. Acheampong, A., Zhang, Y., Xu, X., Kumah, D.A.: A review of the current task offloading algorithms, strategies and approach in edge computing systems. *Comput. Model. Eng. Sci.* **134**(1), 35–88 (2023)
 36. Yang, J., Shah, A.A., Pezaros, D.: A survey of energy optimization approaches for computational task offloading and resource allocation in MEC networks. *Electron. (Basel)*. **12**, 3548 (2023)
 37. Mustafa, E., Shuja, J., uz Zaman, S.K., Jehangiri, A.I., Din, S., Rehman, F., et al.: Joint wireless power transfer and task offloading in mobile edge computing: a survey. *Cluster Comput.* **25**, 2429–2448 (2022)
 38. Ahmed, M., Raza, S., Mirza, M.A., Aziz, A., Khan, M.A., Khan, W.U., et al.: A survey on vehicular task offloading: Classification, issues, and challenges. *J. King Saud University-Computer Inform. Sci.* **34**, 4135–4162 (2022)
 39. Zhang, S., Yi, N., Ma, Y.: A survey of computation offloading with task types. *IEEE Trans. Intell. Transp. Syst.* **25**(8), 8313–8333 (2024)
 40. Xia, Q., Ye, W., Tao, Z., Wu, J., Li, Q.: A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Comput.* **1**, 100008 (2021)
 41. Brecko, A., Kajati, E., Koziorek, J., Zolotova, I.: Federated learning for edge computing: A survey. *Appl. Sci.* **12**, 9124 (2022)
 42. Al-Quraan, M., Mohjazi, L., Bariah, L., Centeno, A., Zoha, A., Arshad, K., et al.: Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges. *IEEE Trans. Emerg. Top. Comput. Intell.* **7**, 957–979 (2023)
 43. Adam, M., Baroud, U.: Federated learning for IoT: Applications, trends, taxonomy, challenges, current solutions, and future directions. *IEEE Open. J. Commun. Soc.* **5**, 7539–7573 (2024)
 44. Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.-C., Yang, Q., et al.: Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutorials.* **22**, 2031–2063 (2020)
 45. Abreha, H.G., Hayajneh, M., Serhani, M.A.: Federated learning in edge computing: a systematic survey. *Sensors.* **22**, 450 (2022)
 46. Diba, B.S., Plabon, J.D., Mowla, T.J., Nahar, N., Mistry, D., Sarker, S., et al.: Open problems and challenges in federated learning for IoT: A comprehensive review and strategic guide. *Comput. Electr. Eng.* **126**, 110515 (2025)
 47. Hasan, M.K., Jahan, N., Nazri, M.Z.A., Islam, S., Khan, M.A., Alzahrani, A.I., et al.: Federated Learning for Computational Offloading and Resource Management of Vehicular Edge Computing in 6G-V2X Network. *IEEE Trans. Consum. Electron.* **70**, 3827–3847 (2024). <https://doi.org/10.1109/TCE.2024.3357530>
 48. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., Group, P.: Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Int. J. Surg.* **8**, 336–341 (2010)
 49. Krisnawijaya, N.N.K., Tekinerdogan, B., Catal, C., van der Tol, R.: Data analytics platforms for agricultural systems: A systematic literature review. *Comput. Electron. Agric.* **195**, 106813 (2022)
 50. Zhang, P., Su, Y., Li, B., Liu, L., Wang, C., Zhang, W., et al.: Deep Reinforcement Learning Based Computation Offloading in UAV-Assisted Edge Computing. *Drones.* **7**, 213 (2023)
 51. Huang, X.Y., Leng, S.P., Maharjan, S., Zhang, Y.: Multi-Agent Deep Reinforcement Learning for Computation Offloading and Interference Coordination in Small Cell Networks. *IEEE Trans. Veh. Technol.* **70**, 9282–9293 (2021). <https://doi.org/10.1109/TVT.2021.3096928>
