

EaaS: A Service-Oriented Edge Computing Framework Towards Distributed Intelligence

Mingjin Zhang, Jiannong Cao, *Fellow, IEEE*, Yuvraj Sahni, Qianyi Chen, Shan Jiang, Tao Wu
 Department of Computing, The Hong Kong Polytechnic University
 {csmzhang, csjcao, cssjiang}@comp.polyu.edu.hk

Abstract—Edge computing has become a popular paradigm where services and applications are deployed at the network edge closer to the data sources. It provides applications with outstanding benefits, including reduced response latency and enhanced privacy protection. For emerging advanced applications, such as autonomous vehicles, industrial IoT, and metaverse, further research is needed. This is because such applications demand ultra-low latency, hyper-connectivity, and dynamic and reliable service provision, while existing approaches are inadequate to address the new challenges. Hence, we envision that the future edge computing is moving towards distributed intelligence, where heterogeneous edge nodes collaborate to provide services in large-scale and geo-distributed edge infrastructure. We thereby propose Edge-as-a-Service (EaaS) to enable distributed intelligence. EaaS jointly manages large-scale cross-node edge resources and facilitates edge autonomy, edge-to-edge collaboration, and resource elasticity. These features enable flexible deployment of services and ubiquitous computation and intelligence. We first give an overview of existing edge computing studies and discuss their limitations to articulate the motivation for proposing EaaS. Then, we describe the details of EaaS, including the physical architecture, proposed software framework, and benefits of EaaS. Various application scenarios, such as real-time video surveillance, smart building, and metaverse, are presented to illustrate the significance and potential of EaaS. Finally, we discuss several challenging issues of EaaS to inspire more research towards this new edge computing framework.

Index Terms—Edge computing, edge as a service, service-oriented architecture, edge intelligence, edge-native applications.

I. INTRODUCTION

The cloud computing paradigm is a service provisioning model that provides user access to scalable computation, networking, and storage resources in the cloud data centers [1]. Cloud service providers provide flexibility and efficiency for end-users by providing services such as software as service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [2]. Cloud computing has tremendously changed how we live, work, and study and has become a mainstream business model for many organizations.

However, cloud computing shows significant disadvantages with the burgeoning of the Internet of Everything. According to Cisco, 80% of data will be generated by massive IoT devices at the network edge by 2025. Processing such large amounts of data in the cloud will lead to high latency and bandwidth costs, as well as privacy concerns. Recently, edge computing is emerging as a promising paradigm that processes data and provides service on edge devices closer to the data sources [3].

Edge computing brings intelligence to the network edge, even embedding it into IoT devices accessible by the end-users.

In the past few years, many research issues have been studied to deploy applications and services at the edge, such as computation offloading [4], content caching [5], and edge AI [6]. However, existing works are inadequate to meet the requirements of emerging advanced application, such as autonomous vehicles [7], industrial IoT [8], and metaverse [9]. This is because those applications demand ultra-low latency, hyper-connectivity, dynamic and reliable service provision, and the support of distributed and collaborative edge computing. More specifically, most existing works focus on edge-cloud collaboration to share data and computation resources while neglecting the collaboration among edge devices. They suffer from the unpredictable latency between the cloud and edge devices and incur privacy concerns. Moreover, they assume a hierarchical and centralized cloud-edge-end architecture, where the cloud dominates and manages the applications and resources of both edge and end devices. However, such a centralized architecture is with poor scalability in achieving large-scale deployment and resource elasticity.

To support those advanced applications and provide ubiquitous intelligence, we envision that with wider availability of edge computing infrastructures, future edge computing is moving towards collaborative distributed intelligence, where heterogeneous edge nodes collaborate with each other in large-scale and geo-distributed edge computing environments for various computation workloads, including collaborative task processing, distributed machine learning .etc.

To enable distributed intelligence, we propose a novel edge computing framework, namely Edge as a Service (EaaS), to facilitate edge autonomy, edge-to-edge collaboration, and large-scale application deployment and resource elasticity. Specifically, EaaS jointly manages the large-scale, geo-distributed, and heterogeneous edge infrastructures to boost the collaboration among edge devices, which enables the flexible and large-scale deployment of services, making computation and intelligence ubiquitous. EaaS also provides holistic and end-to-end services for developing and deploying applications at the edge, including IaaS, PaaS, and SaaS.

Some works also consider the idea of edge computing as a service. Davy et al. [10] proposed the concept of edge-as-a-service, which aims to leverage the network virtualization to facilitate the flexible usage of access network infrastructure and generate revenue for network operators. Varghese et

al. [11] presented an EaaS platform with a node discovery protocol to enable the selection and deployment of applications on edge. Differently, our work considers edge-as-a-service from a broader perspective. It is a cross-layer service-oriented computing framework integrating large-scale and geo-distributed resource management, edge-native task scheduling, various service modules, and application development throughout the infrastructure, platform, and application layers.

The main contributions of this work are as follows.

- We envision that the future edge computing is moving towards distributed intelligence, where edge nodes collaborate with each other to support various applications in large-scale, heterogeneous, and geo-distributed edge infrastructure.
- We propose EaaS, a novel service-oriented edge computing framework, to enable future edge computing by facilitating edge autonomy, edge-to-collaboration, and large-scale service deployment.
- Motivations, software framework, benefits, and potential application scenarios of EaaS are discussed.
- We present several open issues and challenges to inspire more research towards distributed intelligence empowered by edge computing.

The rest of the paper is organized as follows. Sec. II discusses existing edge computing research and the motivation for proposing EaaS. Sec. III introduces in detail about the software framework and the benefits of EaaS. In Sec. IV, we show the potential application scenarios of EaaS, including real-time video surveillance, smart building, autonomous vehicles, and metaverse. Sec. V discusses the open issues and challenges associated with EaaS. In Sec. VI, we conclude this work.

II. MOTIVATION FOR FUTURE EDGE COMPUTING

In this section, we first introduce the existing research and systems of edge computing and then describe the limitations of existing solutions to meet the requirements of future edge computing applications. Finally, we depict the vision and key features of future edge computing.

A. Introduction to Current Edge Computing

Edge computing has attracted lots of attention from both academics and industry in recent years. The different works in literature have studied several problems with respect to computation offloading, edge AI, designing applications for different domains, and developing edge computing systems.

Computation offloading, also known as task offloading, aims to offload the computation tasks from end devices to edge devices or the cloud. [4], [12], [13] offload computing-intensive tasks of end device to an edge server for processing with the objective of minimizing the task completion time. However, the edge device may easily get overloaded and lead to degraded performance considering the limited edge resources. It is beneficial to incorporate the cloud into the offloading decision. [14]–[16] jointly considers the computation resources and networking capacities of the cloud, edge devices, and user devices to explore the possible offloading

options. Nevertheless, the collaboration between cloud and edge devices suffers from unpredictable latency and causes privacy issues.

