Artificial Intelligence of Things: Architectures, Applications and Challenges

Ehsan Ahvar[*], Shohreh Ahvar[**], Gyu Myoung Lee[***]

[*] LDR research Lab, ESIEA Graduate Engineering School, Paris, France

[**] ISEP Graduate Engineering School, Paris, France

[***] Liverpool John Moores University, Liverpool, UK

Abstract-Internet of Things (IoT), with the help of some technologies such as 5G, Fog Computing and Artificial Intelligence (AI), is driving a significant shift to support emerging applications with specific requirements (e.g., real-time processing of a high volume of data) with greater flexibility and efficiency. In this regard, AI subfields such as data analytics and learning techniques play an important role to support the emerging IoT-related applications. AI integrated with the IoT heralds the era of Artificial Intelligence of Things (AIoT). While AI functions traditionally need centralized data collection and processing, it may not be feasible for modern AIoT-related applications because of some features such as the high volume of generated data, scalable and distributed data sources, high scalability of networks and data privacy concerns. In recent years, we can see noticeable efforts in the AIoT domain ranging from AI efficiency (e.g., proposing lightweight AI units) to system (e.g., proposing more powerful AIoT end devices) and architecture efficiency (e.g., proposing new AI deployment methods for AIoT architecture). This chapter presents recent progress in AIoT research domain from AI, system and architecture perspectives. We then introduce some recent and promising applications of AIoT. Finally, the important challenges facing AIoT as well as some potential research opportunities will be presented.

1. Introduction

The Internet of Things (IoT), a term originally introduced by Kevin Ashton [9], refers to a global intelligent network that provides cyber-physical interactions by connecting a large number of things (devices) with the ability to perceive, compute, execute, and communicate with the Internet [6].

It is expected that the number of IoT devices will be massively increased in the coming years. According to Cisco, IoT devices will account for 50 percent (i.e., 14.7 billion) of all global networked devices by 2023 [7].

To analyze the generated data from ubiquitous IoT devices and enabling intelligent IoT applications (e.g., smart healthcare, smart transportation), artificial intelligence (AI) techniques have been widely used [4]. The combination of AI technologies with the IoT infrastructure refers to Artificial Intelligence of Things (AIoT). AIoT can achieve more efficient IoT operations, improve human-machine interactions and enhance data management and analytics [10].

AI techniques traditionally need centralized data collection and processing. In traditional IoT architectures, data are generated by IoT devices and transmitted to a remote cloud where they are processed by the AI functions. While storing and analyzing data on centralized cloud brings some benefits such as cost efficiency and high computing and storage capabilities, this centralized architecture may be ineffective or infeasible for modern AIoT applications because of high volume of generated data, high scalability of their networks and data privacy issues. For example, it can be infeasible for real-time IoT systems to transmit the big data, generated by a high number of IoT devices, over complex environments to the cloud data centers for AI training [4] [5] [8].

The research in AI/IoT has been striving to find appropriate solutions for this problem. Edge computing is considered as an extension of cloud computing to move cloud services to the proximity of end users [14]. The combination of edge computing and AI defines a new term, called “edge intelligence” or simply “edge AI”. Rather than entirely relying on the cloud, edge AI makes the most of the widespread edge resources to gain AI insight [11].

Edge AI can solve the limitations of the cloud-based architecture when it comes to distributed data analytics. The main idea is to utilize the storage and computing power of edge servers and end devices to train the machine learning models closer to the source of the data [5] [12].

To enable the modern AIoT applications to meet their requirements such as latency and privacy, three main solutions have been proposed. 1) Considering the recent progress in hardware technology, one solution is to run AI functions on the IoT end devices. However, many recent AIoT applications utilize computation intensive algorithms requiring powerful processors. This may outweigh the capacity of even modern IoT devices [14]. 2) Edge server-based architecture is considered as another solution where data from the end devices are transferred to edge servers for processing. However, processing requirements for AI are sharply increasing. AI training executions are doubling every 3.5 month and, since 2012, the processing requirements have increased by more than 300,000 times [8][17]. While edge computing can often satisfy the latency and privacy, running high resource requirements AI functions (e.g., deep learning algorithms) on not very powerful edge compute resources can be sometimes an issue. 3) joint computation among end devices, edge servers, and the cloud is another idea. In this case, offloading a part of the processing workload intelligently from IoT devices to the fog nodes/edge servers and cloud can satisfy latency while leveraging AI capacities for processing a large volume of data [6] [13].

As a result, research efforts for AIoT can be mainly categorized to three following directions:

- Architecture: Proposing efficient architectures and AI deployment methods for AIoT;
- System: Designing more powerful IoT edge and end devices (e.g., hardware and software acceleration);
- AI: Designing light-weight AI functions to be used by resource-limited AIoT end devices;
- Implementation: Some tools and platforms have been proposed for implementing AIoT systems.

This chapter presents recent progress in AIoT research from the three above mentioned perspectives. We discuss the opportunities created by recent advances in IoT's hardware technology and AI related techniques such as Federated Learning (FL) to support AIoT applications with greater flexibility and efficiency. Finally, the recent deployment and applications as well as the important challenges facing AIoT and some potential research opportunities will be presented.

2. Architecture: Deployment methods of AI units in AIoT

For AIoT data analysis, we can see that there are three main players: data, processing (i.e., AI units) and features. Depending on applications, data can be centralized or distributed. In distributed datasets, depending on applications, we may consider a similar or different set of features for sub-datasets. Here, the main question is how to map AI processing units to better support various types of AIoT applications considering their data and feature distribution methods.

In this part, we present potential scenarios for deployment of AI units in the AIoT ecosystem. We then discuss which scenario(s) can be more appropriate for AIoT related applications. As machine learning (including neural network and deep learning) is considered as one of the main subsets of AI, we focus on possible methods to deploy training and inference units. However, the scenarios can be easily modified for other AI subdomains.

As Figure 1 shows, deployment methods of AI units in AIoT, depending on training and inference location, can be generally divided into 2 groups: centralized and distributed. In the centralized scenarios, AI is completely deployed on one AIoT node (i.e., only on end device or on edge device or on cloud server). Distributed AI deployment methods distribute AI units in different locations.

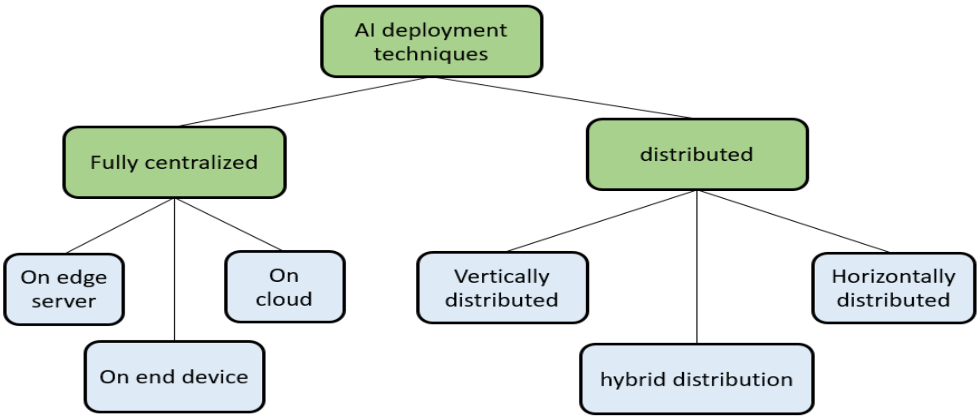


Figure 1. A taxonomy of AI deployment techniques

Fully centralized AI deployment

In traditional IoT architectures, AI units are deployed in the centralized cloud and AIoT devices generate and transmit data to the remote cloud (see Figure 2(a)). Analyzing data on a centralized cloud has some benefits such as cost efficiency and high computing and storage capabilities.

Challenge - The majority of modern AIoT applications generate high volume of data using distributed AIoT devices. Migrating the big data, generated by a high number of IoT devices, over complex environments to the cloud data centers for processing is considered as a big concern in aspects of privacy, latency and bandwidth cost for many AIoT applications [4] [5] [8].

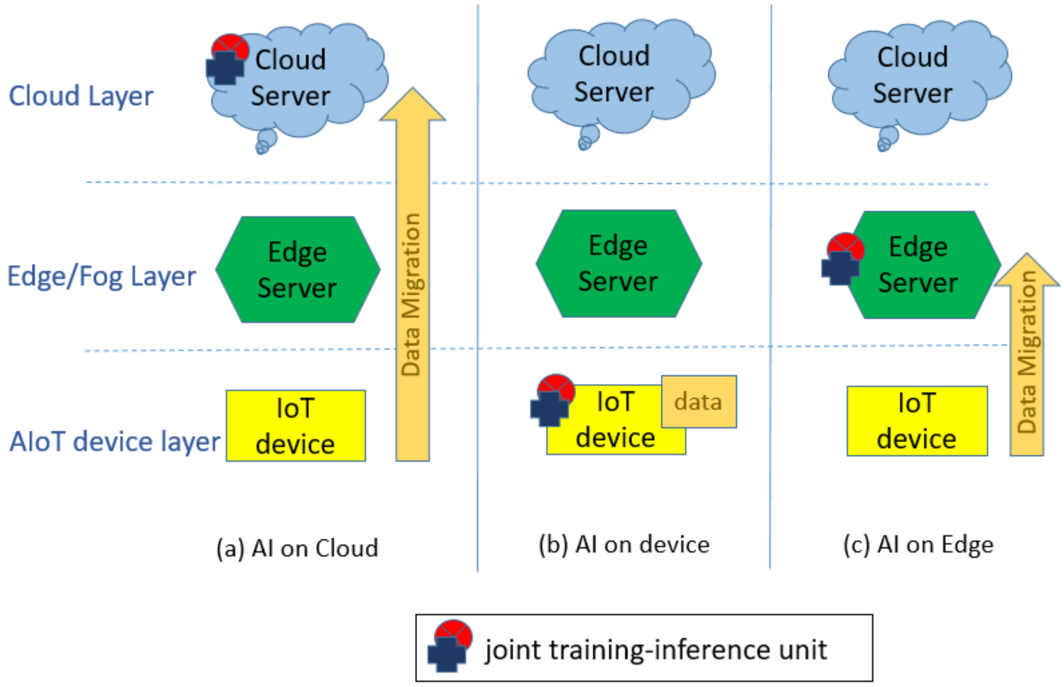


Figure 2. Centralized AI deployment techniques

One solution is to run AI functions, including both training and inference, on each IoT end device (see Figure 2(b)). In recent years, AI-specific hardware accelerators have enhanced high-performance processing in embedded and resource-limited AIoT devices. Embedded ML is a new domain where AI-based sensor data analytics is proceeded near to the data source in real time [8].

