applied sciences

MDPI

Review

Applicability of Deep Reinforcement Learning for Efficient Federated Learning in Massive IoT Communications

Prohim Tam [1,†], Riccardo Corrado [2,3,†], Chanthol Eang [1,†] and Seokhoon Kim [1,4,*]

[1] Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea
[2] Department of Information and Communications Technology, American University of Phnom Penh, Phnom Penh 12106, Cambodia
[3] Cambodian Ministry of Post and Telecommunications, Phnom Penh 12200, Cambodia
[4] Department of Computer Software Engineering, Soonchunhyang University, Asan 31538, Republic of Korea
* Correspondence: seokhoon@sch.ac.kr
† These authors contributed equally to this work.

Abstract: To build intelligent model learning in conventional architecture, the local data are required to be transmitted toward the cloud server, which causes heavy backhaul congestion, leakage of personalization, and insufficient use of network resources. To address these issues, federated learning (FL) is introduced by offering a systematical framework that converges the distributed modeling process between local participants and the parameter server. However, the challenging issues of insufficient participant scheduling, aggregation policies, model offloading, and resource management still remain within conventional FL architecture. In this survey article, the state-of-the-art solutions for optimizing the orchestration in FL communications are presented, primarily querying the deep reinforcement learning (DRL)-based autonomy approaches. The correlations between the DRL and FL mechanisms are described within the optimized system architectures of selected literature approaches. The observable states, configurable actions, and target rewards are inquired into to illustrate the applicability of DRL-assisted control toward self-organizing FL systems. Various deployment strategies for Internet of Things applications are discussed. Furthermore, this article offers a review of the challenges and future research perspectives for advancing practical performances. Advanced solutions in these aspects will drive the applicability of converged DRL and FL for future autonomous communication-efficient and privacy-aware learning.

Keywords: communication-efficient learning; deep reinforcement learning; federated learning; massive Internet of Things; policy optimization; self-organizing networks

Citation: Tam, P.; Corrado, R.; Eang, C.; Kim, S. Applicability of Deep Reinforcement Learning for Efficient Federated Learning in Massive IoT Communications. *Appl. Sci.* **2023**, *13*, 3083. https://doi.org/10.3390/app13053083

Academic Editors: Joon-Min Gil and Jisu Park

Received: 26 January 2023
Revised: 19 February 2023
Accepted: 26 February 2023
Published: 27 February 2023

1. Introduction

5G and beyond cellular networks have experienced an advancement in their enabler technologies and dense deployment, drawing major attention from time-sensitive Internet of Things (IoT) applications with mission-critical Quality of Service (QoS) expectations. IoT deployment is expected to reach 30.9 billion connected devices by 2025, generating exponential data growth from different application source taxonomies, mission criticalities, and privacy constraints [1]. The challenges are raised in real-world scenarios regarding data privacy, automation of big data management, and latency-efficient connectivity. In this context, the capability of data exposure in privacy-restricted sectors, e.g., Internet of Healthcare Things, Internet of Vehicles (IoV), and Industrial IoT (IIoT), needs to tackle privacy-preserving awareness following the General Data Protection Regulation (GDPR) [2]. In massive IoT services with enormous volume, velocity, and variety of data features, artificial intelligence (AI) algorithms are used to enhance operational efficiency, make smart predictions, respond to instantaneous actions, and enhance service scalability [3,4]. Additionally, AI aims to enable the practicability of empowering intelligent management and orchestration for IoT communications within wired/wireless networks

and software-defined networking (SDN) in various aspects such as, and not limited to, resource allocation, offloading decision, routing optimization, caching placement, and network function chaining policy [5–7].

In this section, the motivational statement of this article is given by mentioning the main reference development of zero-touch network and service management (ZSM) and edge intelligence (EI), which provide the high-applicability principles of closed-loop network automation and collaborative computing architecture for intelligence on the edge. The problem statement presents the existing challenges, such as data privacy, self-adaptive resource placement, offloading decisions, high latency to convergence, and fronthaul congestion, on ZSM and EI scenarios.

Furthermore, the goals of this paper are outlined to overcome the problem statement, consisting of (1) deep reinforcement learning (DRL) for autonomous experience-aware policy control and (2) federated learning (FL) for a privacy-enhanced collaborative AI framework. The confluence of DRL autonomy and FL systems is given to motivate the applicability of a complete zero-touch edge model deployment for massive AI-based IoT applications and communication efficiencies.

1.1. Motivational Statement

To drive intelligent next-generation network management, ZSM and EI are notable frameworks that appraise a complete self-organizing service lifecycle and competence of AI execution in distributed areas. We consider ZSM and EI as the foremost objectives, which later lead to the necessity of DRL-based FL approaches to (1) tackle several challenging issues and (2) bring auxiliary autonomy and privacy-aware functional capabilities.

1.1.1. Zero-Touch Network and Service Management

Standard telecommunication entities suggest self-organizing networks (SON) with AI-assisted automation control by releasing technical specifications, testbeds, and application programming interfaces (APIs). Specifically, the 3rd Generation Partnership Project (3GPP) presents an enabler for autonomous networks by presenting network data analytics function to associate with service-based architecture and expose data gathering/analysis capability for applying data-driven AI models in 5G applications [8].

From active phases of generic autonomic networking architecture (GANA) and experiential networked intelligence (ENI) using closed-loop AI architecture for SON, the European Telecommunications Standards Institute (ETSI) keeps emphasizing the ZSM aspect to describe the use cases in network functions virtualization (NFV), edge computing (EC), autonomous policy configuration, and service automation [9–11]. The Internet Engineering Task Force (IETF) has developed an autonomic networking integrated model and approach (ANIMA) for enabling the applicability of self-functions in configuration, optimization, protection, and healing [12]. ANIMA motivates AI-assisted approaches for autonomic orchestration in networking infrastructure, control plane, and slice management. Moreover, The International Telecommunication Union (ITU) organized a focus group on machine learning (ML) for future networks, including 5G, to study and plan the execution and evaluation of data handling, APIs, system architecture, standard protocols, and intelligence-level future network management [13].

To drive the adoption between 5G end-to-end (E2E) architecture and AI-enabled applicability, next-generation mobile networks (NGMN) alliance has presented documentation and capabilities on emerging use cases, requirements, and enabler specifications as an autonomic networking framework [14]. Furthermore, deep learning (DL)-based network slicing approaches are further referenced by NGMN documentation for assisting E2E procedures of various 5G usage services, including enhanced mobile broadband, ultra-reliable and low-latency communications, and massive machine-type communications [15]. Additionally, a joint testbed federation between ITU, ETSI, and the Institute of Electrical and Electronics Engineers (IEEE) is expected to be established for 5G and beyond networks with applied AI models in the system architecture, interoperability, and reference APIs [16].

1.1.2. Edge Intelligence

With the proliferation of IoT, data-driven AI solutions are applied for (1) handling the enormous data features and heterogeneous IoT taxonomies, (2) analyzing the underlying hidden patterns of structured, semi-structured, quasi-structured, and unstructured data, and (3) making reliable decision outcome in terms of classifications, predictions, and recommendations [17,18]. To make this solution sufficient in real-world applications, the development process has to consider computational resource placement, leading to the confluence of AI with adequate cloud computing and EC capacities, known as cloud intelligence (CI) and EI, respectively [19]. Figure 1 illustrates the overview between CI and EI, each of them consisting of three primary tiers: end devices (ED), edge, and cloud. In CI, model training and inference are performed in central cloud servers, which raised various challenging drawbacks while gathering the local data, including backhaul congestion, high latency, privacy leakage, and insufficient bandwidth consumption.

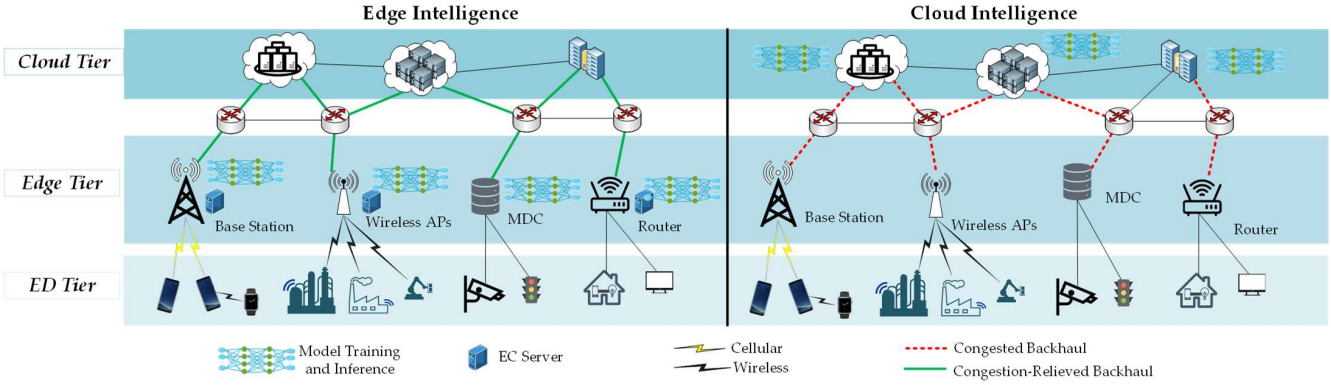

Figure 1. Cloud and edge intelligence.

EC brings solutions for traditional cloud computing by allowing IoT nodes to offload the tasks wholly/partially toward computing entities in close proximity, effectively alleviating the backbone congestion, lowering the computing delay, and saving local IoT power utilization [20]. Based on different service deployments, EC servers are attached to particular access points (APs) such as base stations, wireless APs, routers, etc. A micro-data center (MDC) at the edge assists the applicability of EI, predominantly for smart cities. The model can be trained and evaluated for clustered ED and services with specific local context awareness by leveraging sufficient computing resources. With EC-enabled collaboration, the advancement of communication, computation, control, and caching (4C) is jointly optimized in edge networks [21,22]. EI tackles these issues by integrating DL and EC to enable efficient edge learning in distributed networks [23–25].

From real-time communication perspectives, integrated DL requires a processing time of critical upper-bound tolerable delay in the execution procedure between input, hidden, and output layers. Within DL modeling, deep neural networks (DNN) are a prominent class of several popular DL algorithms such as convolutional neural networks, artificial neural networks, multi-layer perceptron, and recurrent neural networks, which necessitates minimizing the abundance of execution time. With adequate support from EC, partial DL model training and inference are executable in edge networks to support practical collaborative learning systems [26].

1.2. Problem Statement

Within ZSM and EI architectures, problems are upraised that downgrade the privacy regulations, practical requirements, and autonomy capabilities. The context of mandatory problems is given in this subsection, which connects to the necessity of solutions from the confluence of DRL and FL.

In the ZSM concept, the essential contribution consists of the enablement of a long-term SON without manual modification or administration. Therefore, the systematic

network architecture requires adequate computing entities and feature-gathering modules to withstand AI control mechanisms. The programmable virtualization environment must be initialized for observing the real-time states of the infrastructure plane, configuring the policy autonomously, and modifying the flow rules proactively/reactively. The communication and computation resources need (1) to be well-allocated adaptively in different congestion states, (2) to be reserved with priority-specified properties for non-mission-critical/mission-critical services, (3) to benefit from an adjusted dynamic placement for instantiating virtual network functions (VNFs), and (4) to experience load balancing to avoid overloading/low-loading, which cause bottlenecks or extravagant placement [27]. Furthermore, in the ZSM perspective, binary/partial IoT task offloading requires intelligent steering toward edge-cloud systems to minimize the total task execution times and prevent the high contingency of task termination, particularly for computation-intensive tasks in time-sensitive applications. By overcoming the limitations and activating the self-managing capabilities of AI-driven ZSM, the network architecture can operate with complete autonomy and be maintainable for facilitating DL model builders at ED-level or edge-level by (1) organizing EC resources and DL selection for intelligent services, (2) offering long-term resource placement for DL computation, and (3) orchestrating matching policies for cloud-edge collaboration in model co-training/inference [27,28].

EI adheres to four primary phases along with different problem types such as (1) edge caching for collecting data from the ED tier and storing it in the edge tier, which causes the problems on what and where to cache, (2) edge training for using the cached data and performing the model learning procedures, which brings several issues in terms of training architecture and optimization of the model key performance indicator (KPI), (3) edge inference for evaluating the trained model in the edge application architecture, which faces challenging drawbacks in multi-service applicability and model designs, and (4) edge offloading for making decisions on edge server selection and orchestrating the computation policy [18,19]. In terms of model deployment, the main aspects for further enhancements are upraised as follows:

- Data privacy: GDPR suggests the local data remain in local authentication or data accessibility terms, which primarily restricts the sharing of privacy-sensitive information to other nodes. In edge-level model training/inference, the training data are gathered to the greatest extent for optimizing the learning parameters and constructing an accurate final model in the edge tier. During the caching process, the raw data from local ED are transferred across nodes and fully uploaded to the edge tier, which completely burdens the fronthaul networks, causes communication overhead, and violates privacy-preserving obligations.
- Self-adaptive resource placement: within IoT architecture, the local energy and computing capacities are constrained for model task completion, which requires edge-cloud assistance with specified resource allocation (e.g., bandwidth, computing, and storage). To contribute the intelligence for future ZSM, self-organizing VNFs and virtual machine (VM) placement solutions should be deployed for resource-aware computation and scalable multi-service parallelism.
- Offloading decisions: to construct models whether in edge or cloud tiers, the offloading of high data volumes causes heavy congestion and consumes abounding communication resources. Therefore, the local tasks must be minimized and offloaded to servers with adequate capacities depending on each streaming timeslot.
- High latency to convergence: with loss optimization in DL (e.g., using gradient descent), the model parameters are iterated through numerous steps until the convergence point. With non-optimal computing capacities, high data drops, and non-applicable model hyperparameters (e.g., from unfamiliar taxonomies and nonstandard data), the convergence expectation can cause long delays. Therefore, the drawback of non-collaborative architecture for model training causes the overall procedures to reach over the upper-bound tolerable delay threshold and brings an unsatisfied quality of experience.

