World Electric Vehicle Journal

MDPI

Review

Exploring Computing Paradigms for Electric Vehicles: From Cloud to Edge Intelligence, Challenges and Future Directions

Sachin B. Chougule [1,2], Bharat S. Chaudhari [1,*], Sheetal N. Ghorpade [2] and Marco Zennaro [3]

[1] Department of Electrical and Electronics Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune 411038, India; sachin.chougule@mitwpu.edu.in
[2] Rubiscape Private Limited, Pune 411045, India; sheetal.ghorpade@rubiscape.com
[3] Science, Technology and Innovation Unit, Abdus Salam International Centre for Theoretical Physics, 34151 Trieste, Italy; mzennaro@ictp.it
* Correspondence: bsc@ieee.org or bharat.chaudhari@mitwpu.edu.in

Abstract: Electric vehicles are widely adopted globally as a sustainable mode of transportation. With the increased availability of onboard computation and communication capabilities, vehicles are moving towards automated driving and intelligent transportation systems. The adaption of technologies such as IoT, edge intelligence, 5G, and blockchain in vehicle architecture has increased possibilities towards efficient and sustainable transportation systems. In this article, we present a comprehensive study and analysis of the edge computing paradigm, explaining elements of edge AI. Furthermore, we discussed the edge intelligence approach for deploying AI algorithms and models on edge devices, which are typically resource-constrained devices located at the edge of the network. It mentions the advantages of edge intelligence and its use cases in smart electric vehicles. It also discusses challenges and opportunities and provides in-depth analysis for optimizing computation for edge intelligence. Finally, it sheds some light on the research roadmap on AI for edge and AI on edge by dividing efforts into topology, content, service segments, model adaptation, framework design, and processor acceleration, all of which stand to gain advantages from AI technologies. Investigating the incorporation of important technologies, issues, opportunities, and Roadmap in this study will be a valuable resource for the community engaged in research on edge intelligence in electric vehicles.

Keywords: electric vehicles; artificial intelligence; edge intelligence; cloud computing; edge computing; internet of things; deep neural networks; energy efficiency; autonomous vehicles

Citation: Chougule, S.B.; Chaudhari, B.S.; Ghorpade, S.N.; Zennaro, M. Exploring Computing Paradigms for Electric Vehicles: From Cloud to Edge Intelligence, Challenges and Future Directions. *World Electr. Veh. J.* **2024**, *15*, 39. https://doi.org/10.3390/wevj15020039

Academic Editors: Biao Yu, Linglong Lin and Jiajia Chen

Received: 31 December 2023
Revised: 17 January 2024
Accepted: 22 January 2024
Published: 26 January 2024

1. Introduction

Four prominent technology trends are playing a pivotal role in driving innovation within the automotive industry: autonomous driving, connected vehicles, electric vehicles, and shared mobility. Electric Vehicles (EVs) are rapidly gaining ground in intelligent transportation systems (ITS) owing to their low driving costs and minimal carbon emissions [1–3]. Artificial Intelligence (AI) stands out as a critical component in enhancing the sophistication of EVs. ITS encompasses a range of technologies, including automation, computers, controls, and communication, all geared towards improving the safety, efficiency, energy efficiency, and environmental friendliness of transportation. The rise of Autonomous Vehicles (AV) introduces challenges in the realm of intelligent decision-making, often perceived as incomprehensible to humans. Such a lack of transparency impedes the widespread acceptance of AV technology within society. In the case of self-driving cars, AI systems face the dual challenge of making real-time and secure decisions while also providing explanations for those decisions, a necessity to comply with legal requirements in various jurisdictions.

A significant portion of AI solutions relies on cloud computing for data storage and algorithmic processing. Hence, the cloud-based Internet of Things (IoT) platform is essential for autonomous vehicles, and cloud computing encounters several challenges, as highlighted by existing research [4]. The surge in interconnected devices necessitates efficient data processing and robust decision-making within strict latency constraints. Despite the efficiency and speed of our networks, transporting the massive volume of information spawned by these devices to the cloud for investigation and storage is impractical. The transfer of such vast data over cloud networks introduces overheads that diminish throughput, escalate energy consumption, increase network traffic, and incur additional costs. The heterogeneous nature of data produced by a large number of IoT sensors and devices in AVs further complicates cloud processing [5].

The cloud's complexity is exacerbated by the diverse and real-time data streams from numerous AVs, significantly increasing the workload on the cloud infrastructure. In response to these challenges, edge computing emerges as a solution to a distributed computation model deployed in nearby proximity to the data source. By deploying an Edge Intelligence (EI) model, the inference computing of AVs can experience substantial improvements in accuracy and latency. This shift toward edge computing addresses the limitations posed by centralized cloud processing and aligns with the demands of processing diverse, real-time data from interconnected devices in AVs. The increasing demands of autonomous driving have brought together machine learning, explicitly AI and Mobile Edge Computing (MEC), giving rise to edge intelligence (EI) or edge AI. This convergence aims to enhance various routine activities [6–8] significantly. The core aim of edge intelligence is to orchestrate the collaboration among numerous edge devices and servers to handle data spawned in close vicinity. Simultaneously, AI seeks to replicate intelligent human behavior in devices and machines by learning from data. The fusion of AI and edge intelligence is a logical progression due to the evident overlap between these two technologies, collectively referred to as edge intelligence.

With AV's perspective, edge intelligence plays a crucial role by enabling the AV to recognize its backgrounds precisely. This is achieved by offloading the data to additional powerful edge server situated at the base station. The substantial volume of information spawned and offloaded to the edge necessitates robust AI algorithms for precise processing, thereby giving rise to the integration of edge intelligence. Consequently, the inference processing capabilities of AVs can be significantly enhanced by installing an edge intelligence model to enhance precision and reduce latency. Nevertheless, the research on edge intelligence is still in its early stages equally in academia and industries. The exploration of this field encounters notable challenges related to transmission, computation within restricted bandwidth, data safety, confidentiality concerns, and energy utilization [9,10]. These hurdles underline the complexity and evolving nature of edge intelligence, indicating the need for further exploration and innovative solutions in the integration of AI and edge computing for autonomous systems. The research journey for this study started with the in depth understanding of six levels of edge intelligence. The analysis focused on four pivotal components essential for enhancing the efficiency of edge intelligence: edge caching, edge training, edge interpretation, and edge offloading. Subsequently, the devised techniques for optimizing these components were examined. The document provides an outline of the applications of edge intelligence in electric smart vehicles. Finally, it discusses the hurdles and prospects related to the adoption of edge intelligence in electric vehicles, enhancements in performance, and emerging directions for future research.

The rest of the paper is organized as follows: In Section 2, we have discussed edge intelligence paradigms, which describe strategies for training and inference on cloud or edge. The advantages and applications of edge intelligence are presented in Section 3. Edge intelligence presents more intriguing opportunities but encounters numerous challenges during its implementation. Section 4 elaborates on challenges and opportunities in edge intelligence. Section 5 presents in-depth analysis of artificial intelligence-based solutions for optimizing the computation of edge intelligence. Architectural layers in the Roadmap

for edge intelligence are discussed in detail in Section 6. Lastly, the paper is concluded in Section 7. The organization of the paper is illustrated in Figure 1.

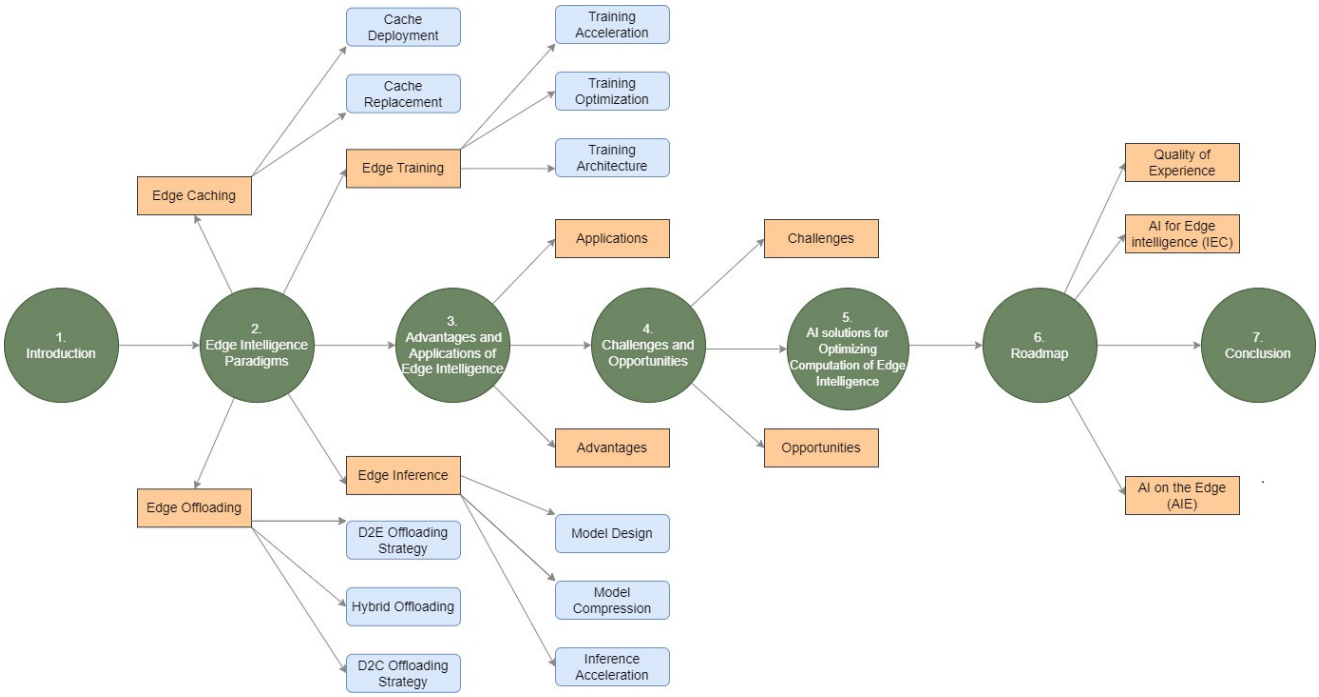

Figure 1. Organization of the paper.

2. Edge Intelligence Paradigms

Edge Intelligence (EI) is the execution of AI algorithms on edge devices using data generated on those devices and on sensor nodes [11,12]. This approach often involves high-performance AI chips but has limitations. It increases energy consumption as well as cost and is inappropriate for aged devices having restricted computation capabilities. However, it is essential to recognize that this narrow definition of EI does not fully leverage the potential of the technology. Recent studies have shown that for Deep Neural Network (DNN) models, an amalgamation of edge and cloud computing can reduce latency and energy consumption compared to local implementations [13–15]. EI should encompass a broader concept, utilizing available data and resources across various levels, as shown in Figure 2, from end nodes and edge devices to cloud data centers, for optimizing the training and inference of DNN models. These levels include:

- **L0_Cloud Intelligence:** Complete DNN model training and interpretation in the cloud.
- **L1_Cloud-Edge Cooperation and Cloud Training:** Train the deep neural network-based model (DNNM) in the cloud, then perform inference in collaboration with the edge, partly offloading new additional data to the cloud.
- **L2_In-Edge Cooperation and Cloud Training:** Train the DNNM in the cloud but perform interpretation at the edge, potentially offloading data on edge devices or else on the adjacent devices.
- **L3_On-Device Interpretation and Cloud Training:** Train the DNNM in the cloud but perform on end nodes for interpretation with partly data offloading from cloud to end nodes.
- **L4_Cloud–Edge Co-training and Interpretation:** Both training and interpretation of the DNNM model occur in cooperation between the cloud and edge.
- **L5_All In-Edge:** Both training and interpretation of the DNNM take place at the edge environment.
- **L6_All On-Device:** Both training and interpretation of the DNNM occur exclusively on the end node.

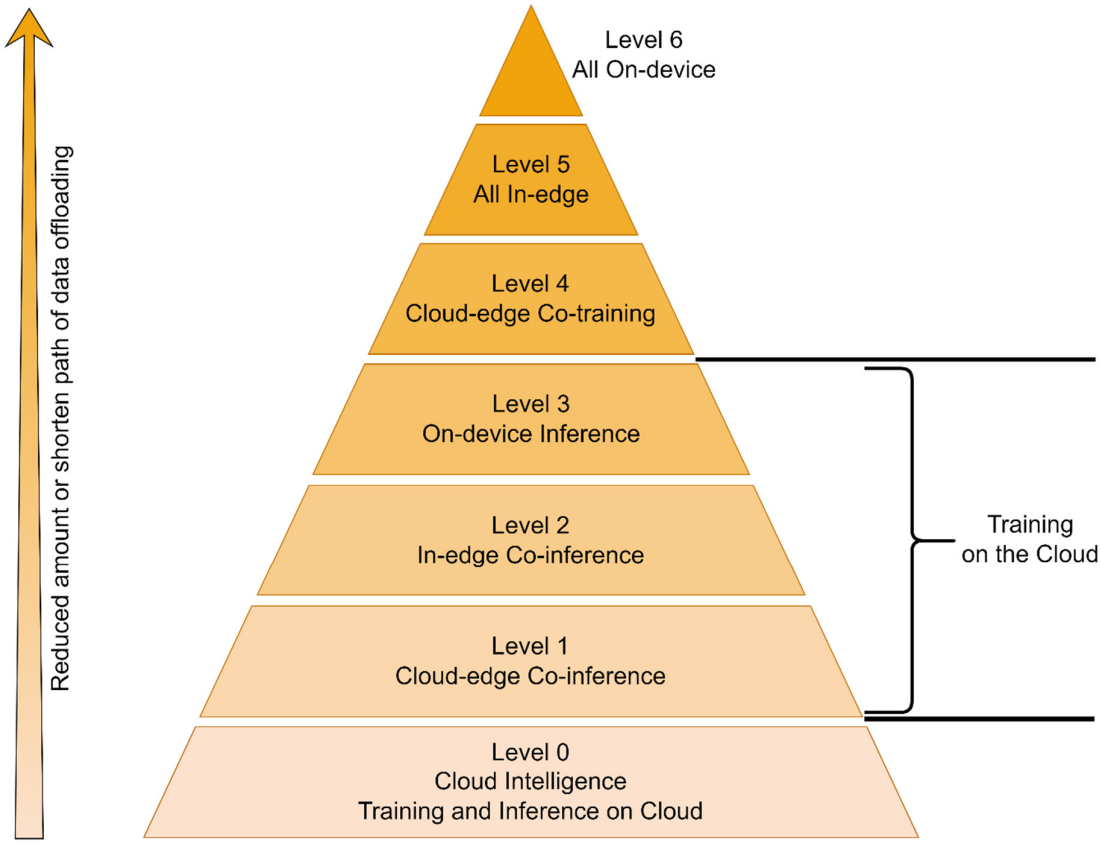

Figure 2. Six leveled ratings for edge intelligence.