 52. Li, J., Yang, Z.P., Wang, X.W., Xia, Y.C., Ni, S.J.: Task offloading mechanism based on federated reinforcement learning in mobile edge computing. *Digit. Commun. NETWORKS.* **9**, 492–504 (2023). <https://doi.org/10.1016/j.dcan.2022.04.006>
 53. Zang, L.Q., Zhang, X., Guo, B.R.: Federated Deep Reinforcement Learning for Online Task Offloading and Resource Allocation in WPC-MEC Networks. *IEEE ACCESS.* **10**, 9856–9867 (2022). <https://doi.org/10.1109/ACCESS.2022.3144415>
 54. Tong, Z., Wang, J.K., Mei, J., Li, K.L., Li, K.Q.: FedTO: Mobile-Aware Task Offloading in Multi-Base Station Collaborative MEC. *IEEE Trans. Veh. Technol.* **73**, 4352–4365 (2024). <https://doi.org/10.1109/TVT.2023.3329146>
 55. Tong, Z., Deng, J., Mei, J., Zhang, Y., Li, K.: Multi-objective DAG task offloading in MEC environment based on federated DQN with automated hyperparameter optimization. *IEEE Trans. Serv. Comput.* **17**(6), 3999–4012 (2024)
 56. Xiao, H., Hu, Z., Zhang, X., Xu, A., Zheng, M., Li, K.: Federated deep reinforcement learning for task offloading in MEC-enabled heterogeneous networks. *IEEE Internet Things J.* **12**(8), 10743–10756 (2025)
 57. Verma, N.K., Naik, K.J.: Graph reinforcement and asynchronous federated learning based task offloading in fog computing. *Phys. Commun.* **72**, 102776 (2025)

58. Shen, S.H., Han, Y.W., Wang, X.F., Wang, Y.: Computation Offloading with Multiple Agents in Edge-Computing-Supported IoT. *ACM Trans. Sens. Netw.* **16** (2020). <https://doi.org/10.1145/3372025>
59. Tong, Z., Wang, J.K., Mei, J., Li, K.L., Li, W.B., Li, K.Q. Multi-type task offloading for wireless Internet of Things by federated deep reinforcement learning. *FUTURE GENERATION COMPUTER, SYSTEMS-THE INTERNATIONAL: JOURNAL OF ESCIENCE* ;145:536–49. (2023). <https://doi.org/10.1016/j.future.2023.04.004>
60. Huang, Y., Li, M., Yu, F.R., Si, P., Zhang, Y.: Performance Optimization for Energy-Efficient Industrial Internet of Things Based on Ambient Backscatter Communication: An A3C-FL Approach. *IEEE Trans. Green. Commun. Netw.* **7**, 1121–1134 (2023). <https://doi.org/10.1109/TGCN.2023.3260199>
61. Liao, H.J., Jia, Z.H., Zhou, Z.Y., Wang, Y., Zhang, H., Mumtaz, S.: Cloud-Edge-End Collaboration in Air-Ground Integrated Power IoT: A Semidistributed Learning Approach. *IEEE Trans. Industr. Inf.* **18**, 8047–8057 (2022). <https://doi.org/10.1109/TII.2022.3164395>
62. Liao, H.J., Wang, Z., Zhou, Z.Y., Wang, Y., Zhang, H., Mumtaz, S., et al.: Blockchain and Semi-Distributed Learning-Based Secure and Low-Latency Computation Offloading in Space-Air-Ground-Integrated Power IoT. *IEEE J. Sel. Top. Signal. Process.* **16**, 381–394 (2022). <https://doi.org/10.1109/JSTSP.2021.3135751>
63. Demil, S., Abdmeziem, M.R.: Incentivizing task offloading in IoT: A distributed auctions-based DRL approach. *Internet Things.* **30**, 101493 (2025)
64. Benila, S., Devi, K.: Federated Synergy: Hierarchical Multi-Agent Learning for Sustainable Edge Computing in IIoT. *IEEE Access* (2025)
65. Elmazi, K., Elmazi, D., Lerga, J.: XDT-FMARL: an explainable federated multi-agent reinforcement learning framework for energy-efficient IoT task offloading. *Int. J. Web Grid Serv.* **21**, 398–423 (2025). <https://doi.org/10.1504/ijwgs.2025.150176>
66. Vishwanath, V.K., Rajendra, A.B., Gururaj, H.L.: Federated deep reinforcement learning with knowledge distillation for QoE-aware task offloading in containerized MEC. *Discover Artif. Intell.* **5**, 393 (2025). <https://doi.org/10.1007/s44163-025-00606-0>
