Multi-Perspective and Energy-Efficient Deep Learning in Edge Computing

Haitao Yuan, *Senior Member, IEEE,* Jing Bi, *Senior Member, IEEE*, Ziqi Wang, *Student Member, IEEE*, Jia Zhang, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, and Rajkumar Buyya, *Fellow, IEEE*

***Abstract*—The deployment of billions of Internet of Things (IoT) devices is driving unprecedented data generation at the network edge, demanding high computational power for real-time deep learning (DL) while raising serious concerns about energy consumption. While edge computing offers a viable paradigm for decentralized DL by preserving data privacy and reducing latency, the substantial energy costs of DL training and inference pose a major challenge for resource-constrained edge devices. This work provides a comprehensive review of state-of-the-art studies that address energy efficiency at the intersection of DL and edge computing. Moving beyond isolated solutions, we analyze the critical need for a co-design approach integrating hardware and software with adaptive resource management to build sustainable systems. The paper systematically examines hardware-level optimizations and software-level techniques for reducing energy consumption while maintaining model accuracy. Furthermore, it investigates how adaptive management of compute, memory, and communication resources is key to dynamic energy savings. Finally, the paper synthesizes recent trends, identifies emerging opportunities, and discusses open challenges, positioning hardware-software co-design as the most promising approach for achieving scalable and energy-efficient deep learning in edge computing.**

***Index Terms*—Edge Computing, Internet of Things, Deep Learning, Energy Optimization, and Deep Neural Networks.**

I. Introduction

Edge computing emerges as a complement to cloud computing, processing data at the network edge. The latter is centralized and can process and analyze a large amount of data from edge devices, *e.g.*, smartphones, music/video players, wearable devices, and game controllers. Yet, it is designed to offer real-time services, primarily due to the much-needed communication between edge devices and cloud data centers (CDCs), which cannot always be guaranteed to be error- and delay-free. Billions of devices have been connected *via* various communications links. Devices collect tremendous amounts of data, whose communication and processing pose high network traffic and computational requirements [1]. Real-time response requirements from the user side force some computation to be shifted from CDCs to the Internet of Things (IoT) devices at the edge. Several studies combine edge computing with cloud computing, enabling the entire system to leverage both. Fig. 1 illustrates an exemplary scenario showing the advantages of edge computing. In traditional CDCs, an image from a camera is first transmitted to the CDC for recognition of a license plate within it. The plate number is extracted and kept in a database along with time/location information. The network bandwidth and response time are increased because image data is sent to CDCs for processing. In the edge computing paradigm, all the processing is finished in the camera, which only needs to transmit the plate number to the CDCs. It is evident that bandwidth requirements have decreased significantly since the image is not processed by CDCs, thereby avoiding a large number of security and privacy attacks. As shown in Fig. 2, edge nodes, *e.g.*, routers, switches, and base stations, are often co-located with cellular base stations, IoT gateways, *etc*.

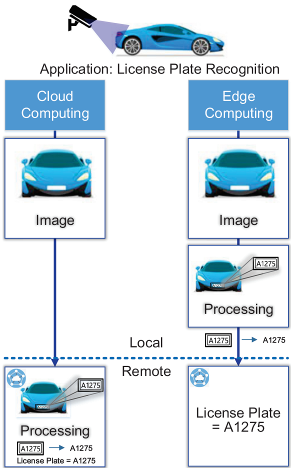

Fig. 1. An exemplary scenario using both edge devices and CDCs.

In recent years, deep learning (DL) has become a crucial analytical method for handling vast amounts of data across various fields, including natural language processing, computer vision, pattern recognition, bioinformatics, intelligent transportation systems, smart cities, and social networks. It is viewed as the most promising method for analyzing real-world IoT data in complex environments that are challenging

This work was supported by the National Natural Science Foundation of China under Grants 62173013 and 62473014, the Beijing Natural Science Foundation under Grants L233005 and 4232049, and the Fundamental Research Funds for the Central Universities under Grant 501XYGG2025103005. (Corresponding author: Haitao Yuan)

H. Yuan is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. (e-mail: yuan@buaa.edu.cn).

J. Bi is with the College of Computer Science, Beijing University of Technology, Beijing 100124, China. (e-mail: bijing@bjut.edu.cn).

Z. Wang is with the School of Software Technology, Zhejiang University, Ningbo 315100, China. (e-mail: wangziqi0312@zju.edu.cn).

J. Zhang is with the Department of Computer Science, Southern Methodist University, Dallas, TX 75206, USA. (e-mail: jiazhang@smu.edu).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA. (e-mail: zhou@njit.edu).

R. Buyya is with the Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: rbuyya@unimelb.edu.au).

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3640292

to tackle with traditional machine learning methods. It is becoming the *de facto* analysis method, with widespread adoption and powering the current big data society. They are suitable for complex and hierarchical structures that learn powerful representations and features from sensory data. There are many types of DNNs, *e.g.*, recurrent neural networks (RNNs) and convolutional neural networks (CNNs). The size of intermediate data in DNNs can be dynamically scaled down by each layer until enough features are learnt. Consequently, they are suitable for edge computing environments, as some of their learning layers can be offloaded to the edge, allowing only the reduced intermediate data to be transmitted to centralized CDCs.

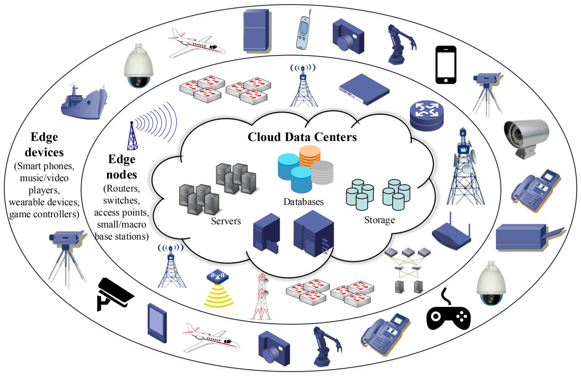
Edge devices
(Smart phones, music/video players, wearable devices, game controllers)
Edge nodes
(Routers, switches, access points, small/macro base stations)
Cloud Data Centers
Servers
Databases
Storage

Fig. 2. Edge devices/nodes and CDCs.

However, many issues exist in achieving DL at edge devices. Among them, one major challenge is that training of DL models needs expensive memory space and computation owing to millions of weight and bias parameters, which are difficult to realize on a variety of resource-limited edge devices, ranging from smartphones with lightweight processors to edge servers with graphics processing units (GPUs). DNN applications are data- and computationally-intensive, and network structures become increasingly larger and deeper to handle more complex data. Training or inference in edge devices can be costly in terms of computation and memory space, as it is computationally intensive and energy-consuming due to possibly high-dimensional input data, *e.g.*, high-resolution images. Thus, limited energy and its usage efficiency are becoming major concerns for edge computing systems. The work in [2] presents an example for analyzing seismic imaging in edge computing. Current subsurface imaging methods for visualizing gas/oil reservoirs or magma/fault movement cannot obtain real-time information. Thousands of seismic sensors are installed to monitor subsurface anomalies. These sensors monitor the ground's vibration and store it in their local storage. The acoustic signals are collected at 16-24 bits, with a frequency range of 100-500 Hz, and can accumulate to a few gigabytes per day in each edge device. It is infeasible to transmit such large amounts of data to CDCs using low-power radio due to bandwidth and energy constraints. Thus, a framework of edge computing is proposed to realize real-time seismic imaging where sensors are located close to raw data, and seismic signals are transmitted to them in real-time.

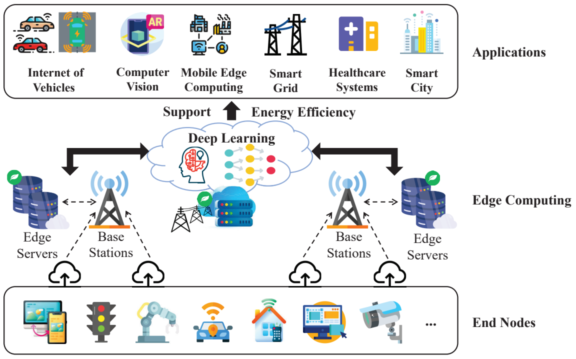

Fig. 3. Energy-efficient and collaborative end-edge-cloud framework with deep learning.

Fig. 3 shows an energy-efficient and collaborative end-edge-cloud framework with deep learning. In Fig. 3, end nodes, *e.g.*, vehicles, cameras, and sensors, collect data and transmit it to nearby edge servers through base stations. The data are further processed in the cloud, where deep learning models are trained using it, thereby supporting applications such as the Internet of Vehicles, computer vision, mobile edge computing, smart grid, healthcare systems, and smart city initiatives. While cloud-based deep learning offers powerful computation for large-scale analysis and intelligent decision-making, it also results in considerable energy consumption due to the enormous computations required. It is predicted that there will be over 80 billion edge devices connected to the Internet, and the amount of global data will reach 260 trillion gigabytes by 2030. More than 90% of data will need to be stored and processed locally. Most of the data is privacy-sensitive, and storing it in CDCs is risky due to the associated communication costs. For example, a location-based app named Waze helps users decrease congestion and select light-traffic roads. However, if users' own locations have to be shared with CDCs, they might not be kept safe. In addition, Waze needs to send a tremendous amount of data about each road to CDCs for producing the optimal route for users, which requires a high communication cost. However, connections between users and CDCs might be frequently unavailable. Moreover, training data collected from edge devices also needs to follow its privacy protection laws. For example, US laws in smart hospitals or Europe's laws require that data and models cannot be kept in remote CDCs. Thus, considering data privacy and network bandwidth, it is unnecessary and impractical to send all data to a remote CDC. In such cases, training of DL has to be performed in edge computing. It tends to consume high amounts of energy, which is usually limited in edge devices. The mismatch between resource-intensive DNN applications and energy-limited edge devices makes it challenging to apply DL in edge computing in an energy-efficient manner.

To the best of our knowledge, there is no survey focusing on energy optimization for DL for IoTs with edge computing. This survey aims to provide a) industrial applications, b) DNN-based energy optimization mechanisms for edge computing, c) trends and challenges in edge computing in Fig. 4. This work discusses key technical enablers and highlights various

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3640292

real-world applications, offering a holistic understanding of opportunities and challenges in building energy-efficient edge systems.

The rest of the paper is organized as follows. Section II gives typical industrial applications of DL technologies in IoTs and edge computing environments. Section III discusses mechanisms that apply DNN to achieve energy-efficient edge computing systems. Section IV discusses new trends, opportunities, and open research challenges of energy optimization for DL in edge computing systems. Section V concludes this work. The nomenclatures are shown in Table II. We first give a taxonomy of energy-efficient deep learning methods in edge computing in Table I. It organizes prior studies into four categories: (i) model-centric approaches, including pruning, quantization, and lightweight architectures; (ii) collaborative and distributed learning strategies like federated and split learning; (iii) hardware-centric solutions, including Processing-In-Memory (PIM), FPGA-based accelerators, memristor arrays, and neuromorphic chips; and (iv) system-level mechanisms, including computation offloading, resource allocation, and energy-aware scheduling.