Challenge - we should consider this point that many recent AIoT applications utilize computation intensive algorithms requiring powerful processors. This may still outweigh the capacity of even modern IoT devices [14].

One straightforward idea is to deploy the AI units only on edge servers (see Figure 2(c)). In this case, the end device transfers its data to a nearby edge server and receives the corresponding results after server processing [13]. As a result, in comparison to deploying AI units on Cloud, the transmission latency of data offloading and required bandwidth reduced and the data privacy increased.

Distributed AI deployment techniques

As we saw in the previous section, each fully centralized AI deployment technique has its related challenges. While deployment of an AI unit only on cloud has some challenges (e.g., privacy and network latency), deployment of AI units on end or edge devices raises other concerns such as limited computation capacity.

Distributed AI deployment techniques, unlike the fully centralized techniques that place AI units on only one end, edge or cloud element, has been proposed to distribute and place AI units on various locations. Existing techniques can be divided into three main categories: vertical, horizontal and hybrid AI distribution techniques.

As shown in Figure 3, while vertical techniques deploy AI units vertically on AIoT ecosystem (i.e., a combination of an end device, edge server and cloud server), horizontally deployment techniques mainly focus on extending number of nodes (e.g., end device-to-end device collaboration).

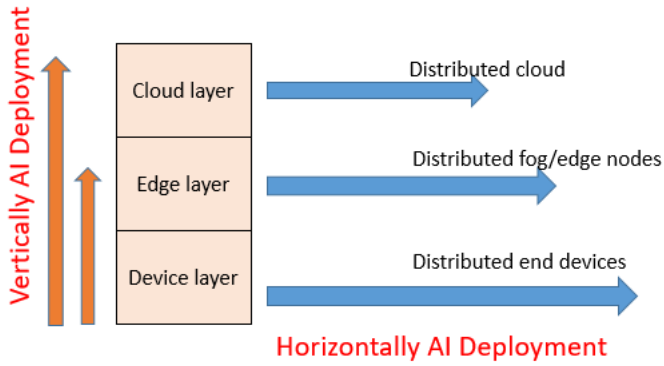


Figure 3. Vertical and horizontal AI distribution

Finally, hybrid techniques try to get benefits from both vertical and horizontal deployment of AI units for AIoT applications.

Vertically distributed AI deployment

As already mentioned, inference and training are the two main computing parts in machine learning and deep learning. In vertically distributed AI deployment, unlike fully centralized one, training and inference units are not deployed on the same machine. In fact, recent studies show that executing AI models with edge-cloud synergy can improve both the end-to-end latency and energy consumption in comparison to the local execution method [11].

In an AIoT system, it is common to deploy and distribute AI units vertically on cloud nodes, fog/edge nodes, and end devices to increase AIoT’s performance. By intelligently distributing AI units, it is possible to leverage machine learning capacities for big data with satisfying latency [6].

Vertically distributed AI deployment methods can be divided into two main groups:

- the methods that do not support co-processing (i.e., co-inference and co-training);
- the methods that support co-processing.

Figure 4 shows potential vertically distributed AI deployment techniques that do not support any co-inference and co-training features.

While the training part is typically processing intensive and needs the availability of high volume of data, the inference can be processed with less processing power [21]. For this reason, it is usually considered that the training part should be deployed on the cloud (see figures 4 (a) and (b)) and, therefore, we can see that many studies focus on the inference part to let cloud-scale models execute on resource-limited devices. However, deploying a training unit on the cloud also means that the large volume of training data should be sent from end or edge devices to the cloud. It can significantly increase communication overhead and the data privacy concern [11] [24].

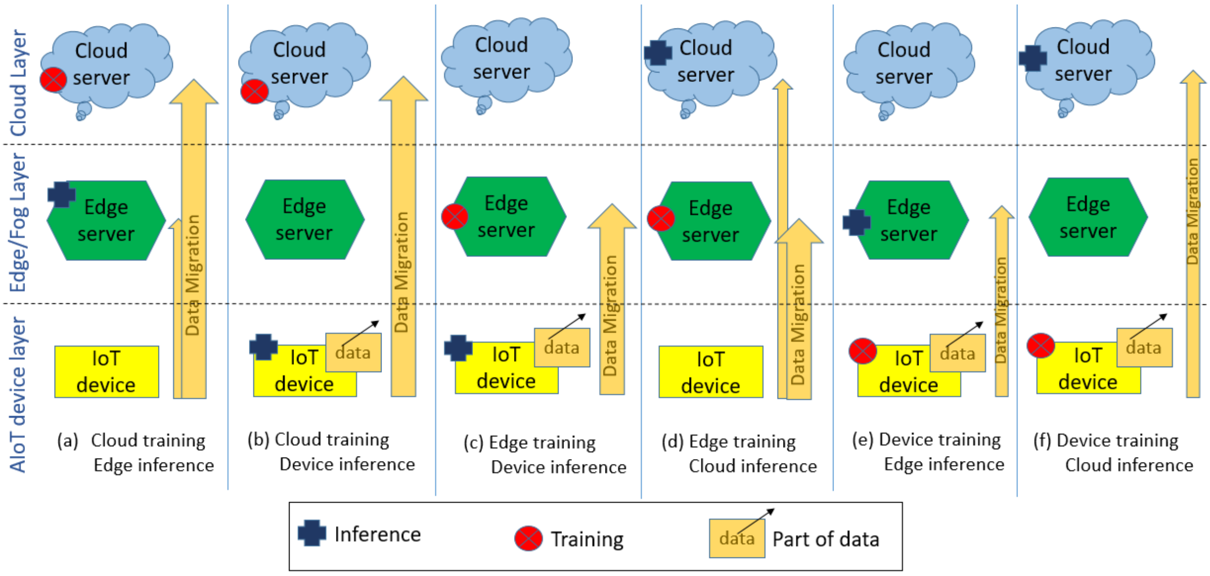


Figure 4. Potential vertically distributed AI deployment techniques

If data size to process is not very large and the end device and edge device (or at least one of them) have acceptable processing capabilities, training models on an edge device or even an end device can be sometimes considered as an alternative (Figures 4 (c)-(f)).

In addition, co-processing (i.e., co-training and co-inference) is considered as one solution to address, at the same time, challenges of AI processing on end/edge devices (e.g., resource limitation) as well as cloud (e.g., bandwidth cost, latency, privacy). In co-processing, instead of training or inference on only one node, two or more nodes can cooperate. Figure 5 highlights centralized and vertically distributed AI deployment characteristics. While in centralized method, Figure 5 (a), both Training Unit (TU) and Inference Unit (IU) are deployed on the same machine (i.e., on end device or edge device or cloud), TU and IU in a vertically distributed method are separately deployed on different vertical locations, Figure 5 (b) (i.e., on end device-edge device or end device-cloud or edge device-cloud).

In the vertically distributed AI deployment methods that support co-processing (e.g., co-inference or co-training), every TU can be divided into two or three sub-TUs where the sub-TUs can be deployed on different places (i.e., end device or edge device or cloud) and cooperate (see Figure 5 (c) and (d)).

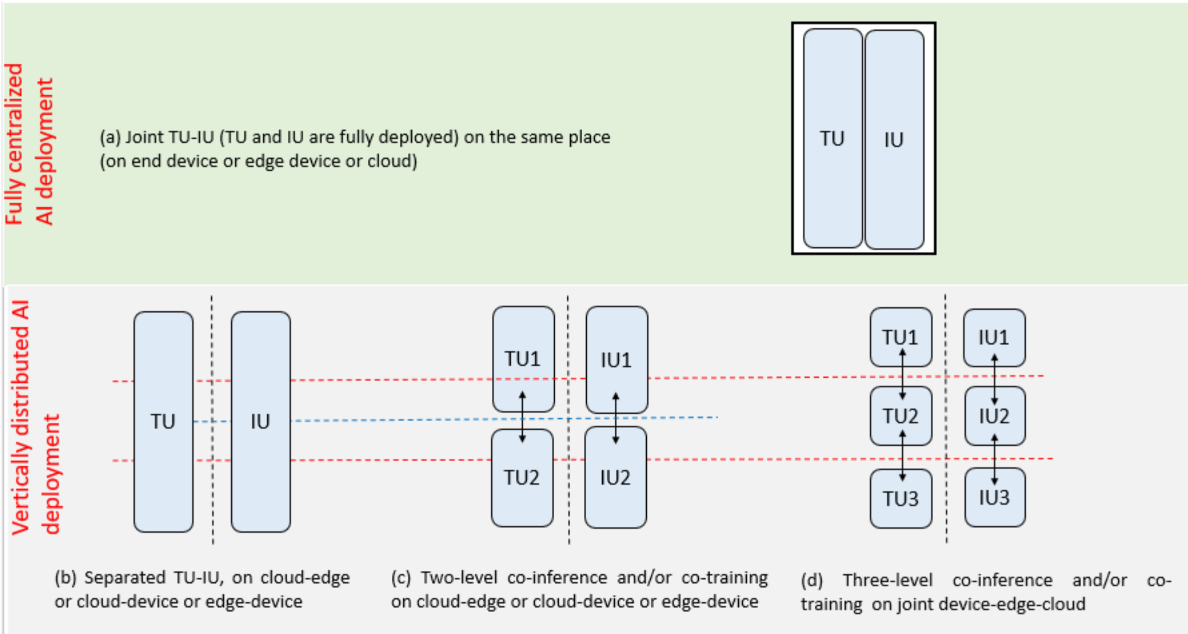


Figure 5. Centralized and vertically distributed AI deployment characteristics

Model parallelism is a technique that can be used for supporting co-processing. It splits a model into several partitions where every partition can be processed on a separate device. Model partitioning can be either vertical (i.e., by applying splits between neural network layers) or horizontal (i.e., by splitting the layers) [26].