Another problem is edge AI. Due to the recent advances in deep learning, various AI applications are emerging to provide advanced analytical functions, such as object detection and tracking and human action recognition. The edge AI aims to facilitate the collaborative inference and training of AI models among cloud, edge, and end. Li et al. [17] model the execution delay and energy consumption of each layer of a neural network model to determine the partition point and distribute the neural network between end and edge, thus accelerating the DNN inference. Ding et al. [18] first used historical data to train the model in the cloud and then shared the shallow network layer EdgeCNN to the edge. Later, the edge used the newly collected real-time data to optimize the model to reduce transmission delay and improve user experience.

Apart from the existing research work, there are also some edge computing systems. Many start-ups and industry leaders in cloud computing, such as Microsoft, Amazon, and Huawei, have developed edge computing platforms to develop and deploy applications on resource-constraint devices, including AWS IoT Greengrass [19], Azure IoT Edge [20], and KubeEdge [21]. However, most of these solutions aim to extend the cloud computing functionalities. These solutions often follow the principles of service-oriented architecture to deploy cloud-based services on edge devices that are close to data sources. Table. I shows a comparison between the existing commercial and open-source edge computing systems. We can see that existing edge computing provides less support to edge autonomy and edge-to-edge collaboration, as they rely on centralized management from the cloud.

B. Increasing Demanding Application Requirements

Though achieving great success in the past few years, current edge computing is inadequate to serve the emerging advanced applications. Specifically, the dramatic development of information and communication technologies has brought striking revolutions to tremendous disciplines, and various new applications have emerged, including autonomous vehicles, industrial IoT, and metaverse. Those advanced applications are expected to be deployed at the network edge and pose great challenges to current edge computing. Those challenges are summarized as follows.

1) *Ultra-low latency*: To achieve safe autonomous vehicles and immersive interactive experience in metaverse, the edge devices are required to perform computation-intensive tasks in milliseconds, such as object detection and tracking, human action recognition, and 3D point cloud reconstruction. Existing works offloading the computation tasks to the cloud may suffer from the unpredictable latency between the cloud and the edge devices and leads to privacy concerns.

2) *Large-scale deployment with hyper-connectivity*: For industrial IoT and metaverse, tremendous end devices such as VR glasses, industrial controllers, and sensors are required to be interconnected for online interaction and data collecting.

TABLE I
COMPARISON OF EXISTING EDGE COMPUTING SYSTEMS.

System	Service-oriented	Programmable	Resource Elasticity	Edge Autonomy	Edge-to-edge Collaboration
IoT Greengrass	yes	yes	supported	supported	limited
Azure IoT Edge	yes	yes	supported	limited	limited
Cloud IoT Edge	yes	yes	supported	limited	limited
KubeEdge	yes	yes	supported	supported	limited
Baetyl	yes	yes	supported	supported	limited
OpenYurt	yes	yes	supported	supported	limited
Edge Foundry	yes	limited	supported	limited	limited
Edgent	yes	yes	supported	limited	limited

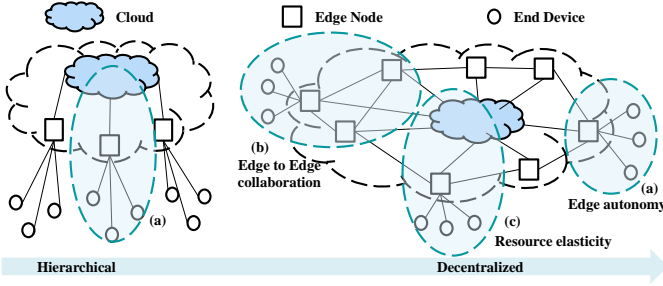


Fig. 1. Architecture of Future Edge Computing.

Existing edge computing approaches usually adopt centralized management, which is with limited scalability and fails to support the hyper-connectivity of end devices. Further, those end devices are usually located in different geographical areas and require collaboration among geo-distributed edge devices to provide large-scale service deployment.

3) *Dynamic and reliable service provision*: Many end devices are with high mobility, such as mobile phones and autonomous vehicles. They frequently pass through distinct geographical areas and demand consistent and reliable service provision. Existing works rely on the centralized cloud to synchronize the service status and make the service migration decisions, which is vulnerable to the dynamic network and is insufficient to provide reliable service provision. The dynamic network connection and service provision require collaboration among geo-distributed edge devices to share both data and computation resources.

C. Vision of Future Edge Computing

To better support emerging advanced new applications and provide ultra-low latency, reliable, and ubiquitous intelligent services, further research to enhance current edge computing is needed. Hence, we envision that the future edge computing is moving towards distributed intelligence, where numerous edge devices collaborate with each other in the large-scale, heterogeneous, and geo-distributed edge infrastructure.

There are three new key properties of future edge computing to enable distributed intelligence, i.e., edge-to-edge collaboration, edge autonomy, and resource elasticity. The three properties can be described as:

- **Edge autonomy**: It implies enabling offline and/or localized operations for reliable service provision without communication with the cloud.
- **Edge-to-edge collaboration**: It implies that each edge device can collaborate with its surrounding edge nodes to form a cluster that has the ability to provide real-time processing and computation-intensive services like AI algorithms. Furthermore, seamless and cross-platform computing can also be accomplished among edge nodes, and dynamic access can be subsequently supported.
- **Resource elasticity**: It refers to the ability of a system to support autonomous scheduling, including provisioning and de-provisioning, of different resources and services for a large-scale, distributed, and heterogeneous edge computing environment.

These three properties are essential to support the different requirements of future edge applications, including large-scale deployment, real-time processing, and dynamic access and reliable service provisioning. The existing solutions are insufficient to support these stringent requirements of future edge applications. Furthermore, The existing platforms currently lack in providing an integrated solution for future edge computing that includes providing infrastructure, platform, and software level services for the end-to-end development of edge computing applications. There is a need for a paradigm shift, as shown in Fig. 1, from the hierarchical architecture used in existing platforms to decentralized architecture for future edge applications. In future edge computing, the cloud still exists but will not play the dominant role in managing all the edge and end nodes.

III. EaaS FRAMEWORK

In this section, we give details about the EaaS framework, which aims to enable collaborative distributed intelligence supporting various advanced applications. We first give an overview of the EaaS software framework and then articulate the benefits of EaaS.

A. EaaS Framework

EaaS is designed to manage the large-scale, geo-distributed, and heterogeneous edge resources and support the collaboration among edge nodes for various emerging advanced applications. It mainly consists of three layers, i.e., infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), as shown in Fig. 2.