The location for training and inferencing at each level is presented in Table 1.

Table 1. Training and Inferencing Location.

Level	Location	Training	Inferencing
Level 0	Coud Intelligence	Cloud	Cloud
Level 1	Cloud-edge Co-inference	Cloud	Cloud-edge
Level 2	In-Edge Co-inference	Cloud	Edge & adjacent devices
Level 3	On-device Inference	Cloud	Edge
Level 4	Cloud-edge Co-training	Cloud-edge	Cloud-edge
Level 5	All In-edge	Edge & adjacent devices	Edge & adjacent devices
Level 6	All On-device	Edge Device	Edge Device

The choice of EI level depends on various factors, including latency, energy efficiency, privacy, and WAN bandwidth cost, making it application-dependent. Four crucial elements of edge intelligence are edge caching, edge training, edge interpretation, and edge offloading, and their subclasses are shown in Figure 3. These elements are elaborated further in this section.

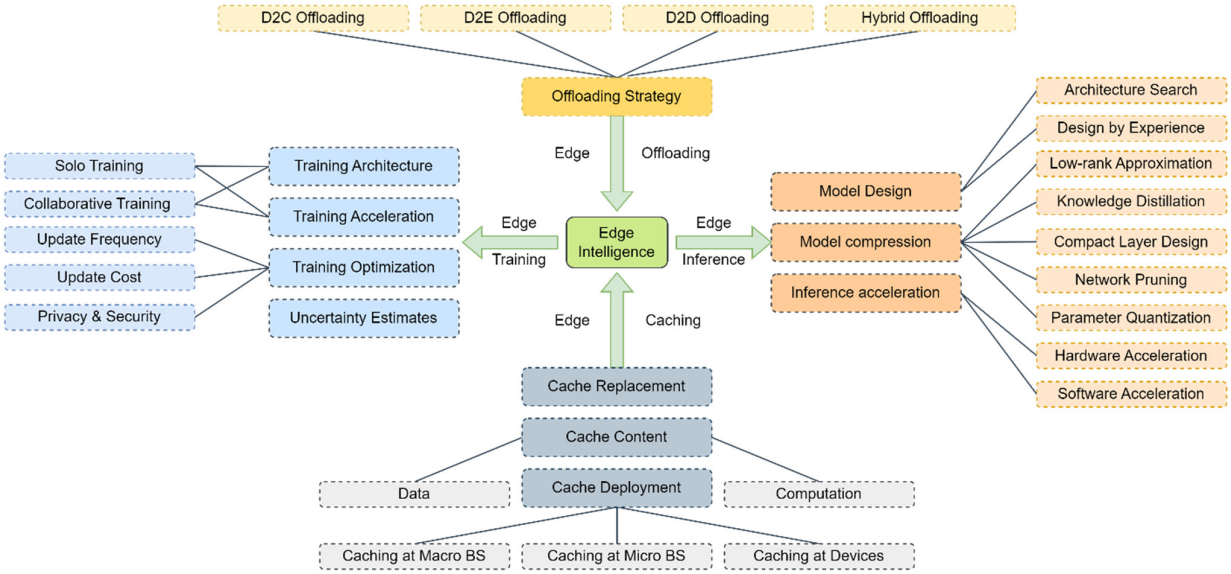

Figure 3. Elements of Edge Intelligence.

2.1. Edge Caching

Edge caching involves storing information generated by edge devices, sensors, and IoT devices closer to users to enhance performance and lower latency. This technique can reduce computational complexity and interpretation time, storing raw sensor data or previous computation results for reuse. Various caching methods have shown significant latency improvements [16–19].

2.1.1. Cache Deployment

Cache elements are deployed at edge or end entities such as macro base stations, micro base stations, and end nodes.

- *Caching at Macro Base Stations*

Broader coverage and substantial cache size are the characteristics of macro base stations. A macro base station typically covers a radius of approximately 500 m [20].

- *Caching at Micro Base Stations*

Micro base stations refer to a group of low-energy access points with a coverage span ranging from 20 to 200 m, including microcells, picocells, and femtocells [21]. By deploying small base stations or hot spots in strategic locations enhances the overall experience. This improvement is attributed to advantages like efficient spatial spectrum reuse, resulting in benefits such as higher end rates [22,23].

- *Caching at Devices*

Utilizing device-level caching takes advantage of the storage capacity within end devices, which can optimize local transmission and computing redundancy. Additionally, they have the capability to retrieve the desired contents or computing outcomes through nearby devices via device-to-device (D2D) transmission. [24,25]. However, this approach could be better for IoT as most of the end nodes in IoT are resource-constrained nodes having very low onboard memory, limited computational capability, and, most importantly battery, battery-operated.

2.1.2. Cache Replacement

In practical situations, the allocation of requests for cache access changes over time, and new content is continuously generated. Therefore, it is crucial to update caches periodically. This updating process is referred to as cache replacement. Various conventional

strategies for cache replacement have been suggested, including first-in-first-out (FIFO), least frequently used (LFU), least recently used (LRU), and its modifications. [26]. The comparison of edge caching placement is shown in Table 2.

Table 2. Cache placement locations comparison.

Cache Locations	MBSs	SBSs	Devices
Coverage Range	500 m	20~200 m	10 m
Cache Size	Larger	Intermediate	Smaller
Number of Users	Massive	Smaller	Fewer
Topology Structure	Stable	Alters Vaguely	Alters Significantly
Redundancy Capability	Higher	Medium	Lower
Computing Power	Higher	Medium	Lower

2.2. Edge Training

Edge training allows devices to learn patterns from cached edge data. It can occur on an edge server or device and includes independent training and collaborative training strategies. Collaborative training involves multiple devices and requires communication updates, posing challenges to data privacy and security. Various factors related to edge training are discussed in this section.

2.2.1. Training Architecture

The training framework relies on the computational capabilities of both edge devices and servers. If a singular edge device or server possesses adequate power, it can employ the identical training structure as a centralized server, performing the training on a single device. Conversely, when the device or server lacks such capabilities, cooperation with other devices becomes imperative. This results in the emergence of two types of training frameworks: individual training, which entails executing training tasks on a lone edge device or server, and cooperative training, where limited devices and servers work together to execute training tasks.

A prevalent example of a cooperative training framework is the master-slave model, as illustrated by federated learning [27]. In federated learning, a server involves numerous devices and delegates training tasks individually. Another form of cooperative training architecture is peer-to-peer, where participants are regarded as equals in the training process.

2.2.2. Training Acceleration

The emphasis is on expediting training at the edge, with some initiatives [28,29] exploring transfer learning to enhance training speed. Transfer learning involves utilizing features learned from previous models, resulting in a significant reduction in learning time. In a cooperative training approach, edge devices have the capability to acquire from one another, thereby enhancing overall knowledge proficiency. A framework known as Recycle ML employs cross-modal transmission to expedite the training of neural networks along mobile platforms throughout the diverse sensory system. Federated learning can also be employed to hasten model training on distributed edge devices, particularly in scenarios where labeled data is insufficient [30].

2.2.3. Training Optimization

Training optimization involves streamlining the training procedure to attain specific goals, viz., minimizing energy consumption, enhancing precision, preserving secrecy, maintaining security, and more. The critical factors involved in optimization are communication frequency, communication cost, privacy, and security issues.

- *Communication Frequency*

Communication Frequency is a crucial aspect of federated learning, where the exchange of information among edge devices and the cloud server plays a vital role. This operation involves update uploading through edge devices to the cloud server and downloading the combined updates from the distributed model to local models. Given the potential for erratic network conditions in edge devices, it is essential to minimize update cycles.

- *Communication Cost*

Besides the frequency of communication, another factor influencing the efficiency of communication among edge devices and the central server is the cost of communication. Minimizing communication costs has the potential to save bandwidth substantially and enhance overall communication efficiency.

- *Privacy and Security Issues*

Upon catching updates from edge devices, the central server is tasked with aggregating these updates to construct unified updates for the distributed universal model. The concern arises that malicious hackers may scrutinize the updates, posing a threat to the privacy of participating edge users. To address this, an aggregation process is implemented to combine updates from all edge devices, rendering individual updates indiscernible by the central server [31]. More precisely, every edge device transmits encrypted updates to the server. The server later combines these encrypted updates. The counteraction of masks occurs when enough edge devices are involved. Consequently, the server gains the ability to unveil the aggregated update by unmasking it. Throughout this aggregation process, exclusive updates remain unscrutinized, and the server can solely approach the combined unmasked updates, thereby efficiently safeguarding the secrecy of participants.

The conventional deep learning approaches for classification and regression lack the ability to account for model uncertainty.

2.3. Edge Interpretation

Edge interpretation occurs during the use of the trained model for computing output on edge devices and servers. However, many deep learning algorithms are designed for high-performance hardware and are not suitable for edge environments. Challenges include designing models for edge deployment and accelerating edge inference for real-time responses. These issues can be addressed through new model designs or model compression techniques.

2.3.1. Model Design

The primary emphasis in model design revolves around creating neural network architectures that are lightweight and appropriate for execution on edge devices with lower hardware demands. This process involves either automated generation of the optimal architecture by machines or manual design by humans.

- *Architecture Search*

Exploration in architecture search is a thriving research field with broad applications in the future. A recent notable advancement in this area is differentiable architecture search (DARTS) [32], which offers the potential to substantially decrease reliance on hardware. DARTS relies on the constant easing of architecture description and employs gradient descent for the process of hunting for architecture.

- *Design by Experience*

The experience-driven design employs two distinct strategies. The initial approach involves the use of comprehensive detachable convolutions, which are utilized to construct a streamlined DNN known as MobileNets. This design specifically caters to the needs of mobile and embedded devices [33]. Another method employed is group convolution,

which serves as an alternative means to diminish computation costs during the process of designing architecture.

2.3.2. Model Compression

Model compression seeks to reduce the dimensions of a model, enhance energy efficiency, and accelerate inference on edge devices with a restricted number of resources, all without compromising precision. The five important methods to model compression are lower Rank approximation, knowledge distillation, compact layer design, parameter quantization, and network pruning [34].

- *Lower Rank Approximation*

The fundamental concept behind low-rank approximation involves the replacement of high-dimensional kernels by the low-rank convolutional kernel multiplication.

- *Knowledge Refinement*

It relies on transfer learning, wherein a smaller NN is trained employing distilled data by initiating a large model. The large, intricate model is known as the mentor model, whereas the more compacted model is described as the student model. The student model gains advantages by assimilating knowledge from the teacher network.

- *Designing Condensed Layers*

In DNN, when weight values approach zero, computational resources are inefficiently utilized. A key strategy to address this issue involves creating a condensed layer in NN, excellently minimizing resource utilization such as memory and computation power. Christian and colleagues suggest addressing this by incorporating sparseness and substituting the entirely linked layers in GoogLeNet. In Residual-Net, an alternative approach is taken by replacing entirely linked layers through global regular merging to decrease resource demands. Substituting a large convolutional layer with several smaller and more compact layers can efficiently decrease the parameter count and subsequently lower computational requirements.

- *Network Pruning*

The fundamental concept behind network pruning involves the removal of less significant parameters, recognizing that not all parameters play a crucial role in extremely accurate DNN. As a result, associations between lower weights are eliminated, transforming a heavy network into a sparser one.

- *Parameter Quantization*

Achieving high performance in neural networks does not always require highly precise parameters, particularly when those parameters are unnecessary. Research has demonstrated that a relatively smaller quantity of parameters suffice for reconstructing a complete network.

2.3.3. Interpretation Acceleration

The primary concept behind accelerating models in interpretation is to decrease the runtime of interpretation on edge devices and achieve instantaneous replies for precise applications based on NN, all deprived of modifying the architecture of the trained model. There are two main categories of acceleration: hardware and software acceleration. The software acceleration approach is centered on enhancing resource management, pipeline structure, and compiler optimization.

- *Hardware Acceleration*

Methods for hardware acceleration concentrate on parallelizing inference tasks across accessible hardware, including CPU, GPU, and DSP. In recent times, the potency of mobile devices has seen a notable rise. A growing number of mobile platforms now feature GPUs. Given that mobile CPUs are less apt for deep neural network computations, leveraging

embedded GPUs becomes a viable strategy to distribute computing tasks and expedite the inference process.