- Fronthaul congestion: by uploading raw original data into the training modules in edge servers, the burden of fronthaul resources and increment of communication costs bring great challenges for AI-based model deployment, particularly for future massive intelligent IoT applications. The completion latency can be higher than the KPI of real-time decision requirements.

1.3. Paper Contributions

With the above-mentioned motivational and problem statements, this paper queries the state-of-the-art approaches, including FL framework and DRL agent, with promising beneficial factors for self-managing collaborative AI model deployment in future intelligent IoT applications. Several existing surveys have presented DRL for IoT, FL in edge networks, CI/EI architectures, and ZSM perspective for 5G networks; however, the correlations between DRL and FL for long-term and closed-loop orchestration are not fully considered in a massive IoT environment. This article aims to contribute an in-depth review of using DRL for optimizing (edge) FL model communications, which subsequently supports a major section of preeminent objectives in ZSM and CI/EI. Table 1 presents the important acronyms with descriptions used in the paper. The summary of our contributions is given as follows:

- We discuss the key elements and execution flows of FL implementation in IoT scenarios. Edge FL is also discussed to illustrate EC-assisted model updates for local IoT devices. Furthermore, existing solutions to tackle data privacy, training, inference, compression, edge aggregation, IoT participant selection, and resource optimization are outlined to specify the competence of (edge) FL-based approaches. Federated optimization is described in this paper for maintaining massive IoT taxonomies, data heterogeneity, and the complexity of practical aggregation.
- We review the DRL components for IoT networks and the system architecture for driving the DRL agent applicable in network virtualization. The observability, configurability, and computability of the network architecture for DRL implementation are presented, which specify the set of states, actions, and immediate/long-term rewards used by researchers to construct autonomous policy management between the agent and IoT environment.
- We discuss DRL as an enabler approach for assisting FL algorithms in massive IoT use cases. DRL-based FL policy optimization is coined by considering various aspects such as resource optimization, model offloading decisions, model update scheduling, and aggregation policies.
- We provide various scenarios of IoT application deployments using DRL and FL to engage in each domain's feature with different state observations, action adjustments, and reward formulations. The heterogeneity of DRL- and FL-enabled IoT services, including IIoT, smart automation, medical services, IoV, and environmental context detection, is reviewed as a guideline for applying in real-world use cases.
- We point out the challenging issues and future research directions for enhancing the applicability, adaptability, privacy, autonomy, and optimality of DRL-based FL in massive IoT network architecture.

1.4. Paper Organizations

The rest of the paper is structured as follows (Figure 2). Section 2 outlines the preliminary studies on FL and DRL. Section 3 discusses the correlations between DRL and FL. Section 4 discusses the promising DRL-based enabler optimization approaches for an efficient FL (eFL) framework in massive IoT. Section 5 provides a wide taxonomy of IoT application deployment strategies using DRL/FL. The challenges and future perspectives are highlighted in Section 6. Finally, Section 7 concludes our paper.

Table 1. List of Important Acronyms.

Abbreviation	Description
AI	Artificial Intelligence
APIs	Application Programming Interfaces
APs	Access Points
CI	Cloud Intelligence
DL	Deep Learning
DNN	Deep Neural Networks
DRL	Deep Reinforcement Learning
E2E	End-to-End
EC	Edge Computing
ED	End Devices
eFL	Efficient Federated Learning
EI	Edge Intelligence
FA	Federated Analytics
FL	Federated Learning
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
IID	Independent-and-Identically-Distributed
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoV	Internet of Vehicle
KPI	Key Performance Indicator
MDC	Micro-Data Center
ML	Machine Learning
NFV	Network Functions Virtualization
QoS	Quality of Service
SDN	Software-Defined Networking
SGD	Stochastic Gradient Descent
SON	Self-Organizing Networks
VM	Virtual Machine
VNFs	Virtual Network Functions
VNF-FGs	Virtual Network Function Forwarding Graphs
ZSM	Zero-Touch Network and Service Management

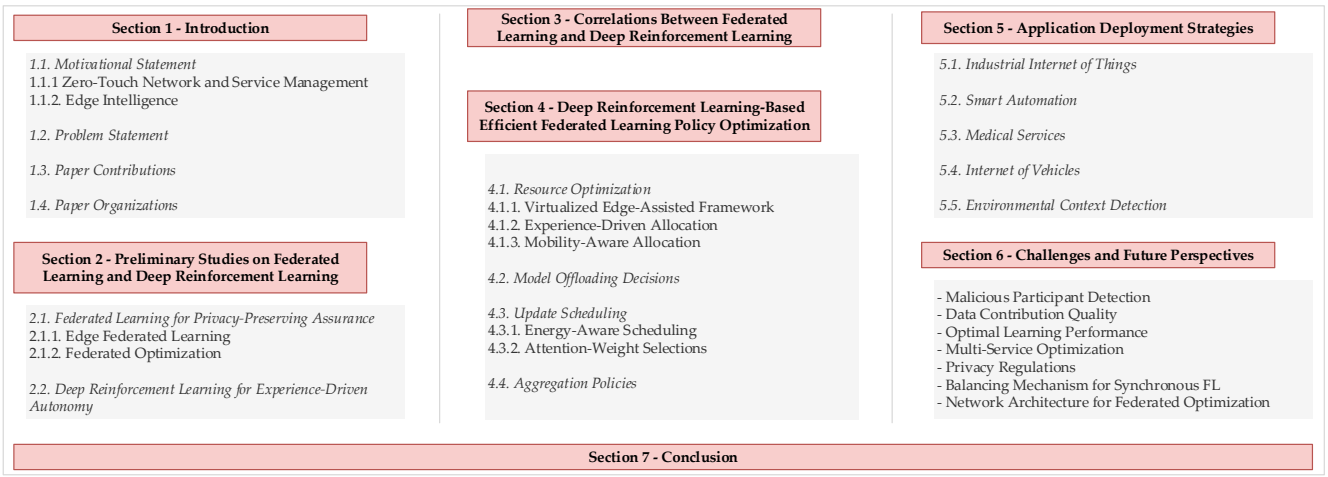

Figure 2. Structure of the paper.

2. Preliminary Studies on Federated Learning and Deep Reinforcement Learning

2.1. Federated Learning for Privacy-Preserving Assurance

Currently, IoT applications consist of privacy-sensitive information belonging to exclusive users, organizations, or business companies, which necessitate obeying GDPR in data gathering or third-party transmission in AI-based services. FL was first introduced in

2016 by Google researchers [29–31] to emphasize the concept of distributing the models to be trained by local data owners instead of uploading and storing raw data centrally. FL is a privacy-preserving, collaborative, and resource-efficient learning framework to aggregate multi-dimensional decentralized local models in a centralized global parameter server for constructing a reliable learning model. Table 2 presents the contributions of existing selected surveys on FL, which is drawing major attention from researchers, deployment engineers, and organizations. There are six main phases to execute iteratively in FL for the IoT environment, including (1) the architecture of collaborative learning system, (2) global model distribution, (3) local model computation, (4) loss optimization, (5) model transmission, and (6) global model aggregation [31,32].

Table 2. Contributions of Existing Selected Surveys on Federated Learning.

Domains	Summary of Contributions	Ref.	Year
Background and classification of practical FL scenarios	The definition, enabling technologies, taxonomy, application types, challenging issues, and future directions of FL	[32]	2021
FL for resource-constrained IoT	Collaborative learning and optimization approaches to efficiently train the models in a heterogeneous IoT environment	[33]	2022
FL for mobile edge networks/FL at mobile edge networks	Providing FL approaches for optimizing mobile edge networks, reviewing frameworks for FL execution, and discussing implementation challenges	[34]	2020
Comprehensive learning on FL	The enabling technologies, frameworks, protocols, application scenarios, and challenges of FL	[35]	2020
FL topics and research domains	An in-depth study on FL architectures/taxonomies, system designs, on-site deployment scenarios, and future research directions	[36]	2021

Conventional FL procedures are modified and enhanced throughout various research studies for optimizing the performance in communication perspectives within resource-constrained IoT environments. The consideration of client prioritization, accuracy performance, and specific data types, e.g., independent- and identically distributed (IID) and non-IID, must be comprehensively emphasized with detailed specifications in network controller and federated settings. Moreover, local model training/updates are not completely sufficient to execute without possible termination before fully interacting with the edge server. System heterogeneity on local participants (e.g., interface or channel) requires further handling schemes. Due to the resource constraints of local devices, deep models cannot completely compute and upload for a satisfying accuracy objective. Therefore, to improve the FL performance, an intelligent computational framework for edge applications with multi-exit-based and greedy-approach-based algorithms is discussed by splitting the main models into sub-models with various scale sizes to prioritize the local participants that have inadequate computing capacities [17,37]. To formulate and handle the problems efficiently, latency-constrained and optimization models must correspond with the observable features of training-time metrics, allocation, and scheduling procedures.

FL introduces a sufficient framework that strongly takes client personalization into consideration, and it is highly efficient for multi-level IoT privacy environments. To fully make this framework applicable, the implementation requires optimizing the resource

allocation, improving the security, and increasing the robustness. A resource-optimized FL for IoT was proposed in [38] by introducing a dispersed framework that aimed to optimize the placement and allocation with the robustness of distributed learning models. The minimization approach using an integer linear problem was implemented in this system architecture. To specifically tackle the problems in fine-grained detail, a model was split into two sub-problems, namely association and resource block allocation. The association problem considered the adjustment of fixed resource blocks within devices, and the resource block allocation problem tackled the association of fixed devices. Moreover, a further improvement on adaptive FL in resource-constrained systems was presented in [39] by tackling the challenging issues of model parameters from numerous edge nodes with its non-IID data. The convergence of gradient descent was analyzed to cooperate with a control scheme for adaptively managing the global aggregation in mission-critical circumstances. The aim of [39] was to minimize the loss within resource-constrained conditions by optimal tradeoffs between two primary resource types, namely in local model update and global aggregation. In [40], an enhanced asynchronous FL technique was proposed for alleviating the overhead communication latency and advancing the FL performance through adjusted weight aggregation. The modification of DNN layers was labeled as shallow and deep, which learned on general features of various tasks/datasets and ad hoc features, respectively. Well-defined weights lead to fast convergence, less latency per communication round, and accuracy improvement.

Furthermore, another use case to be addressed is for non-IID, which possibly degrades the FL convergence expectation. By emphasizing each local model accuracy, user-based service with non-IID data distribution can perform well. To illustrate the applicability in this scenario, [41] proposed a multi-task FL for individual DNNs. The proposed scheme modified the conventional FL by authorizing the participants to customize their DNNs that fit the best for their non-IID data distribution. A new KPI of FL performance on user model accuracy was given for extending the measurement quality. However, privacy-enhanced collaborative architecture and model customization come with a tradeoff against communication costs within FL training phase. To outcome these issues, [42] offered robust and communication-efficient FL solutions in non-IID scenarios, which used a sparse ternary compression framework. The proposed compression scheme modified the existing compression method and handled two major problems: (1) the compression of downstream and (2) caching updates for participant synchronization. In [43], an optimized FL with DRL was proposed for handling the client selection in non-IID data by leveraging the self-managing capability of the agent. The agent observed the state of the global model weights from each device in a particular communication round. The actions aimed to select a subset of clients for participating in the current training iteration by sampling the top-batch size of optimal clients. The reward valuation considered the possibility of accuracy increment. Based on FL and DRL convergence, the control policies can speed up the convergence, self-maintain the update/aggregation, and intelligently select an optimal participant batch following experience-driven and state-considered information in different congestion statuses. The bias and loss can be well-optimized, particularly for non-IID.

2.1.1. Edge Federated Learning

Edge FL consists of edge aggregator orchestration to reduce the number of direct round communication between local participants and server, which is significant for communication-critical and computation-limited IoT clients. There are complementary studies to be outlined in this subsection for analyzing the deployment applicability of edge FL in massive IoT. Edge FL leverages the EC resources for enabling a converged beneficial factor of FL and edge computation offloading [44–47]. A three-tier architecture between participant, edge, and central FL allows the round communications of participant updates and global aggregation to be client-edge and edge-cloud, respectively. To improve client-specific service, each label can aggregate within the distributed edge server for understanding the service efficiency and alleviating unimportant client–cloud communications. The layers of

the defined model are separated for co-training between selected participants and edge servers. The computation-intensive and lightweight layers are operated in edge entities and local devices, respectively. Therefore, the upper-bound execution time of FL training expectation is decreased and acceptable for practical use cases. Figure 3 shows the primary procedures of edge FL, including (1) global model initialization from the parameter server, which is executed in the edge cloud, (2) global model distribution, which is executed in the client–edge tier, (3) local training and loss minimization, which are co-processed between client and edge, (4) loss-optimized local model updates based on labels, (5) multi-model edge aggregation at edge tier, (6) multi-model aggregated updates between edge and cloud, and finally, (7) multi-class global models averaging for the next iteration.