67. Khan, L.U., Guizani, M., Muhaidat, S., Ayyash, M.: QoS-Enabled Wireless Split Federated Learning: A Reinforcement Learning and Optimization Approach. *IEEE Trans. Consum. Electron.* **71**, 8968–8978 (2025). <https://doi.org/10.1109/TCE.2025.3587176>
68. Prathiba, S.B., Raja, G., Anbalagan, S., Dev, K., Gurumoorthy, S., Sankaran, A.P.: Federated Learning Empowered Computation Offloading and Resource Management in 6G-V2X. *IEEE Trans. Netw. Sci. Eng.* **9**, 3234–3243 (2022). <https://doi.org/10.1109/TNSE.2021.3103124>
69. Sethi, V., Pal, S.: FedDOVe: A Federated Deep Q-learning-based Offloading for Vehicular fog computing. *FUTURE GENERATION Comput. SYSTEMS-THE Int. J. ESCIENCE.* **141**, 96–105 (2023). <https://doi.org/10.1016/j.future.2022.11.012>
70. Shinde, S., Bozorgchenani, A., Tarchi, D., Ni, Q.: On the Design of Federated Learning in Latency and Energy Constrained Computation Offloading Operations in Vehicular Edge Computing Systems. *IEEE Trans. Veh. Technol.* **71**, 2041–2057 (2022). <https://doi.org/10.1109/TVT.2021.3135332>
71. Pan, C., Wang, Z., Liao, H.J., Zhou, Z.Y., Wang, X.Y., Tariq, M., et al.: Asynchronous Federated Deep Reinforcement Learning-Based URLLC-Aware Computation Offloading in Space-Assisted Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **24**, 7377–7389 (2023). <https://doi.org/10.1109/TITS.2022.3150756>
72. Shen, S., Shen, G., Dai, Z., Zhang, K., Kong, X., Li, J.: Asynchronous Federated Deep-Reinforcement-Learning-Based Dependency Task Offloading for UAV-Assisted Vehicular Networks. *IEEE Internet Things J.* **11**, 31561–31574 (2024)
73. Zhao, X., Wu, Y., Zhao, T., Wang, F., Li, M.: Federated deep reinforcement learning for task offloading and resource allocation in mobile edge computing-assisted vehicular networks. *J. Netw. Comput. Appl.* **229** (2024). <https://doi.org/10.1016/j.jnca.2024.103941>
74. Fan, C., Wang, Q.: Adaptive federated deep reinforcement learning for edge offloading in heterogeneous AGI-MEC networks. *Appl. Intell.* **55**, 585 (2025)
75. Jia, M., Wu, J., Wang, X., Guo, Q.: Federated deep reinforcement learning based computation offloading in a low Earth orbit satellite edge computing system. *Front. Inform. Technol. Electron. Eng.* **26**, 805–815 (2025)
76. Zhang, P., Wang, E., Guizani, M., Liu, K., Wang, J., Tan, L.: Privacy-Preserving Task Offloading in Vehicular Edge Computing Using Federated Multi-Agent Reinforcement Learning. *IEEE Trans. Veh. Technol.* **74**, 19642–19654 (2025). <https://doi.org/10.1109/TVT.2025.3588204>
77. Abishu, H.N., Seid, A.M., Jhaveri, R.H., Gadekallu, T.R., Erbad, A., Guizani, M.: Blockchain-Empowered Resource Allocation in HAPS-Assisted IoV Digital Twin Networks: A Federated DRL Approach. *IEEE Trans. Intell. Veh.* **10**, 4710–4726 (2025). <https://doi.org/10.1109/TIV.2024.3492015>
78. Shi, J.L., Li, C.Y., Guan, Y.C., Cong, P.Y., Li, J.: Multi-UAV-assisted computation offloading in DT-based networks: A distributed deep reinforcement learning approach. *Comput. Commun.* **210**, 217–228 (2023). <https://doi.org/10.1016/j.comcom.2023.07.041>
79. Cheng, Z.P., Gao, Z.B., Liwang, M.H., Huang, L.F., Du, X.J., Guizani, M.: Intelligent Task Offloading and Energy Allocation in the UAV-Aided Mobile Edge-Cloud Continuum. *IEEE Netw.* **35**, 42–49 (2021). <https://doi.org/10.1109/MNET.010.2100025>