- *Software Acceleration*

Software acceleration primarily centers on enhancing resource allocation, refining pipeline design, and optimizing compilers. Methods for software acceleration aim to optimize the utilization of limited resources to achieve faster performance, but this can sometimes result in a reduction in accuracy in specific scenarios.

2.4. Edge Offloading

It is a distributed computation paradigm that furnishes computation services for edge caching, training, and interpretation. It allows tasks to be processed in cloud servers when edge hardware lacks capability. Four offloading strategies exist, including device-to-cloud (D2C), device-to-edge server (D2E), device-to-device (D2D), and hybrid offloading, each with its adaptiveness and resource utilization.

- *D2C Offloading Strategy*

In the D2C offloading strategy, devices transfer input data, such as audio or images, to a cloud server. Powerful computers perform high-accuracy inference using a large neural model, and the outputs are sent backward via the identical network. However, it has some primary drawbacks. Mobile devices need to communicate large volumes of information to the cloud, creating a bottleneck in the overall process [35]. The execution is reliant on internet connectivity. The transmitted information from mobile devices might inhibit users' personal information, such as personal photos, making it susceptible to attacks by mischievous hacks while the interpretation on the cloud server [36]. Various considerations, including energy efficiency, latency, and privacy, can guide the design of model partitioning and layer scheduling in this context.

- *D2E Offloading Strategy*

In contrast to D2C offloading, which involves transferring inferencing to a central server in the cloud, D2E offloading shifts inferencing to an Edge server. An Edge server, in this context, denotes robust servers that are physically close to mobile devices and possess greater processing power than typical edge devices.

- *D2D Offloading Strategy*

In the strategy of Device-to-Device (D2D) offloading, devices like smartwatches are connected to smartphones or home gateways and have the capability to delegate model interpretation tasks to more influential connected devices. Binary decision-based offloading and partial offloading exist in this context. Binary decision offloading involves deciding upon the execution of the task locally or offloading it. On the other hand, partial offloading entails breaking down the interpretation task into various subtasks and offloading a handful of them to connected devices.

- *Hybrid Offloading*

The hybrid computing framework efficiently leverages cloud services, edge computing, and mobile devices in a comprehensive approach. Distributed Deep Neural Networks (DDNNs) derived from this holistic computing architecture represent a hybrid offloading technique that strategically allocates portions of a DNN across a distributed computing hierarchy [36]. The collective training of these segments occurs in the cloud and aims to reduce communication and resource usage on edge devices. During the interpretation phase, individual edge devices perform local computations, and the resultant outputs are combined to generate the final results.

3. Edge Intelligence: Advantages and Applications

In recent times, there has been a noticeable trend towards enhanced intelligence in various aspects of life, ranging from smartwatches to automobiles, agriculture to industrial

processes, and even urban environments, and additionally benefiting from the general benefits of edge intelligence, like reduced latency and bandwidth consumption. Edge intelligence can help enterprises make quicker data-driven decisions responding to the requirements of their clients. Reduced storage requirements in edge intelligence led to improved operational cost savings for enterprises.

3.1. Advantages

- *Enriching AI with Richer Data and Application Scenarios:*

Recent advancements in deep learning have been driven by four factors: algorithms, hardware, data, and application scenarios. Data plays a pivotal role in enhancing AI performance. As IoT grows, vast quantities of information will be spawned at the edge, challenging cloud-based processing due to bandwidth constraints. Edge intelligence addresses this challenge by enabling low-latency data processing closer to the data source, potentially boosting AI performance. Edge intelligence and AI counterparts one another technically and also with regard to application and adoption [37].

- *Key Infrastructure for AI Democratization:*

AI has made significant strides in digital products and services, from online shopping to self-driving cars. Major IT companies envision democratizing AI, making it accessible to everyone and everywhere. Edge intelligence is well-suited to this goal and offers diverse application scenarios. Thus, Edge intelligence serves as a crucial enabler for ubiquitous AI [38].

- *Popularizing Edge Intelligence with AI Applications:*

Edge intelligence is already bringing about significant transformations across various industries, such as manufacturing, energy, healthcare, agriculture, logistics, and transportation. [39–41]. Real-time video analytics, built on computer vision, emerges as a killer application for Edge intelligence due to its high computational demands, bandwidth requirements, privacy concerns, and low-latency needs [42]. Multiple benefits of Edge intelligence have created a path for expanded progression in the near future [43].

3.2. Applications of Edge Intelligence for Electric Vehicles

Gartner anticipates a significant surge in the adoption of edge intelligence use cases in the coming years. The projection is that by 2024, over fifty percent of the potential enterprises will have implemented a minimum of six edge intelligence use cases. This marks a remarkable expansion in comparison with the scenario in 2019, in which merely one percent of larger enterprises had very few edge intelligence deployments. Presently, some key applications of edge computing include:

3.2.1. Smart Vehicles

An intelligent vehicle is defined as a vehicle with computing capabilities, storage, and communication facilities that enables learning from its environment and making conclusions consequently. Sensors and multi-interface cards are used for equipping vehicles inside and outside. The increasing prevalence of smart vehicles endowed with onboard wireless devices and sensors like radar and lidar has led to a focus on efficient management and transportation applications. The goal is to improve traffic flow by reducing travel time and preventing jamming.

Smart vehicles possess a range of novel features, including information exchange and location info. These functionalities assist specialized applications, such as security communication and warnings. Vehicles inside a Vehicular Edge Computing (VEC) system typically have onboard wireless devices, particularly Onboard Units (OBUs). During disaster alarm schemes, sensors play a crucial role in verifying if airbags were deployed during an accident.

3.2.2. Smart Vehicle Services

Smart vehicles offer a diverse range of services. Some key services, such as assisted driving, autonomous vehicles, platooning, and parking solutions, are discussed below.

- *Assistant Driving*

In contemporary times, vehicles such as cars, buses, and trains are designed with the capability to convey valuable information, including details about accidents, road closures, and traffic congestion. This is achieved through the integration of sensors, actuators, and processors, enhancing safety and navigation for these vehicles. The data on traffic patterns, made available by these intelligent features, can prove advantageous for all types of organizations [44]. Intelligent vehicles are categorized into five layers by the National Highway Traffic Safety Administration [45].

- *Autonomous Vehicles*

As smart vehicles progress towards autonomous driving, establishing robust connectivity amongst smart vehicles becomes imperative. Vehicular networks, on the rise due to this evolution, play a pivotal role in shaping intelligent transportation systems and smart cities. These systems are anticipated to support a spectrum of advanced applications, ranging from road safety and enhanced traffic efficiency to automated driving and seamless admittance to Internet facilities [46,47].

The global acceptance of automated vehicles has sparked a transformation in the automobile sector. Nevertheless, challenges such as invulnerability, fidelity, and secrecy persist in realizing completely automated vehicle editions. Notably, the susceptibility of automated vehicles to security threats is a concern; a single attack on the software of an Autonomous Vehicle (AV) could lead to multiple mishaps. Additionally, interconnected systems on the Internet face risks of unauthorized access, presenting unknown threats. Vehicle design addresses safety-critical issues by enabling the vehicle to anticipate and respond to potential dangers while continuously monitoring road conditions throughout the journey. The assumption in the design is that the driver provides the destination or navigation. However, it may not be in regulation throughout the excursion, emphasizing the role of automated vehicular systems in ensuring safe operations [48]. While automated vehicular systems differ from connected vehicular technology, they share some similarities.

- *Platoon*

A platoon refers to a cluster of smart vehicles equipped with driving assistant schemes in which one vehicle follows another. The formation of a platoon involves several vehicles driven by technology, interconnected through shared communication. This collaborative driving concept, known as platooning, has become feasible because of the advancement of technologies. These technologies, fortified with sensors and actuators, enable modern vehicles to engage in cooperative platooning.

Cooperative platooning offers significant advantages, particularly in improving the fuel efficacy of heavy vehicles. By anticipating speed changes, the vehicles within a platoon can maintain a steady speed, leading to enhanced fuel efficiency. Since carbon dioxide emissions are directly linked to fuel consumption, cooperative platooning has the indirect effect of reducing environmental pollution. Additionally, this form of platooning contributes to the improvement of road safety. In emergencies, messages are transmitted to all vehicles in a platoon, triggering appropriate actions by the automated system [49].

- *Smart Parking*

In metropolitan regions, the number of vehicles parked in parking lots is substantial, distributed across various locations such as street parking and outer parking. Unlike moving vehicles, parked vehicles remain stationary for extended periods. Although they do not transport info from location to location, parked vehicles equipped with wireless communication devices and rechargeable batteries as part of Smart Street Vehicles (SSVs) serve as communication infrastructures with unique characteristics. This allows parked

SSVs to transmit information among themselves and also link to neighboring moving SSVs, functioning as static backbones to enhance communication amongst vehicles. The number of parked vehicles in a parking slot and their duration of stay are critical factors influencing their role as communication infrastructures [50]. Collaboration among parked SSVs, particularly in parking lots, enables the execution of heavy computation tasks under favorable communication conditions. Individual vehicles, constrained by limited resources, may struggle to meet substantial computation demands. Parked SSVs address this challenge by providing powerful and underutilized computation resources, efficiently accomplishing allocated tasks in less time. This environment can be likened to tiny data centers capable of handling intricate tasks that require significant computational capability.

3.2.3. Smart Vehicle Applications

The emergence of Vehicular Edge Computing (VEC) and the utilization of smart vehicles as infrastructures have paved the way for a multitude of associated vehicular applications. These applications span various domains, including driving safety, Augmented Reality (AR), infotainment services, and video streaming. Particularly in scenarios wherever higher computational processing is essential, VEC networks plays a crucial role in accelerating computing processes, in this manner curtailing delays. For instance, in the event of an accident, quick computations are needed to formulate solutions such as rescheduling traffic lights and efficiently dissipating a large traffic backlog. Meeting such demands places an exceptional requirement on computational resources [51]. In this context, applications are categorized into two groups: safety and non-safety. VEC proves to be supportive of both types of applications, as discussed below:

- *Safety Applications*

It emphasizes enhancing security by minimizing the likelihood of accidents. These applications monitor the driving environment and alert drivers to potentially harmful situations to prevent accidents. One such application involves the use of a Global Camera Sensor mounted at a traffic monitoring signal, capable of detecting movement in its region by recognizing number plates within the detection field. This sensor records the location and vehicle number, sending this information to the local edge server. A smart Local Camera Sensor (LCS) positioned at the front of the vehicle observes the driver's activities. The LCS issues warning messages to the driver for such activities, aiding in accident prevention. Repetitive warnings at appropriate times help drivers avoid hazardous situations and ensure their safety. The LCS is equipped to generate these warning messages and, later broadcasting a specific number, informs the edge server about any interrupting activities involving the vehicle. This report involves activity evidence and vehicle identification [52–56].

Context-aware systems are also employed, utilizing information related to the user to adapt operations based on environmental conditions. Context-aware applications adjust their operations according to the user's context, sensing information specific to the environment. These applications involve Context Acquisition, Processing, and Acting [57]. Leveraging contextual knowledge allows the generation of concise, context-aware information, reducing the radio resource requirements for transmission. Users can extract coveted content from the context using suitable decoders and big-data analytics techniques such as Natural Language Processing (NLP) [58,59].

- *Non-safety Applications*

Applications of Vehicular Edge Computing (VEC) extend beyond safety services to include the development of non-safety applications, such as multimedia services like video streaming, Augmented Reality (AR), and infotainment. The surge in streaming applications has notably contributed to a significant portion of network traffic, particularly in IoT communication, where video streaming plays a pivotal role [60]. This is particularly evident in smartphone applications like video crowdsourcing [61]. The Internet of Vehicles (IoV) supports various applications, including intelligent transportation systems and mobile

multimedia. In IoV, users connect their mobile devices to the internet to access multimedia content from remote servers. However, maintaining Quality of Service (QoS) becomes challenging, given factors like jitter, buffering, throughput, and transmission delays in video streaming applications, exacerbated by the high mobility of vehicles in IoV. A proposed solution in [62] introduces distributed reliable real-time streaming in vehicular cloud-fog networks. A utility function is utilized to improve QoS and fairness in resource reservation among mobile devices, considering content provision for streaming and the number of tokens for content reservation from service providers, edge, and cloud. Mobile devices in the network query their probable location, the amount of data for streaming, and required tokens for content provisioning, facilitating effective reservation of streaming content from computing service providers and enhancing streaming utility reliability.

Addressing the parking lot monitoring issue, [63] proposes an edge computing-based scheme where each vehicle uploads street contents collected by the camera for video analytics. This enables ParkMaster to estimate precise locations and track parked vehicles using information from the vehicle's camera, GPS, and inertial sensors.

Augmented Reality (AR) is an evolving multimedia application that seamlessly integrates real scenes into virtual scenes, overlaying virtual content onto the real environment to enhance traditional image information. AR can improve traffic awareness for vehicles or pedestrians near drivers, with the head-up display (HUD) reducing distractions and enhancing driving safety. An exploration of the HUD-based navigation system with AR-based content is detailed in [64], illustrating its potential for safety and convenience services [65]. A novel application, walk navigation, utilizes a camera and GPS for a car navigation system with AR technology, providing real-time navigation without compromising safety. The device's camera output is analyzed by an edge computing application to overlay viewed objects with AR content. Given the intricate storage and processing demands of AR, VEC is considered the optimal solution to meet the specific requirements of AR applications in a vehicular network, including mobility, location awareness, and low latency.