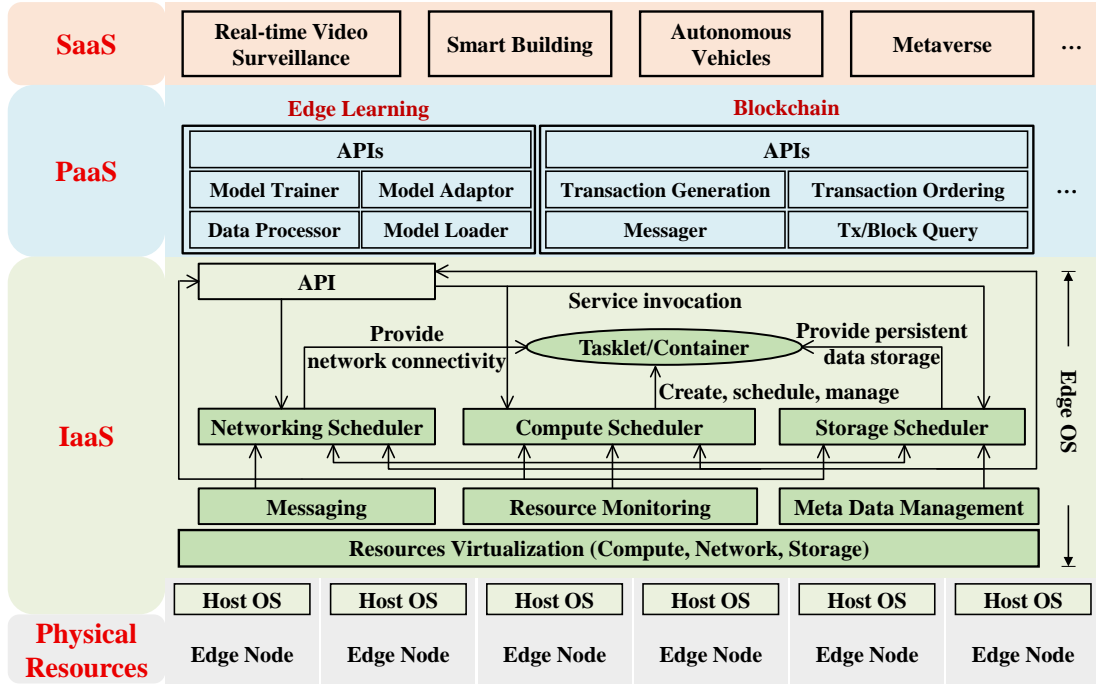


Fig. 2. Framework of Edge as a Service.

1) *Infrastructure as a Service*: In edge computing scenarios, there are various distinct edge nodes, such as edge servers, edge gateways, and intelligent vehicles, which are with heterogeneous computation, storage, and networking capabilities. Services and applications are deployed on those geo-distributed nodes connected with low-bandwidth and intermittent networks.

A fundamental problem of IaaS is to manage the large-scale, heterogeneous, and geo-distributed edge resources and schedule application workloads among edge nodes for efficient task execution. Different from the resource management and workload scheduling in the cloud environment, where cloud servers with abundant computation resources are connected with a high-bandwidth and stable network in a data center, resource management and workload scheduling in edge computing is more challenging.

First, edge-native applications are usually performance-aware, demanding high throughput and low latency. The workload scheduling in the cloud environment is mainly to ensure resource provision of workloads, such as the capacity of requested memory and CPU cores. It lacks support to meet the performance requirements of edge-native applications. Second, edge-native applications are with inner dependencies. Many intelligent edge applications are resource-greedy and complex, consisting of lots of inter-dependent components which are usually deployed to multiple edge nodes considering the constraint resource of a single node. However, the workload scheduling in a cloud environment usually schedules the application to a server with abundant resources and fails to consider the application's inner structure. Third, the data, computation, and networking resources are heterogeneous and coupled with

each other. Application deployed on heterogeneous edge nodes experiences distinct performance, and the coupled resources require joint orchestration. However, cloud resource management concentrates on orchestrating computation resources without jointly considering the data locality and networking resources, which may lead to underutilized resources and poor performance of workloads.

Hence, the new solutions of resource management and workload scheduling need to jointly consider the heterogeneous and coupled data, networking, and computation resources and make optimized scheduling decisions to meet the performance requirements of edge-native applications. In the infrastructure layer, we design a distributed edge computing operating system, namely EdgeOS, to manage the cross-node coupled edge resources and facilitate efficient workload scheduling among the geo-distributed edge nodes.

To manage the cross-node and coupled edge resources of computation, networking, and storage, EdgeOS adopts lightweight virtualization technology, i.e., containers, to abstract the heterogeneous resources and enable seamless service migration among geo-distributed edge nodes. Above the abstracted resources, we design three task schedulers, i.e., networking scheduler, compute scheduler, and storage scheduler, to manage the networking, compute, and storage resources, respectively. The three schedulers collaborated with each other to decide and maintain the resource allocation strategies for the incoming computation tasks. Moreover, EdgeOS provides APIs for the platform services to easily call the functions.

2) *Platform as a Service*: Above the edge infrastructures, many platform-level services, such as edge learning service and blockchain service, can be developed to support various

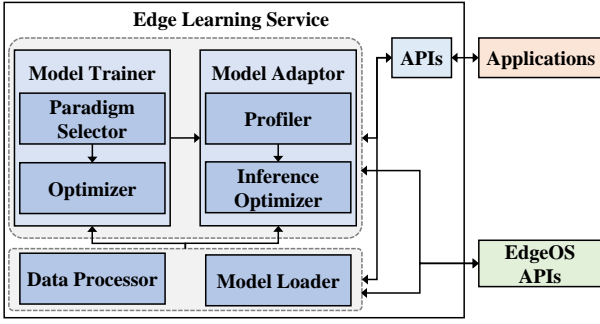


Fig. 3. Framework of Edge Learning Service.

applications.

One important service is edge learning service. Edge learning, also terms as edge AI, refers to the training and inference of AI models near the users at the network edge. Edge AI is essential to support applications requiring short communication delays and high response speed. There are many edge AI applications deployed at the network edge. Edge learning service is designed to support the whole life-cycle of AI applications, including data preparation, algorithm development, model training, and model inference. Edge learning service is in the platform layer. It provides APIs for easy and fast application development in the upper layer and calls the APIs provided by the infrastructure management layer to achieve efficient application deployment and resource utilization over heterogeneous and geo-distributed edge nodes.

The functional module design of the edge learning service is shown in Fig. 3. There are mainly four modules of edge learning service, namely, Data Processor, Model Loader, Model Trainer, and Model Adaptor. The data processor is responsible for data preprocessing, including data sampling, extracting, and transformation. It also supports data self-labeling and human labeling. The model loader supports algorithm development for users. It provides various programming libraries for AI model development and light-weighted pre-trained models, such as Resnet-50 and AlexNet. The model trainer is designed for conducting distributed model training. It provides multiple training paradigms, including federated learning, gossip learning [22], and E-tree learning [23]. Further, an optimizer is designed to support automatic parameters optimization of distributed model training, such as batch size, learning rate, and model aggregation frequency. With intelligent optimization algorithms, the distributed model training can adapt to the dynamic resources of the edge environments and achieve improved accuracy and convergence speed. The model adaptor is mainly responsible for optimizing the model inference strategies. It adopts various optimization methods, including model compression [24], model partition [25], and model early exit [26], to enable resource-aware model inference to meet the applications' requirements, such as latency, energy, and fault tolerance.