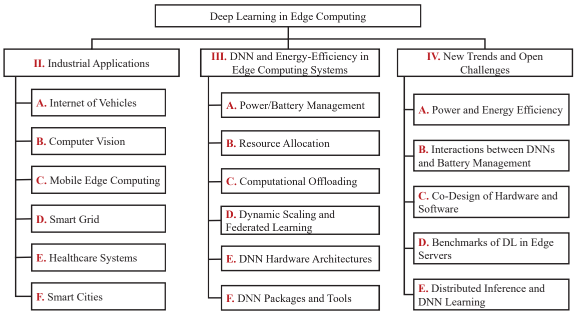
Deep Learning in Edge Computing
II. Industrial Applications
A. Internet of Vehicles
B. Computer Vision
C. Mobile Edge Computing
D. Smart Grid
E. Healthcare Systems
F. Smart Cities
III. DNN and Energy-Efficiency in Edge Computing Systems
A. Power/Battery Management
B. Resource Allocation
C. Computational Offloading
D. Dynamic Scaling and Federated Learning
E. DNN Hardware Architectures
F. DNN Packages and Tools
IV. New Trends and Open Challenges
A. Power and Energy Efficiency
B. Interactions between DNNs and Battery Management
C. Co-Design of Hardware and Software
D. Benchmarks of DL in Edge Servers
E. Distributed Inference and DNN Learning

Fig. 4. Logical framework. Industrial applications raise practical requirements for building energy-efficient edge systems. The needs are met by implementing energy-efficient mechanisms in DNNs. The drawbacks in the mechanisms bring new trends and open challenges for future research.

II. Energy-efficient DL Applications in Industrial Scenarios

A. Internet of Vehicles (IoVs)

There are many existing studies on applying energy-efficient DL techniques in edge computing. Emerging DL technologies can provide intelligence for real-time decision making and achieve energy efficiency for computational facilities and vehicles. Vehicular edge computing (VEC) is an application of mobile edge computing (MEC) to vehicular scenarios [3]–[6] summarized in Table III. Wang *et al.* [3] design a fog-cloud computational offloading method in IoVs for minimizing both the energy consumption of vehicles and that of computational facilities. An offloading problem is formulated as an NP-hard problem and solved using a heuristic algorithm incrementally. Specifically, a predictive combination transmission mode is designed for vehicles, and a DL model is established for computational facilities for determining the optimal allocation of workload. Ning *et al.* [4] consider a challenge of how to meet the quality of experience (QoE) of users in intelligent networks with limited computing abilities of vehicular fog nodes. They develop a three-layer offloading framework for IoVs to minimize their total energy consumption while meeting the delay constraints of users. An optimization problem is formulated and decomposed into two parts: flow redirection and offloading decision. Then, a deep reinforcement learning (DRL)-based mechanism is proposed to address this issue. Kong *et al.* [5] propose a joint caching and computing framework by using an algorithm of deep deterministic policy gradient (DDPG). They consider IoVs with mobile networks. An optimization problem is formulated to minimize the system energy consumption, which is achieved by DDPG. Hussain *et al.* [6] propose a reinforcement learning-based model to optimize energy efficiency and routing. A reward matrix and the Bellman equation are used to determine the most energy-efficient route from source to destination.

B. Computer Vision

Deep learning can enable energy-efficient and computer vision-related problems for edge devices, thereby overcoming their energy limitations [9]–[12]. Fig. 5 illustrates a hybrid energy-powered deep learning mechanism in an end-edge–cloud computing system. In Fig. 5, the cloud layer is powered by the main grid and protected by uninterruptible power systems (UPS), the edge layer is supplied through UPS-backed stable power, and the end devices rely on local batteries or direct connections. Through hierarchical collaboration among cloud-side centralized training, edge-side adaptive inference, and end-side lightweight sensing, the system efficiently reduces redundant computation and data transmission, thereby achieving overall energy-efficient and sustainable operations. A Lyapunov technique is adopted to transform a constrained MDP into a regular one, which is further solved by an asynchronous advantage actor-critic algorithm. Then, energy saving is achieved while QoE is also enhanced.

Lim *et al.* [9] propose an energy-efficient communication method in edge computing with DL, and it decreases the energy consumed by image transmission with edge computing. An energy-efficient IoT camera called CamThings is implemented by using communication and periodic on-off scheduling. CamThings outperforms a method that only adopts periodic on-off scheduling regarding both lifetime and energy consumption. Choi *et al.* [10] propose VisionScaling, a method that jointly optimizes learning models and resources for mobile vision applications. By integrating multiple learning models within an online convex optimization framework, they address the challenge of improving performance while considering energy consumption and Processed Frames Per Second (PFPS) in edge computing, achieving higher PFPS, greater energy efficiency, and better adaptability. Albanese *et al.* [11] design a generic and modular system that controls unmanned aerial vehicles (UAVs) by using vision-based deep learning tasks running in energy-constrained UAVs. Two vision-based navigation configurations with LeNet-5 and MobileNetV2 models are used, and the system energy consumption is decreased without reducing the quality of service. Zawish *et al.* [12] design a DRL-based pruning method, which compresses CNNs adaptively according to the energy management policy and

TABLE I
TAXONOMY OF ENERGY-EFFICIENT DL APPROACHES IN EDGE COMPUTING.

Category	**Subcategory**	**Representative approaches**
Model-centric	Structure reduction	Pruning [12], lightweight architectures / NAS [37]
	Numeric compression	Quantization and mixed-precision computation [65], low-bit computing [63]
	Knowledge transfer	Knowledge distillation and teacher-student compression [32]
Distributed learning	Federated training	Energy-aware FL, client selection, and adaptive aggregation [56], [61], [62]
	Split / partial learning	Split learning [60]
	Collaborative inference	Model partitioning and edge-cloud inference cooperation [58], [59]
Hardware-centric	Memory-centric computing	Processing-In-Memory (PIM) and ReRAM-based accelerators [38], [40], [63]
	FPGA/ASIC design	FPGA accelerators and custom ASIC/systolic-array designs [64], [65]
	Neuromorphic computing	Spiking-neural processors and neuromorphic chips [69]
System-level mechanisms	Resource management	Computation offloading and resource allocation [47], [48], [50]–[52], [55]
	Dynamic energy control	DVFS/AVFS and energy-aware scheduling [58]
	Execution optimization	Adaptive batching and operator fusion *via* accelerator/toolchain support [33], [69], [71]–[73]

TABLE II
NOMENCLATURES

Abbreviation	Explanation	Abbreviation	Explanation	Abbreviation	Explanation
DL	Deep Learning	IoT	Internet of Things	TPU	Tensor Processing Unit
CDCs	Cloud Data Centers	RNNs	Recurrent Neural Networks	CNNs	Convolutional Neural Network
IoVs	Internet of Vehicle	VEC	Vehicular Edge Computing	MEC	Mobile Edge Computing
DRL	Deep Reinforcement Learning	HMEC	Hybrid MEC	GVs	Ground Vehicles
UEs	User Equipments	FPGA	Field Programmable Gate Array	MDP	Markov Decision Process
AI	Artificial Intelligence	MARL	Multiagent Reinforcement Learning	PFPS	Processed Frames Per Second
RIS	Reconfigurable Intelligent Surface	PD	Partial Discharge	FDIA	False Data Injection Attacks
DFL	Deep Federated Learning	PBDL	Permissioned Blockchain and DL	SSVAE	Stacked Sparse Variational AE
LSTM	Long Short-Term Memory	SA	Self Attention	GenLS	Generative Latent Space
DRAM	Dynamic Random-Access Memory	DDNNs	Distributed DNNs	DNN	Deep Neural Networks
ISARA	Island-style Systolic Array Reconfigurable Accelerator	RRAM	Resistive Random Access Memory	SP-PIM	Super-Pipelined Processing-In-Memory
SDN	Software-Defined Networking	GPUs	Graphics Processing Units	QoE	Quality of Experience
UAVs	Unmanned Aerial Vehicles	DDPG	Deep Deterministic Policy Gradient	IIoT	Industrial Internet of Things
ECG	Electrocardiogra	CPU	Central Processing Unit	GNN	Graph Neural Network

TABLE III
SUMMARY OF IOV APPLICATIONS

Studies	Evaluation parameters	Expected latency	Number of edge devices	DNN models
[3]	Power consumption and delay	40-120 sec.	20	CNN with 2-channel input
[4]	Average energy consumption	N/A	5	Deep Q-Network
[7]	Runtime and average energy consumption	2 sec.	10-100	DNN with a scheduling layer
[8]	Power consumption and inference time	2-5 sec.	N/A	VGG-16, DenseNet-128-10

accuracy requirements for IoT applications. It is shown that CNNs with DRL-driven pruning consume relatively higher energy than their counterparts while maintaining accuracy.

C. Mobile Edge Computing

The time-varying dynamics of mobile edge computing make it challenging to achieve energy-efficient offloading and resource allocation. DL can address the challenge and develop automatic computation offloading and resource allocation strategies to optimize the energy consumption of edge devices [13]–[17]. Table IV summarizes industrial applications for MEC. Jin *et al.* [13] investigate a multi-user MEC system, and propose computation offloading and resource allocation policies with DRL to minimize energy consumption in a dynamic environment. Zhu *et al.* [14] propose a computation offloading mechanism to decrease the completion time of applications and the energy consumed by mobile devices. An optimal strategy for energy and time-optimized computation offloading is obtained with deep Q-learning. It outperforms local execution and random offloading in terms of both energy consumption and workflow completion time.