Although edge intelligence presents more intriguing opportunities compared to cloud computing, organizations encounter numerous challenges during its implementation. Opportunities and challenges associated with edge intelligence are discussed in the next section.

4. Challenges and Opportunities for Edge Intelligence

Edge intelligence is yet in its early stages, and currently, there is yet to be an established framework to strengthen it. Such frameworks must encounter specific requisites, including the ability to develop applications for instantaneous processing on edge nodes. While existing cloud computing frameworks can handle data exhaustive purposes, enabling instantaneous data treatment at the network edge remains an area of ongoing research [66–69]. Moreover, we must gain a deep understanding of installing application capabilities on edge nodes, including strategies for workload placement, policies for connecting to edge nodes, and management of distinct node types when deploying applications at the edge.

4.1. Challenges

To create such a framework, five key research challenges spanning the hardware, middleware, and software layers are identified.

4.1.1. Enabling Generic Computing on Edge Nodes

In principle, the concept of Edge intelligence involves utilizing various nodes linking the edge device and the cloud. For illustration, base stations are equipped with specialized Digital Signal Processors (DSPs) designed to manage specific tasks. Nevertheless, in practical terms, base stations might not be ideal for handling critical assignments due to the fact that DSPs are not devised for versatile computation tasks. Furthermore, the situation remains uncertain since these nodes may not be able to execute additional computations along with the primary functions.

Research is underway to enhance the computing capabilities of edge nodes to assist generic tasks. For instance, it is possible to upgrade a wireless home router to handle added tasks [70]. Intel's Smart Cell Platform17 utilizes virtualization to accommodate supplementary tasks. An alternative solution involves exchanging specific DSPs with equivalent generic CPUs, although it demands substantial investment.

Cutting-edge AI techniques like neural networks have shown great potential in solving various challenges using remarkable precision. Nevertheless, this often arises at the expense of higher computation and memory demands. As a result, NN typically executes these algorithms on high-powered GPUs, which consume significant power. In contrast, embedded processors and DSPs propose a more power-efficient remedy and are capable of fixed-point processes [71]. To make neural networks functional for deployment on mobile devices, we need less complex CNN models that can execute on embedded processors without sacrificing precision. Additionally, it is essential to enhance both the efficacy of the inherent processes executed by neural networks and their overall structure to make them more suitable for resource-competent procedures.

4.1.2. Exploring Edge Node Discovery

The exploration of resources and services within a distributed computing environment is an established field. It is accomplished in both tightly and loosely connected setups through various techniques integrated into monitoring tools [72,73] and service brokerages [74,75]. These methods, like benchmarking, create the foundation for yielding decisions about allocating tasks to highly suitable resources to enhance performance.

Nevertheless, the challenge arises when we aim to control the capability of the network's edge. In a decentralized cloud configuration, discovering appropriate nodes necessitates mechanisms that go beyond manual intervention due to the sheer number of available devices at this level. Additionally, such mechanisms should accommodate diverse devices from diverse generations and adapt to prevailing workloads like newly added exhaustive machine learning tasks. Benchmarking methods must rapidly communicate the attainability and capabilities of resources. It is also desirable for them to handle node failures consistently and independent recovery.

In this context, the conventional methods used in the cloud for discovering edge nodes, such as resource management and task scheduling, face limitations:

- *Resource Management*

It entails guaranteeing an ample supply of resources within the edge network, as exemplified in smart parking setups in which sensor data is seamlessly and dependably transmitted to edge devices [76]. Essential components of resource management include dynamic load balancing [77] and the creation of platforms for resource allocation [78]. Nevertheless, addressing on-demand resource requirements, fluctuating workloads, and data streams originating from diverse devices across extensive geographical areas may require making trade-offs between computing power and communication speed [79].

- *Resource Management and Task Scheduling*

Edge devices exhibit numerous models, diverse hardware architectures, various operating systems, and inconsistent creation environments. Established edge intelligence platforms struggle to effectively incorporate and administer such diverse edge devices, particularly when it comes to supporting AI workloads. Managing and orchestrating AI workloads, which have distinct characteristics compared to web loads, is a pressing challenge. Consequently, Edge intelligence platforms must introduce novel resource perceptions tailored to the challenges of AI workloads, including GPU support, capability extension, and task dependence handling.

- *Customized AI Algorithms*

While model compression preserves to foster AI execution on the edge, usually leads to a deficit of model precision. Static model compression methods fail to amend the

dynamical hardware configurations and loads of edge nodes. Thus, there is a growing need for dynamic compression methods tailored to the complex conditions of edge nodes. Additionally, current model-splitting techniques utilize the hierarchical constitution of deep learning models. Future research should focus on developing partitioning methods tailored to the specific characteristics of AI applications.

Furthermore, data availability presents another formidable challenge for edge devices attempting to process raw data for edge training. The usability of data is crucial, and raw data captured from edge devices often cannot be directly used for model training and inference due to potential bias. While federated learning offers a partial solution, the synchronization of training procedures across devices and communication remains a challenging aspect. In conclusion, discovering and effectively utilizing edge nodes in a decentralized cloud setup poses unique challenges that necessitate innovative approaches beyond traditional cloud-based methods.

4.1.3. Dividing and Delegating Discovery

The advancement of distributed computation environments leads to the creation of various methods for dividing tasks to be implemented in numerous geographical localities [80]. One instance is the partitioning of workflows to be executed in different places [81]. Task splitting is typically conveyed obviously using a semantic or administration tool. However, using edge nodes for offloading computations presents the challenge of efficiently dividing computational assignments inevitably, deprived of essentially compelling appropriate definitions of the competencies or locations of edge nodes.

With the increasing global demand for mobile applications, mobile devices face growing constraints such as limited resources and reduced battery life. Mobile fog architectures have been discussed in the context of mobile cloud computing and code offloading mechanisms. Existing investigations have predominantly depended on simulation techniques for examining task offloading. However, this methodology has limitations because it cannot accurately depict the authentic characteristics of AI workloads in industrial settings. AI algorithms utilized in diverse industrial sectors exhibit distinct model structures and processing steps.

Consequently, when devising a task offloading algorithm, it is crucial to customize the offloading strategy to align with the processing steps of the AI application and its model structure. Current research is transitioning towards a synergy between cloud and Edge intelligence. In the future, the emphasis may pivot towards cooperative computing among edge nodes, where multiple edge nodes can collect information from various perspectives, contributing to heightened analysis and decision-making capabilities.

4.1.4. Unwavering Quality of Service and User Experience

The value delivered by edge nodes can be measured using Quality of Service (QoS), while Quality of Experience (QoE) assesses the quality experienced by users. In the realm of Edge intelligence, a critical principle to embrace is the avoidance of overburdening nodes with computationally demanding tasks [82,83]. The difficulties lie in ensuring that these nodes maintain high throughput and reliability while accommodating surplus workloads from data centers or other edge devices. Even though an edge node is fully utilized, users of edge devices and data centers rightfully expect a baseline stage of service. For instance, overloading a base station can negatively impact the service given to connected edge devices. It is imperative to have a comprehensive understanding of peak usage hours for edge nodes so that tasks can be effectively divided and organized in a versatile manner. While an administration framework could be beneficial, it also advances concerns associated with supervising, scheduling, and rescheduling at all levels.

Collaborative training is also an important task to be considered. Edge intelligence employs two AI training methods; distributed training and federated learning. In distributed machine learning, data analysis tasks are performed on nodes that create data, with models and data exchanged among different nodes [84,85]. Google has introduced

federated learning as a privacy-preserving technique, which has found applications in sensitive areas like healthcare and finance. Federated learning and distributed training are distinct. Generally, distributed training emphasizes utilizing data at the network's edge, whereas federated learning places a greater emphasis on safeguarding data privacy. When dealing with diverse edge devices that vary in computing power and communication protocols, adapting models and ensuring serviceability poses challenges. The same methods may yield different learning outcomes when applied to different device clusters. Establishing robust, flexible, and secure synchronization between edge devices, servers, and cloud resources at both hardware and software levels is of utmost importance. There are significant research opportunities in developing a standardized API/IO interface for edge learning across various ubiquitous edge devices.

4.1.5. Utilizing Edge Nodes Safely and Publicly

Hardware assets held by data centers, supercomputing facilities, and private entities using virtualization have the potential to be repurposed to provide computing services as a utility. This approach involves assessing the associated risks for both providers and users, ultimately enabling pay-as-you-go computing. Consequently, a competitive market has emerged, offering a multitude of options to cater to computing consumers while adhering to Service Level Agreements (SLAs) [86].

Nevertheless, when contemplating the use of alternative devices like switches, routers, and base stations as publicly accessible edge nodes, several challenges must be confronted. Firstly, there's a necessity to clearly delineate and communicate the associated risks for both public and private organizations owning these devices and those planning their deployment. Secondly, it is imperative to ensure that the primary function of the device, such as a router managing internet traffic, remains unaltered when repurposed as an Edge intelligence node. Thirdly, realizing multi-tenancy on edge nodes demands technology that prioritizes security; for instance, containers, which are potentially lightweight technology for edge nodes, must exhibit more robust security features [87]. Fourthly, a baseline level of service must be assured for users of the edge node. Lastly, diverse factors like workloads, computation, data location and transfer, maintenance costs, and energy expenses need consideration when formulating appropriate pricing models for facilitating access to edge nodes.

The significance of data privacy and security cannot be overstated. Artificial Intelligence (AI) serves as an effective tool in identifying malicious attacks and preventing privacy breaches. However, edge devices face constraints in computing resources, presenting a substantial challenge in designing lightweight and efficient AI algorithms suitable for Edge intelligence (EC).

4.2. Opportunities

Despite the difficulties that arise in the implementation of Edge intelligence, several promising opportunities exist. We have identified five such opportunities.

4.2.1. Establishing Standards, Benchmarks, and a Marketplace

The practical realization and public accessibility of Edge intelligence hinge on the clear articulation of duties, associations, and consequences among all involved parties. Various efforts have been made to define cloud standards, including those by organizations such as the National Institutes of Standards and Technology (NIST) in 2021, the IEEE Standards Association, the International Standards Organization (ISO), the Cloud Standards Customer Council (CSCC), and the International Telecommunication Union (ITU). However, these standards must now be revisited to account for added stakeholders, such as public and private organizations that acknowledge edge nodes, to address the communal, legitimate, and moral aspects of edge node utilization. This is undeniably a complex task that demands devotion and investment from both public and private organizations as well as academic institutions.

The implementation of standards relies on the ability to benchmark the performance of edge nodes beside established metrics. Benchmarking efforts for the cloud have been undertaken by organizations like the Standard Performance Evaluation Corporation (SPEC) and several academic scientists. In an environment as unpredictable as the cloud, benchmarking presents substantial difficulties. Benchmarking edge nodes will pose even greater challenges but will open up additional opportunities for research.

Utilizing edge nodes becomes an enticing prospect when duties, associations, and consequences are well-defined. Much like a cloud marketplace, the creation of an Edge intelligence marketplace offering a heterogeneity of edge nodes on a pay-as-you-go basis is reasonable. Research is needed to establish Service Level Agreements (SLAs) for edge nodes and develop pricing models to facilitate the creation of such a marketplace.

4.2.2. Frameworks and Languages

There are numerous possibilities for running applications within the cloud paradigm. Besides widely used programming languages, there's a diverse range of offerings available for deploying cloud-based applications. In scenarios where resources beyond the cloud are utilized, such as running a bioinformatics workload on the public cloud with data sourced from a private database, a typical approach involves the use of workflows. Research has extensively explored software frameworks and toolkits for creating extensive workflows in a distributed environment [88]. However, as edge nodes capable of supporting general-purpose computing become more prevalent, the development of new frameworks and toolkits becomes necessary.

The potential applications of edge analytics are expected to vary significantly from established workflows, which have mainly been explored in scientific fields such as bioinformatics [89] or astronomy [90]. As edge analytics becomes relevant in user-driven scenarios, the current frameworks may not be well-suited for representing edge analytics workflows. The programming model created to leverage the capabilities of edge nodes should be capable of handling task and data-level parallelism while executing workloads across various hierarchical levels of hardware.

Additionally, the programming language that supports this model should take into account the diverse hardware landscape and resource capacities present in the workflow. In cases where edge nodes are highly specific to a particular vendor, the frameworks supporting the workflow must be adaptable. This level of complexity goes beyond that of existing models designed to make cloud computing accessible.

4.2.3. Utilizing Lightweight Libraries and Algorithms

In contrast to large servers, edge nodes face limitations in supporting resource-intensive software due to hardware constraints. For instance, consider a small cell base station equipped with Intel's T3K Concurrent Dual-Mode system-on-chip (SoC). This device typically features a 4-core ARM-based CPU and limited memory, making it inadequate for executing complex data processing tools like Apache Spark. Apache Spark demands a minimum of 8 CPU cores and 8 gigabytes of memory for optimal performance. In the context of edge analytics, there is a need for lightweight algorithms capable of performing reasonable machine learning or data processing tasks [91].

One example of a lightweight library is Apache Quarks, which can be utilized on compact edge devices such as smartphones to enable real-time data analytics. However, Quarks primarily supports basic data processing functions, such as filtering and windowed aggregations, which may not suffice for advanced analytical tasks like context-aware recommendations. There is a demand for machine learning libraries that consume less memory and disk space, benefiting data analytical tools designed for edge nodes.

TensorFlow is another framework to consider, supporting deep learning algorithms and heterogeneous distributed systems, although its potential for edge analytics remains to be explored.