We take the model inference as an example to show the workflow of the edge learning service. The model adaptor

takes the model inference request as input and generates the execution strategies. Specifically, the inference request defines the data sources, the AI models, and the performance requirements, such as throughput and latency. The profiler first profiles the execution time of AI models on heterogeneous edge nodes and generates the profile information. The profile information and the underlying resource status accessed from EdgeOS APIs, as well as the inference optimization techniques, including model compression, partition, and early exit, will be jointly considered to decide the inference strategies. The inference strategies define how to compress the model, or how to partition the model, and where to allocate the inference tasks so as to meet the performance requirements. The strategies will be executed by the EdgeOS to achieve efficient model inference.

Edge blockchain as a service (EBaaS) is another essential service besides edge learning [27]. In EBaaS, the blockchain components are deployed at edge devices near the users and provided as easy-to-use services [28]. In this manner, the blockchain services can be delivered to the users with great convenience and low latency. The ultimate goals of EBaaS are delivering blockchain services with high modularity, flexibility, scalability, reliability, and security [29].

The current development of blockchain technology needs to go through two stages towards EBaaS: service-oriented and edge deployment. In the current stage, the companies are developing blockchain applications and hosting blockchain platforms by themselves [30], [31]. They analyze the specific requirements of different applications, e.g., privacy and efficiency [32], design variant blockchain solutions, develop heterogeneous blockchain components, and deploy the blockchain applications on separated hardware. Although the applications share similar requirements and blockchain components, the companies rarely considered integrating the different applications in a unified blockchain infrastructure, e.g., cloud and edge computing environments. Note that there are some popular blockchain platforms, e.g., Bitcoin [33], Ethereum [34], and Hyperledger Fabric [35]. However, they are provided as blockchain solutions rather than services. The users still must adapt them for different applications and host the blockchain components.

More recently, there is emerging research on blockchain as a service, aiming at developing a service-oriented blockchain infrastructure that can be used conveniently [29], [36]–[39]. For example, Jiang et al. developed PolyChain, a generic blockchain as a service platform [29]. They decompose a logical blockchain node into four components: application, consensus, storage, and network. Then, the components of a logical blockchain are deployed on different physical machines considering the resource demand of components and supply from machines. The techniques are named component modularization, distributed deployment, and resource optimization. With blockchain as a service, a generic blockchain infrastructure can be set, on which numerous applications can be deployed [40].

Traditionally, the blockchain as a service infrastructure is

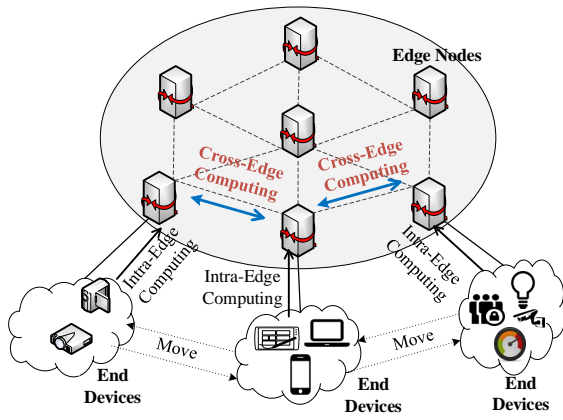


Fig. 4. Hybrid Architecture of EaaS.

deployed on the cloud, and the blockchain services become cloud-based [41]. However, the cloud is far from the end-users, limiting the application performance [42]. With the technological development of edge computing, EBaaS comes out to supply blockchain services near the end-users. We believe EBaaS will be favorable and trendy to support the applications demanding high security, privacy preservation, great flexibility, and low latency, which are not feasible currently.

3) *Software as a Service*: SaaS is defined as an application hosted by cloud servers to offer specific services such as email (Gmail) and storage (Dropbox) via a subscription-based licensing model. EaaS enables SaaS providers to deliver faster, privacy-preserving services for their customers by leveraging the outstanding benefits of edge computing. However, EaaS also poses new challenges to the way that SaaS providers develop their applications and provide services.

First, the application should adopt a modular and stateless design, which decomposes the functions of the applications into a set of dependent modules. Considering the limited resources of a single edge node, deploying the whole application on an edge node may be infeasible. Hence, the application is required to be partitioned and deployed on multiple edge nodes. Dataflow programming model [43] could be one option to enable efficient modular application design. Further, the frequent application state migration may lead to network congestion and degrade the applications. Stateless programming is beneficial in developing SaaS applications.

Second, the application development is dependent on the underlying PaaS and IaaS. Specifically, edge nodes are usually heterogeneous with different architectures, such as x86, ARM. Some edge nodes may be equipped with TPU, FPGA. Considering all the hardware characteristics may be challenging for developers, while the platform services may abstract the heterogeneity and provide unified programming interfaces. Similarly, many applications incorporate 5G capability into their functions, which requires the networking programming abstractions from the IaaS.

B. Benefits of the EaaS

Compared with existing edge computing frameworks, EaaS has outstanding benefits and significance as follows.

Large-scale connection and deployment. Current edge computing usually adopts the hierarchical cloud-edge-end architecture. It is a centralized architecture with the cloud to manage the whole system, which suffers from the limited scalability while many applications require large-scale deployment. To achieve the goal of pervasive intelligence, EaaS adopts a hybrid architecture based on our previous proposed Edge Mesh framework [44], integrating both centralized and decentralized resource management manner. As shown in Fig. 4, the edge node manages the end devices in a centralized manner, while edge nodes collaborate with each other in a decentralized manner considering the unpredictable latency between cloud and edge nodes. The hybrid architecture enables large-scale connections between edge nodes and end devices and among edge nodes.

Coupled resource management. In collaborative edge computing environments, data locate on geo-distributed and heterogeneous edge nodes. Considering the limited resources of a single edge node, the computation-intensive applications are more likely to be partitioned and deployed in distributed edge nodes, which may lead to frequent communication through a low-bandwidth network. Hence, the data, computation, and networking resources are coupled with each other and need joint management and orchestration [45], [46]. This is beneficial to improve resource utilization and meet the performance requirements of edge-native applications. Most previous works only consider orchestrating computation resources but neglect the resource heterogeneity and network resources, such as bandwidth allocation and customized routing of data flows. We envision that future edge computing is required to jointly orchestrate the heterogeneous computation and networking resources with the consideration of the data locality.