Li *et al.* [15] investigate a multi-user MEC system in which many UEs realize energy/delay-optimized computation offloading through wireless channels connected to an MEC server. The weighted cost of energy consumption and delay for UEs is formulated as an optimization objective. A DRL-based Q-learning framework is proposed to jointly optimize computational resources in MEC systems. Ansere *et al.* [16] jointly optimize stochastic computation offloading, dynamic resource allocation, and content caching to maximize the en-

TABLE IV
SUMMARY OF INDUSTRIAL APPLICATIONS FOR MEC

Studies	Evaluation parameters	Expected latency	Number of edge devices	DNN models
[13]	Cost and capacity	N/A	5	Fully connected DNN with two hidden layers
[14]	Energy consumption and completion time	10-50 sec.	1	Deep Q-learning
[15]	Cost	N/A	3-7	DRL
[18]	Energy consumption and Execution time	10 ms	15-30	DDPG with actor-critic networks

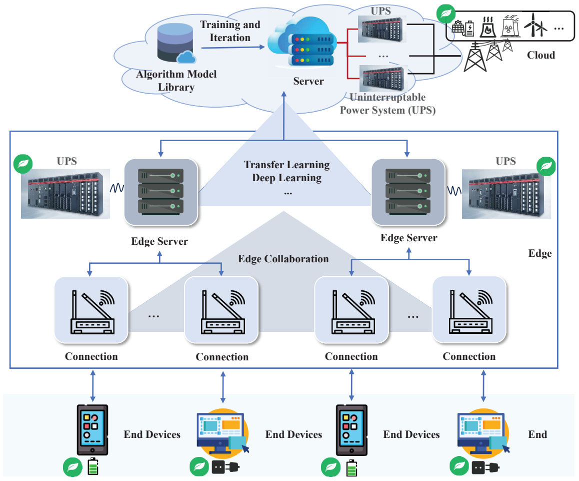

Fig. 5. A hybrid energy-powered deep learning mechanism in an end-edge–cloud computing system. End devices generate raw data, which is transmitted to nearby edge nodes for initial processing. Edge servers collaborate with each other to share intermediate features or models, enabling distributed inference and transfer learning close to data sources. Processed information and partially trained models are synchronized with the cloud, where centralized servers perform global training. The integrated models are deployed back to the edge, forming a loop of end collection, edge collaboration, and cloud-based global training.

ergy efficiency of mobile edge computing. They design a quantum deep reinforcement learning algorithm to exponentially increase the speed of caching learning and content caching efficiency in multi-dimensional large-action and continuous spaces. A faster Grover algorithm is used to improve processing efficiency and data retrieval, thereby outperforming other benchmarks in maximizing energy efficiency subject to energy consumption, transmission power, and transmission latency. Xiao *et al.* [17] design a multi-agent reinforcement learning-based and energy-efficient collaborative inference scheme in MEC, which enables mobile devices to select the partition point of deep learning and their collaborative edge based on the channel conditions, the image quantity, and the performance of previous inference. An exchange mechanism for learning experiences investigates the Q-values of neighboring mobile devices to speed up policy optimization while consuming less energy in MEC.

D. Smart Grid

A growing number of edge devices are incorporated to collect information about a smart grid, and DL can increase the energy efficiency of the smart grid by capturing higher-order statistical information of its complex data [19]–[22]. Lv *et al.* [19] adopt several deep learning algorithms to analyze distributed renewable energy generation and consumer power data for edge computing-supported smart grid, thereby improving the efficiency of information transmission and processing in smart grid. Dong *et al.* [20] design a learning-based decision-making method for economic energy dispatch of smart grid based on the cloud-edge computing architecture. The well-trained model is adopted locally at edge devices keeping long-term parameters fixed for realizing the real-time energy dispatch. Cloud resources are adopted to solve the optimal dispatch decision over historical operating patterns of smart grid. Dong *et al.* [21] consider a risk-aware energy scheduling problem for a smart grid-powered MEC network. It considers the conditional value-at-risk measurement for both energy generation and consumption, thereby minimizing the expected residual of scheduled energy for MEC networks. An asynchronous advantage actor-critic algorithm with shared neural networks based on multi-agent deep reinforcement learning is proposed to mitigate the dimensionality curse of the state space and chooses the optimal policy among agents. Su *et al.* [22] propose an efficient and secure federated-learning-enabled artificial intelligence of Things scheme for sharing of private energy data in smart grids with collaboration of edge and cloud. Specifically, a federated learning framework assisted by edge and cloud for privacy-preserving and communication-efficient energy data sharing of users in smart grids. A two-layer deep reinforcement-learning-based incentive algorithm is designed to increase participation of energy data owners and high-quality model contribution.

E. Healthcare Systems

There are several studies on applications of DL in building energy-efficient healthcare edge systems [23]–[26]. Zhang *et al.* [23] propose a self-adaptive power-control-based, efficiency-aware method to reduce the energy consumption of the edge system and improve battery lifetime and reliability. A joint DL and Internet of Medical Things (IoMT) framework is proposed to process cardiac images from remote elderly patients. Then, a DL-driven layered architecture for IoMT is designed, and a battery model is proposed that leverages features of body posture and the wireless channel. Osman [24] provides an improved and reliable IoMT method to reduce interference from other transmitting devices sharing the same IoMT spectrum in hospitals and medical institutions. It introduces an interference-avoidant distributed deep learning method for IoMT to medical receptionists. The system's signal-to-interference-plus-noise ratio is maintained while achieving the highest energy efficiency. Sornalakshmi *et al.* [25] present an energy-aware heart disease prediction system by using improved spider monkey optimization and a weight-optimized neural network for reactive healthcare and IoT-based systems. It includes phases of energy-efficient data transmission and heart disease prediction. Simulation results demonstrate that the proposed mechanism extends the network lifespan while consuming less energy than existing state-of-the-art techniques. Gowri and Baranidharan [26] present a hybrid fog server classification model based on dueling deep Q-learning and chaotic Lévy flight. A fog-enabled secure healthcare system is built, and it includes layers of IoT, fog, and cloud. The performance of the proposed algorithm is evaluated in terms of several important indicators, including energy consumption, computational cost, traffic overhead, makespan, latency, and network utilization.

F. Smart Cities

DL has been used to build MEC-enabled, energy-efficient smart cities recently [27]–[31]. Udayakumar *et al.* [27] propose an integrated method combining deep learning with edge computing to optimize energy consumption and enhance sustainability in IoT-based smart cities. The framework uses edge devices for local data analysis, reducing latency and enabling real-time decision-making. It uses deep learning models to analyze intricate data relationships and provide precise predictions of energy consumption patterns. The integration of deep learning and edge computing addresses challenges posed by the massive data generated by IoT devices while ensuring energy efficiency and responsiveness in smart cities. The work in [28] presents a DL model for efficient prediction of short-term energy consumption while keeping effective communication between energy users and providers. The proposed model comprises multiple stacked spatiotemporal modules, each comprising a temporal transformer and a spatial transformer to capture temporal and spatial information. Haseeb *et al.* [29] introduce a secure and intelligent edge-enabled computing framework for building sustainable cities with energy-efficient IoT, and develop a communication strategy to reduce liability in energy management and data security for data transport. The proposed method uses deep learning to select optimal features for data routing, training the sensors to predict the best routes to edge servers. The work in [30] adopts a deep learning method to efficiently manage energy for building sustainable smart cities. It proposes a framework for optimizing energy use in IoT-enabled smart cities by leveraging deep learning algorithms. It uses real-time data from various sources, including sensors, devices, and smart grids, to support smart energy-saving and efficiency decisions, enabling a more environmentally conscious and effective urban future. Alkhalifa *et al.* [31] present a hybrid deep learning-based and energy-efficient intrusion detection for edge computing by applying a starfish optimization algorithm, thereby providing intelligent edge computing in smart cities with advanced optimization models. It integrates a convolutional neural network and a bidirectional gated recurrent unit with a cross-attention mechanism to achieve fast and computation-efficient classification.

III. DNN and Energy-Efficiency in Edge Computing Systems

The performance improvement of DNNs is achieved with increased energy consumption, which grows significantly as DNNs become deeper and larger in scale. It is becoming more difficult for edge devices to run such DNNs. Thus, the energy efficiency of limited-edge devices and the computational complexity of DNNs have gaps that need to be optimized carefully for achieving real-time DL applications. Table V summarizes representative frameworks with columns Framework, Hardware, Energy-Efficiency Techniques, and Use Case. Supported by the technologies summarized in this section, recent empirical studies have quantitatively demonstrated efficiency gains achieved by energy-oriented deep learning methods at different levels. At the software level, model-centric approaches, such as pruning, quantization, lightweight architectures, and neural architecture search, efficiently reduce the energy consumption of edge systems [35]–[37]. For example, Bouzidi *et al.* [35] show that HADAS achieves up to 57% energy savings on CIFAR-100 and ImageNet, while Gong *et al.* [37] achieve 30–35% efficiency improvement through mixed-precision neural architecture search. At the hardware level, accelerator-based solutions, including Processing-In-Memory, ReRAM-based designs, and systolic-array architectures, reach up to 2–3× improvement in energy efficiency and 30–50% latency reduction compared with state-of-the-art GPU or CPU implementations [38]–[40]. For instance, Xiang *et al.* [38] report that NOR Flash analog arrays deliver more than 2× energy efficiency, and Heo *et al.* [40] demonstrate a 45% latency reduction with the SP-PIM super-pipelined accelerator.

A. Power/Battery Management

The power/battery management has been attracting significant attention in recent years [41]–[46]. Yang *et al.* [41] consider a joint problem of minimizing energy and latency to distribute hierarchical machine learning tasks in MEC. They propose a framework that allows end devices with shallow neural network models to offload energy-intensive, latency-sensitive tasks to nearby servers equipped with powerful DNN models. The offloading strategy is formulated as a piecewise convex optimization problem, and an optimal partial offload method is analytically derived to minimize the weighted sum of energy consumption and latency. Kang *et al.* [42] design a joint optimization approach for distributed energy-efficient data centers. It adopts an LSTM algorithm to increase the prediction accuracy of green energy for a long period, and an unsupervised DL method to optimize coordinated frequency scaling and right-sizing. In addition, a macro- and micro-scale management method for data center management is presented to decrease wake-up transition overhead and improve high energy efficiency. Xu *et al.* [43] adopt an energy harvesting method and realize dynamic access control for MEC. It aims to maximize the long-term average rate of uplink transmission and minimize the energy consumption of transmission for green IoT networks. Each IoT device is powered by a battery, which utilizes energy from its surrounding environment. The problem is formulated as an MDP with unknown system

TABLE V
REPRESENTATIVE FRAMEWORKS FOR ENERGY-EFFICIENT DL AT THE EDGE.

Framework	Hardware	Energy-efficiency Techniques	Use Case
HADAS [35]	Edge SoC/accelerators	Hardware-aware NAS, early-exit, & DVFS co-optimization	Vision inference in edge devices, model-centric energy reduction
RL-based DVFS [58]	Mobile/edge+cloud	DRL for DVFS, feature offloading, & distributed inference	Collaborative inference with load/latency constraints
SP-PIM [40]	28nm PIM chip	Super-pipelined PIM, local error prediction, & sparsity handling	On-device training/inference with high throughput and power efficiency
ISARA [39]	RRAM PE islands/systolic array	Bit-fusion ADC, reconfigurable PE tiles, & reduced data movement (PIM/memristors)	Edge AI inference for CNNs
EAFL+ [62]	Mobile/IoT clients	Energy-aware FL, power control, & adaptive participation	Federated training with device lifetime preservation
QoE-driven offloading [52]	MEC	DRL-based offloading & latency/energy-aware allocation	Real-time services with QoE constraints
FPGA toolchains (Vitis AI/OpenVINO) [33]	FPGA/heterogeneous edge	Quantization, operator fusion, & pipeline scheduling	Deployment of customized accelerators with power budgets
MLPerf Tiny/Inference, EEMBC MLMark [34]	TinyMCU / Mobile–Edge / Embedded	Standardized benchmarks, & partial energy reporting	Cross-platform evaluation of energy and latency trade-offs

dynamics. An LSTM-based deep Q-network is proposed to achieve optimal access control.