AI algorithms play a pivotal role in extracting valuable insights from big data. Nevertheless, the information extracted by existing algorithms is somewhat limited. In the case of supervised learning, manual data labeling can introduce unknown errors. Furthermore, the future data acquisition systems for smart medical applications will predominantly rely on wearable devices. The rapid analysis and response to collected data on these wearables present a significant challenge in terms of energy supply. Balancing the accuracy and lightweight nature of AI models is an area that warrants further investigation.

4.2.4. Micro Operating Systems and Virtualization

Research into micro-operating systems or microkernels presents a potential avenue for addressing challenges associated with deploying applications on diverse edge nodes. These nodes, unlike traditional servers, typically have limited resources. Therefore, it is essential to optimize the general-purpose computing environment at the edge by conserving resources. Advantages such as rapid deployment, shorter boot-up times, and resource isolation are highly desirable [92]. Initial studies suggest that mobile containers, which distribute device hardware functions among multiple virtual devices, can offer performance comparable to native hardware [93]. Container technologies, such as docker, are advancing and enabling swift application deployment on various platforms. However, further research is needed to establish containers as a suitable method for deploying applications on edge nodes.

Virtualization plays a vital role in the evolution of IT technologies, enabling the simultaneous operation of multiple operating systems or numerous applications on a single server [94]. Its key function is to diminish the reliance on physical servers, leading to substantial reductions in power consumption and cooling costs. The growing prevalence of IoT, mobile devices, and sensors has amplified the requirement for remote data centers [95]. Consequently, there exists an opportunity to relocate applications and intelligence from the cloud to the edge network. This transition can usher in a new type of virtualization at the edge, wherein a physical server can provide adaptable and dedicated storage and cache resources.

4.2.5. Energy Efficiency

The rapid proliferation of edge devices in urban areas has worsened the global energy crisis and the issue of global warming. One potential method to mitigate this problem involves harnessing renewable energy sources to power these edge devices. Given that these devices are dispersed throughout the city, adopting distributed renewable energy generators can significantly reduce the reliance on conventional energy sources. However, this approach is not without its challenges. It must address issues like minimizing the use of conventional energy while ensuring the uninterrupted operation of edge devices and establishing a complementary power system for various edge devices [96,97]. In the context of an Energy Internet system, the energy router, which serves as a control center, requires a certain level of computational capacity [98]. Hence, a plausible avenue for future research is to explore the integration of energy routers with edge intelligence.

5. AI Solutions for Optimizing Computation of Edge Intelligence

As discussed, major problems in computing for edge Intelligence are computing offload, resource allocation, privacy, and security. Enhancement in conventional approaches or hybridization can help improve and optimize computing, resource allocation, privacy, and security for Edge Intelligence. A detailed review of the techniques proposed by the researchers to address these objectives is presented in Table 3.

Table 3. Summary of Solutions for Optimizing Computing for Edge Intelligence.

Problem Addressed	Objective	Technique/Algorithm	Details of Technique/Algorithm
Computing offloading optimization	Reduction in energy consumption and latency.	Deep Reinforcement Learning (DRL) based offloading scheme [99]	Lack of prior familiarity with transmission delay and energy consumption models reduces the complexity of the state space by employing Deep Reinforcement Learning (DRL) to augment the understanding speed. Additionally, consider the Energy Consumption (EC) scenario involving energy harvesting.
		Deep Reinforcement Learning (DRL) based computing offloading algorithm [100]	Utilizes a Markov decision process for the portrayal of computational offloading, employing Deep Reinforcement Learning (DRL) to acquire insights into network dynamics.
		A hybrid approach based on Q-function breakdown and double DQN [101]	Utilized a double deep Qnetwork for the attainment of optimal computing offloading in the absence of prerequisite, employed a novel function approximator founded deep neural network (DNN) model which is designed to address high-dimensional state spaces.
		Reinforcement learning utilizing neural network architectures [102]	A continuous-time Markov decision process with an infinite horizon and average rewards is employed to model the optimization issue. Additionally, a novel value function approximator is introduced to address the challenges posed by high-dimensional state spaces.
	Optimization of the hardware structure for edge devices	Binary Weight Convolutional Neural Network based Algorithm [103]	Static random-access memory (SRAM) is designed for binary weight convolutional neural networks (CNNs) with the aim of minimizing memory data output, facilitating parallel implementation of CNN operations.
		Approach based on DNN and FPGA [104]	Expeditor for weed species categorization utilizes a binarized deep neural network, which is employed on field programmable gate arrays (FPGA).
	Reduction in energy consumption	Distributed Deep Learning based offloading technique [105]	Built a model by adding the cost of varying local implementation assignments in the cost function.
	Reduction in latency	DL based Smart-Edge-CoCaCo [106]	An approach based on joint optimization of wireless communication, combined filter caching and computation offloading, is developed to reduce the latency.
		A heuristic offloading technique [107]	Using electronic communication networks to estimate the distance between origin and destination, along with heuristic searching, to identify the most effective scheme for reducing the communication lag of deep learning tasks.
		Cooperative Q-learning [108]	Noticeable improvement in the searching pace of the conventional Q-learning approach.
		TD learning involves a method that incorporates post-decision states and utilizes a semi-gradient descent approach [109]	Utilized approximate dynamical planning as a strategy to tackle the difficulties established by the curse of dimensionality.
		Online Reinforcement Learning [110]	Unique arrangements of state transitions are designed to address the difficulties raised by the curse of dimensionality. Moreover, it takes into account the energy-harvesting aspect of Edge intelligence.

Table 3. *Cont.*

Problem Addressed	Objective	Technique/Algorithm	Details of Technique/Algorithm
Security of Edge Intelligence		Hypergraph clustering [111]	Improves the identification rate by modeling the association among edge nodes and DDoS through hypergraph clustering.
		Extreme Learning Machine [112]	Demonstrate quicker convergence rates and enhanced generalization capabilities of the Extreme Learning Machine classifier compared to the majority of traditional algorithms.
		Distributed Deep Learning [113]	Eases the load of training the model while enhancing its accuracy.
		An algorithm based on restricted Boltzmann machines [114]	Enhances the ability to identify unfamiliar attacks through the incorporation of active learning features.
		Deep PDS-Learning [115]	Accelerate the training process by incorporating supplementary details, such as the energy consumption of edge devices.
Resource allocation optimization		Actor-critic RL [116]	Introduced an additional Deep Neural Network (DNN) for expressing a parameterized stochastic policy, aiming to enhance both performance and convergence speed. Additionally, incorporated a natural policy gradient approach to mitigate the risk of local convergence.
		DRL-based resource allocation scheme [117]	Enhanced the Quality of Service (QoS) through the integration of supplementary SDN
		Multi-task DRL [118]	Modifies the final layer of a Deep Neural Network (DNN) responsible for estimating the Q-function to accommodate action spaces with increased dimensions.
Privacy protection		Generative adversarial networks (GAN) [119]	An algorithm for objective perturbation and another for output perturbation, both ensuring adherence to the principles of differential privacy.
		Edge Sanitizer: A deep inference framework [120]	Proposes maximum utilization of data while ensuring privacy protection.
		Deep Q-learning [121]	Generate trust values through uncertain reasoning and prevent local convergence by regulating the learning rate.
Other ways to reduce energy consumption	Control device operational condition	DRL-based joint mode selection and resource management approach [122]	Minimizes energy consumption in the medium and long term by managing the communication mode of the operator apparatus and regulating the active state of processors.
	Merging into energy Internet	Model-based DRL [123]	Solves the energy supply issue of the multi-access edge server.
		Reinforcement Learning [124]	A fog computing device operating on energy generated from a renewable source.
		Minimax-Q learning [125]	Gradually learns the optimal strategy by raising the spectral efficiency throughput.
		Online learning [126]	Minimized bandwidth utilization by selecting the server with the highest reliability.
		Multiple Artificial Intelligence based algorithms [127]	Developed a mechanism for selecting AI algorithms intelligently to choose the most suitable algorithm for a given task.

6. Architectural Layers in the Roadmap for Edge Intelligence

The architectural layers in the Roadmap for edge intelligence, distinguishing between two main directions, viz. AI for the edge and AI on the edge as shown in Figure 4. Using a bottom-up strategy, our focus in Edge intelligence research is on dividing efforts into topology, content, and service segments, all of which stand to gain advantages from AI technologies. Conversely, a top-down method dissects AI research on the edge into model

adaptation, framework design, and processor acceleration. Prior to exploring AI for the edge and AI on the edge as distinct entities, it is essential to establish a shared objective, termed Quality of experience (QoE), which consistently takes precedence. The detailed discussion of QoE, AI for the edge, and AI on the edge is discussed further in this section.

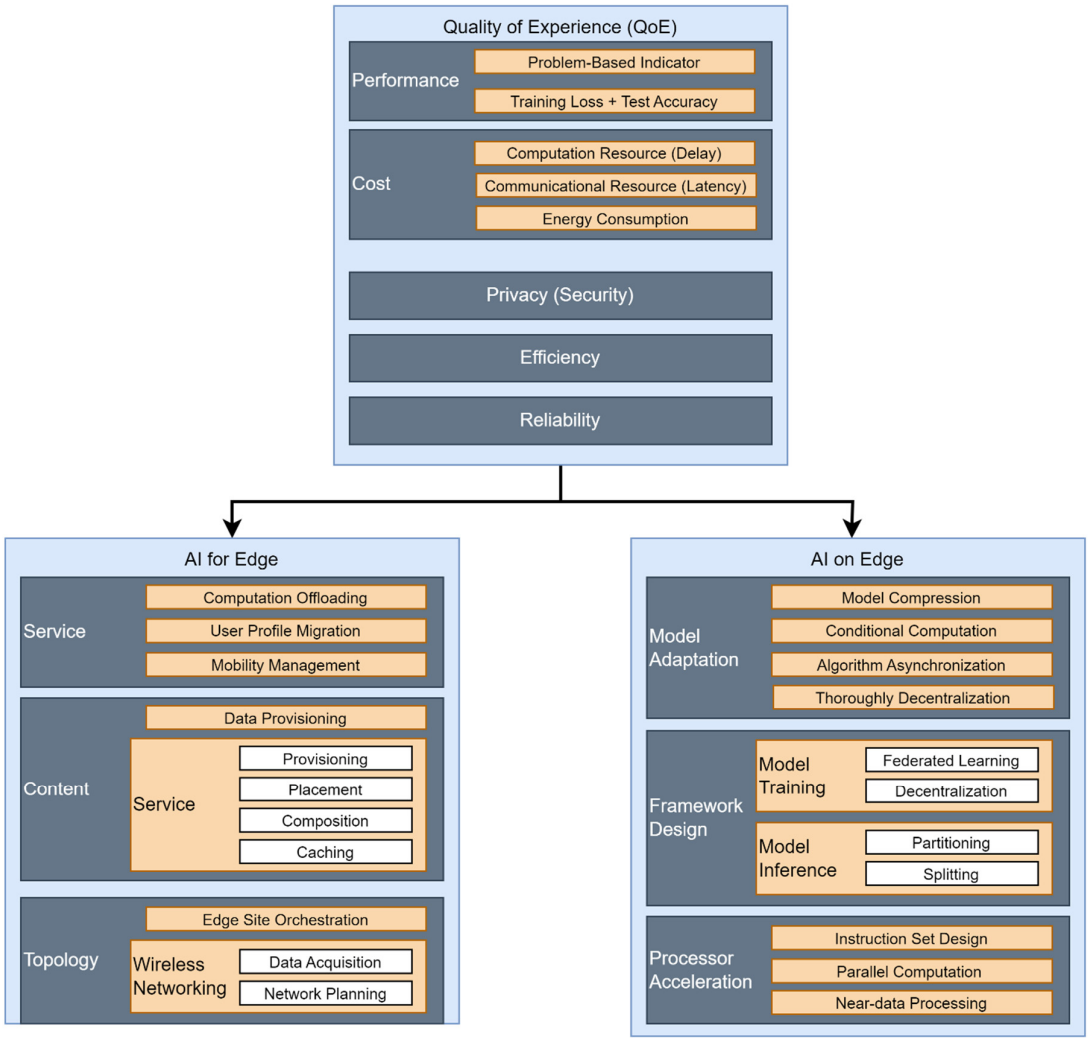

Figure 4. The architectural layers in the Roadmap.

6.1. Quality of Experience

We believe that QoE should be tailored to specific applications and should be established by contemplating multiple criteria: performance, cost, privacy (security), efficiency, and reliability.

- *Performance*

Performance criteria differ between AI for the edge and AI on the edge. In the case of the former, performance metrics are tailored to specific problems. For instance, it might encompass metrics like the successful offloading ratio in computation offloading challenges or the efficient optimization of revenue and hiring costs for base stations in service placement issues. On the other hand, for the latter, performance primarily centers around training loss and inference accuracy, both critical for AI models. Despite the transition from cloud clusters to a system integrating devices, edge, and cloud, these criteria continue to be significant.

- *Cost*

Cost considerations generally encompass computation cost, communication cost, and energy consumption. Computation cost indicates the need for computing resources, including factors like CPU cycle frequency and allocated CPU time. Communication cost deals with the resource requirements for communication, considering aspects like power, frequency band, and access time, with a focus on minimizing delays arising from the allocation of computation and communication resources. Energy consumption is particularly crucial, especially for mobile devices with restricted battery capacity. The importance of cost reduction cannot be overstated, as Edge intelligence holds the potential for substantial decreases in delay and energy consumption, simultaneously addressing critical challenges in realizing 5G capabilities.