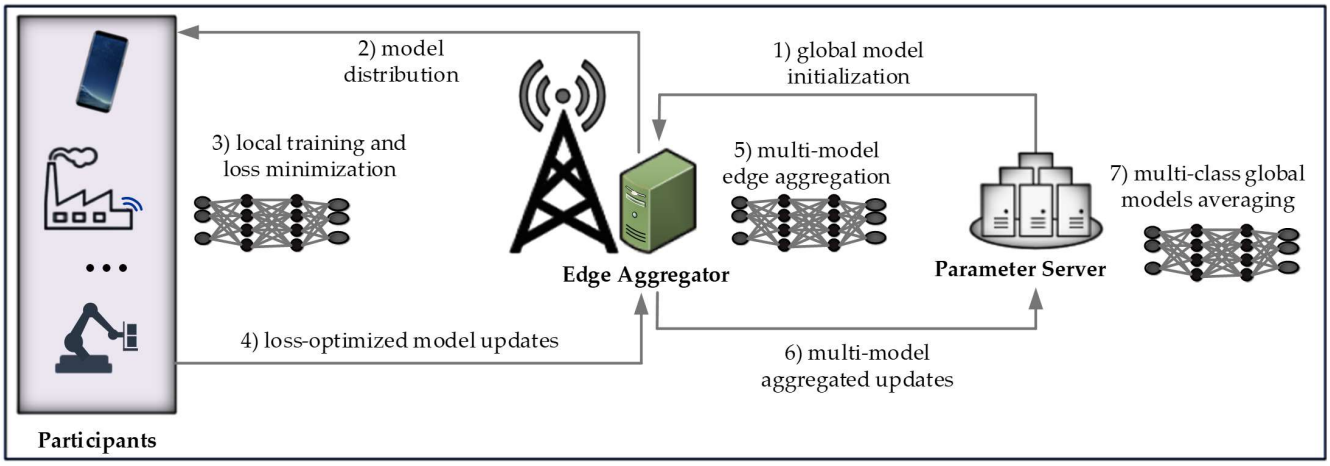

Figure 3. Transition procedures between participants, edge, and parameter server in edge FL.

The primary objectives of edge FL are to reduce global communication cost/frequency and reliability of FL performances. There are complementary studies that explored the promising edge-assisted paradigm. In [48], the impact of global model transmission costs was analyzed in the process of training a model with various bandwidth states and hyperparameter values (e.g., batch sizes and epoch numbers). The authors also provided guidelines about parameter configuration to obtain better learning accuracy and less resource consumption. Furthermore, in [49], the authors introduced edge-assisted FL by showing the efficiencies of integrated design/delay valuation, offloading decisions, and elasticity in different scenarios. The threshold-based offloading strategies were formulated to minimize the tolerable delay under the edge-assisted framework. However, in a heterogeneous IoT environment, the variety of device taxonomies can cause high-complexity offloading path decisions. An experience-driven paradigm or state-detailed observation of the agent can be applied to efficiently optimize the performance in numerous network conditions.

2.1.2. Federated Optimization

A converged concept of FL and federated analytics (FA) introduces federated optimization. By aiming to protect the user's personalization, this distributed scheme seeks to ensure a collaborative statistical report or model learning from decentralized data without raw sharing. By successfully deploying this paradigm, the efficiencies in backhaul communications, IoT data complexities, and differential privacy levels are delivered to the service providers [50]. Each aspect is tackled individually before the convergence procedure. FA, a statistical analysis, or data-science-based application requires collaborative querying data from local users, executes the local computation on clients, and generates the output aggregation on cloud tier. These executions give major benefits to data exploration and analysis visualization for specific application strategies. Since FL focuses on constructing deep models, FA can use it for serving data-science-based services.

In a massive IoT environment, federated optimization must consider the robustness, fairness, customization, and privacy level of multi-service prioritization. Secure computation and transmission are significant objectives to obtain in FA and FL. Malicious clients must be detected and secured from the participant selection process, which was studied in [51]. The authors proposed a secure aggregation procedure that aimed to support numerous clients to achieve better communication and computation performances. Furthermore, [52] presented a system design in high-level architecture in which multiple applications were described, including next-word prediction, content suggestions, and item ranking. The beneficial factors and implementation guidelines offer interesting knowledge for further field expansion of research and development.

From a communication perspective, to fully support the applicability of federated optimization, there are several modifications that need to be made in network architecture from infrastructure to the application layer. The operations on quick data queries and model building require zero-touch management to handle heavy-congestion states and future massive IoT participants.

2.2. Deep Reinforcement Learning for Experience-Driven Autonomy

In this subsection, we provide a background review of DRL, which specifically refers to the handling of long-term network performance and real-time agent interactivity in the service deployment of IoT networks. This background presents the assisting capability of DRL and leads to the key integration with FL for self-organizing capabilities.

Markov decision process models an optimization problem with four-tuple components, namely $\{S, A, R, P\}$, which represents the set of states, actions, rewards, and transition probability [53,54]. At a particular time t, state s_t represents the characteristic features of the environment, which are observed for feeding the agent. Action a_t consists of the updating parameters or configuration settings that the agent applies for purposes of improving the state s_t condition. Within general DRL, a_t is selected by function approximator using DNN. By modified DRL configuration (e.g., deep Q-networks), function approximators can be split into two primary networks, namely online and target. The separation procedure prevents an error of overoptimism. In the experience replays, denoted as $e_t(s_t, a_t, r_t, s_{t+1})$, the agent feeds (s_t, a_t, r_t) for current loss optimization and s_{t+1} for q-maximized action approximation. Over numerous iterations of exploration and exploitation, the non-optimal or near-optimal values are iteratively altered through the gradient process. Eventually, the optimal agent policy will be obtained. In every defined step, the weights are exchanged between online and target networks. To evaluate how well action a_t is applied in state s_t, the agent used the reward function that was formulated by the specific performance targets for the state–action pair. The transition probability indicates the estimation after configuring a_t into s_t, then transits the environment into s_{t+1}. To classify the diversity of DRL types, Figure 4 lists the differences between used/unused criteria (e.g., value, policy, and model).

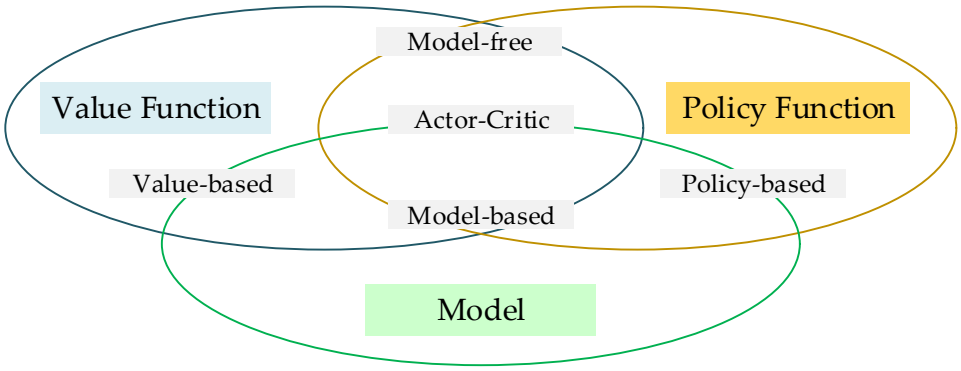

Figure 4. Overview taxonomies of DRL.

The features and functions are varied depending on application scenarios and performance objectives. The concept of a dynamic problem-solving algorithm, DRL, is used for modeling an optimal self-learning agent by interacting throughout exploration, understanding the possible maximum expected long-term rewards, and later exploitation. Optimal policy π^* is the policy-maximized long-term reward expectation. The value function can be formulated by the expectation of increasing reward summation following the policy π from state s. The standard optimal q-value function follows the Bellman equation and epsilon ε indication [53].

Table 3 presents the selected existing surveys on DRL that have been applied in communications and IoT applications. The review studies show how efficient DRL is in optimizing and self-organizing network control and orchestration. In [55], a multi-agent model-free DRL approach was proposed for assisting the framework of joint multi-channel and traffic control in SDN-based IoT environments. The system models were defined to show the optimization problems and DRL applicability. Within DRL specifications, the agent observed the state features as follows: (1) weights for channel capacity, transmission, and utilization, (2) channel/task numbers at a particular timeslot, (3) size of a particular task, (4) signal-to-interference-and-noise ratio between channels, and (5) state information of a particular timeslot's channel. The agent applied the actions of task index determination and the factorial of task numbers. A reward valuation was formulated by jointly considering the delay, throughput, and packet loss. As a result, the performance of a (multi-agent) DRL-based approach is highly efficient for experience-driven core network autonomy.

Table 3. Contributions of Existing Selected Surveys on Deep Reinforcement Learning.

Domains	Summary of Contributions	Ref.	Year
Applications of DRL in communications and networking	The overview, in-depth components, analysis, taxonomies, and comparisons of DRL techniques for optimizing communications and networking	[56]	2019
DRL for IoT applications	DRL algorithms for handling the problems of communication, computation, caching, control, domain-oriented applications, and privacy in IoT environment	[57]	2021
DRL in wireless IoT environment	The overview of wireless networks and DRL, taxonomy of IoT problem models, DRL-based approaches in IoT, and challenging issues of DRL in IoT networking	[58]	2021
Applicability of DRL for resource management	DRL-based optimization approaches for renewable energy, network slicing, spectrum scarcity, improving transmission rate, and big data in 5G heterogeneous networks	[59]	2019

3. Correlations between Federated Learning and Deep Reinforcement Learning

In this section, the correlations between the DRL agent and FL environment are given by emphasizing the usage of DRL-assisted zero-touch management to optimize the learning procedures in IoT networks. Figure 5 is given as an overall concept of modeling the interaction interfaces between the DRL agent and FL environment, including the key features from the main entities such as parameter server, edge aggregator, and local participants. Different scenarios define different condition-aware states, actions, and reward formulations. In DRL-based eFL, there are complementary studies that outline various scenarios with defined state–action relativity and long-term orchestration goals from agent.

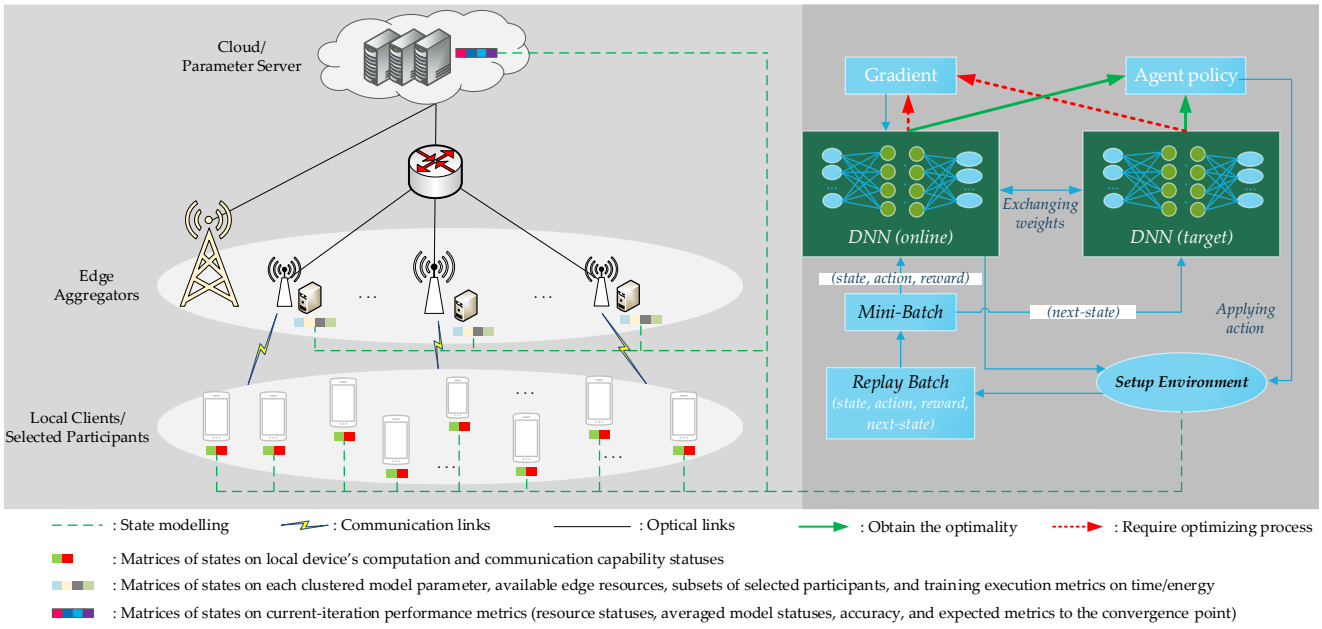

Figure 5. Key interaction and components between DRL agent and FL environment.