Dhull *et al.* [44] tackle the challenges of implementing neural networks with quantized synaptic weights in nonvolatile memory devices, which suffer from limited programmable states and inherent stochasticity. They demonstrate on-chip training and inference of neural networks using a quantized magnetic domain wall-based synaptic array integrated with complementary metal-oxide semiconductor circuits. A comprehensive synaptic model is developed, accounting for device variations and write stochasticity, while domain-wall pinning through physical constrictions ensures stable weight representation. Hong *et al.* [45] address the challenge of balancing energy efficiency and content freshness in high-definition map caching for vehicular edge networks. They propose a prioritized double DRL algorithm to jointly optimize roadside sensor energy consumption and map content freshness through coordinated edge updating and roadside unit transmission resource allocation. By modeling the problem as a Markov decision process, their algorithm integrates DRL with prioritized experience sampling to minimize long-term operational costs related to energy usage and content staleness. Zhang *et al.* [46] propose solutions to the dual challenges of energy scarcity and computational limitations in IoT nodes by integrating wireless power transmission with mobile edge computing. They introduce a joint optimization framework to maximize the sum computation rate in wireless-powered edge networks, coordinating power transfer duration, task offloading time allocation, and energy partitioning.

Overall, existing studies on power and battery management demonstrate diverse strategies, from optimization of task offloading and renewable energy prediction to energy harvesting and wireless power transfer. Their strengths lie in extending device lifetime and improving sustainability, but many approaches rely on accurate workload or energy availability prediction and may face scalability issues in highly dynamic environments. Among them, hybrid solutions that integrate prediction-based management with adaptive reinforcement learning are particularly promising, as they can strike a balance between long-term energy availability and real-time adaptability.

B. Resource Allocation

There are some investigations focusing on resource allocation in edge computing systems [47]–[51]. Lei *et al.* [47] design time and energy-efficient methods for content delivery at a network edge. Two resource allocation problems are formulated to minimize transmission time and energy in content delivery. They are formulated as mixed-integer linear programs and solved using learning-based methods, including fully connected DNNs and CNNs, to provide a computationally efficient and high-quality solution. Dai *et al.* [48] adopt DRL to obtain an optimal resource allocation and computation offloading strategy for minimizing the energy consumption of the system by using network information, including computation resources, available bandwidth, and wireless channel state. A multi-user edge computing architecture for heterogeneous networks is presented, considering the strong relationships among devices based on application needs or radio access. A joint resource allocation and computation offloading problem is formulated as a DRL one, and a new DRL-inspired algorithm is proposed to minimize the energy consumption of the system.

Gu *et al.* [49] propose a framework for efficient training of DDNNs that affects the best configuration of a training cluster with heterogeneous computing resources. It adopts pre-training with a limited number of training steps and predicts training time, energy, and energy-delay product for each configuration of a training cluster. It performs training of DNN models for the remaining steps using a selected best cluster configuration according to the preferences of DNN service providers, such as energy efficiency and training time. Wu *et al.* [50] focus on long-term optimization of joint task offloading and resource allocation in multi-UAV edge computing networks. The problem is formulated as an MDP with hybrid discrete-continuous action spaces, where continuous parameters are mapped to discrete offloading decisions. To address this challenge, they propose a Deep Deterministic Policy Gradient algorithm enhanced with multi-head self-attention mechanisms, which enables the coordinated learning of resource allocation and task distribution strategies. Li *et al.* [51] address energy-efficient task offloading and resource allocation in dynamic small-cell mobile edge computing networks, where time-varying channels and decentralized infrastructure pose coordination challenges. They propose a distributed multi-agent Proximal Policy Optimization framework to minimize total energy consumption under delay constraints. It is modeled as a partially observable MDP, allowing each small-cell base station to independently learn policies based on local observations, guided by a collaboratively designed global reward function. A state normalization mechanism is also introduced to stabilize training and improve policy performance.

In summary, resource allocation studies in edge computing have explored both optimization-based formulations and learning-driven strategies. Traditional optimization ensures high solution quality and interpretability, while DRL and multi-agent methods offer adaptability to dynamic environments and heterogeneous resources. However, optimization models often suffer from high complexity and poor scalability, whereas learning-based approaches may incur long training time and stability issues. Promising directions lie in hybrid frameworks that combine model-based guarantees with DRL adaptability, enabling efficient and stable resource allocation under realistic and time-varying conditions.

C. Computational Offloading

Existing approaches for computational offloading can be categorized into centralized and decentralized strategies [52]–[57]. Among centralized approaches, Lu *et al.* [52] propose a deep deterministic policy gradient algorithm to trade off latency, energy consumption, and task success rate in edge-enabled IoT. Xu *et al.* [53] exploit UAV and reconfigurable intelligent surface, and design a centralized algorithm based on deep neural networks to optimize energy efficiency for device-to-device users such that the quality of service is met, yielding energy-efficient communication and green transmission. Fabiani *et al.* [54] propose an unsupervised and centralized learning framework to optimize the power allocation of downlink in edge networks with the power control learning. Deep neural networks are trained to predict power coefficients while handling the constraints of power budgets and pilot contamination. Decentralized and energy-driven solutions focus on adaptive or distributed learning. Shuai *et al.* [55] design a collaborative offloading scheme for satellite edge networks, where distributed DNNs dynamically adapt to mobility and link variations, jointly optimizing energy and latency. Hasan *et al.* [56] address vehicular edge computing by integrating federated learning into offloading decisions, thereby reducing data transfer overhead while improving energy efficiency. Tripathy *et al.* [57] introduce a DRL-based task prioritization algorithm for multi-modal IoT systems, modeling energy and delay trade-offs under heterogeneous modalities. These decentralized methods enhance adaptability and robustness, but require device-side intelligence and higher training complexity.

In addition, some works discuss the trade-off between energy consumption and latency. For example, the work in [52] emphasizes the benefits of energy-aware and QoE-driven offloading strategies, highlighting that delay and task success rates deteriorate when energy savings are prioritized. On the other hand, the work in [55] minimizes latency through digital twin synchronization and distributed DNN-based algorithms, yet it may require higher device power budgets to support frequent communication and local model updates. Recent studies such as [56] and [57] jointly optimize both objectives, formulating the offloading as a multi-objective problem that balances energy savings with acceptable service latency.

Building upon these adaptive offloading mechanisms, recent research has started to explore distributed inference as an effective extension of energy-efficient edge–cloud collaboration. Distributed inference partitions DNN execution across heterogeneous devices such that early layers with lower computational intensity are processed locally at the edge, while deeper layers are transmitted to more capable clouds for completion [58], [59]. This collaborative design achieves a flexible balance between computation and communication energy. Under low-load conditions, local inference minimizes transmission overhead, whereas during high-demand or congested periods, selective offloading alleviates device-side thermal and power stress. Furthermore, model partitioning and dynamic resource scaling techniques, such as DVFS and elastic model slicing, enhance energy proportionality across distributed tiers [10], [60]. Incorporating distributed inference into the analysis of energy efficiency provides a more holistic understanding of how computation and communication jointly determine the overall energy footprint of edge–cloud intelligent systems.

The overall energy efficiency of edge DL systems is jointly determined by interactions between computation, communication, and hardware operation. Computation-intensive tasks consume significant on-device power, whereas aggressive offloading increases transmission energy and network congestion, leading to higher end-to-end latency [52]. Similarly, minimizing communication overhead through local execution results in elevated thermal stress and reduced inference throughput. Therefore, achieving sustainable energy efficiency requires holistic optimization that considers computation placement, communication scheduling, and dynamic energy management simultaneously. Recent approaches leverage multi-objective optimization and adaptive control to find Pareto-optimal operating points that balance device power, network cost, and model accuracy under varying workloads [10], [56], [57]. Recognizing and quantifying these trade-offs is essential for designing next-generation edge–cloud systems for stable and energy-aware deep learning.

D. Dynamic Scaling and Federated Learning

Dynamic scaling is an emerging mechanism for controlling energy consumption of DNNs by adaptively adjusting model complexity or resource usage according to runtime conditions. Bouzidi *et al.* [35] propose HADAS, a hardware-aware dynamic neural architecture search framework that jointly optimizes early-exit points and DVFS settings, achieving up to 57% energy gains on CIFAR-100 across edge platforms. Zhang *et al.* [58] integrate reinforcement learning with collaborative edge–cloud inference to co-optimize computing frequencies and feature offloading, reducing energy consumption. Lim *et al.* [60] present a Lyapunov-based dynamic DNN partitioning scheme that adaptively adjusts model split between devices and servers, effectively improving the tradeoff of energy and latency under varying workloads. Above dynamic scaling techniques prove significant energy savings. However, they require complex runtime profiling or controller training and are unstable when workloads fluctuate sharply.

Federated learning (FL) represents another promising paradigm for reducing energy overhead in edge AI systems by minimizing the transmission of costly raw data. Hu *et al.* [61] propose a dynamic scheduling algorithm for federated edge learning that optimizes device participation under energy and latency constraints, thereby improving long-term learning performance. Arouj *et al.* [62] develop EAFL+, an energy-aware FL scheme that adapts device transmission power and participation frequency to extend battery lifetime, demonstrating notable energy savings. These FL approaches provide privacy preservation and communication efficiency, but they may suffer from slower convergence and fairness issues under heterogeneous devices, requiring careful scheduling and parameter tuning.

While federated and centralized learning paradigms both aim to achieve high model performance, they exhibit distinct trade-offs in terms of accuracy and energy consumption. FL eliminates centralized data aggregation and reduces communication with the cloud, thereby improving data privacy and lowering transmission energy [56], [62]. However, local model training on multiple heterogeneous edge devices introduces computation overhead and accuracy degradation due to non-independent and identically distributed data distributions and limited local updates [61]. In contrast, centralized training or inference leverages powerful servers for global optimization and consistent accuracy, but often at the cost of higher communication energy and latency [58]. Recent empirical studies report that FL frameworks reduce end-to-end energy consumption by approximately 25–40% compared with fully centralized systems, with only 2–5% accuracy deviation with well-aggregated mechanisms [56], [62]. Thus, balancing model accuracy and energy efficiency requires hybrid optimization strategies that adapt training frequency, aggregation intervals, and participant selection based on real-time system conditions.