- *Privacy and Security*

With growing apprehensions about data leaks, safeguarding privacy has gained significant attention. Consequently, Federated Learning has emerged as a solution involving the aggregation of local machine-learning models from distributed devices while actively preventing data leakage [128]. Security is intricately linked with privacy preservation and holds implications for the resilience of middleware and edge systems.

- *Efficiency*

Whether in the realm of AI for the edge or AI on the edge, achieving high efficiency is paramount to achieving outstanding performance with minimal overhead. The pursuit of efficiency is crucial for the improvement of existing algorithms and models, especially in the context of AI on the edge. Numerous strategies, including model compression, conditional computation, and asynchronous algorithms, have been suggested to enhance the efficiency of training and inference processes for deep AI models.

- *Reliability*

System reliability plays a pivotal role in ensuring the continuous operation of Edge intelligence over specified durations, a critical element for user experience. In the domain of edge intelligence, system reliability holds particular importance for AI on the edge. This is especially true when considering that model training and inference frequently take place in a distributed and synchronized manner, and local users may encounter obstacles related to wireless network congestion when attempting to complete model uploads and downloads.

6.2. Edge Intelligence/Intelligent Edge Computing

The Roadmap, as illustrated in Figure 4, pertains to AI for Edge intelligence, which we refer to as intelligent edge computing (IEC). AI offers potent tools for addressing intricate challenges in learning, planning, and decision-making. We adopt a bottom-up approach to categorize the primary concerns in Edge intelligence into three layers: topology, content, and service.

- *Topology*

In terms of topology, our attention is directed towards orchestrating edge sites (OES) and wireless networking (WN). Within our framework, an edge site is defined as a micro data center hosting deployed applications and connected to a small-cell base station (SBS). OES focuses on the deployment and configuration of wireless telecom equipment and servers. Notably, recent years have seen a surge in interest surrounding the management and automation of unmanned aerial vehicles (UAVs). These UAVs, equipped with a small server and access point, can be viewed as mobile edge servers with exceptional maneuverability. Consequently, numerous studies explore scheduling and trajectory planning challenges with the aim of minimizing UAV energy consumption.

For instance, Chen et al. [129] investigated power consumption by caching popular content based on predictions, introducing a conceptor-based echo state network (ESN) algorithm to learn user mobility patterns. Leveraging this efficient machine learning

technique, their algorithm significantly outperforms benchmarks in terms of transmission power and user satisfaction. On the other hand, WN encompasses data acquisition and network planning. The former focuses on swiftly acquiring data from widely distributed sources at edge devices, while the latter concentrates on network scheduling, operation, and management. Fast data acquisition involves elements such as multiple access, radio resource allocation, and signal encoding/decoding. Network planning explores efficient management through protocols and middleware.

Significantly, recent years have witnessed a rising trend in intelligent networking, utilizing AI technologies to construct intelligent wireless communication mechanisms. As an example, Zhu et al. [130] proposed learning-driven communication, exploiting the synergy between communication and learning in edge systems.

- *Content*

The focus lies on several vital aspects, including data provisioning, service provisioning, service placement, service composition, and service caching. In the realms of data and service provisioning, resources can be drawn from remote cloud data centers and edge servers. Recent initiatives have concentrated on developing lightweight Quality of Service (QoS) aware service-based frameworks. Alternatively, shared resources may originate from mobile devices with suitable incentive mechanisms in place.

Service placement complements service provisioning and delves into the location and method of deploying complex services on potential edge sites. In recent times, numerous studies have approached service placement from the perspective of application service providers (ASPs). For instance, Chen et al. [131] endeavored to deploy services within a limited budget on fundamental communication and computation infrastructure. Subsequently, they applied the multi-armed bandit (MAB) theory, a branch of reinforcement learning, to optimize service placement decisions.

Service composition involves the selection of candidate services for composition, taking into account energy consumption and Quality of Experience (QoE) for mobile end users. This domain presents opportunities for leveraging AI technologies to generate more effective service selection strategies. Service caching, akin to service provisioning, revolves around designing a caching pool to store frequently accessed data and services. It can also be explored in a cooperative manner, offering research prospects for applying multi-agent learning to enhance QoE in large-scale Edge intelligence systems.

- *Services*

Regarding services, our focus is on computation offloading, user profile migration, and mobility management. Computation offloading addresses the load balancing of various computational and communication resources, involving edge server selection and frequency spectrum allocation. Recent research has concentrated on dynamically managing radio and computational resources for multi-user, multi-server Edge intelligence systems, utilizing Lyapunov optimization techniques. Computation offloading decisions are also being optimized through Deep Q-Network (DQN), modeling the problem as a Markov Decision Process (MDP) to maximize long-term utility performance, encompassing the aforementioned Quality of Experience (QoE) indicators.

User profile migration involves adjusting the location of user profiles, encompassing configuration files, private data, and logs, as mobile users are constantly on the move. User profile migration is often intertwined with mobility management. For instance, the JCORM algorithm, proposed in [132], optimizes computation offloading and migration through cooperative networks, presenting research opportunities for the application of more advanced AI technologies to enhance optimality.

Mobility management, viewed through the lens of statistics and probability theory, is another area of interest. There is a notable inclination toward realizing mobility management with the assistance of AI.

6.3. AI on the Edge (AIE)

The right side of the Roadmap focuses on AI on edge, referred to as AIE, involving the study of implementing AI model training and inference on the network edge. This area is categorized into four research endeavors: framework design, model training, inference and adaptation, conditional computation, and processor acceleration. Given that model adaptation builds upon existing training and inference frameworks, the initial focus will be on introducing framework design.

- *Framework Design*

Framework design aims to establish improved training and inference architecture for the edge without altering existing AI models. Researchers strive to create new frameworks for both model training and model inference.

- *Model Training*

Presently, the predominant frameworks for model training are largely distributed, with the exception of those based on knowledge distillation. Distributed training frameworks can be categorized based on data splitting and model splitting methods [133]. Data splitting involves master-device, helper-device, and device-device splitting, differing in how training samples are sourced and how the global model is aggregated. Model splitting involves separating neural network layers onto various devices, relying on complex pipelines. Knowledge distillation-based frameworks may or may not be decentralized, utilizing transfer learning technologies to enhance the accuracy of shallower student networks [119]. This process entails training a basic network on a standard dataset and transferring the learned features to student networks, which are then trained on their respective datasets by multiple mobile end devices. There is significant potential for exploration in knowledge distillation-based frameworks for model training at the edge.

The leading approach in model training is Federated Learning, designed to preserve privacy while training Deep Neural Networks (DNNs) in a distributed manner [134]. It trains local models on multiple clients, optimizing a global model by averaging trained gradients. Due to limited resources in edge nodes, training a comprehensive model there is impractical, making distributed training more feasible. However, coordination between edge nodes becomes essential. The challenge lies in optimizing the global gradient from distributed local models. Regardless of the learning algorithms used, stochastic gradient descent (SGD) plays a crucial role in model training. Edge nodes utilize SGD to update local gradients based on their datasets, sending updates to a central node for global model enhancement. Balancing model performance and communication overhead is critical. Selective transmission of local gradients showing significant improvements can ensure global model performance while reducing communication overheads, preventing network congestion caused by simultaneous transmissions from all edge nodes.

- *Model Inference*

While splitting a model during training poses challenges, it is a preferred method during model inference. Model splitting, or partitioning, serves as a framework for model inference, and various techniques, including model compression, input filtering, early exit, etc., are adaptations from existing frameworks. A well-described example of model inference on the edge can be found in reference [129], where a Deep Neural Network (DNN) is divided into two parts, each processed collaboratively. The computationally intense segment operates on the edge server, while the other part functions on the mobile device. The challenge lies in determining the optimal layer to split and when to exit the intricate DNN while maintaining inference accuracy.

- *Model Adaptation*

Model adaptation is a process that fine-tunes existing training and inference frameworks, particularly in the context of Federated Learning, to better suit Edge intelligence. While Federated Learning can potentially operate on the edge, its traditional version demands significant communication efficiency, as complete local models are transmitted back

to the central server. Consequently, many researchers are concentrating on developing more efficient model updates and aggregation policies. Their endeavors aim to minimize costs, enhance robustness, and ensure system performance. Approaches to achieving model adaptation include techniques such as model compression, conditional computation, algorithm synchronization, and comprehensive decentralization.

Model compression exploits the inherent sparsity structure of gradients and weights. Potential strategies encompass quantization, dimensional reduction, pruning, precision downgrading, components sharing, and cutoff, among others. Implementation methods involve techniques like Singular Value Decomposition (SVD), Huffman coding, Principal Component Analysis (PCA), and similar approaches.

- *Conditional Computation*

Conditional computation serves as an alternative approach to reduce computations by selectively disabling less crucial calculations in Deep Neural Networks (DNNs). Feasible methods include component shutdown, input filtering, early exit, and results caching. This concept can be likened to block-wise dropout [135,136]. Additionally, random gossip communication can help reduce unnecessary calculations and model updates. Asynchronization in algorithms aims to aggregate local models asynchronously, mitigating the inefficiency and lengthy synchronous steps of model updates in Federated Learning. Thorough decentralization involves eliminating the central aggregator to prevent potential data leaks and address issues stemming from the central server's malfunction. Strategies for achieving complete decentralization encompass blockchain technologies, game-theoretical approaches, and similar methods.

- *Process Acceleration*

Processor acceleration aims to optimize the structure of DNNs, specifically targeting the frequently used computation-heavy multiply-and-accumulate operations for improvement. Strategies for enhancing DNN computations on hardware involve various methods, such as Creating specialized instruction sets for DNN training and inference, developing highly parallel computing paradigms, and implementing near-data processing to bring computation closer to memory. Highly parallelized computing paradigms can be categorized into temporal and spatial architectures. Temporal architectures like CPUs and GPUs can be accelerated by reducing the number of multiplications and increasing throughput. Spatial architectures, on the other hand, can be accelerated by enhancing data reuse with data flows.

7. Conclusions

Autonomous vehicles represent a significant milestone in the latest wave of technological advancements, serving as a crucial indicator of technological evolution. AVs can significantly contribute to decreasing traffic accidents and enhance road safety by eliminating hazardous driving behaviors like fatigued driving. This article extensively examines the essential technologies indispensable for realizing autonomous vehicles, highlighting the interconnections between these technologies and the development of AVs. Edge intelligence, a fusion of AI and edge computing, plays a pivotal role in empowering vehicles to make intelligent decisions swiftly. The critical components of edge intelligence include edge caching, edge training, edge inference, and edge offloading are well evaluated and analysed in this research. Efficient deployment of cache is essential to make optimal use of both base station and device resources, and it is crucial to optimize the cache replacement strategy to enhance overall performance. In the context of edge training, the focus lies on collaborative training, accelerating and optimizing communication frequency and costs, all while maintaining a high standard of security and privacy. While AutoML techniques, specifically architectural search, play a role in edge interpretation, human expertise remains vital for designing optimal models. Techniques such as knowledge distillation and pruning are employed to achieve model compression. Leveraging a distributed computation

paradigm through edge offloading can significantly improve both model training and interpretation performance.

The efficient deployment of AI to the network's edge hinges on improving the efficacy of AI algorithms with limited computing and energy resources, necessitating the design of lightweight AI models. Beyond the significance of individual technologies, this study also delves into the challenges that must be overcome and the areas that require reinforcement. Investigating the incorporation of important technologies, issues, opportunities, and roadmap in this study will be a valuable resource for the community engaged in research on edge intelligence in EV.

Author Contributions: All the authors have contributed equally towards the conceptualization, methodology, analysis, investigation, and resources. The original draft and changes were also performed by all the authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest. Sachin B. Chougule and Sheetal N. Ghorpade are employees of Rubiscape Private Limited, Pune, India. The paper reflects the views of the scientists and not the company.