Horizontally distributed and hybrid AI deployment

A lot of AIoT infrastructures consist of a set of geographically distributed end devices. The geographically distributed AIoT end devices may generate a large volume of distributed data.

On the other hand, classical AI methods (i.e., machine learning) have been considered to work with centralized data. In this case, in order to process distributed data, we usually need to collect them in a database. However, it can be usually either ineffective or infeasible for the following reasons [22].

- The cost of maintaining (i.e., storing) a central large-size data set can be much more than the total cost of maintaining its smaller sub-data sets.
- The processing cost of a central database is much more than the total cost of processing its smaller sub-data sets.
- Transferring a large amount of data over the network has some noticeable challenges such as privacy and transferring time.
- Because of the data security and privacy issues, legal restrictions (e.g., Cybersecurity Law of the People's Republic (CLPR) of China, General Data Protection Regulation (GDPR) in European Union, Personal Data Protection Act (PDP) in Singapore, California Consumer Privacy Act (CCPA) and Consumer Privacy Bill of Rights (CPBR) in the United States) have been defined and put in practice which makes data aggregation from distributed devices, multiple regions, or organizations, almost impossible [25].

Putting all pieces together, designing AI deployment methods to support horizontally distributed data in an effective and efficient way is necessary. Here, we present horizontally distributed AI deployment techniques to process the distributed data. We will show that some studies proposed hybrid AI deployment methods.

Before talking about horizontally distributed and hybrid AI deployment techniques to support the distributed data, it is necessary to briefly review the following related topics:

- Data (and feature) distribution methods;
- Training cycles;
- Distributed machine learning topologies.

Data distribution methods

As Figure 6 shows, we can define several common methods of data distribution or fragmentation [22]:

- Horizontal distribution in which sub-data sets (i.e., subparts of instances) are stored at different locations. Here, an instance includes the values of several attributes;
- Vertical distribution in which subparts of attributes or features of instances are stored at different locations;
- Hybrid (mixed) distribution in which every sub-data set and its related features are stored at different locations.

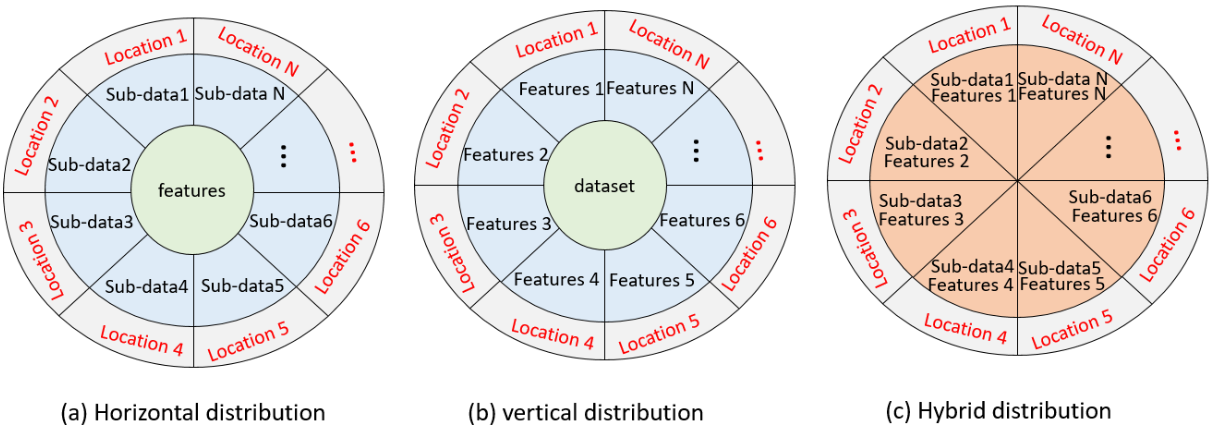
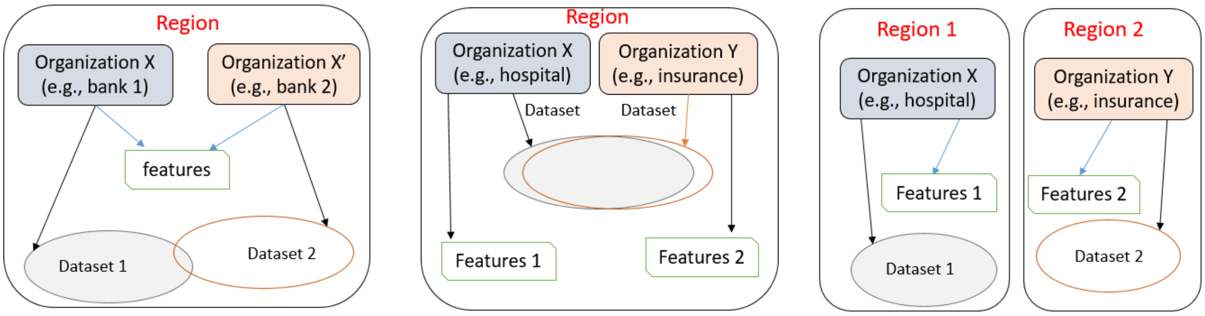


Figure 6. Common methods of data and feature distribution

Datasets are often distributed horizontally as it constitutes the most appropriate and natural way for many applications. As an example, two regional banks can have different groups of clients from their respective locations and the intersection set of their clients can be very small (see Figure 7(a)). However, they can have similar feature spaces as their business is quite similar [23].



(a) Horizontal distribution example: Similar organizations, same region, similar features, almost different clients

(b) Vertical distribution example: different organizations, same region different features, almost similar clients

(c) Hybrid distribution example: different organizations, different regions, different features, different clients

Figure 7. Data and feature distribution methods examples

On the other hand, as Figure 7(b) shows, two different regional organizations that have almost similar clients. However, the features can be different. If these two different organizations are located in two different regions, their clients can be also different (Figure 7(c)). In this case, considering our application, we may need to process a set of sub-datasets with different features.

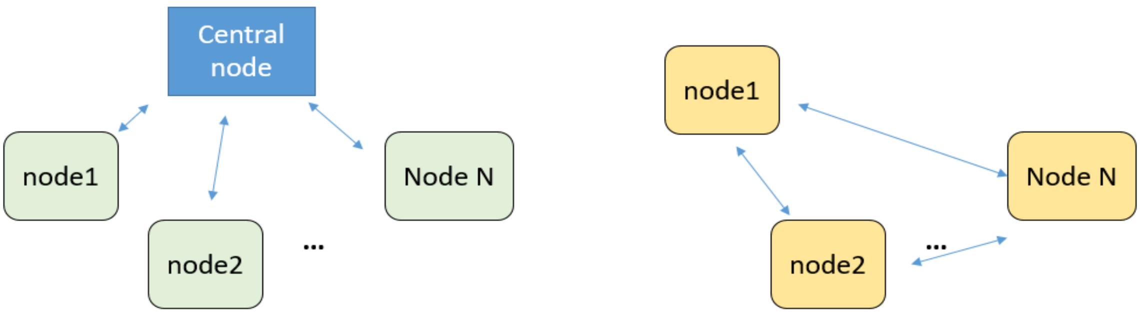
Distributed Machine Learning Topologies

The training procedure can be divided into two distinct cycles [26].

- Gradient Computation cycle that computes parameter gradients based on the current model parameters by applying backpropagation on mini-batches drawn from the training data.
- Optimization cycle that consumes the computed gradients in the Gradient Computation cycle to update model parameters.

Based on two defined cycles, as Figure 8 shows, there are two main network topologies to map an execution model onto a cluster of independent devices [26].

- Centralized optimization wherein the optimization cycle is run in a central node and the gradient computation code is replicated onto the AIoT end devices.
- Decentralized optimization wherein both cycles are replicated in each AIoT end device and some form of synchronization is done to let the distinct optimizers to work cooperatively.



(a) Centralized optimization

(b) Decentralized optimization

Figure 8. Two main distributed machine learning topologies

In 2017, Federated Learning (FL) [27] was presented to enable local and distributed machine learning training at the level of edge nodes or end devices.

Based on methods of data and feature distribution that we already described, we can divide FL into three small classes [4].