80. Islam, S., Badsha, S., Khalil, I., Atiquzzaman, M., Konstantinou, C.: A Triggerless Backdoor Attack and Defense Mechanism for Intelligent Task Offloading in Multi-UAV Systems. *IEEE Internet Things J.* **10**, 5719–5732 (2023). <https://doi.org/10.1109/JIOT.2022.3172936>
81. Ji, B., Zhang, M., Huang, J., Wang, Y., Xing, L., Li, T., et al.: Research on Offloading Strategy of Twin UAVs Edge Computing Tasks for Emergency Communication. *IEEE Trans. Netw. Serv. Manage.* **21**, 684–696 (2024). <https://doi.org/10.1109/TNSM.2023.3310486>
82. Agarwal, N., Joshi, S.: Federated learning-based task offloading in a uav-aided cloud computing mobile network. *IEEE Trans. Veh. Technol.* **73**, 15751–15756 (2024)
83. Consul, P., Budhiraja, I., Garg, D.: Task offloading in AIoT-enabled UAV-assisted MEC network: A digital twin-empowered approach With FedRL. *IEEE Trans. Consum. Electron.* **71**(1), 1945–1956 (2024)
84. Wei, Z., Mao, J., Li, B., Zhang, R.: Privacy-preserving hierarchical reinforcement learning framework for task offloading in low-altitude vehicular Fog computing. *IEEE Open. J. Commun. Soc.* **6**, 3389–3403(2024)
85. Li, C., Jiang, K., He, G., Bing, F., Luo, Y.: A computation offloading method for multi-UAVs assisted MEC based on improved federated DDPG algorithm. *IEEE Trans. Industr. Inf.* **20**(12), 10185–10196 (2024)
86. Arsalan, A., Umer, T., Rehman, R.A., Kim, B.-S.: Next-Gen Internet of Drones: Federated Learning and Digital Twin Synergy for Energy-Efficient Task Allocation and Seamless Service Migration. *IEEE Access* (2025)
87. Wang, C., Liu, K., Yuan, Y., Peng, S., Li, G.: Joint trajectory and offloading optimization in UAV-assisted MEC via federated multi-agent reinforcement learning and potential fields. *Comput. Netw.* **272**, 111681 (2025). <https://doi.org/10.1016/j.comnet.2025.111681>

88. Wang, S.H., Chen, M.Z., Yin, C.C., Saad, W., Hong, C.S., Cui, S.G., et al.: Federated Learning for Task and Resource Allocation in Wireless High-Altitude Balloon Networks. *IEEE Internet Things J.* **8**, 17460–17475 (2021). <https://doi.org/10.1109/JIOT.2021.3080078>
89. Rahbari, D., Alam, M.M., Le Moullec, Y., Jenihhin, M.: Fast and Fair Computation Offloading Management in a Swarm of Drones Using a Rating-Based Federated Learning Approach. *IEEE ACCESS*. **9**, 113832–113849 (2021). <https://doi.org/10.1109/ACCESS.2021.3104117>
90. Liu, L., Zhao, Y.K., Qi, F., Zhou, F.Q., Xie, W.L., He, H.R., et al.: Federated Deep Reinforcement Learning for Joint AeBSs Deployment and Computation Offloading in Aerial Edge Computing Network. *Electron. (Basel)*. **11** (2022). <https://doi.org/10.3390/electronics11213641>
91. Khan, S., Ali-Pour, A., Avgeris, M., Gascon-Samson, J., Leivadreas, A.: FEDORA: Federated Ensemble Reinforcement Learning for DAG-Based Task Offloading and Resource Allocation in MEC. *IEEE Internet Things J.* **12**, 44228–44242 (2025). <https://doi.org/10.1109/JIOT.2025.3596467>
92. Joshi, N., Budhiraja, I., Bansal, A., Kumar, N., Almuhaideb, A., Unhelkar, B.: Energy Efficient Task-Offloading for DT-Powered IRS-Aided Vehicular Communication Network Underlying UAV. *IEEE Trans. Intell. Transp. Syst.* **26**, 18019–18033 (2025). <https://doi.org/10.1109/TITS.2025.3565621>