References

1. Elvas, L.B.; Ferreira, J.C. Intelligent Transportation Systems for Electric Vehicles. *Energies* **2021**, *14*, 5550. [CrossRef]
2. Ahmad, K.; Khujamatov, H.; Lazarev, A.; Usmanova, N.; Alduailij, M.; Alduailij, M. Internet of Things-Aided Intelligent Transport Systems in Smart Cities: Challenges, Opportunities, and Future. *Wirel. Commun. Mob. Comput.* **2023**, *2023*, 7989079. [CrossRef]
3. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. GWO Model for Optimal Localization of IoT-Enabled Sensor Nodes in Smart Parking Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1217–1224. [CrossRef]
4. Khayyam, H.; Javadi, B.; Jalili, M.; Jazar, R.N. Artificial intelligence and internet of things for autonomous vehicles. In *Nonlinear Approaches in Engineering Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 39–68.
5. Pradhan, N.M.; Chaudhari, B.S.; Zennaro, M. 6TiSCH Low Latency Autonomous Scheduling for Industrial Internet of Things. *IEEE Access* **2022**, *10*, 71566–71575. [CrossRef]
6. Zhang, K.; Zhu, Y.; Leng, S.; He, Y.; Maharjan, S.; Zhang, Y. Deep learning empowered task offloading for mobile edge computing in urban informatics. *IEEE Internet Things J.* **2019**, *6*, 7635–7647. [CrossRef]
7. Chaudhari, B.; Borkar, S. Design Considerations and Network Architectures for Low-Power Wide-Area Networks. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 15–35, ISBN 9780128188804. [CrossRef]
8. Dai, Y.; Xu, D.; Maharjan, S.; Qiao, G.; Zhang, Y. Artificial intelligence empowered edge computing and caching for internet of vehicles. *IEEE Wirel. Commun.* **2019**, *26*, 12–18. [CrossRef]
9. Mendez, J.; Bierzynski, K.; Cuéllar, M.; Morales, D.P. Edge Intelligence: Concepts, architectures, applications and future directions. *ACM Trans. Embed. Comput. Syst. (TECS)* **2022**, *21*, 1–41. [CrossRef]
10. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S.; Saeed, R.A.; Alhumyani, H.; Abdel-Khalek, S. Enhanced Differential Crossover and Quantum Particle Swarm Optimization for IoT Applications. *IEEE Access* **2021**, *9*, 93831–93846. [CrossRef]
11. Li, Y. Deep reinforcement learning. *arXiv* **2018**, arXiv:1810.06339.
12. Shakya, A.K.; Pillai, G.; Chakrabarty, S. Reinforcement learning algorithms: A brief survey. *Expert Syst. Appl.* **2023**, *231*, 120495. [CrossRef]
13. Stefenon, S.F.; Yow, K.-C.; Nied, A.; Meyer, L.H. Classification of distribution power grid structures using inception v3 deep neural network. *Electr. Eng.* **2022**, *104*, 4557–4569. [CrossRef]
14. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. IoT-based hybrid optimized fuzzy threshold ELM model for localization of elderly persons. *Expert Syst. Appl.* **2021**, *184*, 115500. [CrossRef]
15. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. Binary grey wolf optimization-based topology control for WSNs. *IET Wirel. Sens. Syst.* **2019**, *9*, 333–339. [CrossRef]
16. Drolia, U.; Guo, K.; Tan, J.; Gandhi, R.; Narasimhan, P. Cachier: Edge-caching for recognition applications. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 276–286.
17. Drolia, U.; Guo, K.; Narasimhan, P. Precog: Prefetching for image recognition applications at the edge. In Proceedings of the Second ACM/IEEE Symposium on Edge Computin, San Jose, CA, USA, 12–14 October 2017; p. 17.

18. Guo, P.; Hu, B.; Li, R.; Hu, W. Foggy Cache: Cross-device approximate computation reuse. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, MobiCom 2018, New Delhi, India, 29 October–2 November 2018; pp. 19–34.
19. Xu, M.; Zhu, M.; Liu, Y.; Lin, F.X.; Liu, X. Deep Cache: Principled cache for mobile deep vision. *arXiv* **2017**, arXiv:1712.01670.
20. Li, T.; Xiao, Z.; Georges, H.M.; Luo, Z.; Wang, D. Performance analysis of co-and cross-tier device-to-device communication underlaying macro-small cell wireless networks. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 1481–1500.
21. Xiao, Z.; Li, T.; Ding, W.; Wang, D.; Zhang, J. Dynamic PCI allocation on avoiding handover confusion via cell status prediction in LTE heterogeneous small cell networks. *Wirel. Commun. Mob. Comput.* **2016**, *16*, 1972–1986. [CrossRef]
22. Xiao, Z.; Liu, H.; Havyarimana, V.; Li, T.; Wang, D. Analytical study on multi-tier 5g heterogeneous small cell networks: Coverage performance and energy efficiency. *Sensors* **2016**, *16*, 1854. [CrossRef]
23. Xiao, Z.; Li, T.; Cheng, W.; Wang, D. Apollonius circles based outbound handover in macro-small wireless cellular networks. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
24. Ji, M.; Caire, G.; Molisch, A.F. Wireless device-to-device caching networks: Basic principles and system performance. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 176–189. [CrossRef]
25. Chen, W.; Li, T.; Xiao, Z.; Wang, D. On mitigating interference under device-to-device communication in macro-small cell networks. In Proceedings of the 2016 International Conference on Computer, Information and Telecommunication Systems (CITS), Kunming, China, 6–8 July 2016; pp. 1–5.
26. Ioannou, A.; Weber, S. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2847–2886. [CrossRef]
27. McMahan, B.; Ramage, D. Federated learning: Collaborative machine learning without centralized training data. *Google Res. Blog* **2017**, *3*, 355–359. [CrossRef]
28. Valery, O.; Liu, P.; Wu, J.-J. CPU/GPU collaboration techniques for transfer learning on mobile devices. In Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), Shenzhen, China, 15–17 December 2017; pp. 477–484.
29. Valery, O.; Liu, P.; Wu, J.-J. Low Precision Deep Learning Training on Mobile Heterogeneous Platform. In Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Cambridge, UK, 21–23 March 2018; pp. 109–117. [CrossRef]
30. Xing, T.; Sandha, S.S.; Balaji, B.; Chakraborty, S.; Srivastava, M. Enabling edge devices that learn from each other: Cross modal training for activity recognition. In Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking, Munich, Germany, 10 June 2018; ACM: New York, NY, USA, 2018; pp. 37–42.
31. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 1175–1191.
32. Liu, H.; Simonyan, K.; Yang, Y. Darts: Differentiable architecture search. *arXiv* **2018**, arXiv:1806.09055.
33. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
34. Choudhary, T.; Mishra, V.; Goswami, A.; Sarangapani, J. A comprehensive survey on model compression and acceleration. *Artif. Intell. Rev.* **2020**, *53*, 5113–5155. [CrossRef]
35. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. Towards Green Computing: Intelligent Bio-Inspired Agent for IoT-enabled Wireless Sensor Networks. *Int. J. Sens. Netw.* **2021**, *35*, 121. [CrossRef]
36. Raval, N.; Srivastava, A.; Razeen, A.; Lebeck, K.; Machanavajjhala, A.; Cox, L.P. What you mark is what apps see. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 26–30 June 2016; ACM: New York, NY, USA, 2016; pp. 249–261.
37. Wendelken, S.; MacGillivray, C. Worldwide and U.S. IoT Cellular Connections Forecast, 2021–2025. Available online: https://www.idc.com/getdoc.jsp?containerId=US47296121 (accessed on 17 February 2022).
38. Wong, T.-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognit.* **2015**, *48*, 2839–2846. [CrossRef]
39. Mittal, S. A survey of FPGA-based accelerators for convolutional neural networks. *Neural Comput. Appl.* **2018**, *32*, 1109–1139. [CrossRef]
40. Manokaran, J.; Vairavel, G. An Empirical Comparison of Machine Learning Algorithms for Attack Detection in Internet of Things Edge. *ECS Trans.* **2022**, *107*, 2403.
41. Watson, D.S. On the Philosophy of Unsupervised Learning. *Philos. Technol.* **2023**, *36*, 28. [CrossRef]
42. Thomos, N.; Maugey, T.; Toni, L. Machine Learning for Multimedia Communications. *Sensors* **2022**, *22*, 819. [CrossRef]
43. Chaudhari, B.S.; Zennaro, M. Introduction to low-power wide-area networks. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 1–13. [CrossRef]
44. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
45. Cheng, X.; Chen, C.; Zhang, W.; Yang, Y. 5G-enabled cooperative intelligent vehicular (5GenCIV) Framework: When Benz Meets Marconi. *IEEE Intell. Syst.* **2017**, *32*, 53–59. [CrossRef]

46. Liang, L.; Peng, H.; Li, G.Y.; Shen, X. Vehicular communications: A network layer perspective. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10647–10659. [CrossRef]
47. Ye, H.; Li, G.Y.; Juang, B.-H.F. Deep reinforcement learning based resource allocation for V2V communications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3163–3173. [CrossRef]
48. A Look at the Future of 5G. Available online: https://spectrum.ieee.org/computing/software/a-look-at-the-future-of-5g/ (accessed on 28 November 2023).
49. Gaudet, B. Review of cooperative truck platooning systems. *Natl. Res. Counc. Can.* **2014**, *10*, 1–79.
50. Hou, X.; Li, Y.; Wu, D.; Chen, S. Vehicularfog computing: A viewpoint of vehicles as the infrastructures. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3860–3873. [CrossRef]
51. Eltoweissy, M.; Olariu, S.; Younis, M. Towards autonomous vehicular clouds. In Proceedings of the Springer Conference on International Conference on Ad Hoc Networks, Edmonton, AB, Canada, 20–22 August 2010; pp. 1–16.
52. Hong, K.; Lillethun, D.; Ramachandran, U.; Ottenw, B.; Koldehofe, B. Mobile fog: A programming model for largescale applications on the internet of things. In Proceedings of the ACM SIGCOMM Workshop on Mobile Cloud Computing, Hong Kong, China, 16 August 2013; pp. 15–20.
53. Kaur, S.; Dhillon, K.K.; Manvi, M.; Singh, R. An automatic system for detecting the vehicle registration plate from video in foggy and rainy environments using restoration technique. *Int. J. Comput. Appl.* **2014**, *97*, 14–19. [CrossRef]
54. Roy, S.; Bose, R.; Sarddar, D. A fog-based DSS model for driving rule violation monitoring framework on the internet of things. *Int. J. Adv. Sci. Technol.* **2015**, *82*, 23–32. [CrossRef]
55. Vashitz, G.; Shinar, D.; Blum, Y. In-vehicle information systems to improve traffic safety in road tunnels. *Transp. Res. Part F Traffic Psychol. Behav.* **2008**, *11*, 61–74. [CrossRef]
56. Miah, S.J.; Ahamed, R. A cloud-based DSS model for driver safety and monitoring on Australian roads. *Int. J. Emerg. Sci.* **2011**, *1*, 634.
57. Vahdat-Nejad, H.; Ramazani, A.; Mohammadi, T.; Mansoor, W. A survey on context-aware vehicular network applications. *Veh. Commun.* **2016**, *3*, 43–57. [CrossRef]
58. Baldauf, M.; Dustdar, S.; Rosenberg, F. A survey on context aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* **2007**, *2*, 263–277. [CrossRef]
59. Bogale, T.E.; Wang, X.; Le, L.B. Machine intelligence techniques for next-generation context-aware wireless networks. *Comput. Sci. Inf. Theory* **2018**. [CrossRef]
60. He, Q.; Liu, J.; Wang, C.; Li, B. Coping with heterogeneous video contributors and viewers in crowdsourced live streaming: A cloud-based approach. *IEEE Trans. Multimed.* **2016**, *18*, 916–928. [CrossRef]
61. Zhuo, G.; Jia, Q.; Guo, L.; Li, M.; Li, P. Privacy-Preserving Verifiable Set Operation in Big Data for Cloud-Assisted Mobile Crowdsourcing. *IEEE Internet Things J.* **2017**, *4*, 572–582. [CrossRef]
62. Huang, C.; Xu, K. Reliable real time streaming in vehicular cloud-fog computing networks. In Proceedings of the IEEE Conference on Communications in China, Chengdu, China, 27–29 July 2016; pp. 1–6.
63. Grassi, G.; Bahl, P.; Jamieson, K.; Pau, G. Park Master: An in vehicle, edge-based video analytics service for detecting open parking spaces in urban environments. In Proceedings of the ACM/IEEE Symposium on Edge Computing, San Jose, CA, USA, 18–21 April 2017; p. 16.
64. Cho, S.Y. Development of an IGVM integrated navigation system for vehicular lane-level guidance services. *J. Position. Navig. Timing* **2016**, *5*, 119–129. [CrossRef]
65. Park, H.S.; Park, M.W.; Won, K.H.; Kim, K.-H.; Jung, S.K. In-Vehicle AR-HUD system to provide driving-Safety information. *ETRI J.* **2013**, *35*, 1038–1047. [CrossRef]
66. Ghorpade, S.; Zennaro, M.; Chaudhari, B. Survey of Localization for Internet of Things Nodes: Approaches, Challenges and Open Issues. *Future Internet* **2021**, *13*, 210. [CrossRef]
67. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S.; Saeed, R.A.; Alhumyani, H.; Abdel-Khalek, S. A Novel Enhanced Quantum PSO for Optimal Network Configuration in Heterogeneous Industrial IoT. *IEEE Access* **2021**, *9*, 134022–134036. [CrossRef]
68. Li, W.; Zhao, Y.; Lu, S.; Chen, D. Mechanisms and challenges on Mobility-augmented Service Provisioning for Mobile Cloud Computing. *IEEE Commun. Mag.* **2015**, *53*, 89–97. [CrossRef]
69. Hromic, H.; Le Phuoc, D.; Serrano, M.; Antonic, A.; Zarko, I.P.; Hayes, C.; Decker, S. Real Time Analysis of Sensor Data for the Internet of Things by Means of Clustering and Event Processing. In Proceedings of the IEEE International Conference on Communications, London, UK, 8–12 June 2015; pp. 685–691.
70. Meurisch, C.; Seeliger, A.; Schmidt, B.; Schweizer, I.; Kaup, F.; Muhlh, M. Upgrading Wireless Home Routers for Enabling Large-scale Deployment of Cloudlets. In *Mobile Computing, Applications, and Services*; Springer: Cham, Switzerland, 2015; pp. 12–29.
71. Shafique, M.; Theocharides, T.; Bouganis, C.S.; Hanif, M.A.; Khalid, F.; Hafz, R.; Rehman, S. An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era. In Proceedings of the 2018 Design, Automation Test in Europe Conference Exhibition (DATE), Dresden, Germany, 19–23 March 2018; pp. 827–832.
72. Povedano-Molina, J.; Lopez-Vega, J.M.; Lopez-Soler, J.M.; Corradi, A.; Foschini, L. DARGOS: A Highly Adaptable and Scalable Monitoring Architecture for Multi-Tenant Clouds. *Future Gener. Comput. Syst.* **2013**, *29*, 2041–2056. [CrossRef]