- Horizontal FL
- Vertical FL
- Federated transfer learning

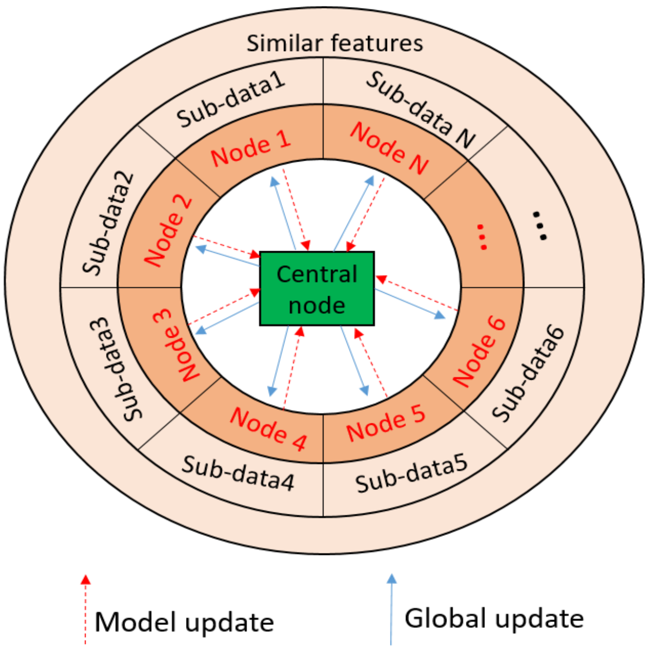


Figure 9. Horizontal FL logical architecture

Horizontal FL, see Figure 9, is utilized to analyze the horizontal data distribution. In horizontal FL, every node (i.e., learning client) has its private dataset. All clients cooperatively train a global FL model using their datasets considering a similar feature space. Then, the central node aggregates all local updates from nodes and computes the new global update without the need to have direct access to local data. After that, the central node sends back the global update to all nodes for the next round of local learning. This procedure continues until the loss function converges or an appropriate accuracy value is achieved.

One IoT-related example of horizontal FL is voice assistants in a smart home. Where persons in the home speak the same keyword or sentence (feature space) with the different types of voice (sample space) on their smartphones. In this case, a central node (i.e., parameter server) averages the local speaking updates to build a global model for voice recognition [4].

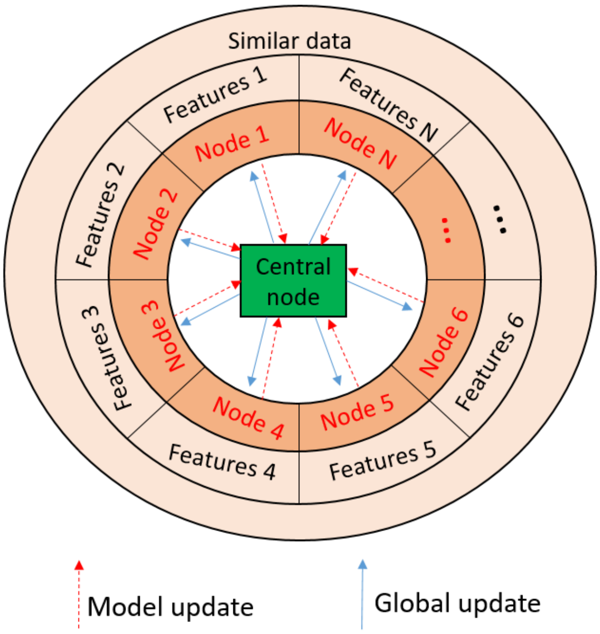


Figure 10. Vertical FL logical architecture

On the other hand, vertical FL supports vertical data distribution where the nodes utilize the same data with different features, as shown in Figure 10. In this case, an entity alignment method is considered to collect overlapped data samples of nodes. These samples are combined to train a common AI model using encryption techniques.

As an example, consider two regional different companies: ecommerce company and bank. They have different data features but very similar clients (i.e., same sample space). They can join a vertical FL process to cooperatively train an AI. Using this model, the FL-based system can estimate the optimal personalized loans for all clients based on their online shopping behaviours.

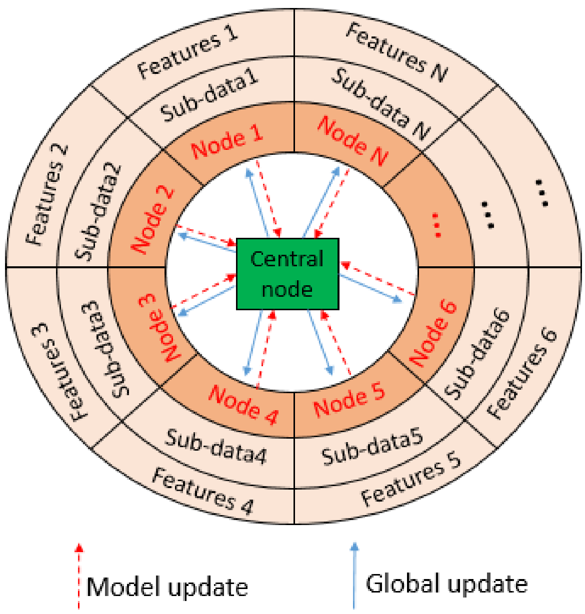


Figure 11. Federated transfer learning logical architecture

Federated transfer learning we have a set of nodes that have datasets with different sample space and different feature space, as shown in Figure 11. In other words, the datasets differ not only in samples but also in feature space.

For example, consider there are two companies in different locations: one bank located in one country and an e-commerce company located in another country. Because they are in two different countries, the client groups have a small intersection. In addition, because of the different businesses, only a small portion of the feature space from both parties overlaps. Federated transfer learning techniques can be utilized to offer solutions for the entire sample and feature space under a federation. Specially, a common representation between the two feature spaces is learned utilizing the limited common sample sets and, after, used to get predictions for samples considering only one-side features. Federated transfer learning is considered as an essential extension to the existing federated learning systems because it deals with problems exceeding the scope of existing FL algorithms [23].

3. System: System opportunities for AIoT (accelerators)

For many recent AIoT applications, we need to build end devices as standalone intelligent machines. The relation between intelligence and embedded systems becomes important.

The main hardware platforms for the deployment of AI units on end devices are:

- Graphics Processing Unit (GPU);
- Field Programmable Gate Array (FPGA);
- Central Processing Unit (CPU);
- Application Specific Integrated Circuit (ASIC);

High processing requirement of advanced AI algorithms is a challenge for general purpose CPUs in AIoT end devices. Therefore, hardware accelerators such as ASIC, FPGA, and GPU have been utilized to improve the performance of advanced AI algorithms.

GPU includes a large number of small cores. It was initially designed for video games. After, it has been deployed in high-performance computing systems to provide powerful processing capability. GPUs are used as hardware accelerators to speed up training and inference of AI. Many operations in AI such as convolution can be done in parallel on GPU, significantly decrease the training and inference time [18]. GPUs are considered as the most widely used hardware accelerators for improving AI processes. This is because of their high memory bandwidth and throughput. However, they are not energy efficient [19]. Therefore, using GPUs in battery operated AIoT end devices is an issue.

On the other hand, FPGA and ASIC hardware accelerators have relatively limited memory, I/O bandwidths, and processing power in comparison to GPU-based accelerators. However, they can work with lower power consumption. While the performance of ASIC design can be increased by utilizing memory hierarchy and assigning dedicated resources, the development cycle, cost, and flexibility are not satisfactory. As a solution, FPGA-based accelerators can be utilized to offer high performance at an acceptable price with low power consumption and reconfigurability [19], [20].

4. AI: AI progress for AIoT

As we already mentioned, the centralized cloud architecture may be ineffective or infeasible for modern AIoT applications because of the high volume of generated data, high scalability of their networks and data privacy issues.

On the other hand, running processing and memory intensive AI algorithms (e.g., neural networks with high number of hidden layers) on limited-resources AIoT end devices or even edge servers are a challenge. We can even see that the gaps between computational complexity and energy efficiency of deep neural networks and the hardware capacity is increasing [15].

For implementing AI functions, especially deep learning algorithms, on powerless AIoT end devices or not powerful edge devices, a lot of works have been done. Model compression is a technique which is used to lighten the learning model, improve energy efficiency, and speed up the inference on resource-limited end or edge devices, without reducing the accuracy. As Figure 12 shows, model compression techniques can be classified into five groups [14].

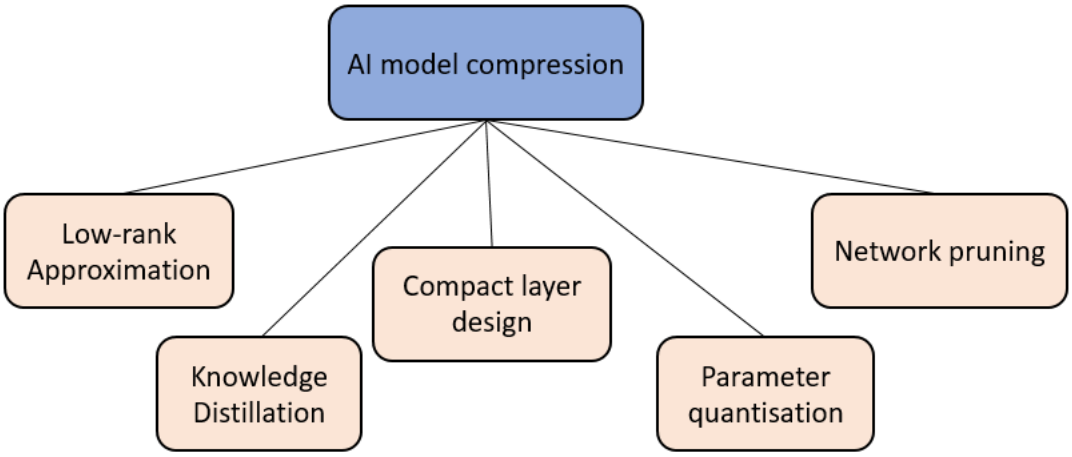


Figure 12. Model compression techniques

Low-rank Approximation: Considering this point that a matrix can be decomposed into the multiplication of multiple matrices of smaller size, this technique utilizes the multiplication of low-rank convolutional kernels to replace kernels of high dimension. This technique can effectively decrease the model size and computation.