In the simulation perspective, federated averaging uses an iterative process and selects a random subset of participants for each iteration training step. Then, the initialization function is executed to sample the server's states. The server distributes the model to selected participants, which later computes the model training locally. The aggregation toward the new state is the output of adding between the initial state and the average of total states from clients. To build federated averaging, the models, clients, and optimizers are required to be known. In the testing phase, the data must be centralized and the validation function must be declared. To evaluate the federated optimization, the developers/researchers must specify the settings in terms of cross-silo/cross-device, differential privacy, or communication types [60–63]. The framework can be evaluated after finalizing the identical hyperparameter setup, the number of selected clients, and communication conditions. In this implementation flow, we can identify the correlation parts with the DRL agent as follows:

- The initialization function of FL can be converged with DRL as state features within the setup environment (e.g., capacities of edge aggregators);
- The actions can be merged with the selected values of participant subset within each FL round communication;
- Reward is the evaluation of federated optimization. By fully integrating the functions together, simulation is highly applicable and opens many key ideas for real-world application scenarios.

DRL has the potential in managing dynamic resources and making smart decisions in IoT environments. Once the application services have computation-intensive tasks to execute, EC is optimally suggested for dealing with adaptive virtual resource allocation [59,64]. Similarly, in FL model tasks, EC can serve as an enabler for resource efficiencies. The resource states (e.g., remaining capacities at the current timeslot) and FL model task sizes can be observed as state features for configuring the allocation amount and edge node offloading destination [65]. The reward can be formulated by the transmission latencies between participant–edge–central and convergence points in the achieved round index. In another aspect of a synchronous FL-IoT condition, high training delays can happen because of a straggler effect. To tackle these issues, [66] introduced a promising architecture that is capable of mapping the client operation with real-time states to a virtual entity and capturing the features of devices for interacting with the agent. Furthermore, an asynchronous FL

was proposed with actor–critic DRL-assisted device selection. The DRL components and observable features in this FL environment are listed as follows:

- ***State***: in the particular timeslot t, the environment samples a set of states including transmission power, the device's available resource for computation, model statuses, and previously selected device statuses at timeslot $t-1$;
- ***Action***: in timeslot t, the agent indicates which device to select for participating in the training process;
- ***Reward***: to evaluate the state–action (device selection efficiency) in timeslot t, the objectives of minimizing the energy utilization and maximizing the FL accuracy are formulated as the reward evaluation metric.

In the process of implementing a DRL-based algorithm for device selection, the parameters of actor, critic, target actor, and target critic networks are required to execute the model iteratively until optimal DNN parameters are defined. In each iteration, the sampling experience replays of an FL environment setup are stored and input into different networks until the final policy satisfies the expectation of the end goal. After selecting the device, k-class(es) of devices are clustered based on the power levels and data contribution sizes. By identifying specific classes of device performance stats, the convergence speed and parallel processing threshold are significantly improved. The straggler effects of massive IoT scenarios are handled in this systematic framework. Local training and global aggregation are executed afterward. Local updates from massive clients can be optimized in the uploading process within a cellular system [67].

In [68], a resource-optimized and trust-aware DRL scheduling for optimal client selection over FL-based IoT environment was studied. Monitoring modules were used for detection and information abstraction between IoT devices and edge nodes. Moreover, edge nodes can play an important role as a management entity of IoT nodes. A stochastic optimization formulation was used as a primary statement to correspond with the DRL-based selection algorithm. To execute a functioning agent, the states were collected from IoT participant's features, and the actions were adjusted to handle the different labels of current-iteration resource/trust at that particular timeslot. The edge nodes were also responsible for initializing the deep model with privacy-free data before declaring the model structures/values. The DRL-based approach can later sample a subset of clients to participate in the procedure. The efficiency of expected performance has relied on the upper-bound execution delay, training latency, resource utilization, and trust levels. The solution on [68] was deployed in a healthcare scenario on top of IoT infrastructure.

A DRL-based FL was further converged for efficient and intelligent orchestration in EC-enabled IIoT applications. Due to the drawback of a centralized resource-allocation scheme, [69] introduced an optimization approach to ideally observe and modify three state spaces, namely (1) task offloading ratio, (2) bandwidth allocation ratio, and (3) transmission power in FL-enabled IIoT environments. Action spaces covered a discrete vector of state modification. The action index altered the configuration of systematic allocation values. After applying the selection action index, the reward of the next-state feedback on the cost-improvement values was determined, whether it turns out positive or negative. The aims of the agent were to alleviate the communication cost, stabilize the energy consumption, and optimize the FL performance.

The correlations of modeling FL environments to interact with a DRL agent introduce efficient solutions in a variety of FL policy domains. The observable states of FL and the reward-maximized action configuration of DRL are integrated to ensure the high possibility of positive feedback in long-term self-organizing management.

4. Deep Reinforcement Learning-Based Efficient Federated Learning Policy Optimization

In this section, the main contribution of the paper is given by presenting complementary studies on applying DRL for eFL policy optimization in various domains in terms of resource, model offloading, update scheduling, and aggregation policies. Each outlined

study is evaluated through the relativity of integrating the network architecture in an interchangeable FL environment and its applicability in terms of DRL-based deployment (state observability, action configurability, and reward computability). Table 4 summarizes the selected works in this section.

Table 4. Summary of Selected Works in Section 4, including Primary Domain Taxonomies, Contributions, and Enabler Paradigms.

Proposed Domain Taxonomies		Contributions	Enabler Paradigms	Ref.	Year
4.1. Resource Optimization	4.1.1. Virtual Edge-Assisted Framework	Leveraged FL paradigm for predictive deep models to optimize VNFs autoscaling in 5G and beyond	VNFs, DL for (multi-step) time series prediction, FL with DNN	[70]	2021
		GNN- and DRL-enabled algorithm for efficient VNF-FGs placement	VNF-FGs, DRL-based provisioning, SDN- and NFV-enabled IoT networks, integer linear programming, GNN	[71]	2022
		Lower- and upper-level DRL based on GNN for optimizing resource-utilization-and-reliability ratio of service chains	Hierarchical DRL, GNN, VNFs, service chains	[72]	2022
		Adaptive FL via dynamic resource (radio and computation) optimization framework	SGD-based FL, Lyapunov stochastic optimization, asymptotic optimality	[73]	2021
		Optimizing computation resources, transmission power, and local model accuracy of FL-based vehicular EC applications	Vehicular EC, greedy, non-linear programming, Lagrangian dual, subgradient projection, min-max optimization	[74]	2021
	4.1.2. Experience-Driven Allocation	DRL approach for optimizing resource scheduling via experienced network states	DRL, policy gradient, E2E network models, VNFs	[75]	2019
		DRL-based computation resource control for FL networks	Joint models (training, energy, and loss), DRL-based allocation (actor-critic), FL	[76]	2020
	4.1.3. Mobility-Aware Allocation	DRL-based decision-making process for optimizing resource placement in EC	Deep Q-network, EC, SDN	[77]	2019
		DRL deployment for optimizing client decisions over energy and channels in FL networks	Deep Q-network, mobility-aware system models, FL network architecture	[78]	2020
4.2. Model Offloading Decisions		Leveraged FL to assist fast- and slow-timescale DRL training procedure for efficient computation offloading and resource allocation	FL for privacy-preserving DRL training, ultra-dense EC framework, hybrid offloading strategy	[79]	2021
		Improved FL communication/computation efficiencies with masked DNN models (non-shared global model), which secure the model offloading for central aggregation	Localized/structured sparse DNN, personalized FL, non-IID setting	[80]	2021

Table 4. *Cont.*

Proposed Domain Taxonomies		Contributions	Enabler Paradigms	Ref.	Year
4.3. Update Scheduling	4.3.1. Energy-Aware Scheduling	Double DRL-based scheduling for optimizing CPU and memory utilization in FL-enabled IoT environment	Edge FL, double Deep Q-learning, trust mechanism, stochastic optimization problem	[81]	2020
		Efficient client scheduling for optimizing communication costs in FL-enabled wireless IoT	Lagrange multiplier method, wireless IoT networks, communication-prioritized FL	[82]	2022
	4.3.2. Attention-Weight Selection	Advanced edge caching selection and replacement via federated DRL in D2D-assisted architecture	Edge caching, attention-weighted federated DRL, long-term mixed integer linear programming	[83]	2021
		Optimized asynchronous FL via modified DRL-based algorithms for client selection/clustering in IIoT	Digital twin-empowered IIoT, DRL-based selection (actor-critic), asynchronous FL	[66]	2022
		DRL-based client selection in FL for optimized resources and trust approaches	Federated (inter-edge) transfer learning, EC, DRL-based selection, trust management, IoT infrastructure	[68]	2022
4.4. Aggregation Policies		Efficient FL in resource-constrained networks via optimized client scheduling and aggregation policies	Asynchronous FL, equal weight, age-aware aggregation, significance-based scheduling, frequency-based scheduling, non-IID/IID	[84]	2021
		Dynamic scheduling algorithm for optimizing global loss in heterogeneous FL	FL, heterogeneity-aware problem, delay/energy models, Lyapunov optimization	[85]	2022
		Collaborative multi-agent architecture for controlling over local training procedure	Federated reinforcement learning, actor-critic proximal policy, gradient function	[86]	2020

4.1. Resource Optimization

4.1.1. Virtualized Edge-Assisted Framework

To enable network virtualization and softwarization, the confluence of SDN, NFV, and EC is expected to be pivotal. With the intelligence of AI-based mechanisms, the virtualized edge-assisted architecture can be further improved for completing self-managing capabilities. An efficient approach is expected to overcome network heterogeneity, different congestion states, newly instantiated/modified network services, and satisfying QoS expectations. In FL environments, virtualized edge-assisted architecture can greatly assist the edge aggregation process, co-training, and virtual computational resource placement in an adaptive manner. VNF autoscaling can enhance multi-level IoT privacy with FL-assisted virtual resource management. In [70], the authors leveraged the privacy-preserving and distributed capabilities of the FL paradigm to generate a learning model that is client specific and accurately predicts the autoscaling of VNFs in 5G-and-beyond networks. Various DL models in both centralized and decentralized aspects were evaluated for the prediction metrics regarding QoS- and cost-prioritized goals. The proposed scheme tackled multi-domain services in both reactive and predictive methods for (multi-step) time series forecasting. The performance comparison between collaborative FL distribution technique and centralized approaches was given.

SDN and NFV are well-known for enabling elastic and virtual network services, which greatly help in adopting the next-generation FL architecture in massive IoT services. Figure 6 illustrates the architecture of decoupled SDN planes with NFV-enabled edge resources to distribute sliced DNNs for softwarized FL-based service controllers. FL applications can be virtualized and instantiated as network services/functions within a set of VNF forwarding graphs (VNF-FGs). This way introduces the elasticity of FL deployment in the future by leveraging the flexibility of flow rule installations/modifications and VM placement within VNFs. FL services can be modified by providers in real time with the expected quality assurance. Figure 7 shows the interactions of DRL-based orchestration by configuring the actions that control the resource properties on VNF instances. Multi-service FLs are linked to efficient VNFs and VM resource pools in EC-enabled NFV infrastructure. To deal with the heterogeneity of IoT taxonomies and services, [71] presented a graph neural network (GNN) and DRL-assisted efficient VNF-FGs placement in SDN- and NFV-enabled IoT environments, which provided a great lesson on the applicability of modifying the FL-based IoT services to be optimized via a DRL-based algorithm. The states of resource capacities and actions on link placement are correlated in NFV-enabled FL service deployment. Furthermore, in [72], a hierarchical DRL that relied on the GNN algorithm was extensively studied for improving resource management for efficient service chain control. The lower- and upper-level DRL were designed for minimizing the resource utilization and blockage probability, respectively. The collaboration between lower and upper levels greatly assists the training execution and outputs a cost-efficient policy.

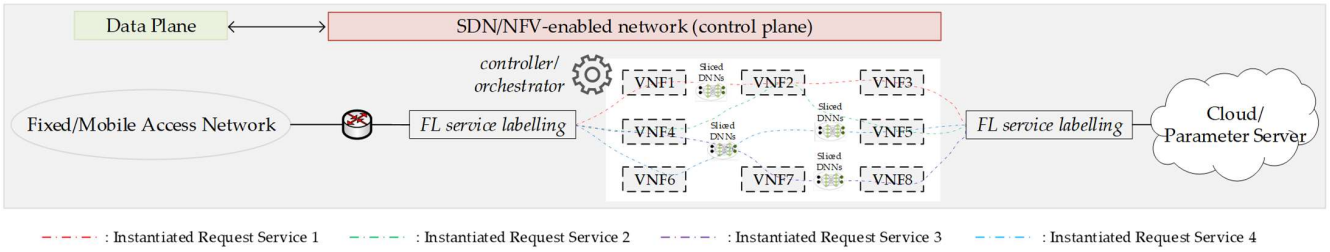

Figure 6. SDN-assisted control architecture for distributing DNNs.