E. DNN Hardware Architectures

DNN hardware architectures are crucial to the performance of DL in edge computing systems, and several studies have been conducted on the design of hardware architectures specifically tailored to DNNs. Optimizations of DNNs need to be supported by hardware accelerators, and there are several studies on accelerators for edge devices [38]–[40], [63]–[65]. Table VI shows the summary of selected studies on DNN hardware architectures. Xiang *et al.* [38] propose a hardware implementation of an analog DNN based on NOR Flash Computing Array. It removes additional analog-to-digital/digital-to-analog transformation between adjacent layers. It is high-speed, energy-efficient, and promising to realize AI at the edge. Chen *et al.* [63] propose a compute-in-memory method that is paired with high-density nonvolatile memory to improve operations of DNNs for AI edge processors. Specifically, a binary-input hardware-driven ternary-weighted network is proposed to achieve smaller energy-hardware cost by using pseudo-binary nonvolatile memory macros and a two-macro DNN. Verhelst and Moons [64] introduce tightly interwoven methods for hardware and software processing for energy efficiency. They show details of the implementation of algorithmic innovations with flexible processing architectures. It also points out implementation challenges in designing efficient data transfer, results interpretation, image slicing, careful algorithm-architecture cooptimization, and new proprietary hardware for DL.

Yang *et al.* [39] propose ISARA, a reconfigurable island-style systolic array accelerator based on memristive resistive random-access memory for edge AI applications. The architecture incorporates flexible processing element tiles inspired by field-programmable gate arrays (FPGAs) to optimize data flow within the systolic array. It also integrates a bit-fusion scheme that dynamically adjusts the resolution of analog-to-digital converters, effectively reducing power consumption and mitigating the impact of RRAM conductance variation. Heo *et al.* [40] introduce SP-PIM, a high-throughput super-pipelined processing-in-memory accelerator designed to address power and latency challenges in on-device machine learning training for edge devices. The architecture incorporates pipeline optimization to eliminate training bottlenecks, hardware-efficient floating-point units to reduce power and area overhead, and dynamic sparsity handling for efficient gradient computation. Kudo *et al.* [65] present a DNN hardware architecture that supports variable and binary bit-width logarithmic quantization, which employs a distributed accumulator for multi-bit serial input processing with a low-overhead single-accumulator circuit for binary operations. Experimental results demonstrate reduced hardware resource usage and energy consumption compared to variable bit-width architectures, while maintaining computational speed, functionality, and inference accuracy.

While the aforementioned accelerator designs demonstrate significant energy savings, their practical applicability varies depending on the characteristics of edge devices. For ultra-low-power IoT nodes, *e.g.*, wearables, smart sensors, lightweight analog computing arrays such as NOR flash-based [38] or ReRAM-based [39], [63] solutions are attractive due to their compact footprint and reduced conversion overhead. However, their limited flexibility and precision may constrain applicability in scenarios requiring complex models. Mid-tier devices, *e.g.*, smartphones, drones, can leverage FPGA-based accelerators [60] or hybrid processing-in-memory schemes [62] that balance programmability with efficiency, making them suitable for frequently updated DNNs in dynamic environments. High-end edge servers or gateways benefit more

TABLE VI
SUMMARY OF SELECTED STUDIES ON DNN HARDWARE ARCHITECTURES

Studies	End devices	Criteria	Speed up approaches	Applications	DNN models
[38]	NOR flash computing array and peripheral circuits	Recognition accuracy Time delay Energy consumption	NOR flash computing array	Analog DNNs	5-layer analog DNN
[63]	Megabit high-density nonvolatile memory ReRAM macro	Latency Energy cost	Hardware-driven binary-input ternary-weighted network	Binary DNNs	CNN with 3×3 kernels
[66]	Xilinx FPGA	Performance Power consumption	Custom inference accelerator	Automatic speech recognition	DNNs for acoustic models
[39]	RRAM-based PE islands	Computational efficiency Latency reduction Power consumption	Island-style systolic array Bit-fusion Reconfigurable ADC	Edge AI	LeNet-5 VGG-16 ResNet-18
[40]	28-nm CMOS chip multi-level pipelined cores	Training speed area efficiency power efficiency	Super-pipelined architecture local error prediction dual-sparsity handling in PIM macros	On-device learning	VGG8B

from architectures emphasizing scalability and throughput, such as SP-PIM [40] or systolic-array accelerators [39], which offer better support for large-scale models at the cost of higher area and power budgets. The choice of architectures depends on the trade-off between energy efficiency, flexibility, and workload scale, suggesting that no single solution is universally the best across various edge systems.

F. DNN Packages and Tools

There are some popular packages and tools that are designed to realize DNNs due to the sharp increase of applications of DNNs in edge computing systems [36], [37], [59], [67]–[70]. Table VII shows the summary of selected studies on DNN packages and tools. Shafique *et al.* [36] introduce current and new trends of secure, reliable, efficient, and scalable machine learning architectures for IoT devices, and also points out challenges in realizing its expected goals. They provide a roadmap for addressing challenges in designing scalable, energy-efficient, and high-performance architectures that enable machine learning on the edge. Gong *et al.* [37] design a method to realize joint optimization of a neural architecture and a quantization space. It finds optimal combinations of precisions and architectures to optimize both hardware energy consumption and prediction accuracy. It automates and improves flows across a design of neural architectures and the deployment of hardware. It provides better energy efficiency than its peers, *i.e.*, efficiency-aware NAS approaches and advanced quantization ones on two datasets. Edstrom *et al.* [67] provide optimization of memory hardware to meet the budget of power in IoT edge devices. It considers many factors, including accuracy, privacy, and power efficiency of different DL systems. According to an analysis of these factors, an integer linear program is formulated to minimize the mean square error. Then, an energy-efficient and near-threshold memory operation is achieved for different privacy needs with a slight reduction in classification accuracy. Vipin [68] presents a Python package that enables the development of fast DNNs on low-cost FPGA platforms. It combines hardware and software designs, enabling pre-trained or on-board trained networks in environments similar to TensorFlow. The resulting DNNs have accuracy close to software implementations at a lower energy cost.

Samajdar *et al.* [69] present a hardware and software prototype of an evolutionary algorithm-based learning system. It consists of a loop learning engine and an inference one. A learning engine can dynamically change the topologies and weights of neural networks in hardware, without requiring backpropagation training or manual optimization. The inference engine interacts with an environment, and it is optimized for efficiently executing irregular neural networks. A prototype is deployed in a suite of environments in OpenAI Gym, demonstrating that energy efficiency is achieved two to five orders of magnitude greater than that of state-of-the-art desktop and embedded GPU and CPU systems. Kang *et al.* [59] design a lightweight scheduler to partition DNN computation among data centers and mobile devices at the granularity of layers of neural networks. It effectively utilizes resources in CDCs and edge nodes, resulting in reduced computing time, lower energy consumption, and higher traffic throughput. Alkendi *et al.* [70] introduce GNN-Transformer, a graph neural network-driven transformer algorithm designed to address noise filtration challenges in neuromorphic vision systems. While neuromorphic cameras offer advantages such as low power consumption and high-speed processing, they are vulnerable to measurement noise that degrades event-based perception. The proposed framework incorporates an EventConv spatiotemporal message-passing mechanism to capture asynchronous event correlations while preserving temporal dynamics. To enable effective supervised training under varying illumination conditions, they present a known-object ground-truth labeling method that generates labeled datasets from experiments conducted in extreme environments, including moonlight-level lighting.

There are several techniques that benefit both practitioners and theorists in deploying energy-efficient DNNs. For the hardware, FPGA packages and toolchains, *e.g.*, Xilinx Vitis AI, and Intel OpenVINO, enable practitioners to prototype and deploy customized accelerators with fine-grained control over resource allocation and power budgets, while also providing researchers with an experimental platform to validate new architectural ideas. For the algorithms, combining neural ar-

TABLE VII
SUMMARY OF SELECTED STUDIES ON DNN PACKAGES AND TOOLS

Studies	End devices	Criteria	Speeding up approaches	Applications	DNN models
[37]	NVIDIA Tesla P100 GPU	Accuracy Energy efficiency	Neural architecture search and mixed precision quantization	Hardware-efficient NNs	CIFAR-100 ImageNet
[67]	Platform with 45-nm CMOS	Privacy Accuracy Power efficiency	Memory hardware optimization	Image recognition	CNNs
[68]	Hybrid FPGA	Accuracy Resource utilization	Hardware-software co-design	FPGA-based edge computing	5-layer NN
[69]	GENESYS SoC in 15nm	Runtime Energy efficiency	Evolution engine and accelerator for dense addition and multiplication	Open AI Gym games	Typical NNs
[59]	NVIDIA Jetson TK1	Latency Energy consumption	Layer-level computation partitioning	Computer vision, speech, and natural language domains	AlexNet
[70]	DAVIS346C DVS	Filtration accuracy Computational time	GNN-Transformer architecture KoGTL labeling	Neuromorphic camera denoising	GNN-Transformer
[71]	Microcontroller units	Latency Energy consumption	Quantization-aware training Micro-operator optimization	On-device inference for embedded IoT	TinyCNN
[72]	Edge sensor nodes	Runtime efficiency Accuracy	End-to-end TinyML pipeline	Data classification and anomaly detection	Small DNN
[73]	ARM-based edge processors	Energy efficiency Accuracy	Neural architecture search for lightweight models	Image recognition and mobile inference	MobileNetV3

chitecture search with quantization has emerged as an effective method to automatically generate lightweight models tailored for edge devices. Such approaches enhance deployment efficiency for practitioners by reducing inference latency and energy consumption, and enable theorists to systematically explore the trade-off between accuracy, complexity, and energy efficiency in the design of DNNs. In addition to general software toolchains, TinyML has recently emerged as a key paradigm for deploying deep learning models directly on microcontrollers and ultra-low-power devices. Lightweight deployment frameworks, such as TensorFlow Lite Micro [71], Edge Impulse [72], and TinyNAS [73], enable end-to-end optimization from model compression and quantization to on-device inference. These frameworks exemplify how neural network compression, pruning, and quantization-aware training can achieve substantial energy and memory savings with minimal accuracy loss. Incorporating such technologies extends energy-efficient deep learning beyond hardware and algorithmic co-design towards practical and real-time intelligence at the edge. This bridges the gap between model-centric optimization and deployment-level realization.

IV. New Trends and Open Challenges

Many challenges exist in optimizing energy for DL in edge computing, particularly in end devices, edge servers, and clouds. Next, some research challenges are discussed. Table VIII shows open challenges and their guidelines.

A. Power and Energy Efficiency

Current AI algorithms have high performance in solving different types of problems with remarkable accuracy. Nevertheless, such accuracy is often achieved at the cost of high memory and computation. These neural networks are executed on powerful GPUs that typically consume a significant amount of power in current applications. On the other hand, digital signal processing and embedded processors offer low-energy solutions for fixed-point operations. To deploy neural networks in edge devices, it is highly necessary to design low-complexity models and algorithms for neural networks that can be executed on embedded processors. In particular, these algorithms require support for fixed-point operations and must guarantee inference accuracy. Furthermore, there is a strong need to enhance the structures and operational efficiency of neural networks to achieve energy-efficient resource management. A promising direction is neuromorphic hardware, which mimics brain-inspired computation to significantly reduce the energy consumption per operation. While current neuromorphic platforms remain limited in programmability and ecosystems, their integration with deep learning may provide a path towards ultra-low-power edge intelligence.