Knowledge Distillation: This technique is based on transfer learning. It trains a smaller size model, called student model, with the distilled knowledge from a larger model, called teacher model. The student model on a resource-constrained device takes the advantage of transferring knowledge from the teacher network [5].

Compact layer design: This technique can effectively improve the utilization of resources (i.e., processing and memory). A big convolution can be replaced with several compact layers to reduce the number of parameters and, therefore, computations.

Network pruning: This technique removes the parameters that are not important. Therefore, by removing connections with low weights, a dense network can be converted to a sparse network.

Parameter quantization: Quantization is considered as an important and very active sub-area of research in the efficient implementation of computations associated with Neural Networks. We can see that it was firstly used for neural network training. In particular, the breakthroughs of half-precision and mixed-precision training have been the principal drivers that enabled an order of magnitude higher throughput in AI accelerators. However, going below half-precision without significant tuning is an issue, and the majority of the quantization studies work on inference [16].

5. AIoT applications

This section classifies the current and future AIoT applications with respect to their requirements. In order to fulfill this task, first, the conventional classification of IoT/AIoT applications is introduced (i.e., utilization-domain-classification) and then a different classification is described. We can see that over the last few years, in the vertical market, the IoT/AIoT applications are mainly categorized based on their utilization domains. Transportation, smart city, health-care, agriculture, environment, retail, and smart home are some examples (see Figure 13). However, the classification of applications based on their utilization domains may result in some conflicts and overlaps [29].

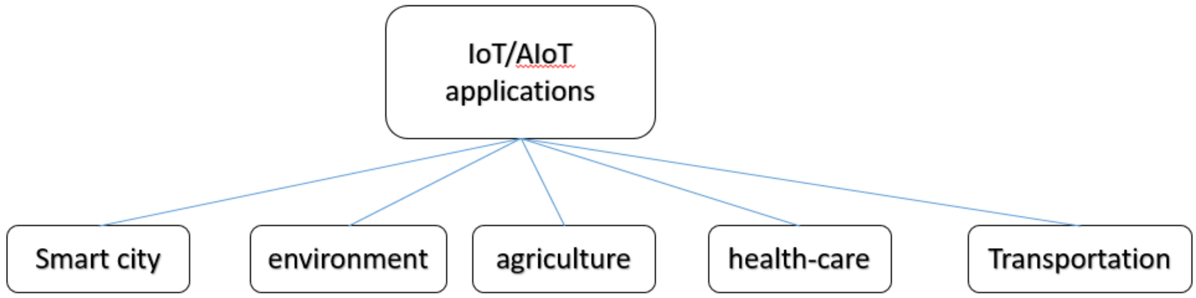


Figure 13. Conventional classification of IoT/AIoT applications based on utilization domains

Ding et al. in [29] proposed a new IoT classification. They first focus on end-user-types of applications and then consider other application requirements (i.e., data-rate, latency, coverage, power, reliability, and mobility) to generalize the proposed classification method. The proposed classification method considers the end-user-type for each application to classify it into one of the two main groups of human-oriented or machine-oriented applications (see Figure 14).

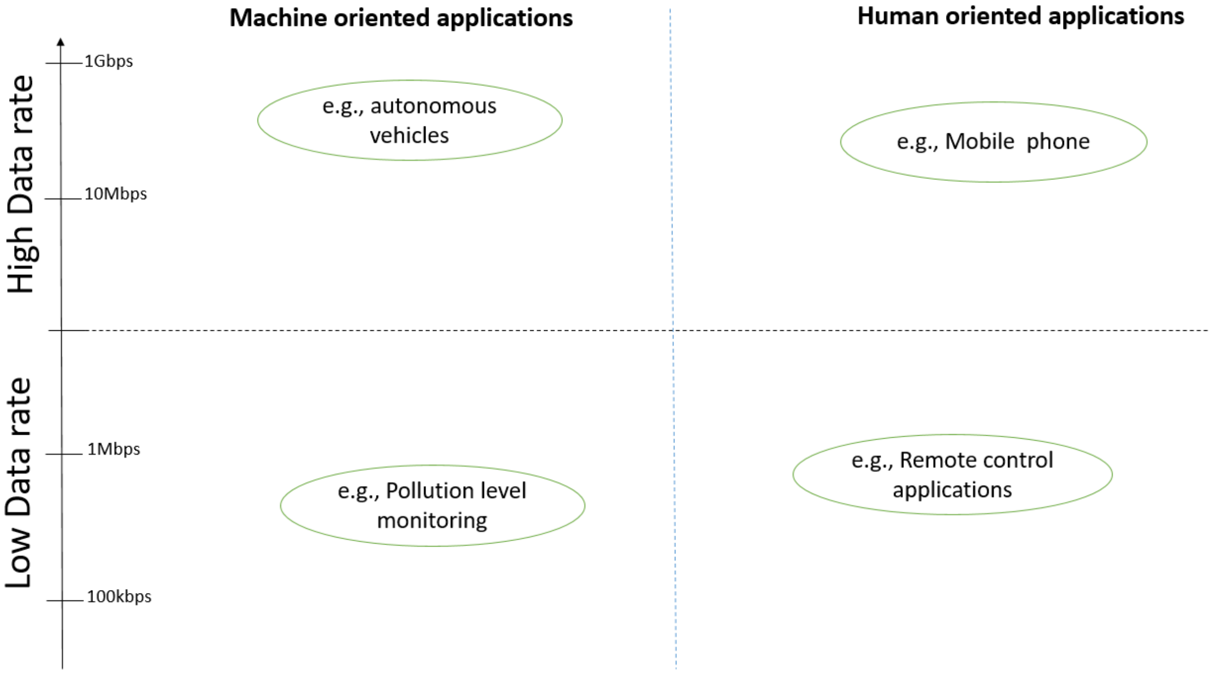


Figure 14. **Classification of IoT/AIoT applications based on end-user-type and data rate**

In human-oriented applications, a person plays an essential role. In other words, human-oriented applications refer to the applications that need human interaction to communicate with a network. Conventional smartphones, security cameras and patient surveillance systems are three examples of human-oriented IoT applications. Human-oriented applications are generally characterized by high data rates (i.e., from tens of Mbps up to tens of Gbps). However, we can also see a few human-oriented applications that need low data rate (e.g., form 1Mbps up to 10Mbps) like intelligent shopping applications that provide information of all items/interactions in a grocery store to the human as its end-user-type [29][30].

However, machine-oriented applications can automatically manage their tasks without requiring human intervention. These applications were historically characterized by low data rate (i.e., up to hundreds of kbps) and power consumption (e.g., wireless sensor network). Even today, many of the applications still require low data rates (as shown in Figure 14). However, some new and complex machine-oriented applications including autonomous vehicles require higher data rates (e.g., tens of Mbps) [31].

We introduce some promising applications of AIoT in the Following.

Life safety and comfort

Flood disaster mitigation can Improve life safety and preserve the environment. To reduce the risk of flood disasters some actions should be taken on special timing. Therefore, in [48], risk of flooding on the slopes of the mountains was analyzed and predicted by measuring the height and velocity of the water and periodically monitoring the rain Gauge sensors using AIoT. The authors proposed to connect several nodes using long range (LoRa) communication technology. LoRa is one of the low-power wide area network (LPWAN) communication systems which has long-distance transmission capabilities. The communication between posts (which include sensors) are carried out by using LoRa and a SIM900 sends data to a cloud server via the internet. Furthermore, all information is available in an application, and warnings are sent via short message service (SMS). This paper finally uses a fuzzy approach to implement various modes of intelligent control.

Pandemic Prevention and Control (COVID19)

In the case of COVID-19, timely screening of infected people can be very helpful. While manual screening takes a lot of time, modern technology can make it much more efficient and faster. AIoT-based wearable technology with Wi-Fi, 5G and Bluetooth helps detect, understand and monitor the Coronavirus as quickly as possible. Other non-portable modules can also be placed in public places such as airports and shopping malls. These modules are used to monitor the respiratory and blood pressure data of suspected patients. The two modules work and alert in the event of a suspicious event in any public place.

In this regard, the intelligent diagnostic system design as a part of AIoT product is critical. This will help to prevent infections of doctors due to direct contacts with patients. In addition, considering the epidemy nature, huge amounts of data, X-ray and CT scans are generated by standard methods of diagnosing COVID-19.

As lung segmentation technology becomes increasingly mature, intelligent COVID-19 diagnostic methods will become even more reliable. Such existing methods using neural networks and digital image processing technology are a great help for the doctors [49].

Helping elderly live

Fall detection, activity classification and health care are examples of AIoT scenarios in providing help for eldery live. These scenarios are achieved by gathering different sources of data. The collected data can be categorized into wearable and non-wearable data including audio, video, sensors gathered data (real-world information in terms of temperature, humidity, illumination, vibration, and human physical status (heart rate, blood pressure, skin conductance, temperature, movement, gait, etc.)), personal data collected from ubiquitous devices (e. g., a smart phone), routinely collected social clinical data, e. g., vital statistics and health data of subjects.

Investigation of data modalities that can be used for AIoT based elderly applications still is not comprehensive. As an example, it is worth exploring the sense of touch (haptics) which can be an essential sensory modality facilitating the ADLs of the elderly [26]. Different methods including signal processing and Machine Learning applied on the collected data helps in this process [50].