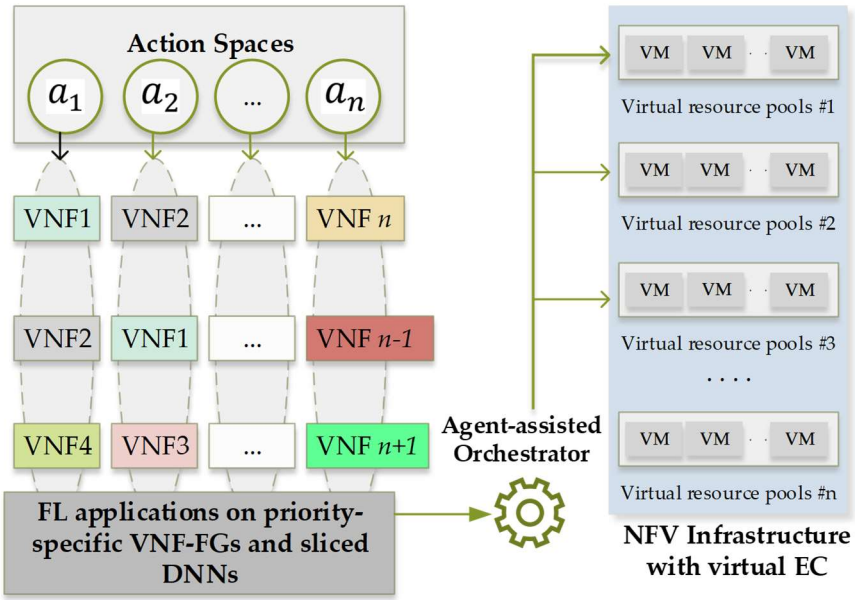

Figure 7. Agent-assisted orchestrator for adaptive action in optimizing FL resources.

In [73], an energy-efficient FL technique was studied at the wireless network edge by aiming to optimize the communication (e.g., power and bandwidth) and computation (e.g., CPU capacities at the central server) resource usage. The weighted models ideally consider

tradeoffs between energy, delay, and FL performance that are delivered in wireless networks. Stochastic gradient descent (SGD)-based FL and Lyapunov stochastic optimization were formulated in the system models to assist the developing phases of the adaptive FL-enabled resource allocation technique. In this perspective, DRL-based control can be used to observe the radio/computation states for reassignment and allocation in an experience-driven orchestration.

An FL-enabled vehicular EC faces the same problems with client (vehicle) selection and resource optimization. In [74], the authors addressed these issues and formulated an optimization model considering geographical position and mobility to achieve sufficient computation resources, transmission power, and accuracy. These features are the primary characteristics of resource optimization in executing FL-based applications. The authors comprehensively designed in-depth system models of autonomous vehicles, FL transitions, computation, and communication models. The resource allocation algorithms are complemented by Lagrange multipliers update. The solutions achieved remarkable performance in optimizing the system costs and fairness. However, in this scenario, DRL-based selection can assist with state-specific resource orchestration to tackle the tradeoffs and feedback in an efficient weighted sum model.

4.1.2. Experience-Driven Allocation

FL struggles to aim for fast convergence speed, accuracy satisfaction, training delays, and local energy utilization, which are important features in mission-critical and resource-constrained IoT networks. Experience- and data-driven knowledge can greatly achieve these features by reactively and proactively analyzing the future congestion states, task execution times, and allocation properties. DRL has driven resource optimization in various aspects, but one of the most resource-critical fields is that of network slicing applications. Ref. [75] leveraged the data-driven DRL capability to offer efficient resource optimization and service reliability in E2E network slicing. By intelligently applying the actions to adjust the resource allocation properties in each slice, the agent can evaluate the performance through the reward function and optimize the policy throughout exploration.

In [76], the authors jointly tackled the model learning and energy consumption by well-formulating a resource allocation problem for FL applications. The learning model primarily interacted with handling the communication overhead and computation resources. The energy model considered the unit energy consumption of each selected client for executing the model updates, and loss function described the model error estimation following the selected algorithm types. The joint problem models were solved by experience-driven actor–critic DRL, which learned the underlying pattern for a well-defined weighted sum balancing policy. The DRL agent was merged at the FL parameter server. Figure 8 shows the architecture of an experience-driven agent in a central FL server, which (1) observes the states of communication/computation features from the clients, (2) applies actions on federated settings and configures sufficient CPU-cycle amount for selected clients, and (3) evaluates the feedback from the environment and stores the batches in experience buffers.

The primary components of the proposed agent are described as follows.

- ***State***: a set of experienced (historical) network bandwidth information was observed as a primary feature, which is the most influential factor in FL model transmission cost. Future allocation requires prior knowledge of available and used properties.
- ***Action***: by determining the values on CPU-cycle frequency, the observed states are altered after applying the new actions. The action selection is indicated through the optimal policy function that is approximated by deep networks.
- ***Reward***: the FL system at t-iteration was feedbacked by r_t of state–action pair (s_t, a_t) efficiency. The evaluation metric aims to measure the system costs and scores according to the min–max possibilities.

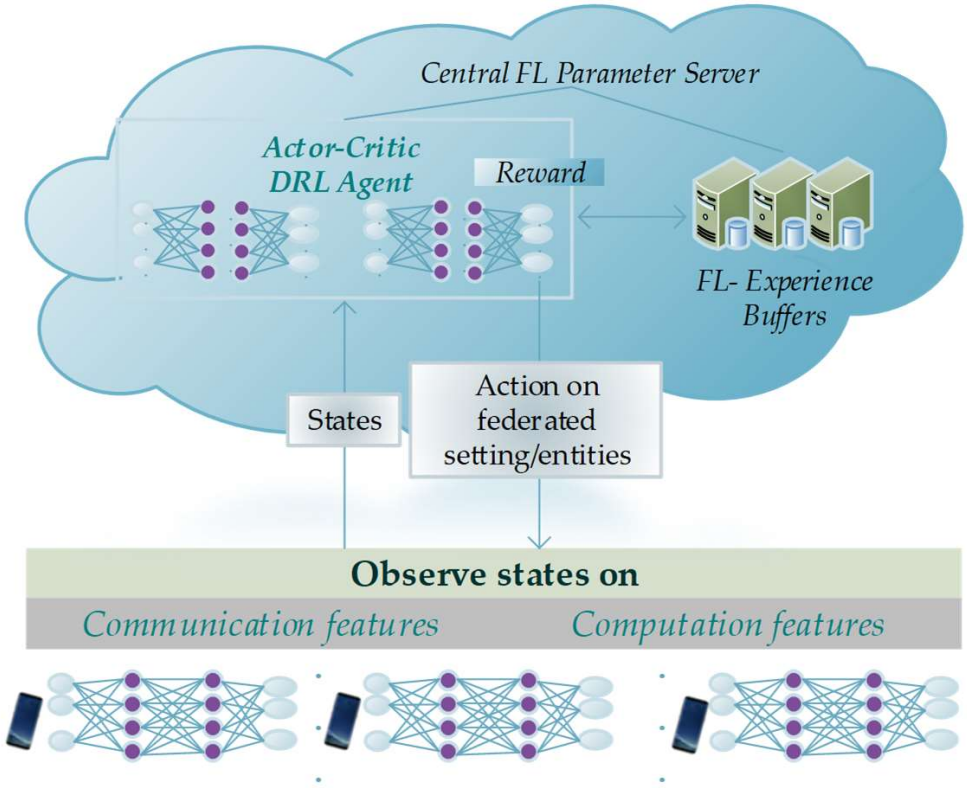

Figure 8. Experience-driven DRL (e.g., actor–critic) in setup environment of FL parameter server.

4.1.3. Mobility-Aware Allocation

Mobile devices/clients collaboratively contribute to the future multi-service FL applications, which cannot ignore the mobility-aware resource optimization; otherwise, the global model can be degraded from model drops and insufficient local computation. Ref. [77] proposed a DRL solution for achieving long-term objectives and learning from experiences of resource allocation in multi-access EC systems. The scheme modeled the system architecture with local and offloading computation. The task transfer policy was obtained by utilizing DRL-based actions on offloading decisions, allocated computation resources, and task transfer decisions. Furthermore, in [78], the authors addressed challenging issues on the energy cost for recharge, which lead to high network resource consumption in FL environments. A mobility-aware deep-Q-networks-based approach was proposed to allow clients to determine an optimal orchestration of energy and channels. By configuring actions on the model owner transmission path, the agent can explore the optimality for the setup architecture. Notable studies in DRL also mentioned double-deep-Q networks for handling over-optimistic problems [87,88], and the target function can be better defined in FL environments accordingly.

4.2. Model Offloading Decisions

Edge FL leverages EC capacities to assist local co-training, reduce client–cloud communication rounds, and offer edge aggregation. However, it comes with several challenging drawbacks in terms of multi-service offloading schedules and resource placement. The interaction of client–edge requires optimal aggregator selection in model label matching for both model update offloading and averaged model downloading. A central FL server with DRL-based decisions on offloading rules and aggregation policies is an optimal technique for ensuring adequately allocated edge resources and appropriate server selection in different IoT network congestion states (Figure 9).

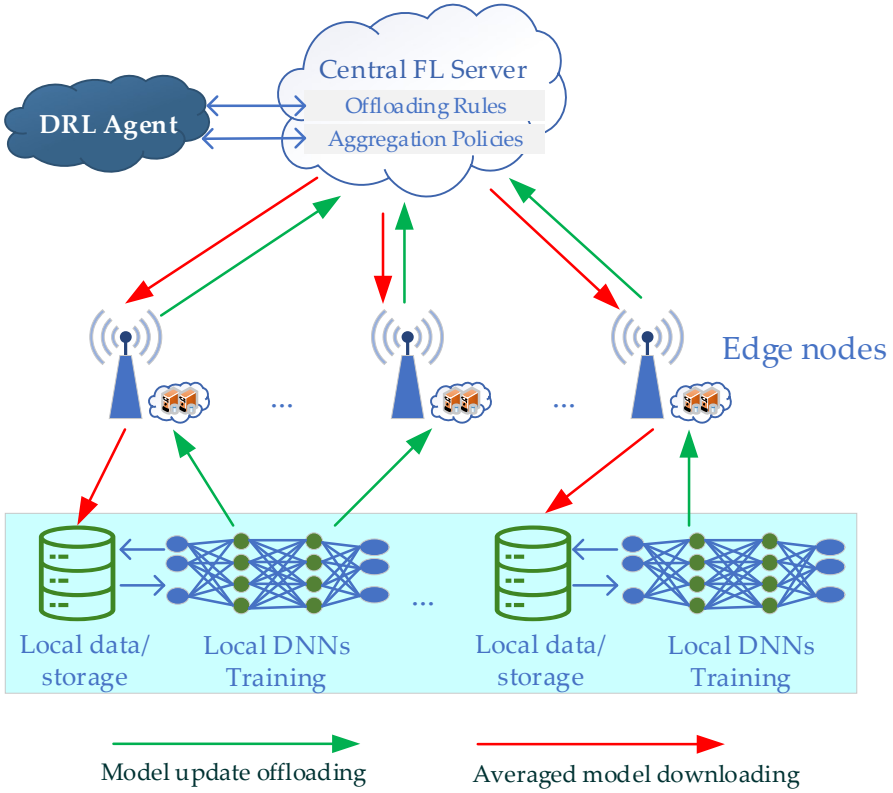

Figure 9. Embedded DRL-based optimization engine for offloading rules and aggregation policies in the central FL server.

In [79], the authors set up a system architecture on intelligent ultra-dense EC with formulated system models on communication (cellular link, D2D link, service caching, and task transmission) and computation offloading. An optimization problem on joint offloading, resource allocation, and service placement was initiated to aim for the main objective of minimizing task execution delays and resource consumption. Modified DRL was used to deal with the problem formulation by labeling two timescale approaches, namely fast and slow timescales. Fast-timescale DRL covered the handling scheme of offloading decisions and resource orchestration. Slow-timescale DRL dealt with a service caching scheme. To securely train these agents, the FL framework is later converged for a personalized model builder. The primary components of the agent and ultra-dense EC environment consist of:

- ***State***: at the particular timeslot t, the setup environment samples state features as follows: (1) available signal frequency (transmission channels), (2) weight matrices of D2D and uplink transmission rates, (3) caching statuses between small-cell base station and ED, (4) task queue, (5) available computational resource, (6) local resources, and (7) required services;
- ***Action***: to alter the environment conditions, the agent consists of three main variables for system modification, namely (1) application partitioning, (2) communication resource allocations, and (3) policy for caching placement;
- ***Reward***: to determine the consequence of applied action, the authors evaluate the positive feedback or negative errors by formulating a correlation model to return the joint task execution delay and resource consumption of the system.

Considering that IoT participants mostly face a resource-constrained status, improved communication/computation efficiencies of on-device DNN training/optimization are greatly beneficial in collaborative FL. The model offloading will be lessened and directed to only essential co-training requirements. Ref. [80] offered masking techniques for efficient FL by allowing participants to execute localized DNN with higher performances and

achievable levels. The proposed scheme optimized the computation costs of gradient execution and federated masking in the local tier. After the end participant optimizes the masking process, the local model owner offloads the masked models to the central FL server, which greatly strengthens the privacy details.

4.3. Update Scheduling

4.3.1. Energy-Aware Scheduling

In client scheduling, different objective statements and algorithm designs can be stated, such as local/edge energy consumption, communication/computation costs, or FL's KPI (e.g., convergence speed and accuracy). Ref. [81] studied a double-DRL-based energy-aware technique for FL client scheduling in an IoT environment. The authors proposed a trust mechanism to observe the states of energy sources and consumption rates. Double DRL aims to obtain the optimal scheduling policy by triggering state features from local IoT clients, including trust variables and energy statuses. The agent applies actions on assigning a central server for training, specifying the required energy units and model transmission costs. The grading metrics consider the effectiveness of the possible selection of trusted clients. By alleviating the malicious (non-trust) clients from the selection probability, FL training is advanced with high-quality and secured contributions from local data owners. In [82], an eFL within a wireless IoT environment was proposed by tackling the communication efficiency metrics, which corresponded with vastly saving energy usage from the high possibility of local model task termination. The main problems were split into scheduling and allocation sub-problems. The Lagrange multiplier method was used to handle the optimization problem. This proposed FL algorithm upgraded the adaptability of power and bandwidth allocation.