Edge-native application development and deployment. Different from cloud-native applications, edge-native applications are usually performance-sensitive and with inner structures, where dependent application modules are deployed on multiple edge nodes due to the limited resources of a single edge node. Further, edge nodes are with heterogeneous computation resources. Existing approaches concentrate on the development of applications while neglecting the deployment issue of applications [47]. They focus less on resource heterogeneity, locality of source data, and inter-communications of edge nodes. Those factors affect a lot of the performance of edge-native applications. Instead, EaaS supports end-to-end support of the application development and deployment. First, EaaS provides universal programming abstractions that enables decomposing the application and explicitly declaring the dependencies of the inner modules. Second, EaaS provides coupled resource management, enabling application profiling on heterogeneous edge nodes and scheduling application workloads for improved distributed execution by jointly considering the coupled data, computation, and networking resources.

Flexible service deployment. The traditional cloud-edge-end three-tier architecture is extensively studied in the literature. However, this hierarchical framework provides less support for edge autonomy, flexible service access, and edge-edge collaboration. In the hierarchical framework, the decision-making for task offloading and data transmission are made on the cloud, which monitors and manages the edge and end nodes. When there is an unstable network connection between cloud and edge, the system shows apparent performance degradation and even cannot work properly. Further, for mobile devices, which usually move among geographical areas, the edge nodes may not be able to provide seamless services. Moreover, the existing framework adopts cloud-edge collaboration, which suffers from the unpredictable latency between cloud and edge and may cause privacy issues due to cloud-edge data transmission. Instead, EaaS provides both cloud-edge and edge-edge collaboration to share both data and computation resources, which enables flexible service access and lower latency. When the cloud is unavailable, the edge nodes can make decisions autonomously by collaborating with other edge nodes. EaaS achieves pervasive and ubiquitous intelligence through flexible server deployment and assessment.

IV. APPLICATION SCENARIOS OF EaaS

This section discusses the promising applications of EaaS, especially the emerging ones.

A. Real-time Video Surveillance

Real-time video surveillance plays a crucial role in various areas in a smart city, including automatic pilot, smart transportation, and public security monitoring. Enormous AI-based video surveillance algorithms have been developed for object tracking, object re-identification, action identification, risk behavior detection, and many more. Edge computing is taken as the promising computing diagram since it can support low-latency, proactive, and secure video surveillance by providing computation near cameras. Most existing edge-based solutions utilize edge-cloud collaboration [48], [49] for the offloading of computation-intensive tasks, such as pedestrian re-identification and risky behavior identification. However, the use of a remote cloud server will degrade the latency and data security. Some studies focus on edge-to-edge cooperation for providing surveillance services jointly [50], [51]. However, they didn't schedule application workloads by jointly considering the coupled resources, such as locality of cameras (data sources), resource heterogeneity and communication overhead among geo-distributed edge nodes.

Our proposed EaaS represents a promising way of bringing together the strength of the aforementioned edge-based solutions, as it supports both edge-to-edge collaboration and resource elasticity while considering the coupled resources. It provides the flexibility to invoke nearby edge nodes and remote cloud servers efficiently for adapting to the ever-change computation requirements in various video surveillance scenarios. For instance, edge-to-edge collaboration is

beneficial for proactive pedestrian re-identification in medium-range areas such as university campuses, compared with edge-cloud collaboration. This is because data (such as pedestrian features) and computation resources sharing among nearby edge devices provides lower latency for reconstructing the pedestrian's trajectory. Instead, suppose the application is to perform pedestrian re-identification on a city scale. Offloading the re-identification task to the cloud is preferable as searching for suspects from numerous surveillance cameras in a city is computation-intensive and requires significant storage space.

B. Smart Building

As a crucial AIoT application, smart building plays a fundamental role in the safe and efficient production of society. It promises the safe and efficient operation of engineering structures, including bridges, high-rise buildings, tunnels, railways, power plants, etc. A smart building consists of multiple subsystems such as structural health monitoring system, drainage condition monitoring system, and power monitoring system. The conditions of each subsystem are identified from the measurement data collected by sensors, including accelerometers, strain gauges, cameras, displacement sensors, GPS, temperature sensors, anemometers, etc [52]. A cloud-based smart building system, as the mainstream solution, can support elastic computation and data storage resources for smart building applications. But they are inapplicable for data-sensitive and latency-sensitive smart building applications. Also, the collaboration between subsystems is dragged down by the centralized cloud server. Therefore, edge computing was introduced to smart building and has been extensively studied recently. Nevertheless, most existing studies focus on optimizing smart building algorithms for edge devices [53], [54]. Works to optimize the edge computing system are very limited. Our proposed EaaS can work as the underlying system for the dedicated smart building applications designed for distributed edge devices. Specifically, EaaS enables efficient communication between subsystems in a smart building through edge-to-edge collaboration. Also, it supports efficient execution of distributed data processing algorithms such as modal analysis [55], multivariate correlation analysis [56], and anomaly detection [57] among multiple edge devices.

C. Autonomous Vehicles

An autonomous vehicle is a vehicle that can sense its surroundings and operate without human intervention. The autonomous vehicles tightly integrated many technologies, including sensing, localization, perception, decision-making, and mechanical control [7]. The primary task of autonomous vehicles is to realize automated decision-making of the running paths, which includes AI computation tasks such as identification and tracking of vehicles and pedestrians, trajectory prediction, and path planning. The decision-making of running path task has demanding requirements of the execution latency, usually less than 5ms. However, due to the limited on-board computation power of a single vehicle, it is essential to offload the running path planning task to other computation

devices. Recently, the vehicular edge computing network has been proposed to enable communication and resource sharing among autonomous vehicles and nearby edge nodes, such as base stations and road-side units [58]. Most existing work considers offloading the computation task to a nearby base station or a road-side unit. However, the connection between vehicles and base stations maybe not always available, and the base station may get overloaded if all tasks are offloaded to a base station [59], [60]. Instead, EaaS jointly manage the resources of base stations, road-side units, and vehicles and enable efficient task offloading among all available edge nodes. It leads to better resource utilization and reduced service latency and is more suitable in unstable network connection.