Recent studies show that energy consumption in edge computing does not scale linearly with device capability or DNN model complexity. In ultra-low-power microcontrollers, even lightweight CNNs or TinyML models exhibit a steep increase in energy per inference once memory and activation reuse saturate [36], [64], [71], [73]. Mid-tier devices, such as smartphones and drones, show quasi-proportional energy growth with model size due to dynamic voltage and frequency scaling and on-chip accelerator utilization [10], [58], [60]. In contrast, edge servers with multi-core CPUs or GPUs exhibit sub-linear energy scaling, where larger models often achieve higher energy efficiency, *e.g.*, energy per operation, because of better hardware parallelization [36], [38]–[40], [49], [63], [64]. Table IX summarizes the approximate energy–complexity proportionality across different edge tiers, compiled from accelerator case studies [38], [40].

B. Interactions between DNNs and Battery Management

It is important to minimize the energy consumed by DL for edge devices powered by batteries, *e.g.*, smartphones. It is demonstrated that reducing computation can lead to

TABLE VIII
SUMMARY OF OPEN CHALLENGES AND THEIR GUIDELINES

Challenges	Causes	Guidelines
Power and energy efficiency	Cost of high memory and computation needs High power consumption	Low-energy solutions for fixed-point operations Underlying operation efficiency improvement
Interactions between DNNs and battery management	High energy consumption Limited battery capacity	Throttling of CPUs Optimization of sensor hardware Understanding interaction between hardware chips
Co-design of hardware and software	Complex relations between hardware and energy usage Conflict of performance efficiency and energy	Joint optimization of hardware and software levels Event-based spiking neural networks
Relation to network abstraction technologies	Growing amount of DL dataflow High energy of edge servers	SDN-enabled controller based on DL Combination of SDN and NFV Optimal network flow management
Benchmarks of DL in edge servers	Rapidly changing models and algorithms Lack of impartial comparison on a specific hardware	Repository construction including benchmark DL models and algorithms
Distributed inference and DNN learning	Difficult to run DNNs in IoT and edge computing	Exchange of decentralized training information Combination of models and algorithms

TABLE IX
ENERGY–PROPORTIONALITY TRENDS ACROSS EDGE TIERS

Edge Tiers	**Representative Devices**	**Energy Scaling v.s. Model Size**	**Typical Range**
Microcontrollers	ARM Cortex-M & ESP32	Super-linear ($>1.2\times$)	0.1–10 mJ/inf.
Mid-tier Devices	Smartphones & drones	Near-proportional ($\approx1.0\times$)	1–100 mJ/inf.
Edge Servers	GPU/FPGA edge nodes	Sub-linear ($<0.8\times$)	10–500 mJ/inf.

decreased energy consumption. However, it is both important and challenging to understand and investigate the relationships between DL computations and the management of battery energy, *e.g.*, the throttling of CPUs, and the optimization of sensor hardware. It is important to detect changes in input data in hardware or software, as this detection helps decrease energy consumption by optimizing the frequency of DNN executions. It is also important for hardware designers to reduce energy consumption by understanding the interactions between hardware chips, *e.g.*, TPUs and GPUs, in edge servers, as well as battery management methods. Additionally, it is crucial to develop hardware units tailored to DNN applications. Neuro-inspired computing chips, memristor-based neuromorphic computing systems, *etc.*, enable promising devices for hardware-level improvement in terms of speed and energy due to their non-volatile memory and analog behavior, thereby building energy-efficient artificial neural networks.

C. Co-Design of Hardware and Software

It is critical to investigate the complexity of underlying hardware and the energy consumption of computing devices. In practice, FPGA-based accelerators, such as Xilinx Vitis AI or Intel OpenVINO toolchains, already enable developers to deploy customized DNN models at the edge with fine-grained control over latency and power. However, FPGA solutions often involve high design complexity and limited portability, making them less accessible for non-specialist developers. Similarly, in-memory computing architectures based on ReRAM or SRAM arrays can drastically reduce data movement and achieve significant gains in energy efficiency. Yet, they suffer from device variability and limited precision, which limits their applicability in large-scale DL applications. Furthermore, neuromorphic approaches, such as IBM TrueNorth and Intel Loihi, have orders-of-magnitude lower energy consumption compared to conventional GPUs, as spiking neural networks consume energy only when spikes are fired. Neuromorphic platforms remain restricted by limited programmability, a lack of mature ecosystems, and challenges in mapping DNN models to spike-based architectures. These limitations hinder their usage in edge applications. Thus, hardware/software co-design becomes necessary where architectures need to be designed considering hardware constraints, and neural network hyperparameters or compression methods need to be co-optimized as well. Sparse DNNs and joint design methods provide near-optimal efficiency while keeping accuracy. Yet, practical deployment still requires balancing accuracy, flexibility, and hardware cost.

Neuromorphic accelerators and architecture chips are currently investigated by large companies, *e.g.*, Intel and IBM, for providing resource-efficient solutions, which are effective in enhancing computational capabilities of edge servers while adhering to energy constraints. Beyond hardware/software optimization efforts, co-design for next-generation edge AI needs to explicitly integrate neural architecture search with hardware constraints, and emerging accelerators such as FPGAs and in-memory computing, which is crucial to bridging the gap between algorithms and energy consumption.

D. Benchmarks of DL in Edge Servers

Models and algorithms for DL are changing rapidly. It is difficult for researchers and programmers to select the most suitable DNN model and deploy energy-efficient DL in edge servers because of a lack of impartial comparison on specific hardware. Current emerging DL studies usually contain a comparison of several existing models. However, these compared models are selected at the discretion of programmers and researchers, and comparison experiments may ignore energy-related factors of hardware platforms, *e.g.*, edge devices. In addition, standalone energy comparison of

DNN models on edge devices in current studies might quickly be outdated because new DNN models emerge. Therefore, a repository including benchmark DL models and algorithms is highly needed, and their energy-related comparison on different hardware is also beneficial to the research community on energy-efficient DL in different types of edge servers, including smartphones, edge servers, and home gateways. This gives a comparative understanding of the energy efficiency of DL models and algorithms on heterogeneous edge devices.

We summarize representative benchmark suites commonly used to evaluate DL on edge devices and servers, and highlight their relevance to energy-efficiency assessment. *MLPerf Tiny* focuses on resource-constrained TinyML platforms and standardizes tasks. *MLPerf Inference* provides standardized inference and scenarios across mobile and edge hardware. Energy metrics are not uniformly mandated across all platforms. *EEMBC MLMark* focuses on embedded inference with metrics including accuracy, latency, throughput. While it offers system-oriented transparency and portability, it provides limited coverage of fine-grained energy instrumentation. These suites are useful for evaluating the energy efficiency of edge platforms. However, a gap remains in unified and cross-suite energy metrics and instrumentation that cover heterogeneous workloads. While existing benchmark suites provide standardized and reproducible evaluations under controlled laboratory settings, the energy efficiency of edge DL systems is highly sensitive to dynamic real-world conditions. Factors such as ambient temperature, wireless transmission load, device aging, battery voltage, and task concurrency can significantly affect power draw and latency. To address this gap, emerging studies suggest incorporating real-scenario indicators (temperature-compensated energy metrics, adaptive workload traces, and mobility-induced communication overheads) into benchmarking [10], [58]. These indicators reflect environment- and context-aware variations that cannot be captured by static test conditions. Integrating such metrics into future benchmark suites will allow a more realistic and holistic evaluation of edge DL energy efficiency in operational environments. Table X compares their scopes, strengths, and limitations from an energy-efficiency perspective.

Community-driven benchmarks evolve towards three key directions: (i) unified energy metrics consistently applied across diverse hardware systems; (ii) open and continuously updated repositories including both emerging lightweight and large-scale DNN models; and (iii) standardized measurement methodologies allowing reproducible comparisons across platforms ranging from IoT devices to edge servers. The above-mentioned directions will enable fairer evaluation, accelerate energy-aware designs, and strengthen collaboration between academia and industries in shaping sustainable edge AI ecosystems. Several open-sourced frameworks and datasets play an essential role in enabling repeatable and verifiable research on energy-efficient edge DL. Frameworks such as TensorFlow Lite Micro, Edge Impulse, and MCUNet provide complete toolchains for quantization, deployment, and on-device energy profiling, while benchmark platforms like MLPerf Tiny and EEMBC MLMark release open reference implementations and measurement scripts. In addition, lightweight public datasets, including Visual Wake Words, Speech Commands, and Person Detection (MLPerf Tiny), support reproducible evaluation of low-power inference tasks. These open resources enhance transparency, facilitate fair comparisons, and allow researchers to replicate and extend energy-efficiency results across heterogeneous edge environments.

E. Distributed Inference and DNN Learning

Distributed computing is receiving a growing amount of attention for executing DNNs in IoT and edge computing systems. To achieve it, training information is exchanged in a decentralized manner. Each device calculates its updates of gradient with its own training data and transmits the updates to its peer devices. A consensus is reached by all devices to finally produce a global DNN model. Based on distributed computing, distributed inference and DNN learning are enabled in IoT and edge computing environments, allowing for fine-grained energy-efficient computation distribution for DNN models and algorithms from edge devices to remote clouds. This can be implemented to achieve energy consumption optimization of edge devices. Models and algorithms for distributed inference and DNN learning can be combined with the aforementioned potential directions and developments. They can achieve significant improvements in performance and efficiency of systems, leading to optimal or near-optimal results. Furthermore, distributed inference is expected to play a more critical role in enabling scalable and energy-efficient DNN execution at the edge. Future work will focus on adaptive partitioning strategies, low-latency communication protocols, and resilience mechanisms balancing accuracy, efficiency, and robustness under dynamic network conditions.

V. Conclusions

Edge computing shifts computation towards the network edge to leverage billions of connected devices such as base stations, routers, and sensors. Deep learning (DL) enables valuable insights from edge data, but its deployment is hindered by the heavy computation and energy demands of deep neural networks (DNNs). This work discusses industrial applications, mechanisms for energy-efficient DNN deployment, and emerging challenges. It is pointed out that software-level methods (*e.g.*, pruning and quantization), and hardware-level methods (*e.g.*, FPGA- and ReRAM-based accelerators) efficiently reduce energy consumption with desired performance (*e.g.*, accuracy and throughput), and adaptive offloading and scheduling also efficiently reduce latency and energy savings on edge systems. These findings in this paper suggest that co-designing hardware and software, along with dynamic resource management, offers the most promising path towards energy-efficient edge intelligence.

Furthermore, this paper points out several emerging promising research directions. First, designing lightweight yet accurate DL models and neuromorphic architectures tailored to edge devices becomes critical to balancing performance and efficiency. Second, closer integration of DL computations with adaptive battery and power management will significantly improve the longevity of energy-constrained devices.