93. Qin, P., Wang, Y., Zhang, X.: Collaborative Cloud-Edge Computing With WPT for Air-Ground Integrated IoMT Network: A URLLC Aware CAFL-Based Approach. *IEEE Trans. Commun.* **73**, 7272–7287 (2025). <https://doi.org/10.1109/TCOMM.2025.3548018>
94. Chen, Y., Zhang, J., Wang, Z., Dai, W., Gao, H., Deng, S.: Privacy-preserving knowledge distillation in latency-critical federated task offloading for consumer iot networks. *IEEE Trans. Consum. Electron.* **71**(1), 239–251 (2024)
95. Fei, Y., Fang, H., Yan, Z., Qi, L., Bilal, M., Li, Y., et al.: Privacy-aware edge computation offloading with federated learning in healthcare consumer electronics system. *IEEE Trans. Consum. Electron.* **71**(4), 4628–4639 (2025)
96. Qin, P., Wang, Y., Zhang, J., Li, P., Fu, Y.: Multi-Type Disaster Scenario Task Offloading in Air-Ground Integrated Search and Rescue Networks: A Blockchain-Assisted MFL Approach. *IEEE Trans. Veh. Technol.* **74**, 19583–19597 (2025). <https://doi.org/10.1109/TVT.2025.3587311>
97. Chen, X., Liu, G.Z.: Federated Deep Reinforcement Learning-Based Task Offloading and Resource Allocation for Smart Cities in a Mobile Edge Network. *SENSORS*. **22** (2022). <https://doi.org/10.3390/s22134738>
98. Chen, Z., Xiong, B., Chen, X., Min, G., Li, J.: Joint computation offloading and resource allocation in multi-edge smart communities with personalized federated deep reinforcement learning. *IEEE Trans. Mob. Comput.* **23**, 11604–11619 (2024)
99. Dai, L., Zeng, F., Kong, H., Cai, J., Jiang, H., Li, K.: Throughput-Aware Cooperative Task Offloading in Dynamic Mobile Edge Computing Systems. *IEEE Trans. Mob. Comput.* **24**, 13276–13292 (2025). <https://doi.org/10.1109/TMC.2025.3592450>
100. Abedi, M.R., Mokari, N., Javan, M.R., Saeedi, H., Jorswieck, E.A., Zorba, N.: Low complexity and mobility-aware robust radio, storage, computing, and cost management for cellular vehicular networks. *IEEE Trans. Veh. Technol.* **74**(2), 3327–3344 (2024)
101. Liu, S., Yang, S.Y., Zhang, H.Z., Wu, W.G.: A Federated Learning and Deep Reinforcement Learning-Based Method with Two Types of Agents for Computation Offload. *SENSORS*. **23** (2023). <https://doi.org/10.3390/s23042243>
102. Almuselem, W.: Secure Latency-Aware Task Offloading Using Federated Learning and Zero Trust in Edge Computing for IoMT. *IEEE Access* (2025)
103. Deng, M., Liang, W., Qiu, B., Xie, X.: Federated deep reinforcement learning-based task offloading strategy with game theory in vehicular edge computing. *Phys. Commun.* **72**, 102802 (2025)
104. Xu, G., Zhang, Y., Xu, X., Peng, X., Li, Q., Ma, T., Inter-Vehicle: DNN Offloading in the IoV: Overcoming Vehicle Heterogeneity With Federated Split Hybrid Learning. *IEEE Trans. Veh. Technol.* **74**, 19523–19538 (2025). <https://doi.org/10.1109/TVT.2025.3585175>
105. Liu, Y., Jiang, L., Qi, Q., Xie, S.L.: Energy-Efficient Space-Air-Ground Integrated Edge Computing for Internet of Remote Things: A Federated DRL Approach. *IEEE Internet Things J.* **10**, 4845–4856 (2023). <https://doi.org/10.1109/JIOT.2022.3220677>
106. Wang, Q., Chen, S.G., Wu, M.: Incentive-Aware Blockchain-Assisted Intelligent Edge Caching and Computation Offloading for IoT. *ENGINEERING*. **31**, 127–138 (2023). <https://doi.org/10.1016/j.eng.2022.10.014>
107. Xiao, L., Shan, H., Zhu, J., Mao, R., Pan, S.: FD3QN: A federated deep reinforcement learning approach for cross-domain resource cooperative scheduling in hybrid cloud architecture. *Informatica* **49**(10), 127–146 (2025)