D. Metaverse

Metaverse is a digital 3D world where people can work, shop, play, socialize, and do many more things they can do in real life. It is regarded as a promising future direction of the Internet and has attracted numerous investments from tech giants. The main objective of metaverse is realizing the physical experience for users in a digital world. To provide users with highly immersive experiences, a high-resolution, interactive, and low-latency digital world is desperately needed. Nevertheless, building such a digital world requires enormous computational and storage resources for the execution of physics calculation, rendering, and AI models [61]. Edge computing has been regarded as a crucial technique for promising the efficient execution of those applications due to its nature of low latency. But the limited computation resources in user's edge devices (such as VR/AR devices, mobile devices, and personal computers) hinder the development of edge-empowered metaverse [62], [63]. Moreover, the large-scale deployed edge devices are heterogeneous and might not be able to provide the same user experience for each people in the metaverse. EaaS provides an integrated solution for running metaverse applications near users. On the one hand, EaaS supports flexible service deployment. The computation workload can be taken by a single edge device in edge autonomy mode, shared with nearby MEC servers through edge-to-edge collaboration, or offloaded to remote cloud servers. The modes are changed automatically regarding the variation of workload, the resources in local edge devices, and available nearby edge resources. On the other hand, EaaS has incorporated multiple native edge learning services for efficient execution of AI models even with limited resources.

V. OPEN CHALLENGES

This section describes the open research challenges that have not been fully explored while designing and developing the services in EaaS.

A. Lightweight and Cross-platform Virtualization

Virtualization is essential for abstracting the heterogeneity of underlying infrastructure resources in developing EaaS. In the past few years, Container has become a popular approach for achieving lightweight virtualization at the edge devices

compared to the alternative approach of virtual machines (VM). However, there are still many issues that need to be resolved to meet the demanding requirements of emerging applications. First, there are few comprehensive solutions for achieving virtualization for embedded edge devices with limited memory, storage, and compute capacity. Docker, a popular container virtualization solution, has minimum requirements on the underlying resources of the host device, for example, at least 4GB RAM. Hence, it is not suitable for many resource-constraint embedded edge devices. One trend for virtualization in the future would be to develop lightweight virtualization solutions that can support resource-constraint embedded edge devices. The second issue is supporting cross-platform application deployment, including different processing architectures, i.e., x86, ARM, etc., different types of OS, i.e., GPOS and RTOS, and different processing units, i.e., CPU, GPU, TPU, etc. The current state-of-the-art Docker solution does not perform well for cross-platform application deployment consisting of heterogeneous devices with both Windows and Linux processing architectures and different processing units. Docker also does not officially support RTOS that is required for the deployment of applications with real-time requirements on embedded systems. The third issue is supporting multi-tenancy, i.e., support management and deployment of multiple application services simultaneously. There have been some solutions by Kubernetes to support multi-tenancy, but these solutions make some compromises on the security while deploying multiple application services. One future direction is balancing the trade-off between multi-tenancy and security.

B. Edge-native Resource Scheduling

Resource scheduling is a fundamental problem that has been widely studied in the context of edge computing and other related computing paradigms. Many works have studied different scheduling problems such as computation offloading, data caching, bandwidth allocation, energy scheduling, service migration, etc., under different system models and objective functions. However, these works have not fully explored the different characteristic features of EaaS, which makes the edge-native resource scheduling problem more challenging. One main characteristic feature of EaaS is the coupled resources that require considering dependency among resources to make joint scheduling decisions [64]. However, most existing solutions often provide a solution for scheduling a single resource. Novel cooperative scheduling solutions are required for the efficient management of coupled resources. Besides resource dependency, another characteristic feature of EaaS is data and resource uncertainty due to network dynamics and devices/link failures that makes the scheduling problem more complex. The scheduling solution has to be designed by modeling the reliability of devices and network links. There are few works that have considered reliability-aware resource scheduling for edge computing environments [65], [66]. However, these works do not focus on the joint task, network, and storage scheduling of coupled resources considering the reliability constraints. The third characteristic

feature of EaaS is a large-scale network with distributed resources that makes it difficult to schedule resources using a centralized controller efficiently. Most existing resource scheduling solutions for edge computing often assume global knowledge of resources that may not be practically feasible. Future works on resource scheduling should focus more on distributed scheduling approaches to enable scalable and high-performance applications.

C. Distributed Data Sharing

The different edge devices need to share and synchronize stateful data to collaborate efficiently with applications in connected healthcare [67] and smart logistics [68]. While the topic of distributed data sharing has been studied in the context of traditional parallel and distributed systems and cloud, It is much more challenging to enable it in the context of edge computing. First, the number of edge devices used for edge computing applications is at a much larger scale compared to the cloud. Furthermore, these devices are heterogeneous, resource-constraint, and highly distributed, which makes it difficult to design suitable scalable yet lightweight solutions. Second, the edge devices are often connected using an unstable wireless network that is dynamic and uncertain. The bandwidth across different links is also heterogeneous. Third, It is difficult to ensure consistency of stateful data as existing consensus algorithms are not suitable for highly distributed, large-scale, and dynamic edge computing environments. Novel lightweight consensus mechanisms are required that can ensure consistency for resource-constraint edge computing environments [69]. Furthermore, the new solutions should also consider addressing different consistency requirements as there could be multiple simultaneously deployed application services with different consistency requirements.

D. Resource-aware Edge AI

Edge AI has been identified as a key technology for 5G/6G and future networks by supporting training and deployment of AI models at edge devices close to data sources to enable several benefits, including low-latency processing, higher scalability, and better privacy. Several emerging IoT applications, such as autonomous vehicles, and metaverse, will be enabled by integrating EdgeAI with underlying 5G/6G networks. A characteristic feature of EdgeAI is that data required for training AI models are distributed at multiple edge devices. There are many challenging issues emerging from data properties that need to be resolved to support edge AI. The first issue is model inaccuracy due to the NonIID data generated on the edge devices. Existing edge AI approaches target learning a generalized model fit for all the edge nodes. In practice, data used for training a generalized model differs a lot from the NonIID data generated on an edge node, causing the model inaccuracy for inference on the node. The second issue is data labeling. The data generated on the edge nodes are mostly unlabeled due to high labeling costs. Existing approaches assuming well-labeled data samples cannot work effectively, and new model training methods need to be developed to

exploit massive unlabeled data. The third issue is the nature of IoT data which are usually streaming data, continuously arriving at the edge nodes. Existing approaches assuming that data samples are available at the start-up time are not suitable for Edge AI. New approaches are required to train an adaptive and lightweight model at resources constraint edge devices. Besides training AI models, there are also many issues in AI model inference as resource-constraint edge devices have limited compute, storage, and memory capacity to run complex AI models. Therefore, services need to be developed as part of EaaS to adapt the model in real-time by leveraging approaches, such as compression, partition, and early exit, while considering the underlying dependent resources.

E. Security and Privacy

The different features in edge computing that enable several benefits also lead to some security and privacy concerns. First, the different edge devices are often connected using a wireless network which leads to security and privacy concerns as malicious attackers can access the private data by eavesdropping. Second, as the edge devices are resource-constraint, it is difficult to deploy traditional security and privacy mechanisms that are compute-intensive. Third, the different edge devices can be under the control of multiple stakeholders unwilling to share data leading to privacy concerns. Fourth, the different geographically distributed edge devices are connected and collaborating with each other, which makes it challenging to ensure security and privacy as an attack at one part of the network can spread across other devices. Fifth, the edge devices are often heterogeneous, making it difficult to design generic security and privacy solution suitable for different application and deployment scenarios. Existing works in literature have also identified several new security and privacy issues emerging due to the unique characteristics in edge computing [70]–[72].