4.3.2. Attention-Weight Selections

An attention-weight federated DRL was proposed for efficient collaborative edge caching [83] by designing the overall procedures in three consecutive phases as follows: (1) model broadcasting, (2) local training, and (3) averaging aggregation. The framework models considered the association between ED and base stations, content popularity, user preference, D2D sharing pattern, content communication, and delay. With these system models, the interaction between each phase is reliable for edge caching services. Base stations distributed the global model to selected participants. In that training iteration, selected devices executed the DRL model following the declared global structure. After finalizing the local execution, the aggregation phase was divided into two sub-phases of evaluation indicators and aggregation weights. Based on values of average reward, loss, and hit rate, the weights were adjusted to emphasize the high-impact factor on the next-iteration global model aggregation. In the aspect of DRL components, the authors mentioned the following: (1) state spaces of content popularity, D2D link, and caching statues, (2) action spaces of cache list replacement via existing content inquiry and replaced status, and (3) reward objective on maximizing the gains of D2D sharing and content fetch. By critically deploying an attention-weight mechanism of FL models, the contributions from local models are comprehensively advanced. In [66], an attention weight on utility performances of IIoT participants in the selection approach was given to advance the FL framework. Furthermore, in [68], the authors proposed an efficient selection approach by weighing the significant factors of trust and execution delays.

4.4. Aggregation Policies

The aggregation process will be executed after the local models from selected participants are offloaded and obtained in the FL parameter server within each iteration of the communication round. By using DRL-based algorithms, device selection and update scheduling are positively optimized as a primary contribution to the final learning performance. Another aspect to improve the efficiency of FL is an adaptive aggregation policy for massive IoT services. Each FL service in heterogeneous networks requires the label-

ing of its criticality and satisfactory expectation to ensure a reliable and on-time model construction. An advanced approach can be migrated from different improving factors such as significance-aware modeling, edge-assisted global aggregation, and a DRL-based control mechanism. Figure 10 presents an eFL procedure transition between clients, edge aggregators, and parameter server throughout the initialization, global model distribution, local model training, model update scheduling, and aggregation policies. DRL-based approaches are greatly assured to be deployed in processes of controlling local training, selecting clients, and adjusting the aggregation [84–86].

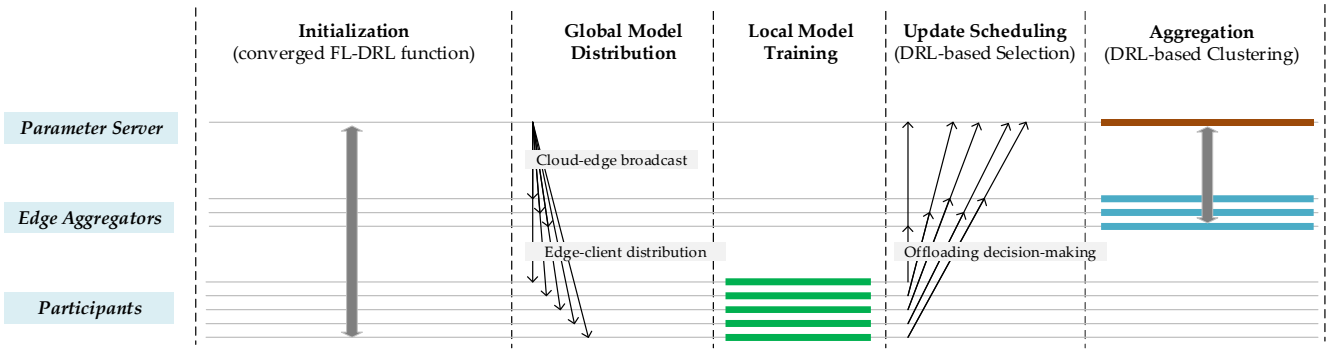

Figure 10. eFL procedures for the aggregation for the averaged global model.

In [84], a cooperative approach for scheduling and aggregation was proposed by evaluating several policies in asynchronous FL. In terms of client scheduling, three significant policies were studied, namely (1) a random policy in which the selection process decides randomly and chooses a subset of qualified participants in the accessible batches, (2) a significance-based algorithm in which a norm formulation of gradient update is used to identify the most suitable client, and (3) a frequency-based algorithm in which the relativity between each communication round is considered by tackling how often the particular client contributes to the aggregation. Aggregation policies considered two detailed approaches, namely (1) the equal weight method in which the weights rely on data contribution and (2) an age-aware aggregation in which the primary weights are based on the metric of the local update's age. In [85], a dynamic scheduling algorithm, termed as DISCO, was proposed by aiming for learning and energy-constrained metrics. The evaluation between multiple policies has been discussed in depth, which leads to a major contingency of applying a DRL agent in a formulated specific-reward assessment. Each objective can be specified into the agent's reward functions, where overall states are considered.

An aggregation approach with an intelligent decision-making agent and self-organizing capabilities will prioritize the mission-critical class in a real-time AI-enabled environment. A weighted-sum model for specific objectives can be deployed as a reward function to evaluate the rightful actions in particular network states.

5. Application Deployment Strategies

In this section, five primary application domains are discussed to give insightful system architectures, processing flows, and significance parameters that recent studies have proposed using DRL/FL-based approaches. Table 5 summarizes the selected works by labeling the application domains and contribution perspectives.

Table 5. Summary of Selected Works in Section 5, including DRL/FL-enabled Domains and Contributions.

DRL/FL-Enabled Domains	Contributions	Ref.	Year
IIoT	An optimized asynchronous FL by intelligent DRL-based client selection/clustering approaches to build models in newly integrated digital twin with IIoT environment	[66]	2022
	Dynamic resource allocation based on DRL for optimizing problems on communication cost, energy consumption, and execution latency of FL-enabled IIoT networks	[69]	2021
	Improved IIoT client selection in eFL framework by evaluating data quality through deep deterministic policy gradient algorithm	[89]	2021
Smart Automation	Handling computation overhead and privacy-preserving scheme of massive IoT participants in lightweight FL framework	[90]	2021
	Blockchain-enabled local training and privacy-preserving analytics for FL-enabled real-time smart grid architecture	[91]	2021
Medical Services	A comprehensive review on FL-enabled smart healthcare, which includes DRL-based experience-driven resource management and assessment metrics on data contributions	[92,93]	2021
	A thorough study on efficiencies of reinforcement learning for smart healthcare systems in tackling the future edge intelligence (collaborative local-edge AI-based applications)	[94]	2020
IoV	Minimizing system costs of FL frameworks by using a greedy algorithm to optimize the vehicular participant selection based on the quality of data contributions	[74]	2021
	DRL for experience-driven generation of IoV data training and FL-based framework for secured IoV models	[95]	2022
Environmental Context Detection	Analysis of waste and natural disasters for contributing to the beneficial factor of decentralized (FL) and centralized learning	[96]	2020

5.1. Industrial Internet of Things

The exponential growth of IIoT development leads to heterogeneous device taxonomies and complexity in management entities. The control mechanism must take the scalability, privacy, and elasticity into deployment schemes for ensuring a long-term support, reliable architecture, and multi-service efficiencies. IIoT in the new era requires automation and robotic systems for intelligent manufacturing, analysis, and high productivity. To gain automation, AI is a well-fitted paradigm; however, it requires gathering historical experiences and data-driven pattern processing. To secure the data of local equipment or industrial organizations, eFL can be proposed and evaluated to develop a collaborative learning model. In the phases of executing eFL framework, DRL-based approaches make great contributions, such as autonomous data management, efficient resource orchestration, and IIoT client selections.

In [66], an actor–critic DRL-based device selection for FL-enabled IIoT was proposed. To make the DRL agent applicable, a setup environment must be well-defined with observability of resourceful features and specific characteristics. The authors modeled the structure of a digital twin in IIoT, which consisted of two layers, namely physical and digital twins. Virtual systems bring connectivity between IIoT and digital models. With well-configured interfaces, the agent can obtain the states for figuring out the underlying patterns and for making smart decisions in client selection within each FL communication round. In [69], the authors initialized the environment and agents for all IIoT equipment in the system architecture, and each piece of equipment consisted of task(s) for FL updates. The proposed agent trained the data to evaluate the feedback scores and then updated the significant weights before aggregation. Furthermore, DRL-assisted FL can be used for ensuring the heterogeneity of IIoT data privacy and management [89].

5.2. Smart Automation

To enable smart automation, data-driven analyses, predictions, and recommendations are well-known outputs of the processing units. However, since IoT is currently being deployed in numerous private-sector entities, the generated data consist of both privacy-insensitive and privacy-sensitive distribution types. In large-scale resource-constrained IoT architectures, a secure FL scheme can be used for assisting the latency/resource overhead of computation and activating future intelligent automation applications. The framework must ensure the personalization of local IoT data during the process of constructing deep models for intelligent applications [90]. In [91], an advanced privacy-preserving computation scheme was proposed for smart grids following the FL processing phases. The proposed system modeled home-area networks with smart IoT devices for the surrounding data collection. The proposed algorithm used blockchain- and FL-based models to reduce the computation heterogeneity and enhance the collaborative model accuracy.

5.3. Medical Services

FL-enabled systems primarily focus on constructing deep models, privacy-preserving mechanisms, distributed learning architecture, and data partitioning. In healthcare systems, an advanced confidential system requires extensive studies because of possible cross-device/silo collaboration. The institutional clients or local medical devices collect highly sensitive data, where a privacy-enhanced FL framework needs to be deployed. FL in medical services requires complementary functions for resource-aware, incentive-aware, and personalized FL [92]. Furthermore, add-on functions for autonomous and accurate decision-making solutions should be extended with DRL [94]. Applications of DRL in IoT networks have become an active topic for research and deployment in the new era and beyond. In medical services, DRL greatly provides the applicability of prioritizing the ultra-low latency service requirements, enabling remote applications, handling heavy congested network states, and contributing intelligent core control. In [93], a converged approach between FL and DRL was proposed for evaluating the weights of client contributions that upgraded the deployment efficiency to a better level, particularly for smart healthcare applications. The distributed edge FL requires ensuring the enhanced confidentiality of practical healthcare systems and edge aggregation processes.

5.4. Internet of Vehicle

In intelligent transportation systems, real-time AI-based applications require optimizing every aspect for ensuring a model that has the capability to be mobility aware, engage in safe driving, be risk free, and have high efficiency. From the FL perspective, the models are constructed throughout numerous communication rounds; therefore, an optimization approach for each phase (e.g., client selection, update scheduling, and clustered aggregation) is obligatory. In [74], the authors gathered the states of vehicle position and velocity for weighting the updating participation. The problem statement was formulated with the objective of enhancing the computation, communication strength, and accuracy of collaborative FL models. The proposed scheme optimized vehicle selection based on its local data quality, which is highly significant for improving the model's performance and reliability. Moreover, in [95], a confluence between DRL and FL was made to bring efficient training data to the IoV environment and secure the state features of selected vehicles. The scheme allowed secure information-sharing between unmanned vehicles.

5.5. Environmental Context Detection

Natural and environmental disasters require accurate/long-term prediction, immediate notifications, and action recommendations to provide multiple options for ensuring people's safety. In the FL-based scenario, [96] discussed the deployment of environmental image datasets, namely natural disaster analysis and waste classification. The proposed methods handled the unlabeled/unannotated training subset. A collaboration learning between ED and a central server was activated in multi-environment use cases. The method-

ology considered the feature extraction of the dataset, active learning, and FL framework. Active learning assisted the labeling procedures (annotation/acquisition) of each data owner. The algorithm was fed by inputting unlabeled images, and the model divided the data into seeds/pools for training. The prediction output the selection processes. FL was used for constructing the final global learning model by collaborating with both the server and client sides.