Automated Wheelchair for Physically Challenged with AIoT modules

Automated Wheel Chair (AWC) can be an assistance for people with a higher degree of impairment, or eldery people who, due to age or infection, are unable to move any of the body's organs save the head. Authors in [51] proposed to expand an AWC by including sensors into its components. In this way the clients of these chairs can be monitored/controlled by their family members and can have more independence.

The Smart Wheel Chair (SWC) proposed by authors can incorporate the microcontroller device to enable widespread electric wheelchair manipulation using head movements. Therefore, the device was equipped with digital and mechanical components. The accelerometer records processing unit is used to create a revolutionary head motion popularity technique. In addition, a Bluetooth device is mounted on the wheel chair, as well as the accelerometer that powers the motors and controls the chair's actions inside the domestic environment. The wheelchair designed by [51] focuses not only on the tool's movement but also on the health of the person who uses it. IoT is the main component of this proposal and sensors are used to detect the cardiac rate and blood oxygen levels. The ultrasonic sensors aid in the detection of impediments, allowing the chair to move in a more exact path. The authors also discuss the use of the SWC to protect disabled people from catastrophic pandemics like COVID-19 or to support them in their treatment of any pandemic-causing sickness.

Predicting arrival time of ships

Maritime transport is considered as one of the principal parts of world trade and economy. It is a complex environment where logistic planning plays an important role to guarantee an efficient use of resources. The maritime community includes a set of actors responsible for different activities related to port management; port authorities need open and neutral platforms that connect multiple systems, and these are known as port community systems [4].

Seaport terminals are one of the most important parts of the logistics network. They are in the center of the supply chain and connect the land-side and seaside, as well as the export and import sides of the supply chain.

As a result, it is very important to ensure a fluid operation of the terminal to guarantee the efficiency of the whole supply chain [8].

Therefore, an accurate estimated time of arrival of ships at ports can be necessary to optimize time, resources, and costs of maritime operations [9].

When a vessel arrives at a port, the necessary operations to be done by the various actors in the chain are scheduled according to this date/time. Therefore, any gap between the estimation and the actuality can result in delays and negative effects on the rest of the chain. One example of this negative impact can be unavailability of the scheduled quay. Because the terminal is operating another vessel or the vessel not being able to dock due to coincidence with the maneuvers of another vessel.

In [32], a cognitive AIoT architecture based has been introduced. Based on the architecture, a cognitive service capable of predicting with high accuracy the vessel port arrival is developed and integrated in a legacy sea traffic management solution. It utilises an automatic identification system and maritime oceanographic data to predict time of arrival of ships. The functionality of the system has been validated using the port of Valencia.

Bat Species Classification

Bats are considered as an important participant of the ecosystems. They are sometimes involved in indirect proliferation of various viruses. Therefore, it is necessary to mitigate the clash between bats and human populations. Tracking bats in real-time may help effectively manage bat populations and reduce risky interactions with bats by preventing any civilian establishments within or near to their habitats [53]. In addition to reducing virus risks, having comprehensive information about bat species and habitats can be useful for the health of our planet and society. However, gathering data related to bat population manually is costly, time consuming, and potentially error-prone.

We know that bats communicate via echolocation calls. In this case, training a neural network to identify the bat species based on their acoustic signals can be a good idea. An AIoT system can be utilized to do automatic bat species classification combining AI techniques (i.e., deep learning) with IoT. For example, Zualkernan et al. in [54] designed and implemented such an AIoT system.

Pests Detection

Presence of pests and plant diseases is considered as one challenge in crop production. In order to avoid the heat, pests often hide behind the leaves of plants during the day. They appear on the leaves in the evening or at night. As a result, it is not easy to observe them on crops during the day. Most of the time farmers detect pests when they have multiplied and spread uncontrollably.

In this situation, a large quantity of pesticides is needed to spray the crops to eliminate the pests and reduce agricultural damage. However, once the crops have been sprayed with pesticides in the growing season, pesticide residues remain even after washing. AIoT technology can be used for an environmental analysis of crop growth and the prediction of pest occurrence to improve the productivity of crops and reduce the size of the agricultural workforce [52].

Virtual shops applications

Online shopping is now considered as a part of our daily life. The online virtual shop systems should provide enough details about the products in the augmented reality (AR) or virtual reality (VR) spaces. In addition, they should enable the intelligent robots to do the joint operations and to provide real-time feedback sensation in real space. For establishing such a virtual shop system, many studies have been done. For example, wearable manipulators (e.g., [35]) can provide an immersive experience to clients with the sensory and haptic feedback functions (e.g., gesture control, direct projection of body motions, and haptic simulators).

Digital twin refers to a digital replica that can provide real-time monitoring or maintenance optimization of physical systems in various applications such as the manufacturing, healthcare, and automotive industry [33][34].

Rapid progress of AIoT technology could pave the way for developing digital-twin-based remote interactive systems for advanced robotic-enabled industrial automation and virtual shopping. For example, we can consider a system to inspect the physical objects with the up-to-date status according to the data obtained from the distributed sensors. It can consider both static properties (i.e., inherent features of the physical objects such as shape, size, color) and dynamic properties, (i.e., the real-time position, movement, gesture, and motion of objects that will change over time). These digital-twin-based remote interactive can enable the advanced interaction virtually, showing the potential in realizing real-time parallel control in unmanned working spaces. They may bring great convenience in different scenarios, such as online shopping and unmanned factories [36].

Floor Monitoring

In addition to deployment of wearable electronics on the human body, versatile ambient-awareness devices can be placed in our living environment to facilitate a higher level of living comforts and building intelligence.

These distributed electronic devices, in comparison to the wearable electronics, can offer additional functionalities in a complementary and generalized manner (e.g., indoor positioning, biometric-based identity authentication, access regulation, amenity automation, smart interaction, security) without asking clients to wear any cumbrous devices. To this end, cameras and motion sensors on our daily interacting floor area can effectively and conveniently obtain the affluent information related to human behaviors. Especially motion sensors can be used without such privacy concerns.

A floor sensor-based monitoring system can capture the correlative information from the interactive signals (e.g., stepping position, walking trajectory and activity status). It can be used to facilitate versatile smart home applications.

Considering AIoT technology, an AIoT-assisted reliable floor sensing system can offer not only robust position and activity information for daily monitoring and automation, but also identity information for personalized health care and access regulation.

For example, a reliable and smart floor monitoring system can be developed by integrating a robust floor mat array with advanced Deep Learning-assisted data analytics [37].

6. AIoT implementation: tools and platforms

In this section, we describe some well-known tools and platforms to implement AIoT systems.

A 3.5-tier container-based edge computing architecture

As we already mentioned, in some special applications because of the privacy issues involved in the service, the processing cannot send the image data to the cloud, and this data must be processed on edge servers. Identification of individuals in person re-identification (Re-ID) systems is one of these examples [38]. In this case, the lightweight and scalable architecture of service design called microservices are suitable for such applications to embedded AIoT environments. Microservices are small applications that can be independently deployed, extended, and tested. Microservices are based on a technology called container technology sharing access to the host kernel which is faster than VM. However, the security issue is a challenge as the applications share the same kernel. This security issue has been addressed in different new attempts such as Kata Container and. Orchestration of these containers can be done using solutions such as Kubernetes. Kubernetes is actually an open-source container-orchestration platform for automating computer application deployment, and management. It provides a container centric platform for microservices to automate auto-scaling, resilience, and high availability [55].

Considering AIoT and implementation at the edge, there is a need to separate the control plane and the compute plane. It is where KubeEdge and OpenYurt are some helpful products in the market. As an example, KubeEdge is the first Kubernetes Native Edge Computing Platform with both Edge and Cloud components open sourced.

In addition, one of the most popular enterprise Java microservices framework is Spring Boot introduced in 2014 and it is part of Spring framework. More than 60% of enterprise applications run on Java and Spring has good integration with almost all popular development libraries. It helps faster development of Java-based microservices with its built-in libraries [55].

Implementing the above-mentioned solution, [38] provided a solution for person re-identification (Re-ID). Re-ID is used to identify the same person from multiple different angles and different directions across multiple cameras. However, because of the privacy issues involved in the identification of individuals, Re-ID systems cannot send the image data to the cloud, and this data must be processed on edge servers. This study designed a Re-ID system at the AIoT edge computing gateway.

The proposed framework deployed in AIoT edge computing gateway in this paper has four layers including the camera communication, tracking, application, and AI accelerated layers. Each layer has its corresponding functions.

The system utilizes a microservice to perform Re-ID calculations on edge computing and balances efficiency with privacy protection. This method helps to handle large computing resources owing to the processing of AI algorithms through edge computing.

Model development was carried out on Torchreid, a Re-ID software framework based on PyTorch with functions to support model development, end-to-end training, and evaluation of Re-ID models such as OSNet, ResNet50, Inception, and MobileNV2. Although the common IoT protocols such as MQTT, XMPP, and CoAP have their own advantages, considering the challenges of TCP/user datagram protocol compatibility, low latency, multiplexing, and other characteristics, the paper selected gRPC which satisfied all requirements. Finally, microservices was implemented through Kubernetes along with Linkerd2 to provide load Balancing.

The experimental results indicated that implementing more servers significantly improves the system performance in terms of the processing speed and response time when compared to only one server. In addition, it can expand and flexibly scale the proposed architecture. When compared to a single PC or traditional server, the AIoT edge computing gateway is more flexible, lightweight, less expensive, and has less power consumption.

Sophon Edge platform

Sophon Edge is an edge-cloud collaborative computing platform for building AIoT applications efficiently [39]. It proposes a pipeline-based computing model that facilitates the programming of computation tasks, with scaling feature. Scaling here can be across multiple edge servers and even different IoT devices. Sophon supports edge-cloud collaboration as well. Sophon was tested using a real-world Smart City application which led to a drastic reduction in coding efforts. The whole application reaches an accuracy of 85% in obtaining its objective.