F. Programming Abstraction

One major issue to support end-to-end service development and deployment in EaaS is to provide programming interfaces that can abstract the underlying complexities for application developers. Application development and deployment at the edge is much more challenging compared to the cloud environment due to issues such as device/network heterogeneity, geo-distributed resources, network dynamics, resource uncertainty, and stateful data. Programming abstractions should be provided in the form of middleware to allow developers to focus on application logic rather than underlying edge platform issues such as resource discovery, monitoring, and management. Since the edge devices are characterized by limited resources, the middleware should be lightweight with low communication, computation, and storage overhead. The middleware should also be generic, modular, and flexible. The generic requirement implies that middleware should not be application-specific with limited primitives. Instead, it should support the development and deployment of different types of platform services and applications, including edge learning

services, blockchain services, and 5G/6G support, for different network and system models. The modular requirement implies that middleware is designed based on the principles of the service-oriented architecture to allow easy replacement of some functionalities and interfaces. Finally, the flexible requirement implies that middleware should be easily extensible to allow the addition of new functions and interfaces.

VI. CONCLUSION

Edge computing is a popular computing paradigm where the data processing and intelligence are performed at the network edge closer to the data sources. Many studies have been done in the past few years to deploy applications at the edge to reduce the service response time and bandwidth costs. However, existing approaches are inadequate to support emerging advanced applications demanding ultra-low latency, large-scale deployment with hyper-connectivity, and dynamic and reliable service provision. To address those challenges, we envision that the future of edge computing is moving towards collaborative distributed intelligence, where heterogeneous edge nodes collaborate with each other in a large-scale and geo-distributed edge computing environment. We hence propose EaaS to enable collaborative distributed intelligence by facilitating edge autonomy, edge-to-edge collaboration, large-scale cross-node and coupled resource management, and end-to-end edge-native application development and deployment. At last, we discussed several challenges and open issues to inspire more research toward collaborative distributed intelligence and finally make intelligence ubiquitous in the IoT world.

VII. ACKNOWLEDGEMENT

This work is supported by the Hong Kong RGC General Research Fund under Grant PolyU 15204921 and PolyU 15220922.

REFERENCES

- [1] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: An overview," in *IEEE International Conference on Cloud Computing*. Springer, 2009, pp. 626–631.
- [2] M. J. Kavis, *Architecting the cloud: design decisions for cloud computing service models (SaaS, PaaS, and IaaS)*. John Wiley & Sons, 2014.
- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [4] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *IEEE Wireless Communications and Networking Conference*. IEEE, 2017, pp. 1–6.
- [5] S. Safavat, N. N. Sapavath, and D. B. Rawat, "Recent advances in mobile edge computing and content caching," *Digital Communications and Networks*, vol. 6, no. 2, pp. 189–194, 2020.
- [6] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.
- [7] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019.
- [8] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.
- [9] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *The 29th ACM International Conference on Multimedia*, 2021, pp. 153–161.
- [10] S. Davy, J. Famaey, J. Serrat, J. L. Gorricho, A. Miron, M. Dramitinos, P. M. Neves, S. Latré, and E. Goshen, "Challenges to support edge-as-a-service," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 132–139, 2014.
- [11] B. Varghese, N. Wang, J. Li, and D. S. Nikolopoulos, "Edge-as-a-service: Towards distributed cloud architectures," *arXiv preprint arXiv:1710.10090*, 2017.
- [12] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [13] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2606–2615, 2017.
- [14] L. Chen, J. Wu, X. Long, and Z. Zhang, "Engine: Cost effective offloading in mobile edge computing with fog-cloud cooperation," *arXiv preprint arXiv:1711.01683*, 2017.
- [15] J. Meng, H. Tan, C. Xu, W. Cao, L. Liu, and B. Li, "Dedas: Online task dispatching and scheduling with bandwidth constraint in edge computing," in *IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2287–2295.
- [16] Z. Han, H. Tan, X.-Y. Li, S. H.-C. Jiang, Y. Li, and F. C. Lau, "Ondisc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2472–2485, 2019.
- [17] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2019.
- [18] C. Ding, A. Zhou, Y. Liu, R. Chang, C.-H. Hsu, and S. Wang, "A cloud-edge collaboration framework for cognitive service," *IEEE Transactions on Cloud Computing*, 2020.
- [19] A. Kurniawan, *Learning AWS IoT: Effectively manage connected devices on the AWS cloud using services such as AWS Greengrass, AWS button, predictive analytics and machine learning*. Packt Publishing Ltd, 2018.
- [20] D. Jensen, *Beginning Azure IoT Edge Computing: Extending the Cloud to the Intelligent Edge*. Apress, 2019.
- [21] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend cloud to edge with kubeedge," in *IEEE/ACM Symposium on Edge Computing*, 2018, pp. 373–377.
- [22] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *IFIP International Conference on Distributed Applications and Interoperable Systems*. Springer, 2019, pp. 74–90.
- [23] L. Yang, Y. Lu, J. Cao, J. Huang, and M. Zhang, "E-tree learning: A novel decentralized model learning framework for edge ai," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 290–11 304, 2021.
- [24] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems*, 2015, pp. 3123–3131.
- [25] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.
- [26] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
- [27] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: A blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.
- [28] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "Fogbus: A blockchain-based lightweight framework for edge and fog computing," *Journal of Systems and Software*, vol. 154, pp. 22–36, 2019.
- [29] S. Jiang, J. Cao, J. Zhu, and Y. Cao, "Polychain: A generic blockchain as a service platform," in *International Conference on Blockchain and Trustworthy Systems*. Springer, 2021, pp. 459–472.
- [30] S. Jiang, J. Cao, H. Wu, and Y. Yang, "Fairness-based packing of industrial iot data in permissioned blockchains," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7639–7649, 2020.