6. Challenges and Future Perspectives

The complete DRL-based eFL or non-complete DRL/FL-assisted systems consist of inevitable challenging aspects to handle to become a practical approach in real-world deployment. The potential challenges and directions for future studies are presented in this section as follows:

- ***Malicious participant detection***: to prevent compromising participants' privacy during vertical/horizontal FL training, non-malicious participant selection is a critical phase to accurately consider in an environment of heterogeneous IoT taxonomies. Local co-training and co-sharing methods between participants for improving model performances are vulnerable to numerous possible attacks. The global learning model can be severely degraded by the false model parameters of malicious devices. A detection method using DL models (e.g., long short-term memory) can be integrated to securely execute predictive FL-enabled intelligence.
- ***Data contribution quality***: data contribution may differ between each participant based on sensing capabilities, hardware resources, or power levels. Data quality management and clustering can be studied to optimally select clients based on how valuably their input data influence the final model. However, it is significant to improve the low-quality data and ensure maximal participation from clients. DL models for data (e.g., images) recovery, visualization, or resolution enhancement can be a cooperative method and a part of a reliable FL framework.
- ***Optimal learning performance***: the convergence of the final learning model is time-consuming for reaching a satisfying accuracy in some real-time/mission-critical services, which leads to inapplicability for deployment. The upper bounds of expected convergence latencies require to be predicted or proactively known for adaptively optimizing the communication and computation resources in terms of model transmission and training rules in every communication round. A joint problem that considers the minimization of (1) system costs, (2) convergence delays, and (3) model accuracy is a prominent research direction.
- ***Multi-service optimization***: FL in multi-service IoT systems demands critical slicing and priority-aware orchestration schemes. The clustering mechanism outputs different labels ranging from mission-critical to non-mission-critical FL applications, which can be differed in terms of bandwidth/computational resource allocation, service prioritization, and aggregation scheduling. In the case of one particular client participating in two or more different FL-based applications, the availability and scheduling policies are interesting domains to research extensively.
- ***Privacy regulations***: a privacy-enhanced contribution is the major role of FL approaches; however, with numerous communication rounds of model updates/distribution between participants, edge aggregators, and parameter servers, sensitive subsets of information can be unmasked. Differential privacy is well-known for ensuring standard individual personalization on multiparty dataset aggregation [97]. Nonetheless, there are tradeoffs between improving each critical privacy and final learning performance. A multi-criteria decision-making model or weighted sum formulation for balancing the privacy regulations and accuracy is considered a great challenge for future improvement, particularly for a multi-service IoT environment [98–100]. A DRL-based optimization approach has a high potential to evaluate the fairness of the system and provide a long-term expected assessment.

- ***Balancing mechanism for synchronous FL***: from the asynchronous FL perspective, there are in-depth studies for (1) enhancing the performance automation with higher efficiency, (2) advancing the privacy/security, and (3) accelerating the convergence speed within digital twin networks and blockchain-enabled approaches [101–105]. However, in each communication round of synchronous FL, the training latency takes up to the slowest participant, which harshly degrades the overall performance and remains doubtful for deployment. To accelerate the slowest selected participant, an enhanced co-training approach with edge-assisted nodes or partial model training per round can become supportive methods in the synchronous FL process. DRL-based edge node selection can contribute to the FL processes by jointly observing the states of the collective edge environment (e.g., remaining resources, queuing sizes, and distance-based features) and evaluating the selection action based on expected co-training delays.
- ***Network architecture for federated optimization***: to operate federated optimization, the network architecture in every layer (e.g., infrastructure, control, and application) mandatorily needs to increase the complexity, intelligent systems, on-device capability, management entities of internet service providers, network operators, and softwarization/virtualization-based structures [106,107]. The constraints on IoT resources, connectivity, and stability should be taken into further consideration in future studies. Furthermore, to take the privacy-preserving framework to an advanced next level, (federated) machine unlearning can be considered by (1) not placing user information into the privacy-agnostic procedures and (2) alleviating the impact of participant's data on the final model [108–110].

7. Conclusions

In this paper, we presented a survey of DRL-based eFL policy optimization and a review of the potential issues regarding DRL/FL deployment. We first offered an introduction with a motivational statement from ZSM and EI in future network automation scenarios. Then, we described the preliminary studies on FL and DRL in IoT networks. The correlation between FL and DRL was provided by outlining the interactivity between the agent and the set-up FL environment. We also discussed the possible states, actions, and reward functions. Afterward, we provided the DRL-based eFL approaches in different optimization aspects, including resource management, model offloading decisions, update scheduling, and aggregation policies. In terms of application scenarios, we summarized the recent and notable studies that used DRL/FL-based schemes in well-known IoT applications, such as IIoT, smart automation, medical services, IoV, and environmental context detection. Finally, we discussed potential challenges and future directions that could be a high-impact research domains for enhancing the applicability of self-organizing FL in practical real-world systems.

Author Contributions: Conceptualization, S.K., R.C. and P.T.; methodology, S.K., R.C. and P.T.; software, P.T.; validation, S.K., R.C. and P.T.; formal analysis, P.T. and C.E.; investigation, S.K. and R.C.; resources, S.K.; data curation, P.T., R.C. and C.E.; writing—original draft preparation, P.T. and R.C.; writing—review and editing, R.C., P.T., C.E. and S.K.; visualization, P.T. and C.E.; supervision, S.K.; project administration, S.K.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-00167197, Development of Intelligent 5G/6G Infrastructure Technology for The Smart City), in part by the National Research Foundation of Korea (NRF), Ministry of Education, through Basic Science Research Program under Grant NRF-2020R1I1A3066543, in part by BK21 FOUR (Fostering Outstanding Universities for Research) under Grant 5199990914048, and in part by the Soonchunhyang University Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. State of the IoT 2020: 12 Billion IoT Connections, Surpassing Non-IoT for the First Time. Available online: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time (accessed on 10 August 2022).
2. Truong, N.; Sun, K.; Wang, S.; Guitton, F.; Guo, Y. Privacy Preservation in Federated Learning: An Insightful Survey from the GDPR Perspective. *Comput. Secur.* **2021**, *110*, 102402. [CrossRef]
3. Ma, X.; Yao, T.; Hu, M.; Dong, Y.; Liu, W.; Wang, F.; Liu, J. A Survey on Deep Learning Empowered IoT Applications. *IEEE Access* **2019**, *7*, 181721–181732. [CrossRef]
4. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
5. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 393–430. [CrossRef]
6. Pan, J.; Cai, L.; Yan, S.; Shen, X.S. Network for AI and AI for Network: Challenges and Opportunities for Learning-Oriented Networks. *IEEE Netw.* **2021**, *35*, 270–277. [CrossRef]
7. Kafle, V.P.; Hirayama, T.; Miyazawa, T.; Jibiki, M.; Harai, H. Network Control and Management Automation: Architecture Standardization Perspective. *IEEE Commun. Stand. Mag.* **2021**, *5*, 106–114. [CrossRef]
8. Sevgican, S.; Turan, M.; Gokarslan, K.; Yilmaz, H.B.; Tugcu, T. Intelligent Network Data Analytics Function in 5G Cellular Networks Using Machine Learning. *J. Commun. Netw.* **2020**, *22*, 269–280. [CrossRef]
9. "Zero-Touch Network and Service Management (ZSM); Landscape," White Paper, ETSI, Sophia Antipolis, France, 2022. Available online: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/004/02.01.01_60/gr_ZSM004v020101p.pdf (accessed on 10 August 2022).
10. "GANA–Generic Autonomic Networking Architecture," White Paper, ETSI, Sophia Antipolis, France, 2016. Available online: https://www.etsi.org/images/files/etsiwhitepapers/etsi_wp16_gana_ed1_20161011.pdf (accessed on 10 August 2022).
11. "Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI" White Paper, ETSI, Sophia Antipolis, France, 2021. Available online: https://www.etsi.org/deliver/etsi_gr/ENI/001_099/004/02.02.01_60/gr_ENI004v020201p.pdf (accessed on 10 August 2022).
12. "An autonomic Control Plane (ACP)," White Paper, Internet Engineering Task Force (IETF), Santa Clara, USA, 2021. Available online: https://www.rfc-editor.org/rfc/rfc8994.pdf (accessed on 10 August 2022).
13. "Focus Group on Machine Learning for Future Networks Including 5G (FG-ML5G)," White Paper, International Telecommunication Union (ITU), Geneva, Switzerland, 2019. Available online: https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-ML5G-2019-PDF-E.pdf (accessed on 10 August 2022).
14. "5G End-to-End Architecture Framework v4.31," White Paper, NGMN Alliance, Frankfurt, Germany, 2020. Available online: https://ngmn.org/wp-content/uploads/201117-NGMN_E2EArchFramework_v4.31.pdf (accessed on 10 August 2022).
15. "5G Smart Devices Supporting Network Slicing v1.1," White Paper, NGMN Alliance, Frankfurt, Germany, 2020. Available online: https://ngmn.org/wp-content/uploads/201214_NGMN_5G_SmartDevicesSupportingNetworkSlicing.pdf (accessed on 10 August 2022).
16. "ITU-ETSI-IEEE Joint SDOs Brainstorming Workshop on Testbeds Federations for 5G and Beyond: Interoperability, Standardization, Reference Model and APIs," Workshops and Seminars, ITU-T, 2021. Available online: https://www.itu.int/en/ITU-T/Workshops-and-Seminars/20210316/Pages/default.aspx (accessed on 10 August 2022).
17. Tang, S.; Chen, L.; He, K.; Xia, J.; Fan, L.; Nallanathan, A. Computational Intelligence and Deep Learning for Next-Generation Edge-Enabled Industrial IoT. *IEEE Trans. Netw. Sci. Eng.* **2022**, 1–13. [CrossRef]
18. Xu, D.; Li, T.; Li, Y.; Su, X.; Tarkoma, S.; Jiang, T.; Crowcroft, J.; Hui, P. Edge Intelligence: Architectures, Challenges, and Applications. *arXiv* **2020**, arXiv:2003.12172.
19. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
20. Liu, Y.; Peng, M.; Shou, G.; Chen, Y.; Chen, S. Towards Edge Intelligence: Multi-Access Edge Computing for 5G and Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 6722–6747. [CrossRef]
21. Ndikumana, A.; Tran, N.H.; Ho, T.M.; Han, Z.; Saad, W.; Niyato, D.; Hong, C.S. Joint Communication, Computation, Caching, and Control in Big Data Multi-Access Edge Computing. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1359–1374. [CrossRef]
22. Tam, P.; Math, S.; Kim, S. Optimized Multi-Service Tasks Offloading for Federated Learning in Edge Virtualization. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 4363–4378. [CrossRef]
23. Rausch, T.; Dustdar, S. Edge Intelligence: The Convergence of Humans, Things, and AI. In Proceedings of the 2019 IEEE International Conference on Cloud Engineering (IC2E), Prague, Czech Republic, 24–27 June 2019.
24. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [CrossRef]

25. Hu, H.; Tang, L. Edge Intelligence for Real-Time Data Analytics in an IoT-Based Smart Metering System. *IEEE Netw.* **2020**, *34*, 68–74. [CrossRef]
26. Wang, X.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]
27. Liyanage, M.; Pham, Q.-V.; Dev, K.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R.; Yenduri, G. A Survey on Zero Touch Network and Service Management (ZSM) for 5G and beyond Networks. *J. Netw. Comput. Appl.* **2022**, *203*, 103362. [CrossRef]
28. Benzaid, C.; Taleb, T. AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. *IEEE Netw.* **2020**, *34*, 186–194. [CrossRef]
29. Konečný, J.; McMahan, B.; Ramage, D. Federated Optimization: Distributed Optimization beyond the Datacenter. *arXiv* **2015**, arXiv:1511.03575.
30. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv* **2016**, arXiv:1610.02527.
31. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. y Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv* **2017**, arXiv:1602.05629.
32. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A Survey on Federated Learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
33. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* **2021**, *9*, 1–24. [CrossRef]
34. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; Miao, C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
35. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef]
36. Abdulrahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A Survey on Federated Learning: The Journey from Centralized to Distributed On-Site Learning and Beyond. *IEEE Internet Things J.* **2021**, *8*, 5476–5497. [CrossRef]
37. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated Learning in Edge Computing: A Systematic Survey. *Sensors* **2022**, *22*, 450. [CrossRef]
38. Khan, L.U.; Alsenwi, M.; Yaqoob, I.; Imran, M.; Han, Z.; Hong, C.S. Resource Optimized Federated Learning-Enabled Cognitive Internet of Things for Smart Industries. *IEEE Access* **2020**, *8*, 168854–168864. [CrossRef]
39. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221. [CrossRef]
40. Chen, Y.; Sun, X.; Jin, Y. Communication-Efficient Federated Deep Learning with Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 4229–4238. [CrossRef]
41. Mills, J.; Hu, J.; Min, G. Multi-Task Federated Learning for Personalised Deep Neural Networks in Edge Computing. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 630–641. [CrossRef]
42. Sattler, F.; Wiedemann, S.; Muller, K.-R.; Samek, W. Robust and Communication-Efficient Federated Learning from Non-I.i.d. Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef]
43. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020.
44. Taik, A.; Cherkaoui, S. Federated Edge Learning: Design Issues and Challenges. *IEEE Netw.* **2020**, *35*, 252–258. [CrossRef]
45. Ahmed, K.; Imteaj, A.; Amini, M.H. Federated Deep Learning for Heterogeneous Edge Computing. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Virtually Online, 13–15 December 2021.
46. Zhou, T.; Li, X.; Pan, C.; Zhou, M.; Yao, Y. Multi-Server Federated Edge Learning for Low Power Consumption Wireless Resource Allocation Based on User QoE. *J. Commun. Netw.* **2021**, *23*, 463–472. [CrossRef]
47. Liu, T.; Di, B.; Wang, B.; Song, L. Loss-Privacy Tradeoff in Federated Edge Learning. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 546–558. [CrossRef]
48. Ye, Y.; Li, S.; Liu, F.; Tang, Y.; Hu, W. EdgeFed: Optimized Federated Learning Based on Edge Computing. *IEEE Access* **2020**, *8*, 209191–209198. [CrossRef]
49. Ji, Z.; Chen, L.; Zhao, N.; Chen, Y.; Wei, G.; Yu, F.R. Computation Offloading for Edge-Assisted Federated Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9330–9344. [CrossRef]
50. Wang, J.; Charles, Z.; Xu, Z.; Joshi, G.; McMahan, H.B.; Arcas, B.A.y; Al-Shedivat, M.; Andrew, G.; Avestimehr, S.; Daly, K.; et al. A Field Guide to Federated Optimization. *arXiv* **2021**, arXiv:2107.06917.
51. Bell, J.; Bonawitz, K.A.; Gascón, A.; Lepoint, T.; Raykova, M. Secure Single-Server Aggregation with (Poly)Logarithmic Overhead. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 9–13 November 2020.
52. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, H.B.; et al. Towards Federated Learning at Scale: System Design. *arXiv* **2019**, arXiv:1902.01046.
53. Sutton, R.S.; Barto, A. *Reinforcement Learning: An Introduction*; The Mit Press: Cambridge, MA, USA; London, UK, 1998; ISBN 9780262039246.
54. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [CrossRef]