TABLE X
REPRESENTATIVE EDGE DL BENCHMARK SUITES FOR ENERGY-EFFICIENCY EVALUATION.

Benchmarks	**Tasks**	**Metrics**	**Strength**	**Limitations**
MLPerf Tiny	TinyML devices, keyword spotting, image classification, and anomaly detection	Accuracy, latency, and optional energy reporting	Standardized tasks and fair cross-device comparison	Limited to lightweight models and varying energy tools
MLPerf Inference	Mobile/edge SoCs and servers, vision, speech, and recommendation	Accuracy, latency, throughput, and partial energy support	Broad hardware coverage and strong ecosystem	Not mandatory energy, and limited ultra-low-power characteristics
EEMBC MLMark	Embedded inference on MCU/SoC devices	Accuracy, latency, throughput and energy	Open implementation and good portability	Lack of fine-grained energy data and limited workload set
Edge AIBench	Edge–cloud collaborative AI workloads including image classification, detection, and recommendation	Throughput, energy, and thermal stability	Heterogeneous edge–cloud workloads and thermal throttling	Limited device diversity and closed-sourced workload sets
AIoTBench	IoT and sensor-edge systems, focusing on smart sensing and anomaly detection	Latency, energy, and communication delay	Integration of network load and environmental influence for realistic evaluation	Restricted task diversity and small-scale dataset coverage
Mobile AIBench	Mobile AI applications (vision, NLP, & AR) on smartphones and tablets	FPS, battery drain rate, and end-to-end energy	Inclusion of real-device traces and battery-level profiling	Hardware-dependent and limited to commercial mobile SoCs

Third, hardware-software co-design can play central roles in optimizing system efficiency. Fourth, establishing unified benchmark suites with standardized energy metrics will be essential for fair evaluation of DL techniques on heterogeneous edge platforms. Finally, distributed inference and collaborative learning strategies will enhance scalability and robustness while keeping energy efficiency. Pursuing these directions can bridge the current gap between rapidly growing demand for intelligent services and limited resources of edge devices, thereby enabling sustainable and scalable edge AI systems.

REFERENCES

[1] H. Zhao, *et al.*, “Online Workload Scheduling for Social Welfare Maximization in the Computing Continuum,” *IEEE Transactions on Services Computing*, vol. 18, no. 4, pp. 2267–2280, Jul. 2025.

[2] G. Kamath, *et al.*, “Pushing Analytics to the Edge,” *2016 IEEE Global Communications Conference*, Washington, DC, USA, 2016, pp. 1-6.

[3] X. Wang, *et al.*, “A Deep Learning Based Energy-efficient Computational Offloading Method in Internet of Vehicles,” *China Communications*, vol. 16, no. 3, pp. 81–91, Mar. 2019.

[4] Z. Ning, *et al.*, ”Deep Reinforcement Learning for Intelligent Internet of Vehicles: An Energy-Efficient Computational Offloading Scheme,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1060–1072, Dec. 2019.

[5] X. Kong, *et al.*, “Deep Reinforcement Learning-Based Energy-Efficient Edge Computing for Internet of Vehicles,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6308–6316, Sept. 2022.

[6] Q. Hussain, *et al.*, “Reinforcement Learning Based Route Optimization Model to Enhance Energy Efficiency in Internet of Vehicles,” *Scientific Reports*, Jan. 2025.

[7] F. Jiang, *et al.*, “Deep Learning Based Joint Resource Scheduling Algorithms for Hybrid MEC Networks,” *IEEE Int. of Things Journal*, doi: 10.1109/JIOT.2019.2954503, Nov. 2019.

[8] C. Lammie, *et al.*, “Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge,” *IEEE Access*, vol. 7, pp. 51171–51184, Apr. 2019.

[9] J. Lim, *et al.*, “CamThings: IoT Camera with Energy-Efficient Communication by Edge Computing based on Deep Learning,” *Proc. 28th International Telecommunication Networks and Applications Conference*, Sydney, NSW, Australia, 2018, pp. 1–6.

[10] P. Choi, *et al.*, “VisionScaling: Dynamic Deep Learning Model and Resource Scaling in Mobile Vision Applications,” *IEEE Int. of Things Journal*, vol. 11, no. 9, pp. 15523-15539, May 2024.

[11] A. Albanese, *et al.*, “Low-power Deep Learning Edge Computing Platform for Resource Constrained Lightweight Compact UAVs,” *Sustainable Computing: Informatics and Systems*, vol. 34, Apr. 2022.

[12] M. Zawish, *et al.*, “Energy-Aware AI-Driven Framework for Edge-Computing-Based IoT Applications,” *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5013–5023, Mar. 2023.

[13] X. Jin, *et al.*, “Computation Offloading and Resource Allocation for MEC in C-RAN: A Deep Reinforcement Learning Approach,” *Proc. 19th International Conference on Communication Technology*, Xi’an, China, 2019, pp. 902–907.

[14] A. Zhu, *et al.*, “Computation Offloading for Workflow in Mobile Edge Computing Based on Deep Q-Learning,” *Proc. 28th Wireless and Optical Communications Conference*, Beijing, China, 2019, pp. 1–5.

[15] J. Li, *et al.*, “Deep Reinforcement Learning Based Computation Offloading and Resource Allocation for MEC,” *Proc. IEEE Wireless Communications and Networking Conference*, Barcelona, Spain, 2018, pp. 1–6.

[16] J. Adu Ansere, *et al.*, “Energy-Efficient Optimization for Mobile Edge Computing With Quantum Machine Learning,” *IEEE Wireless Communications Letters*, vol. 13, no. 3, pp. 661–665, Mar. 2024.

[17] Y. Xiao, *et al.*, “Reinforcement Learning Based Energy-Efficient Collaborative Inference for Mobile Edge Computing,” *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 864–876, Feb. 2023.

[18] S. Kurma, *et al.*, “RIS-Empowered MEC for URLLC Systems With Digital-Twin-Driven Architecture,” *IEEE Transactions on Communications*, vol. 72, no. 4, pp. 1983–1997, Apr. 2024.

[19] L. Lv, *et al.*, “An Edge-AI Based Forecasting Approach for Improving Smart Microgrid Efficiency,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7946–7954, Nov. 2022.

[20] W. Dong, *et al.*, “Machine-Learning-Based Real-Time Economic Dispatch in Islanding Microgrids in a Cloud-Edge Computing Environment,” *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13703–13711, Sept 2021.

[21] M. S. Munir, *et al.*, “Risk-Aware Energy Scheduling for Edge Computing With Microgrid: A Multi-Agent Deep Reinforcement Learning Approach,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3476–3497, Sept. 2021.

[22] Z. Su, *et al.*, “Secure and Efficient Federated Learning for Smart Grid With Edge-Cloud Collaboration,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1333–1344, Feb. 2022.

[23] T. Zhang, *et al.*, “A Joint Deep Learning and Internet of Medical Things Driven Framework for Elderly Patients,” *IEEE Access*, vol. 8, pp. 75822–75832, Apr. 2020.

[24] R. A. Osman, “Internet of Medical Things (IoMT) Optimization for Healthcare: A Deep Learning-Based Interference Avoidance Model,” *Computer Networks*, vol. 248, Jun. 2024.

[25] M. Sornalakshmi, *et al.*, “An Energy-aware Heart Disease Prediction System Using ESMO and Optimal Deep Learning Model for Healthcare

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3640292

Monitoring in IoT," *Journal of Biomolecular Structure and Dynamics*, vol. 43, no. 7, pp. 3542–3556, Jan. 2024.

[26] V. Gowri and B. Baranidharan, "An Eergy Efficient and Secure Model Using Chaotic Levy Flight Deep Q-learning in Healthcare System," *Sustainable Computing: Informatics and Systems*, vol. 43, Sept. 2023.

[27] R. Udayakumar, *et al.*, "An Integrated Deep Learning and Edge Computing Framework for Intelligent Energy Management in IoT-Based Smart Cities," *Proc. International Conference for Technological Engineering and its Applications in Sustainable Development*, Al-Najaf, Iraq, 2023, pp. 32–38.

[28] M. Abdel-Basset, *et al.*, "Energy-Net: A Deep Learning Approach for Smart Energy Management in IoT-Based Smart Cities," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 12422–12435, Aug. 2021.

[29] K. Haseeb, *et al.*, "Intelligent and Secure Edge-enabled Computing Model for Sustainable Cities Using Green Internet of Things," *Sustainable Cities and Society*, vol. 68, May 2021.

[30] A. Aljohani, *et al.*, "Deep Learning-based Optimization of Energy Utilization in IoT-enabled Smart Cities: A Pathway to Sustainable Development," *Energy Reports*, vol. 12, pp. 2946–2957, Dec. 2024.

[31] A. K. Alkhalifa, *et al.*, "Leveraging Hybrid Deep Learning with Starfish Optimization Algorithm Based Secure Mechanism for Intelligent Edge Computing in Smart Cities Environment," *Scientific Reports*, vol. 15, Sept. 2025.

[32] M. Hartmann, *et al.*, "Distilled Deep Learning based Classification of Abnormal Heartbeat Using ECG Data through a Low Cost Edge Device," *Proc. IEEE Symposium on Computers and Communications*, Barcelona, Spain, 2019, pp. 1068–1071.

[33] M. Ling, *et al.*, "Vina-FPGA-Cluster: Multi-FPGA Based Molecular Docking Tool With High-Accuracy and Multi-Level Parallelism," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 18, no. 6, pp. 1321–1337, Dec. 2024.

[34] J. Sun, *et al.*, "An Analytical Model for Performance-Carbon Co-Optimization of Edge AI Accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, doi: 10.1109/TCAD.2025.3602746.

[35] H. Bouzidi, *et al.*, "HADAS: Hardware-Aware Dynamic Neural Architecture Search for Edge Performance Scaling," *2023 Design, Automation & Test in Europe Conference & Exhibition*, Antwerp, Belgium, 2023, pp. 1–6.

[36] M. Shafique, *et al.*, "An Overview of Next-generation Architectures for Machine Learning: Roadmap, Opportunities and Challenges in the IoT Era," *Proc. Design, Automation & Test in Europe Conference & Exhibition*, Dresden, Germany, 2018, pp. 827-832.

[37] C. Gong, *et al.*, "Mixed Precision Neural Architecture Search for Energy Efficient Deep Learning," *Proc. IEEE/ACM International Conference on Computer-Aided Design*, Westminster, CO, USA, 2019, pp. 1–7.

[38] Y. C. Xiang, *et al.*, "Analog Deep Neural Network Based on NOR Flash Computing Array for High Speed/Energy Efficiency Computation," *Proc. IEEE International Symposium on Circuits and Systems*, Sapporo, Japan, 2019, pp. 1–4.

[39] F. Yang, *et al.*, "ISARA: An Island-Style Systolic Array Reconfigurable Accelerator Based on Memristors for Deep Neural Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 33, no. 4, pp. 963–975, Apr. 2025.