- [31] J. Singh and J. D. Michels, "Blockchain as a service (baas): Providers and trust," in *IEEE European Symposium on Security and Privacy Workshops*. IEEE, 2018, pp. 67–74.
- [32] S. Jiang, J. Cao, J. A. McCann, Y. Yang, Y. Liu, X. Wang, and Y. Deng, "Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain," in *IEEE International Conference on Blockchain*. IEEE, 2019, pp. 405–410.
- [33] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.
- [34] S. K. Kim, Z. Ma, S. Murali, J. Mason, A. Miller, and M. Bailey, "Measuring ethereum network peers," in *ACM Internet Measurement Conference*, 2018, pp. 91–104.
- [35] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *The Thirteenth EuroSys Conference*, 2018, pp. 1–15.
- [36] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li, and R. Chen, "Nutbaas: a blockchain-as-a-service platform," *IEEE Access*, vol. 7, pp. 134422–134433, 2019.
- [37] Q. Lu, X. Xu, Y. Liu, I. Weber, L. Zhu, and W. Zhang, "ubaas: A unified blockchain as a service platform," *Future Generation Computer Systems*, vol. 101, pp. 564–575, 2019.
- [38] H. Chen and L.-J. Zhang, "Fbaas: Functional blockchain as a service," in *International Conference on Blockchain*. Springer, 2018, pp. 243–250.
- [39] A. A. Mamun, F. Yan, and D. Zhao, "Baash: lightweight, efficient, and reliable blockchain-as-a-service for hpc systems," in *The International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–18.
- [40] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "Blocksdn: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Network*, vol. 34, no. 2, pp. 83–91, 2020.
- [41] N. Weerasinghe, T. Hewa, M. Liyanage, S. S. Kanhere, and M. Ylianttila, "A novel blockchain-as-a-service (baas) platform for local 5g operators," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 575–601, 2021.
- [42] W.-T. Tsai, X. Sun, and J. Balasooriya, "Service-oriented cloud computing architecture," in *The Seventh International Conference on Information Technology: New Generations*. IEEE, 2010, pp. 684–689.
- [43] W. M. Johnston, J. P. Hanna, and R. J. Millar, "Advances in dataflow programming languages," *ACM Computing Surveys*, vol. 36, no. 1, pp. 1–34, 2004.
- [44] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed intelligence in internet of things," *IEEE Access*, vol. 5, pp. 16441–16458, 2017.
- [45] Y. Sahni, J. Cao, and L. Yang, "Data-aware task allocation for achieving low latency in collaborative edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3512–3524, 2018.
- [46] Y. Sahni, J. Cao, L. Yang, and Y. Ji, "Multihop offloading of multiple dag tasks in collaborative edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4893–4905, 2020.
- [47] M. Satyanarayanan, G. Klas, M. Silva, and S. Mangiante, "The seminal role of edge-native applications," in *IEEE International Conference on Edge Computing*. IEEE, 2019, pp. 33–40.
- [48] Y. Zhao, Y. Yin, and G. Gui, "Lightweight deep learning based intelligent edge surveillance techniques," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1146–1154, 2020.
- [49] Y. Chen, T. Yang, C. Li, and Y. Zhang, "A binarized segmented resnet based on edge computing for re-identification," *Sensors*, vol. 20, no. 23, p. 6902, 2020.
- [50] X. Zeng, B. Fang, H. Shen, and M. Zhang, "Distream: scaling live video analytics with workload-adaptive distributed edge intelligence," in *The 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 409–421.
- [51] Z. Xu, S. Sinha, S. Harshil S, and U. Ramachandran, "Space-time vehicle tracking at the edge of the network," in *The Workshop on Hot Topics in Video Analytics and Intelligent Edges*, 2019, pp. 15–20.
- [52] Y. Sahni, J. Cao, and X. Liu, "Midshm: a middleware for wsn-based shm application using service-oriented architecture," *Future Generation Computer Systems*, vol. 80, pp. 263–274, 2018.
- [53] Q. Chen, J. Cao, and Y. Xia, "Physics-enhanced pca for data compression in edge devices," *IEEE Transactions on Green Communications and Networking*, 2022.
- [54] R.-T. Wu, A. Singla, M. R. Jahanshahi, E. Bertino, B. J. Ko, and D. Verma, "Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 9, pp. 774–789, 2019.
- [55] X. Liu, J. Cao, W.-Z. Song, P. Guo, and Z. He, "Distributed sensing for high-quality structural health monitoring using wsns," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 738–747, 2014.
- [56] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, "A review of deep reinforcement learning for smart building energy management," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12046–12063, 2021.
- [57] R. A. Sater and A. B. Hamza, "A federated learning approach to anomaly detection in smart buildings," *ACM Transactions on Internet of Things*, vol. 2, no. 4, pp. 1–23, 2021.
- [58] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Collaborative vehicular edge computing networks: Architecture design and research challenges," *IEEE Access*, vol. 7, pp. 178942–178952, 2019.
- [59] M. Cui, S. Zhong, B. Li, X. Chen, and K. Huang, "Offloading autonomous driving services via edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10535–10547, 2020.
- [60] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26652–26664, 2019.
- [61] S. Dhelim, T. Kechadi, L. Chen, N. Aung, H. Ning, and L. Atzori, "Edge-enabled metaverse: The convergence of metaverse and mobile edge computing," *arXiv preprint arXiv:2205.02764*, 2022.
- [62] M. Xu, D. Niyato, J. Kang, Z. Xiong, C. Miao, and D. I. Kim, "Wireless edge-empowered metaverse: A learning-based incentive mechanism for virtual reality," *arXiv preprint arXiv:2111.03776*, 2021.
- [63] W.-S. Kim, "Edge computing server deployment technique for cloud vr-based multi-user metaverse content," *Journal of Korea Multimedia Society*, vol. 24, no. 8, pp. 1090–1100, 2021.
- [64] Y. Sahni, J. Cao, L. Yang, and Y. Ji, "Multi-hop multi-task partial computation offloading in collaborative edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1133–1145, 2020.
- [65] J. Liu, A. Zhou, C. Liu, T. Zhang, L. Qi, S. Wang, and R. Buyya, "Reliability-enhanced task offloading in mobile edge computing environments," *IEEE Internet of Things Journal*, 2021.
- [66] Q. Peng, H. Jiang, M. Chen, J. Liang, and Y. Xia, "Reliability-aware and deadline-constrained workflow scheduling in mobile edge computing," in *IEEE International Conference on Networking, Sensing and Control*. IEEE, 2019, pp. 236–241.
- [67] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "Blochie: a blockchain-based platform for healthcare information exchange," in *IEEE International Conference on Smart Computing*. IEEE, 2018, pp. 49–56.
- [68] H. Wu, J. Cao, Y. Yang, C. L. Tung, S. Jiang, B. Tang, Y. Liu, X. Wang, and Y. Deng, "Data management in supply chain using blockchain: Challenges and a case study," in *The 28th International Conference on Computer Communication and Networks*. IEEE, 2019, pp. 1–8.
- [69] H. Wu, J. Cao, S. Jiang, R. Yang, Y. Yang, and J. Hey, "Tsar: a fully-distributed trustless data sharing platform," in *IEEE International Conference on Smart Computing*. IEEE, 2018, pp. 350–355.
- [70] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.
- [71] F. Al-Doghman, N. Moustafa, I. Khalil, Z. Tari, and A. Zomaya, "Ai-enabled secure microservices in edge computing: Opportunities and challenges," *IEEE Transactions on Services Computing*, 2022.
- [72] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.