108. Singh, P., Hazarika, B., Singh, K., Huang, W.-J., Duong, T.Q.: GenAI-enhanced federated multi-agent DRL for digital twin-assisted IoV networks. *IEEE Internet Things J.* **12**(5), 4834–4851 (2025)
109. Devarajan, G.G., Thangam, S., Alenazi, M.J.F., Kumaran, U., Chandran, G., Bashir, A.K.: Federated learning and blockchain-enabled framework for traffic rerouting and task offloading in the Internet of Vehicles (IoV). *IEEE Transact. Consume. Electron.* **71**(2), 3817–3831 (2025)
110. Huang, J., Zhang, M., Wan, J., Chen, Y., Zhang, N.: Joint data caching and computation offloading in UAV-assisted Internet of Vehicles via federated deep reinforcement learning. *IEEE Trans. Veh. Technol.* **43**(11), 14920–14934 (2024)
111. Li, Q.Q., Cui, Y.P., Song, T., Zheng, L.J.: Federated Multiagent Actor-Critic Learning Task Offloading in Intelligent Logistics. *IEEE Internet Things J.* **10**, 11696–11707 (2023). <https://doi.org/10.1109/JIOT.2023.3244783>
112. Su, C., Wei, J.G., Lin, D.Y., Kong, L.H., Guan, Y.L.: Task Offloading and Resource Allocation for Fog Computing in NG Wireless Networks: A Federated Deep Reinforcement Learning Approach. *IEEE Internet Things J.* **11**, 6802–6816 (2024). <https://doi.org/10.1109/JIOT.2023.3314496>
113. Chen, Z., Wang, X.: Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach. *EURASIP J. Wirel. Commun. Netw.* **2020**, 188 (2020)
114. Moon, S., Lim, Y.: Federated Deep Reinforcement Learning Based Task Offloading with Power Control in Vehicular Edge Computing. *SENSORS*. **22** (2022). <https://doi.org/10.3390/s22249595>
115. Zhang, Q.X., Wen, H., Liu, Y., Chang, S., Han, Z.: Federated-Reinforcement-Learning-Enabled Joint Communication, Sensing, and Computing Resources Allocation in Connected Automated Vehicles Networks. *IEEE Internet Things J.* **9**, 23224–23240 (2022). <https://doi.org/10.1109/JIOT.2022.3188434>
116. Song, X., Chen, Q., Wang, S., Song, T.: Cross-domain resources optimization for hybrid edge computing networks: Federated DRL approach. *Digit. Commun. Networks*. **11**, 1797–1808 (2025). <https://doi.org/10.1016/j.dcan.2024.03.006>
117. Zhang, J., Huang, F., Zhu, S., Xiao, X.: A Resource Allocation Strategy in Internet of Vehicles Based on Multi-Task Federated Learning and Incentive Mechanism. *IEEE Trans. Intell. Transp.*

- Syst. **26**, 17686–17697 (2025). <https://doi.org/10.1109/TITS.2025.3528969>
118. Li, C., Deng, C., Zhang, Y., Wan, S.: Federated Meta-Learning Based Computation Offloading Approach With Energy-Delay Tradeoffs in UAV-Assisted VEC. *IEEE Trans. Mob. Comput.* **24**, 10978–10991 (2025). <https://doi.org/10.1109/TMC.2025.3573278>
 119. Wu, H., Gu, A., Liang, Y.: Federated reinforcement learning-empowered task offloading for large models in vehicular edge computing. *IEEE Trans. Veh. Technol.* **43**(11), 14920–14934 (2024)
 120. Chen, X., Xiao, B., Lin, X., Chen, Z., Min, G.: Multi-agent collaboration for vehicular task offloading using federated deep reinforcement learning. *IEEE Trans. Mob. Comput.* **24**(9), 8856–8869 (2025)
 121. Wang, X., He, C., Jiang, W., Wang, W., Liu, X.: Generative AI based dependency-aware task offloading and resource allocation for UAV-assisted IoV. *IEEE Open. J. Commun. Soc.* **6**, 2125–2138 (2025)