55. Wu, T.; Zhou, P.; Wang, B.; Li, A.; Tang, X.; Xu, Z.; Chen, K.; Ding, X. Joint Traffic Control and Multi-Channel Reassignment for Core Backbone Network in SDN-IoT: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 231–245. [CrossRef]
56. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.-C.; Kim, D.I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [CrossRef]
57. Chen, W.; Qiu, X.; Cai, T.; Dai, H.-N.; Zheng, Z.; Zhang, Y. Deep Reinforcement Learning for Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1659–1692. [CrossRef]
58. Frikha, M.S.; Gammar, S.M.; Lahmadi, A.; Andrey, L. Reinforcement and Deep Reinforcement Learning for Wireless Internet of Things: A Survey. *Comput. Commun.* **2021**, *178*, 98–113. [CrossRef]
59. Lee, Y.L.; Qin, D. A Survey on Applications of Deep Reinforcement Learning in Resource Management for 5G Heterogeneous Networks. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019.
60. Nandury, K.; Mohan, A.; Weber, F. Cross-Silo Federated Training in the Cloud with Diversity Scaling and Semi-Supervised Learning. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 25–30 April 2021.
61. Tam, P.; Math, S.; Lee, A.; Kim, S. Multi-Agent Deep Q-Networks for Efficient Edge Federated Learning Communications in Software-Defined IoT. *Comput. Mater. Contin.* **2022**, *71*, 3319–3335. [CrossRef]
62. Majeed, U.; Hassan, S.S.; Hong, C.S. Cross-Silo Model-Based Secure Federated Transfer Learning for Flow-Based Traffic Classification. In Proceedings of the 2021 International Conference on Information Networking (ICOIN), Jeju, Republic of Korea, 13–16 January 2021.
63. Chen, J.; Li, J.; Huang, R.; Yue, K.; Chen, Z.; Li, W. Federated Transfer Learning for Bearing Fault Diagnosis with Discrepancy-Based Weighted Federated Averaging. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3514911. [CrossRef]
64. Chen, Y.; Liu, Z.; Zhang, Y.; Wu, Y.; Chen, X.; Zhao, L. Deep Reinforcement Learning-Based Dynamic Resource Management for Mobile Edge Computing in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4925–4934. [CrossRef]
65. Tam, P.; Math, S.; Nam, C.; Kim, S. Adaptive Resource Optimized Edge Federated Learning in Real-Time Image Sensing Classifications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10929–10940. [CrossRef]
66. Yang, W.; Xiang, W.; Yang, Y.; Cheng, P. Optimizing Federated Learning with Deep Reinforcement Learning for Digital Twin Empowered Industrial IoT. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1884–1893. [CrossRef]
67. Choi, J.; Pokhrel, S.R. Federated Learning with Multichannel ALOHA. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 499–502. [CrossRef]
68. Rjoub, G.; Wahab, O.A.; Bentahar, J.; Cohen, R.; Bataineh, A.S. Trust-Augmented Deep Reinforcement Learning for Federated Learning Client Selection. *Inf. Syst. Front.* **2022**. [CrossRef]
69. Guo, Y.; Zhao, Z.; He, K.; Lai, S.; Xia, J.; Fan, L. Efficient and Flexible Management for Industrial Internet of Things: A Federated Learning Approach. *Comput. Netw.* **2021**, *192*, 108122. [CrossRef]
70. Subramanya, T.; Riggio, R. Centralized and Federated Learning for Predictive VNF Autoscaling in Multi-Domain 5G Networks and Beyond. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 63–78. [CrossRef]
71. Xie, Y.; Huang, L.; Kong, Y.; Wang, S.; Xu, S.; Wang, X.; Ren, J. Virtualized Network Function Forwarding Graph Placing in SDN and NFV-Enabled IoT Networks: A Graph Neural Network Assisted Deep Reinforcement Learning Method. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 524–537. [CrossRef]
72. Li, B.; Zhu, Z. GNN-Based Hierarchical Deep Reinforcement Learning for NFV-Oriented Online Resource Orchestration in Elastic Optical DCIs. *J. Light. Technol.* **2022**, *40*, 935–946. [CrossRef]
73. Lorenzo, P.D.; Battiloro, C.; Merluzzi, M.; Barbarossa, S. Dynamic Resource Optimization for Adaptive Federated Learning at the Wireless Network Edge. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 25–30 April 2021.
74. Xiao, H.; Zhao, J.; Pei, Q.; Feng, J.; Liu, L.; Shi, W. Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11073–11087. [CrossRef]
75. Wang, H.; Wu, Y.; Min, G.; Xu, J.; Tang, P. Data-Driven Dynamic Resource Scheduling for Network Slicing: A Deep Reinforcement Learning Approach. *Inf. Sci.* **2019**, *498*, 106–116. [CrossRef]
76. Zhan, Y.; Li, P.; Guo, S. Experience-Driven Computational Resource Allocation of Federated Learning by Deep Reinforcement Learning. In Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 18–22 May 2020.
77. Din, N.; Chen, H.; Khan, D. Mobility-Aware Resource Allocation in Multi-Access Edge Computing Using Deep Reinforcement Learning. In Proceedings of the 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 16–18 December 2019.
78. Nguyen, H.T.; Luong, N.C.; Zhao, J.; Yuen, C.; Niyato, D. Resource Allocation in Mobility-Aware Federated Learning Networks: A Deep Reinforcement Learning Approach. *arXiv* **2019**, arXiv:1910.09172.
79. Yu, S.; Chen, X.; Zhou, Z.; Gong, X.; Wu, D. When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multi-Timescale Resource Management for Multi-Access Edge Computing in 5G Ultra Dense Network. *IEEE Internet Things J.* **2020**, *8*, 2238–2251. [CrossRef]

80. Li, A.; Sun, J.; Zeng, X.; Zhang, M.; Li, H.; Chen, Y. FedMask: Joint Computation and Communication-Efficient Personalized Federated Learning via Heterogeneous Masking. In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, Sydney, NSW, Australia, 21–24 April 2021.
81. Rjoub, G.; Abdel Wahab, O.; Bentahar, J.; Bataineh, A. A Trust and Energy-Aware Double Deep Reinforcement Learning Scheduling Strategy for Federated Learning on IoT Devices. In *Service-Oriented Computing, Proceedings of the 18th International Conference, ICSOC 2020, Dubai, United Arab Emirates, 14–17 December 2020*; Springer: Cham, Switzerland, 2020; pp. 319–333. [CrossRef]
82. Chen, H.; Huang, S.; Zhang, D.; Xiao, M.; Skoglund, M.; Poor, H.V. Federated Learning over Wireless IoT Networks with Optimized Communication and Resources. *IEEE Internet Things J.* **2022**, *9*, 16592–16605. [CrossRef]
83. Wang, X.; Li, R.; Wang, C.; Li, X.; Taleb, T.; Leung, V.C.M. Attention-Weighted Federated Deep Reinforcement Learning for Device-To-Device Assisted Heterogeneous Collaborative Edge Caching. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 154–169. [CrossRef]
84. Hu, C.-H.; Chen, Z.; Larsson, E.G. Device Scheduling and Update Aggregation Policies for Asynchronous Federated Learning. *arXiv* **2021**, arXiv:2107.11415.
85. Guo, K.; Chen, Z.; Yang, H.H.; Quek, T.Q.S. Dynamic Scheduling for Heterogeneous Federated Learning in Private 5G Edge Networks. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 26–40. [CrossRef]
86. Lim, H.-K.; Kim, J.-B.; Heo, J.-S.; Han, Y.-H. Federated Reinforcement Learning for Training Control Policies on Multiple IoT Devices. *Sensors* **2020**, *20*, 1359. [CrossRef]
87. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
88. van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. *arXiv* **2015**, arXiv:1509.06461. [CrossRef]
89. Zhang, P.; Wang, C.; Jiang, C.; Han, Z. Deep Reinforcement Learning Assisted Federated Learning Algorithm for Data Management of IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8475–8484. [CrossRef]
90. Wei, Z.; Pei, Q.; Zhang, N.; Liu, X.; Wu, C.; Taherkordi, A. Lightweight Federated Learning for Large-Scale IoT Devices with Privacy Guarantee. *IEEE Internet Things J.* **2021**, *10*, 3179–3191. [CrossRef]
91. Singh, P.; Masud, M.; Hossain, M.S.; Kaur, A.; Muhammad, G.; Ghoneim, A. Privacy-Preserving Serverless Computing Using Federated Learning for Smart Grids. *IEEE Trans. Ind. Inform.* **2021**, *18*, 7843–7852. [CrossRef]
92. Nguyen, D.C.; Pham, Q.-V.; Pathirana, P.N.; Ding, M.; Seneviratne, A.; Lin, Z.; Dobre, O.A.; Hwang, W.-J. Federated Learning for Smart Healthcare: A Survey. *arXiv* **2021**, arXiv:2111.08834. [CrossRef]
93. Zhao, J.; Zhu, X.; Wang, J.; Xiao, J. Efficient Client Contribution Evaluation for Horizontal Federated Learning. *arXiv* **2021**, arXiv:2102.13314.
94. Coronato, A.; Naeem, M.; Pietro, G.D.; Paragliola, G. Reinforcement Learning for Intelligent Healthcare Applications: A Survey. *Artif. Intell. Med.* **2020**, *109*, 101964. [CrossRef]
95. Ding, T.; Liu, L.; Zhu, Y.; Cui, L.; Yan, Z. IoV Environment Exploring Coordination: A Federated Learning Approach. *Digit. Commun. Netw.* **2022**. [CrossRef]
96. Ahmed, L.; Ahmad, K.; Said, N.; Qolomany, B.; Qadir, J.; Al-Fuqaha, A. Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification. *IEEE Access* **2020**, *8*, 208518–208531. [CrossRef]
97. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security-CCS'16, Vienna, Austria, 24–28 October 2016.
98. Chen, L.; Tang, H.; Zhao, Y.; You, W.; Wang, K. A Privacy-Preserving and Energy-Efficient Offloading Algorithm Based on Lyapunov Optimization. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 2490–2506.
99. Su, Y.; Gao, H.; Zhang, S. Hybrid Resource Allocation Scheme in Secure Intelligent Reflecting Surface-Assisted IoT. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 3256–3274.
100. Singh, G.; Joshi, P.; Raghuvanshi, A.S. A Novel Duty Cycle Based Cross Layer Model for Energy Efficient Routing in IWSN Based IoT Application. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 1849–1876.
101. Qu, Y.; Gao, L.; Xiang, Y.; Shen, S.; Yu, S. FedTwin: Blockchain-Enabled Adaptive Asynchronous Federated Learning for Digital Twin Networks. *IEEE Netw.* **2022**, *36*, 183–190. [CrossRef]
102. Qu, Y.; Xu, C.; Gao, L.; Xiang, Y.; Yu, S. FL-SEC: Privacy-Preserving Decentralized Federated Learning Using SignSGD for the Internet of Artificially Intelligent Things. *IEEE Internet Things Mag.* **2022**, *5*, 85–90. [CrossRef]
103. Xu, C.; Qu, Y.; Eklund, P.W.; Xiang, Y.; Gao, L. BAFL: An Efficient Blockchain-Based Asynchronous Federated Learning Framework. In Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 5–8 September 2021.
104. Li, C.; Yuan, Y.; Wang, F.-Y. Blockchain-Enabled Federated Learning: A Survey. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 1 July 2021; pp. 286–289.
105. Xu, C.; Qu, Y.; Xiang, Y.; Gao, L. Asynchronous Federated Learning on Heterogeneous Devices: A Survey. *arXiv* **2022**, arXiv:2109.04269.
106. Zhang, X.; Xia, C.; Ma, T.; Zhang, L.; Jin, Z. Optimizing Energy-Latency Tradeoff for Computation Offloading in SDIN-Enabled MEC-Based IIoT. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 4081–4098.

107. Liang, X.; Wu, Y.; Huang, Y.; Ng, D.W.K.; Li, P.; Yao, Y. Performance Optimization and Analysis on P2P Mobile Communication Systems Accelerated by MEC Servers. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 188–210.
108. Nguyen, T.T.; Huynh, T.T.; Nguyen, P.L.; Liew, A.W.-C.; Yin, H.; Nguyen, Q.V.H. A Survey of Machine Unlearning. *arXiv* **2022**, arXiv:2209.02299.
109. Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C.A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; Papernot, N. Machine Unlearning. *arXiv* **2020**, arXiv:1912.03817.
110. Liu, G.; Ma, X.; Yang, Y.; Wang, C.; Liu, J. Federated Unlearning. *arXiv* **2021**, arXiv:2012.13891.