Sophon Edge platform includes two main parts: Edge Nodes and Edge Hubs. Edge Nodes are computing servers and mainly responsible for managing the connection and operation on IoT devices (e.g., forwarding significant data to the cloud). They are deployed at the network edge. In order to facilitate device access, Sophon Edge supports a set of communication protocols such as Modbus, MQTT, COAP, AMQP, ONVIF, RTSP, RTMP, UVC, OPCUA, HLS and GB28281 which covers the current mainstream devices in the AIoT domain. In addition, AIoT devices generate raw data which are gathered and converted into a data stream. It can facilitate subsequent data processing through the exploitation of a multimedia gateway or a message bus. The multimedia gateway collects real-time stream media data (e.g., images, videos) from multimedia AIoT devices, while the message bus collects structured data from ordinary IoT devices.

An Edge Node manages several pipelines that are either received from the cloud or created locally. The instances of the pipelines are run by the stream processing engine. In addition, an Edge Node can control various AI models received from the cloud, and these models can be utilized to create inferences in the deployed pipeline instances. The edge infrastructure mainly supports local data storage and transportation to the cloud end.

Edge Hubs are considered as cloud servers. They control Edge Nodes and do computing-intensive tasks such as model training. In addition, an Edge Hub can control pipelines, known as shared pipelines.

By increasing the scale of an AIoT application, the number of Edge Nodes could increase. It can make it possible to run the same computation task on multiple Edge Nodes. In order to reduce cost, instead of creating at least one pipeline on each Edge Node, the shared pipelines are managed on the Edge Hub and delivered to the Edge Nodes later on.

The model warehouse stores all the trained models as container images. They can be easily deployed on the Edge Nodes.

Edge performance benchmarking and AI platforms

In edge computing, where resources are located near users, a large-scale, geographically dispersed, and resource-rich distributed system will be created. This kind of loosely coupled complex systems have different operational conditions over time. In this context, the performance characteristics of such systems will need to be captured rapidly. This performance characteristics capture is called performance benchmarking, for application deployment, resource orchestration, and adaptive decision-making. Edge performance benchmarking is a new research topic that has started gaining momentum [40].

Performance benchmarks have played a vital role in the evolution of the computing landscape and represent an important field of research and development. CPU or processor-related benchmarks have existed since several decades. To mention an example, the Whetstone benchmark was designed to measure the floating-point arithmetic performance [41] and Dhrystone was developed in the 1980s to evaluate the performance of integers [42].

Post-1990 the HPC benchmarking emerged, then in post-2000 grid benchmarking, and post-2010 cloud benchmarking were hop topics. And post-2015 edge benchmarking achieved major milestones. This trend seems to suggest that benchmarks for more loosely coupled distributed systems are gaining prominence as a result of the decentralization and distribution of resources within the computing landscape [40].

Starting from 2015, edge performance benchmarking has been under development. These benchmarks cover a wide range of resources, including (i) end devices, such as IoT sensors, smartphones, and user gadgets, including wearables; (ii) computational resources located at the edge of a wired network, including routers, switches, gateways, and dedicated resources, including embedded computers and micro clouds; and (iii) cloud resources [40].

When AI is merged with edge computing, edge performance benchmarks are faced with much more challenges. Not many edge performance benchmarks consider AI platforms.

Three successful edge performance benchmarks consider AI platforms. Edge AIBench [43] provides four AI benchmark applications that can reflect complex scenarios of edge computing. These scenarios refer to intensive care unit patient monitoring, surveillance cameras, smart homes, and autonomous vehicles [40]. These test models can be executed using a federated learning framework in a publicly available test bed. LEAF [44] is another benchmarking framework which is modular for evaluating learning in federated settings. LEAF consists of open-source datasets, statistical and system metrics, and reference implementations. AIoTBench [45] was developed to facilitate evaluating the AI capabilities of edge devices for image classification, speech recognition, transformer translation, and micro workloads.

7. AIoT challenges and future

AI is considered as one main part of AIoT which also plays an important role for future of AIoT. AI is currently utilized in AIoT mostly for [46]:

- Increasing operational efficiency. For example, Google brings the power of AI into IoT to improve its data center cooling costs;
- Offering better risks management. As an example, Fujitsu ensures worker safety using AI for analyzing data sourced from connected wearable devices;
- Triggering new and enhanced products/services. For example, Rolls Royce utilizes AI methods in the implementation of IoT-enabled airplane engine maintenance amenities;
- Improving the scalability of IoT.

It is expected that AI procedures and strategies continue to be integrated into a few IoT compositional layers. This can pave the way for the development of more AIoT applications. In the future, we expect AI to be also used for designing data reduction approaches for AIoT on embedded edge node and mimicking human-like personality in designing smart-objects. AI can be applied to IoT to enhance security, protect resources, and lessen vulnerabilities. As a result, AI will be used to make self-governing activities [46].

However, similar to other emerging technologies, there are still challenges that must be addressed with AIoT innovations. In the following, we list a set of main challenges which AIoT faces with them.

- **Data and device heterogeneity**: AIoT systems typically include a large number of heterogeneous end devices (including scalar sensors, vector sensors, and multimedia sensors) that generate a huge volume of data stream having different formats, sizes, and timestamps. In this case, processing, transmission, and storage of these data is a challenge and needs more studies.
- **AI on end and edge devices**: Deployment of AI models on edge and even devices is essential, especially for real-time data stream processing application. However, we saw that these devices have limited resources. Therefore, finding some solutions such as designing lightweight AI as well as more powerful AIoT devices is still a big challenge.
- **Lack of an appropriate and comprehensive architecture:** we saw that there are different architectures and AI deployments techniques. When the level of edge intelligence goes higher (i.e., more powerful edge and end devices supporting AI functions), the amount and path length of data offloading reduce. In this case, the data transmission delay decreases and data privacy increases. However, computational latency can be increased. This conflict shows that generally there is no "best-level". In other words, the "best-level" edge intelligence depends on an application, and it can be found by considering several criteria (e.g., network and computation latency, energy efficiency, privacy and network bandwidth cost) [11].
- **Green clouds and datacenters:** datacenters consume a large amount of electricity and generate a noticeable volume of carbon emission. Therefore, energy consumption improvement in datacenters is needed for a sustainable future. Running processing-intensive AI functions on datacenters can consume a noticeable amount of energy (i.e., for both processing and cooling aspects). Edge intelligence can help large-size datacenters and even core network elements to reduce their energy consumption and carbon emission.

- **Security and privacy:** AIoT can be deployed to achieve more efficient IoT operations. As a result, it can improve human-machine interactions and offer better data analysis. AI approaches are utilized to transform IoT data into useful information for the better decision-making processes, and it further increases the overall usability of the system. As already mentioned, AIoT systems can be used for different applications ranging from security and surveillance system to intelligent transportation system, smart farming, healthcare monitoring, industrial automation, and eCommerce. However, AIoT frameworks may have data security and privacy issues as they are vulnerable to various types of information security-related attacks. These issues further cause the serious consequences such as the unauthorized data leakage and data update. As a solution, Blockchain can offer more security as compared to the traditional security mechanisms. It is a specific type of database and a digital ledger of transactions, which is duplicated and distributed across the entire network of computer systems. It can save data in the form of some blocks which are then chained together. Therefore, blockchain can be integrated in many AIoT applications to offer more security [47].

References

[1] IoT Analytics, State of IoT Q4/2020 & Outlook 2021. November 2020. Available online: https://iot-analytics.com/product/state-of-iot-q4-2020-outlook-2021/ (accessed on 28 July 2021).

[2] Fraga-Lamas, P.; Lopes, S.I.; Fernández-Caramés, T.M. Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case. *Sensors* 2021, *21*, 5745. https://doi.org/10.3390/s21175745

[3] Artificial intelligence in Internet of things, CAAI Transactions on Intelligence Technology, 2018, Vol. 3, Iss. 4, pp. 208–218.

[4] Nguyen, DC, Ding, M, Pathirana, Pubudu, Seneviratne, A, Li, J and Poor, HV, Federated Learning for Internet of Things: A Comprehensive Survey, IEEE Communications Surveys and Tutorials, pp. 1-38, doi: 10.1109/COMST.2021.3075439.

[5] O. A. Wahab, A. Mourad, H. Otrok and T. Taleb, "Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future Directions in Communication and Networking Systems," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1342-1397, Second quarter 2021, doi: 10.1109/COMST.2021.3058573.

[6] J. Zhang and D. Tao, "Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things," in IEEE Internet of Things Journal, vol. 8, no. 10, pp. 7789-7817, 15 May15, 2021, doi: 10.1109/JIOT.2020.3039359.

[7] Cisco Annual Internet Report (2018–2023) White Paper, March 9, 2020. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[8] Fraga-Lamas, Paula, Sérgio I. Lopes, and Tiago M. Fernández-Caramés 2021. "Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case" Sensors 21, no. 17: 5745. https://doi.org/10.3390/s21175745

[9] K. Ashton et al., “That ‘internet of things’ thing,” RFID Journal, vol. 22, no. 7, pp. 97–114, 2009.

[10] Technical Paper on Artificial Intelligence of Things (AIoT), INTERNATIONAL TELECOMMUNICATION UNION (ITU)- SG20-C979, October 2021.

[11] Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. Proc. IEEE 2019, 107, 1738–1762.

[12] 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.

[13] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, Aug. 2019, doi: 10.1109/JPROC.2019.2921977.