[40] J. Heo, *et al.*, "SP-PIM: A Super-Pipelined Processing-In-Memory Accelerator With Local Error Prediction for Area/Energy-Efficient On-Device Learning," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 8, pp. 2671–2683, Aug. 2024.

[41] B. Yang, *et al.*, "Joint Communication and Computing Optimization for Hierarchical Machine Learning Tasks Distribution," *Proc. IEEE Symposium on Computers and Communications*, Barcelona, Spain, 2019, pp. 1–6.

[42] D. Kang, *et al.*, "Deep Learning-Based Sustainable Data Center Energy Cost Minimization With Temporal MACRO/MICRO Scale Management," *IEEE Access*, vol. 7, pp. 5477–5491, Dec. 2019.

[43] L. Xu, *et al.*, "Deep Reinforcement Learning for Dynamic Access Control with Battery Prediction for Mobile-Edge Computing in Green IoT Networks," *Proc. 11th Int. Conf. on Wireless Communications and Signal Processing*, Xi'an, China, 2019, pp. 1–6.

[44] S. Dhull, *et al.*, "Quantized Magnetic Domain Wall Synapse for Efficient Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 3, pp. 4996–5005, Mar. 2025.

[45] G. Hong, *et al.*, "Joint Content Update and Transmission Resource Allocation for Energy-Efficient Edge Caching of High Definition Map," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5902–5914, Apr. 2024.

[46] S. Zhang, *et al.*, "DRL-Based Partial Offloading for Maximizing Sum Computation Rate of Wireless Powered Mobile Edge Computing Network," *IEEE Trans. on Wireless Communications*, vol. 21, no. 12, pp. 10934–10948, Dec. 2022.

[47] L. Lei, *et al.*, "Learning-Based Resource Allocation: Efficient Content Delivery Enabled by Convolutional Neural Network," *Proc. IEEE 20th Int. Workshop on Signal Processing Advances in Wireless Comm.*, Cannes, France, 2019, pp. 1–5.

[48] Y. Dai, *et al.*, "Deep Reinforcement Learning for Edge Computing and Resource Allocation in 5G Beyond," *Proc. IEEE 19th International Conference on Communication Technology*, Xi'an, China, 2019, pp. 866–870.

[49] B. Gu, *et al.*, "A Framework for Distributed Deep Neural Network Training with Heterogeneous Computing Platforms," *Proc. IEEE 25th International Conference on Parallel and Distributed Systems*, Tianjin, China, 2019, pp. 430–437.

[50] G. Wu, *et al.*, "Joint Task Offloading and Resource Allocation in Multi-UAV Multi-Server Systems: An Attention-Based Deep Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 11964–11978, Aug. 2024.

[51] H. Li, *et al.*, "Collaborative Task Offloading and Resource Allocation in Small-Cell MEC: A Multi-Agent PPO-Based Scheme," *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 2346–2359, Mar. 2025.

[52] H. Lu, *et al.*, "Edge QoE: Computation Offloading with Deep Reinforcement Learning for Internet of Things," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2020.2981557, Mar. 2020.

[53] Q. Xu, *et al.*, "RIS-Assisted UAV-Enabled Green Communications for Industrial IoT Exploiting Deep Learning," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26595–26609, Aug. 2024.

[54] M. Fabiani, *et al.*, "Unsupervised Learning for Distributed Downlink Power Allocation in Cell-Free mMIMO Networks," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 644–658, Apr. 2025.

[55] J. Shuai, *et al.*, "Dynamic Satellite Edge Computing Offloading Algorithm Based on Distributed Deep Learning," *IEEE Int. of Things Journal*, vol. 11, no. 16, pp. 27790–27802, Aug. 2024.

[56] M. K. Hasan, *et al.*, "Federated Learning for Computational Offloading and Resource Management of Vehicular Edge Computing in 6G-V2X Network," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3827–3847, Feb. 2024.

[57] S. S. Tripathy, *et al.*, "Toward Multi-Modal Deep Learning-Assisted Task Offloading for Consumer Electronic Devices Over an IoT-Fog Architecture," *IEEE Trans. on Consumer Electronics*, vol. 70, no. 1, pp. 1656-1663, Feb. 2024.

[58] Z. Zhang, *et al.*, "DVFO: Learning-Based DVFS for Energy-Efficient Edge-Cloud Collaborative Inference," *IEEE Trans. on Mobile Computing*, vol. 23, no. 10, pp. 9042–9059, Oct. 2024.

[59] Y. Kang, *et al.*, "Neurosurgeon: Collaborative Intelligence between the Cloud and Mobile Edge," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, Apr. 2017.

[60] J. A. Lim, *et al.*, "Cutting-Edge Inference: Dynamic DNN Model Partitioning and Resource Scaling for Mobile AI," *IEEE Trans. on Services Computing*, vol. 17, no. 6, pp. 3300–3316, Nov. 2024.

[61] C. H. Hu, *et al.*, "Dynamic Scheduling For Federated Edge Learning With Streaming Data," *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops*, Rhodes Island, Greece, 2023, pp. 1–5.

[62] A. Arouj and A. Abdelmoniem, "Towards Energy-Aware Federated Learning via Collaborative Computing Approach," *Computer Communications*, vol. 221, pp. 131–141, May 2024.

[63] W. Chen, *et al.*, "A 65nm 1Mb Nonvolatile Computing-in-memory ReRAM Macro with Sub-16ns Multiply-and-accumulate for Binary DNN AI Edge Processors," *Proc. IEEE International Solid - State Circuits Conference*, San Francisco, CA, USA, 2018, pp. 494–496.

[64] M. Verhelst and B. Moons, "Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 4, pp. 55–65, Fall 2017.

[65] T. Kudo, *et al.*, "Area and Energy Optimization for Bit-Serial Log-Quantized DNN Accelerator with Shared Accumulators," *Proc. 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, Hanoi, 2018, pp. 237–243.

[66] M. Ahn, *et al.*, "AIX: A High Performance and Energy Efficient Inference Accelerator on FPGA for a DNN-based Commercial Speech Recognition," *Proc. Design, Automation & Test in Europe Conference & Exhibition*, Florence, Italy, 2019, pp. 1495–1500.

[67] J. Edstrom, *et al.*, "Memory Optimization for Energy-Efficient Differentially Private Deep Learning," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 307–316, Feb. 2020.

[68] K. Vipin, "ZyNet: Automating Deep Neural Network Implementation on Low-Cost Reconfigurable Edge Computing Platforms," *Proc. International Conference on Field-Programmable Technology*, Tianjin, China, 2019, pp. 323–326.

[69] A. Samajdar, *et al.*, "GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware," *Proc. 51st Annual IEEE/ACM International Symposium on Microarchitecture*, Fukuoka, Japan, 2018, pp. 855–866.

[70] Y. Alkendi, *et al.*, "Neuromorphic Camera Denoising Using Graph Neural Network-Driven Transformers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 4110–4124, Mar. 2024.

[71] R. David, *et al.*, "Tensorflow Lite Micro: Embedded Machine Learning for Tinyml Systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, Jun. 2021.

[72] S. Hymel, *et al.*, "Edge Impulse: An Mlops Platform for Tiny Machine Learning," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 254–268, May 2023.

[73] J. Lin, *et al.*, "Mcunet: Tiny Deep Learning on IoT Devices," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11711–11722, Jun. 2020.

Haitao Yuan (S'15–M'17–SM'21) received the Ph.D. degree in Computer Engineering from New Jersey Institute of Technology (NJIT), Newark, NJ, USA in 2020. He is currently a Deputy Director in the Department of Strategic Development, Wenchang International Aerospace City, Hainan, China. He is currently an Associate Professor at the School of Automation Science and Electrical Engineering at Beihang University, Beijing, China, and he is named in the world's top 2% of Scientists List since 2022. His research interests include the Internet of Things, edge intelligence, deep learning, data-driven optimization, and computational intelligence algorithms. He received the Chinese Government Award for Outstanding Self-Financed Students Abroad, the 2021 Hashimoto Prize from NJIT, the Best Work Award in the 17th International Conference on Networking, Sensing and Control, and the Best Student Work Award Nominee in the 2024 IEEE International Conference on Systems, Man, and Cybernetics. He is an associate editor for IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Internet of Things Journal, and Expert Systems With Applications.

Jing Bi (Senior Member, IEEE) received her B.S., and Ph.D. degrees in Computer Science from Northeastern University, Shenyang, China, in 2003 and 2011, respectively. She is currently a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. She has over 170 publications in international journals and conference proceedings. Her research interests include distributed computing, cloud & edge computing, large-scale data analytics, machine learning, industrial internet, and performance optimization. She is now an Associate Editor of IEEE Transactions on Systems, Man, and Cybernetics: Systems. She is a senior member of the IEEE.

Ziqi Wang is currently pursuing the Ph.D. degree with the School of Software Technology, Zhejiang University, China. Before that, he received the B.S. degree in Internet of Things in 2022 and the M.S. degree in Software Engineering in 2025, both from Beijing University of Technology, China. His research interests include mobile edge computing, task scheduling, intelligent optimization algorithms, and deep learning. He received the Best Paper Award in 2024 ICAIS & ISAS and the Best Application Paper Award in the 21st IEEE ICNSC.

Jia Zhang received a Ph.D. degree in computer science from the University of Illinois at Chicago. She is the Cruse C. and Marjorie F. Calahan Centennial Chair in Engineering and a professor of the Department of Computer Science in the Lyle School of Engineering at Southern Methodist University. Her research interests emphasize applying machine learning and information retrieval methods to tackle data science infrastructure problems, with a recent focus on scientific workflows, provenance mining, software discovery, knowledge graphs, and interdisciplinary applications of all these interests in earth science. She is a senior member of the IEEE.

MengChu Zhou (S'88-M'90-SM'93-F'03) received his Ph. D. degree from Rensselaer Polytechnic Institute, Troy, NY in 1990 and then joined New Jersey Institute of Technology where he is now a Distinguished Professor. His interests are Petri nets, automation, Internet of Things, cloud/edge computing, and AI. He has 1000+ publications, including 12 books, 700+ journal papers (600+ in IEEE transactions), 31 patents, and 30 book chapters. He is Fellow of IFAC, AAAS, CAA and NAI.

Rajkumar Buyya (Fellow, IEEE) is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He received a B.E and M.E in Computer Science and Engineering from Mysore and Bangalore Universities in 1992 and 1995, respectively, and a Ph.D. in Computer Science and Software Engineering from Monash University, Melbourne, Australia, in 2002. He was a Future Fellow of the Australian Research Council from 2012 to 2016. He has authored 1,294 publications and seven textbooks. He is one of the highly cited authors in computer science and software engineering worldwide, with over 166,100 citations and an h-index of 176. He was recognized as a "Web of Science Highly Cited Researcher" from 2016 to 2021 by Thomson Reuters.