 122. Hou, W.J., Wen, H., Song, H.H., Lei, W.X., Zhang, W.: Multi-agent Deep Reinforcement Learning for Task Offloading and Resource Allocation in Cybertwin-Based Networks. *IEEE Internet Things J.* **8**, 16256–16268 (2021). <https://doi.org/10.1109/JIO T.2021.3095677>
 123. Jin, Z.Y., Wang, Y.J., Lv, J.Y.: Edge Computing Task Offloading of Internet of Vehicles Based on Improved MADDPG Algorithm. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*; 18:327–347. (2024). <https://doi.org/10.3837/tiis.2024.02.004>
 124. Fathima, A.S., Basha, S.M., Ahmed, S.T., Khan, S.B., Asiri, F., Basheer, S., et al.: Empowering consumer healthcare through sensor-rich devices using federated learning for secure resource recommendation. *IEEE Trans. Consum. Electron.* **71**(1), 1563–1574 (2025)
 125. Shabir, B., Rahman, A.U., Malik, A.W., Buyya, R., Khan, M.A.: A federated multi-agent deep reinforcement learning for vehicular fog computing. *J. SUPERCOMPUTING.* **79**, 6141–6167 (2023). <https://doi.org/10.1007/s11227-022-04911-8>
 126. Song, X.Q., Cheng, M.Q., Lei, L., Yang, Y.: Multitask and Multiobjective Joint Resource Optimization for UAV-Assisted Air-Ground Integrated Networks Under Emergency Scenarios. *IEEE Internet Things J.* **10**, 20342–20357 (2023). <https://doi.org/10.1109/JIOT.2023.3284425>
 127. Ye, W.H., Zheng, K., Wang, Y.Y., Tang, Y.L.: Federated Double Deep Q-Learning-Based Computation Offloading in Mobility-Aware Vehicle Clusters. *IEEE ACCESS.* **11**, 114475–114488 (2023). <https://doi.org/10.1109/ACCESS.2023.3324718>
 128. Quan, H., Zhang, Q., Zhao, J.: Federated learning assisted intelligent IoV mobile edge computing. *IEEE Trans. Green. Commun. Netw.* **9**, 228–241 (2024)
 129. Liu, Z., Li, D., Yang, P.: KECO: toward efficient task offloading in mobile edge computing with federated knowledge distillation. *PeerJ Comput. Sci.* **11**, e3021 (2025)
 130. Paul, A., Singh, K., Li, C.-P.A., Multi-Agent: Federated DRL Model for Vehicular Task Offloading in WPT-Aided eROAD Environment. *IEEE Trans. Intell. Transp. Syst.* **26**, 17723–17739 (2025). <https://doi.org/10.1109/TITS.2025.3537983>
 131. Shahidinejad, A., Farahbakhsh, F., Ghobaei-Arani, M., Malik, M.H., Anwar, T.: Context-Aware Multi-User Offloading in Mobile Edge Computing: a Federated Learning-Based Approach. *J. Grid Comput.* 2021;19. <https://doi.org/10.1007/s10723-021-09559-x>
 132. Zhang, X.J., Wu, W.G., Wang, J.Y., Liu, S.: BiLSTM-based Federated Learning Computation Offloading and Resource Allocation Algorithm in MEC. *ACM Trans. Sens. Netw.* **19** (2023). <https://doi.org/10.1145/3579824>
 133. Sun, F., Zhang, Z.J., Chang, X.L., Zhu, K.G.: Toward Heterogeneous Environment: Lyapunov-Orientated ImpHetero Reinforcement Learning for Task Offloading. *IEEE Trans. Netw. Serv. Manage.* **20**, 1572–1586 (2023). <https://doi.org/10.1109/TNSM.2023.3266779>
 134. Dai, Y.Y., Zhao, J.T., Zhang, J., Zhang, Y., Jiang, T.: Federated Deep Reinforcement Learning for Task Offloading in Digital Twin Edge Networks. *IEEE Trans. Netw. Sci. Eng.* **11**, 2849–2863 (2024). <https://doi.org/10.1109/TNSE.2024.3350710>
 135. Song, X., Chen, Q., Wang, S., Song, T., Xu, L.: Hybrid multi-server computation offloading in air-ground vehicular networks empowered by federated deep reinforcement learning. *IEEE Trans. Netw. Sci. Eng.* **11**(6), 5175–5189 (2024)
 136. Kumar, A.S., Zhao, L., Fernando, X.: Asynchronous federated based vehicular edge computation offloading. *IEEE Trans. Veh. Technol.* **73**(12), 19350–19361 (2024)