[14] Dianlei Xu, Tong Li, Yong Li, Xiang Su, Sasu Tarkoma, Tao Jiang, Jon Crowcroft and Pan Hui, "Edge intelligence: Architectures, challenges, and applications," arXiv preprint arXiv:2003.12172, 2020.

[15] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," Nature Electronics, vol. 1, no. 4, p. 216, 2018.

[16] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630, 2021.

[17] Amodei, D.; Hernandez, D. AI and Compute. May 2018. Available online: https://openai.com/blog/ai-and-compute/ (accessed on 28 July 2021).

[18] Ang, L.M.; Seng, K.P. "GPU-Based Embedded Intelligence Architectures and Applications, Electronics 2021, 10, 952.

[19] A. Shawahna, S. M. Sait and A. El-Maleh, "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review," in *IEEE Access*, vol. 7, pp. 7823-7859, 2019, doi: 10.1109/ACCESS.2018.2890150.

[20] L. Du, Y. Du, Y. Li, J. Su, Y.-C. Kuan, C.-C. Liu, and M.-C. F. Chang, "A reconfigurable streaming deep convolutional neural network accelerator for internet of things," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 1, pp. 198–208, 2018.

[21] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, Jan S. Rellermeyer, "A Survey on Distributed Machine Learning", ACM Computing Surveys, Volume 53, Issue 2, 2020.

[22] Peteiro-Barral, D., Guijarro-Berdiñas, B. A survey of methods for distributed machine learning. Prog Artif Intell 2, 1–11 (2013). https://doi.org/10.1007/s13748-012-0035-5

[23] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 2, pp. 1–19, 2019.

[24] Akhil Mathur, Daniel J. Beutel, Pedro Porto Buarque de Gusmão, Javier Fernandez-Marques, Taner Topal, Xinchi Qiu, Titouan Parcollet, Yan Gao, Nicholas D. Lane, "On device Federated Learning with Flower", arXiv preprint arXiv:2104.03042, 2021.

[25] Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, D. Dou, "From Distributed Machine Learning to Federated Learning: A Survey", arXiv:2104.14362.

[26] Matthias Langer, Zhen He, Wenny Rahayu, Yanbo Xue, "Distributed Training of Deep Learning Models: A Taxonomic Perspective," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 12, pp. 2802-2818, 1 Dec. 2020, doi: 10.1109/TPDS.2020.3003307

[27] H. B. McMahan, E. Moore, D. Ramage, S. Hampson et al., "Communication-efficient learning of deep networks from decentralized data," in Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[28] B. -W. Chen, Y. -S. Jhang, H. -T. Pai, S. -H. Wang, M. -H. Sheu and T. -H. Chen, "AIoT-based Audio Recognition System for Smart Home Applications," *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2021, pp. 1-2, doi: 10.1109/ICCE-TW52618.2021.9603103.

[29] J. Ding, M. Nemati, C. Ranaweera and J. Choi, "IoT Connectivity Technologies and Applications: A Survey," in IEEE Access, vol. 8, pp. 67646-67673, 2020, doi: 10.1109/ACCESS.2020.2985932.

[30] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "Iot applications on secure smart shopping system," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 1945–1954, 2017.

[31] R. Fukatsu and K. Sakaguchi, "Millimeter-wave v2v communications with cooperative perception for automated driving," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), pp. 1–5, IEEE, 2019.

[32] Valero, C.I.; Ivancos Pla, E.;Vaño, R.; Garro, E.; Boronat, F.; Palau, C.E. Design and Development of an AIoT Architecture for Introducing a Vessel ETA Cognitive Service in a Legacy Port Management Solution. Sensors 2021, 21, 8133.

[33] A. El Saddik, "Digital Twins: The Convergence of Multimedia Technologies," in *IEEE MultiMedia*, vol. 25, no. 2, pp. 87-92, Apr.-Jun. 2018, doi: 10.1109/MMUL.2018.023121167

[34] Fei Tao, Fangyuan Sui, Ang Liu, Qinglin Qi, Meng Zhang, Boyang Song, Zirong Guo, Stephen C.-Y. Lu & A. Y. C. Nee (2019) Digital twin-driven product design framework, International Journal of Production Research, 57:12, 3935-3953, DOI: 10.1080/00207543.2018.1443229

[35] Zhu M, Sun Z, Zhang Z, Shi Q, He T, Liu H, Chen T, Lee C. Haptic-feedback smart glove as a creative human-machine interface (HMI) for virtual/augmented reality applications. Sci Adv. 2020 May 8;6(19). doi: 10.1126/sciadv.aaz8693. PMID: 32494718; PMCID: PMC7209995.

[36] Sun Z, Zhu M, Zhang Z, Chen Z, Shi Q, Shan X, Yeow RCH, Lee C. Artificial Intelligence of Things (AIoT) Enabled Virtual Shop Applications Using Self-Powered Sensor Enhanced Soft Robotic Manipulator. Adv Sci (Weinh). 2021 Jul;8(14):e2100230. doi: 10.1002/advs.202100230. Epub 2021 May 26. PMID: 34037331; PMCID: PMC8292889.

[37] Qiongfeng Shi, Zixuan Zhang, Yanqin Yang, Xuechuan Shan, Budiman Salam, and Chengkuo Lee, "Artificial Intelligence of Things (AIoT) Enabled Floor Monitoring System for Smart Home Applications", ACS Nano 2021 *15* (11), 18312-18326

[38] Ching-Han Chen, Chao-Tsu Liu, A 3.5-tier container-based edge computing architecture, Computers & Electrical Engineering, Volume 93, 2021, 107227, ISSN 0045-7906.

[39] Rong, G., Xu, Y., Tong, X. *et al.* An edge-cloud collaborative computing platform for building AIoT applications efficiently. *J Cloud Comp* 10, 36 (2021). https://doi.org/10.1186/s13677-021-00250-w

[40] Varghese, B., Wang, N., Bermbach, D., Hong, C-H., de Lara, E., Shi, W., & Stewart, C. (2021). A Survey on Edge Performance Benchmarking. ACM Computing Surveys, 54(3), [66]. https://doi.org/10.1145/3444692

[41]. HJ Curnow, BA Wichmann, A Synthetic Benchmark. The Comput. J. 19, 43–49 (1976).

[42] RP Weicker, Dhrystone: A Synthetic Systems Programming Benchmark. Commun. ACM 27, 1013–1030 (1984).

[43] C Luo, et al., AIoT Bench: Towards Comprehensive Benchmarking Mobile and Embedded Device Intelligence in Benchmarking, Measuring, and Optimizing, eds. C Zheng, J Zhan. (Springer International Publishing), pp. 31–35 (2019).

[44] T Hao, et al., Edge AIBench: Towards Comprehensive End-to-end Edge Computing Bench marking in Proceedings of the First BenchCouncil International Symposium on Benchmark ing, Measuring, and Optimizing. pp. 23–30 (2018).

[45]. S Caldas, et al., LEAF: A Benchmark for Federated Settings. CoRR abs/1812.01097 (2018).

[46] S. C. Mukhopadhyay, S. K. S. Tyagi, N. K. Suryadevara, V. Piuri, F. Scotti and S. Zeadally, "Artificial Intelligence-Based Sensors for Next Generation IoT Applications: A Review," in IEEE Sensors Journal, vol. 21, no. 22, pp. 24920-24932, 15 Nov.15, 2021, doi: 10.1109/JSEN.2021.3055618.

[47] Mohammad Wazid, Ashok Kumar Das and Youngho Park et al. Blockchain-Envisioned Secure Authentication Approach in AIoT: Applications, Challenges, and Future Research. Wirel. Commun. Mob. Comput.. Vol. 2021:1-19. DOI: 10.1155/2021/3866006

[48] Sung, Wen-Tsai & Devi, Ihzany & Hsiao, Sung-Jung. (2021). Early Warning of Impending Flash Flood Based on AIoT. 10.21203/rs.3.rs-850371/v1.

[49] Shu-Wen Chen, Xiao-Wei Gu, Jia-Ji Wang, Hui-Sheng Zhu, "AIoT Used for COVID-19 Pandemic Prevention and Control", *Contrast Media & Molecular Imaging*, vol. 2021, Article ID 3257035, 23 pages, 2021. https://doi.org/10.1155/2021/3257035

[50] K. Qian, Z. Zhang, Y. Yamamoto and B. W. Schuller, "Artificial Intelligence Internet of Things for the Elderly: From Assisted Living to Health-Care Monitoring," in *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 78-88, July 2021, doi: 10.1109/MSP.2021.3057298

[51] Shrivastava, Sanatan. (2021). EasyChair Preprint Automated Wheelchair for Physically Challenged with AIoT Modules Automated Wheelchair for Physically Challenged with AIoT modules.

[52] C. -J. Chen, Y. -Y. Huang, Y. -S. Li, C. -Y. Chang and Y. -M. Huang, "An AIoT Based Smart Agricultural System for Pests Detection," in *IEEE Access*, vol. 8, pp. 180750-180761, 2020, doi: 10.1109/ACCESS.2020.3024891.

[53] P. C. Y. Woo and S. K. P. Lau, “Viruses and Bats,” Viruses, vol. 11, no. 10, Art. no. 10, Sep. 2019, doi: 10.3390/v11100884.

[54] I. Zualkernan, J. Judas, T. Mahbub, A. Bhagwagar and P. Chand, "An AIoT System for Bat Species Classification," *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, 2021, pp. 155-160, doi: 10.1109/IoTaIS50849.2021.9359704.

[55] Srivastava, R., Cloud Native Microservices with Spring and Kubernetes: Design and Build Modern Cloud Native Applications using Spring and Kubernetes (English Edition), 9789390684311, 2021, BPB Publications.