73. Perez-Espinoza, J.A.; Sosa-Sosa, V.J.; Gonzalez, J.L.; Tello-Leal, E. A Distributed Architecture for Monitoring Private Clouds. In Proceedings of the 2015 26th International Workshop on Database and Expert Systems Applications (DEXA), Valencia, Spain, 1–4 September 2015; pp. 186–190. [CrossRef]
74. Grozev, N.; Buyya, R. Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey. *Softw. Pract. Exp.* **2014**, *44*, 369–390. [CrossRef]
75. Garg, S.K.; Versteeg, S. A Framework for Ranking of Cloud Computing Services. *Future Gener. Comput. Syst.* **2013**, *29*, 1012–1023. [CrossRef]
76. Singh, P.; Kaur, A.; Gill, S.S. Machine learning for cloud, fog, edge and serverless computing environments: Comparisons, performance evaluation benchmark and future directions. *Int. J. Grid Util. Comput.* **2022**, *13*, 447–457. [CrossRef]
77. Stacker, L.; Fei, J.; Heidenreich, P.; Bonarens, F.; Rambach, J.; Stricker, D.; Stiller, C. Deployment of deep neural networks for object detection on edge ai devices with runtime optimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1015–1022.
78. Iftikhar, S.; Ahmad, M.M.M.; Tuli, S.; Chowdhury, D.; Xu, M.; Gill, S.S.; Uhlig, S. Hunterplus: Ai based energy-efficient task scheduling for cloud–fog computing environments. *Internet Things* **2023**, *21*, 100667. [CrossRef]
79. Mousavi, S.; Mood, S.E.; Souri, A.; Javidi, M.M. Directed Search: A New Operator in Nsga-Ii for Task Scheduling in IoT Based on Cloud-Fog Computing. *IEEE Trans. Cloud Comput.* **2022**, *11*, 2144–2157. [CrossRef]
80. Ghafariana, T.; Javadi, B. Cloud-aware Data Intensive Workflow Scheduling on Volunteer Computing Systems. *Future Gener. Comput. Syst.* **2015**, *51*, 87–97. [CrossRef]
81. Tang, W.; Jenkins, J.; Meyer, F.; Ross, R.; Kettimuthu, R.; Winkler, L.; Yang, X.; Lehman, T.; Desai, N. Data-Aware Resource Scheduling for Multicloud Workflows: A Fine-Grained Simulation Approach. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science, Singapore, 15–18 December 2014; pp. 887–892.
82. Beck, M.T.; Maier, M. Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms. In Proceedings of the International Conference on Advanced Engineering Computing and Applications in Sciences, Rome, Italy, 24–28 August 2014; pp. 65–70.
83. Simoens, P.; Van Herzeele, L.; Vandeputte, F.; Vermoesen, L. Challenges for Orchestration and Instance Selection of Composite Services in Distributed Edge Clouds. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, ON, Canada, 11–15 May 2015; pp. 1196–1201.
84. Gupta, S.; Chaudhari, B.S.; Chakrabarty, B. Vulnerable Network Analysis Using War Driving and Security Intelligence. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; IEEE: Coimbatore, India, 2016; pp. 1–5. [CrossRef]
85. Valerio, L.; Passarella, A.; Conti, M. A communication efficient distributed learning framework for smart environments. *Pervasive Mob. Comput.* **2017**, *41*, 46–68. [CrossRef]
86. Baset, S.A. Cloud SLAs: Present and Future. *ACM SIGOPS Oper. Syst. Rev.* **2012**, *46*, 57–66. [CrossRef]
87. Bui, T. Analysis of Docker Security. *arXiv* **2015**, arXiv:1501.02967. Available online: http://arxiv.org/abs/1501.02967 (accessed on 3 June 2023).
88. Deelman, E.; Vahi, K.; Juve, G.; Rynge, M.; Callaghan, S.; Maechling, P.J.; Mayani, R.; Chen, W.; da Silva, R.F.; Livny, M.; et al. Pegasus: A Workflow Management System for Science Automation. *Future Gener. Comput. Syst.* **2015**, *46*, 17–35. [CrossRef]
89. Serrano-Solano, B.; Fouilloux, A.; Eguinoa, I.; Kalaš, M.; Grüning, B.; Coppens, F. Galaxy: A decade of realizing CWFR concepts. *Data Intell.* **2022**, *4*, 358–371. [CrossRef]
90. Ruiz, J.; Garrido, J.; Santander-Vela, J.; Snchez-Expsito, S.; Montenegro, L.V. Astro Taverna-Building Workflows with Virtual Observatory Services. *Astron. Comput.* **2014**, *78*, 3–11. [CrossRef]
91. Kartakis, S.; McCann, J.A. Real-time Edge Analytics for Cyber Physical Systems Using Compression Rates. In Proceedings of the International Conference on Autonomic Computing, Philadelphia, PA, USA, 18–20 June 2014; pp. 153–159.
92. Xu, L.; Wang, Z.; Chen, W. The Study and Evaluation of ARM based Mobile Virtualization. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 310308. [CrossRef]
93. Andrus, J.; Dall, C.; Hof, A.V.; Laadan, O.; Nieh, J. Cells: A Virtual Mobile Smartphone Architecture. In Proceedings of the ACM Symposium on Operating Systems Principles, Cascais, Portugal, 23–26 October 2011; pp. 173–187.
94. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. Localization approaches for internet of things. In *Optimal Localization of Internet of Things Nodes*; Springer: Cham, Switzerland, 2022; pp. 17–50.
95. Morabito, R.; Beijar, N. Enabling data processing at the network edge through lightweight virtualization technologies. In Proceedings of the 2016 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops), London, UK, 27 June 2016; pp. 1–6.
96. Barker, A.; Varghese, B.; Ward, J.S.; Sommerville, I. Academic Cloud Computing Research: Five Pitfalls and Five Opportunities. In Proceedings of the USENIX Conference on Hot Topics in Cloud Computing, Philadelphia, PA, USA, 17–18 June 2014.
97. Liu, Y.; Yang, C.; Jiang, L.; Xie, S.; Zhang, Y. Intelligent edge computing for IoT-based energy management in smart cities. *IEEE Netw.* **2019**, *33*, 111–117. [CrossRef]
98. Liang, H.; Hua, H.; Qin, Y.; Ye, M.; Zhang, S.; Cao, J. Stochastic optimal energy storage management for energy routers via compressive sensing. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2192–2202. [CrossRef]

99. Min, M.; Xiao, L.; Chen, Y.; Cheng, P.; Wu, D.; Zhuang, W. Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1930–1941. [CrossRef]
100. Cheng, X.; Feng, L.; Quan, W.; Zhou, C.; He, H.; Shi, W.; Shen, X. Space/aerial-assisted computing offloading for IoT applications: A learning-based approach. *IEEE J. Sel. Area. Comm.* **2019**, *37*, 1117–1129. [CrossRef]
101. Chen, X.; Zhang, H.; Wu, C.; Mao, S.; Ji, Y.; Bennis, M. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* **2019**, *6*, 4005–4018. [CrossRef]
102. Lei, L.; Xu, H.; Xiong, X.; Zheng, K.; Xiang, W.; Wang, X. Multiuser resource control with deep reinforcement learning in IoT edge computing. *IEEE Internet Things J.* **2019**, *6*, 10119–10133. [CrossRef]
103. Biswas, A.; Chandrakasan, A.P. CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks. *IEEE J. Solid-State Circ.* **2019**, *54*, 217–230. [CrossRef]
104. Lammie, C.; Olsen, A.; Carrick, T.; Azghadi, M.R. Low-power and high-speed deep FPGA inference engines for weed classification at the edge. *IEEE Access* **2019**, *7*, 51171–51184. [CrossRef]
105. Huang, L.; Feng, X.; Feng, A.; Huang, Y.; Qian, L. Distributed deep learning-based offloading for mobile edge computing networks. *Mobile Netw. Appl.* **2018**, *66*, 6353–6367. [CrossRef]
106. Hao, Y.; Mian, Y.; Hu, L.; Hossain, M.S.; Muhammad, G.; Amin, S.U. Smart-edge-coCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT. *IEEE Netw.* **2019**, *33*, 58–64. [CrossRef]
107. Xu, X.; Li, D.; Dai, Z.; Li, S.; Chen, X. A heuristic offloading method for deep learning edge services in 5G networks. *IEEE Access* **2019**, *7*, 67734–67744. [CrossRef]
108. Kiran, N.; Pan, C.; Wang, S.; Yin, C. Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks. *J. Commun. Netw.* **2020**, *22*, 1–11. [CrossRef]
109. Lei, L.; Xu, H.; Xiong, X.; Zheng, K.; Xiang, W. Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system. *IEEE Internet Things J.* **2019**, *6*, 5345–5362. [CrossRef]
110. Xu, J.; Chen, L.; Ren, S. Online learning for offloading and autoscaling in energy harvesting mobile edge computing. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 361–373. [CrossRef]
111. An, X.; Su, J.; Lu, X.; Lin, F. Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system. *J. Wirel. Comm. Netw.* **2018**, *1*, 249–258. [CrossRef]
112. Kozik, R.; Ficco, M.; Choraś, M.; Palmieri, F. A scalable distributed machine learning approach for attack detection in edge computing environments. *J. Parallel Distrib. Comput.* **2018**, *119*, 18–26. [CrossRef]
113. Abeshu, A.; Chilamkurti, N. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Commun. Mag.* **2018**, *56*, 169–175. [CrossRef]
114. Chen, Y.; Zhang, Y.; Maharjan, S.; Alam, M.; Wu, T. Deep learning for secure mobile edge computing in cyber-physical transportation systems. *IEEE Netw.* **2019**, *33*, 36–41. [CrossRef]
115. He, X.; Jin, R.; Dai, H. Deep PDS-learning for privacy-aware offloading in MEC-enabled IoT. *IEEE Internet Things J.* **2019**, *6*, 4547–4555. [CrossRef]
116. Wei, Y.; Yu, F.; Song, M.; Han, Z. Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning. *IEEE Internet Things J.* **2019**, *6*, 2061–2073. [CrossRef]
117. Wang, J.; Zhao, L.; Liu, J.; Kato, N. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach. *IEEE Trans. Emerg. Top. Comput.* **2019**, *9*, 1529–1541. [CrossRef]
118. Chen, J.; Chen, S.; Wang, Q.; Cao, B.; Feng, G.; Hu, J. iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks. *IEEE Internet Things J.* **2019**, *6*, 7011–7024. [CrossRef]
119. Du, M.; Wang, K.; Xia, Z.; Zhang, Y. Differential privacy preserving of training model in wireless big data with edge computing. *IEEE Trans. Big Data* **2020**, *6*, 283–295. [CrossRef]
120. Xu, C.; Ren, J.; She, L.; Zhang, Y.; Qin, Z.; Ren, K. EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics. *IEEE Internet Things J.* **2019**, *6*, 5140–5151. [CrossRef]
121. He, Y.; Yu, F.; He, Y.; Maharjan, S.; Zhang, Y. Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications. *IEEE Wirel. Commun.* **2019**, *25*, 103–109. [CrossRef]
122. Sun, Y.; Peng, M.; Mao, S. Deep reinforcement learning-based mode selection and resource management for green fog radio access networks. *IEEE Internet Things J.* **2019**, *6*, 1960–1971. [CrossRef]
123. Munir, M.S.; Abedin, S.F.; Tran, N.H.; Hong, C.S. When edge computing meets microgrid: A deep reinforcement learning approach. *IEEE Internet Things J.* **2019**, *6*, 7360–7374. [CrossRef]
124. Conti, S.; Faraci, G.; Nicolosi, R.; Rizzo, S.A.; Schembra, G. Battery management in a green fog-computing node: A reinforcement-learning approach. *IEEE Access* **2017**, *5*, 21126–21138. [CrossRef]
125. Wang, B.; Wu, Y.; Liu, K.R.; Clancy, T.C. An anti-jamming stochastic game for cognitive radio networks. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 877–889. [CrossRef]
126. Li, B.; Chen, T.; Giannakis, G.B. Secure mobile edge computing in IoT via collaborative online learning. *IEEE Trans. Signal Process.* **2019**, *67*, 5922–5935. [CrossRef]
127. Wang, Y.; Meng, W.; Li, W.; Liu, Z.; Liu, Y.; Xue, H. Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5101. [CrossRef]

128. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, 9–11 May 2016; pp. 1273–1282.
129. Li, E.; Zhou, Z.; Chen, X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In Proceedings of the Workshop Mobile Edge Communications (MECOMM@SIGCOMM), Budapest, Hungary, 20 August 2018; pp. 31–36.
130. Wang, J.; Zhang, J.; Bao, W.; Zhu, X.; Cao, B.; Yu, P.S. Not just privacy: Improving performance of private deep learning in mobile cloud. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2407–2416. [CrossRef]
131. Chen, M.; Challita, U.; Saad, W.; Yin, C.; Debbah, M. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3039–3071. [CrossRef]
132. Zhang, C.; Zhao, H.; Deng, S. A density-based offloading strategy for IoT devices in edge computing systems. *IEEE Access* **2018**, *6*, 73520–73530. [CrossRef]
133. Park, J.; Samarakoon, S.; Bennis, M.; Debbah, M. Wireless network intelligence at the edge. *arXiv* **2018**, arXiv:1812.02858. [CrossRef]
134. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
135. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
136. Lee, J.; Eshraghian, J.K.; Cho, K.; Eshraghian, K. Adaptive precision CNN accelerator using radix-X parallel connected memristor crossbars. *arXiv* **2019**, arXiv:1906.09395.