 137. Wu, C., Li, L., Zhang, L., Gao, C., Wu, X., Xiao, S.: Efficient GAN-Based Federated Optimization for Vehicular Task Offloading With Mobile Edge Computing in 6G Network. *IEEE Internet Things J.* **12**, 2736–2748 (2025)
 138. Chen, Z., Huang, Z., Zhang, J., Cheng, H., Li, J.: Resource allocation and collaborative offloading in multi-UAV-assisted IoV with federated deep reinforcement learning. *IEEE Internet Things J.* **12**(5), 4629–4640 (2024)
 139. Zou, W., Zhang, R., Xun, Y.: On a Federated-Learning-Based Computation Offloading Strategy for Nonterrestrial-Network-Assisted Internet of Medical Things. *IEEE Internet Things J.* **12**, 21280–21289 (2025). <https://doi.org/10.1109/JIOT.2025.3546812>
 140. Zhao, H., Li, Y., Pang, Z., Ma, Z.: Federated Multi-Agent DRL for Task Offloading in Vehicular Edge Computing. *Electron. (Basel).* **14**, 3501 (2025)
 141. Kaur, M., Cheng, H., Kourtessis, P.: Quantized Graph-Based Personalized DRL for Dependency-Aware Task Offloading in Heterogeneous Edge Networks. *IEEE Trans. Consum. Electron.* **71**, 8626–8640 (2025). <https://doi.org/10.1109/TCE.2025.3598355>
 142. Wang, D., Wang, T., Feng, X., Wu, Y.: Task offloading and computing in vehicular edge networks: A joint of dynamic game and federated learning. *Alexandria Eng. J.* **133**, 688–699 (2025). <https://doi.org/10.1016/j.aej.2025.11.012>
 143. McClellan, M., Cervelló-Pastor, C., Sallent, S.: Deep learning at the mobile edge: Opportunities for 5G networks. *Appl. Sci.* **10**, 4735 (2020)
 144. Choudhury, A., Ghose, M., Islam, A.: Machine learning-based computation offloading in multi-access edge computing: A survey. *J. Syst. Architect.* **148**, 103090 (2024)
 145. Sadatdiyev, K., Cui, L., Zhang, L., Huang, J.Z., Salloum, S., Mahmud, M.S.: A review of optimization methods for computation offloading in edge computing networks. *Digit. Commun. Networks.* **9**, 450–461 (2023)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Kong Peng received the master degree in computer science from Hefei University of Technology in 2011. He is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak. His primary research interests include federated learning, mobile edge computing, and game theory.



Kartinah Zen received the Ph.D. degree in sensor network from Edith Cowan University (ECU), Australia. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS). She is also a Research Fellow with the Centre of Excellence for Rural Informatics, where she is working on project in wireless sensor networks. Her research interests include wireless sensor networks, data transmission, and sensor-related network technology and applications.



Lau Sei Ping is a senior lecturer at the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS), and currently serves as the Deputy Director of the Institute of Social Informatics and Technological Innovations (ISITI). He obtained his PhD from the University of Southampton, UK, in 2015. His research focuses on wireless sensor networks, smart cities, energy-efficient systems, and applied computing. Dr. Lau has contributed

significantly to projects involving intelligent street lighting and rural electrification. With over 15 years of experience in research and teaching, he has published widely and remains actively engaged in both academic and practical technological innovation.