ICN Research Group   J.K. Choi
Internet Draft   J.S. Han
Intended status: Informational   G.H. Lee
Expires: June 28, 2023   H.S. Park
                                                              KAIST
                                                       Dec 27, 2022

Requirements and Challenges for User-level Service Managements of IoT Network by utilizing Artificial Intelligence
draft-choi-icnrg-aiot-10

Abstract

This document describes the requirements and challenges to employ artificial intelligence (AI) into the constraint Internet of Things (IoT) service environment for embedding intelligence and increasing efficiency.

The IoT service environment includes heterogeneous and multiple IoT devices and systems that work together in a cooperative and intelligent way to manage homes, buildings, and complex autonomous systems. Therefore, it is becoming very essential to integrate IoT and AI technologies to increase the synergy between them. However, there are several limitations to achieve AI enabled IoT as the availability of IoT devices is not always high, and IoT networks cannot guarantee a certain level of performance in real-time applications due to resource constraints.

This document intends to present a right direction to empower AI in IoT for learning and analyzing the usage behaviors of IoT devices/systems and human behaviors based on previous records and experiences. With AI enabled IoT, the IoT service environment can be intelligently managed in order to compensate for the unexpected performance degradation often caused by abnormal situations.

Table of Contents

1. Introduction

The document explains the effects of applying artificial intelligence /machine learning (AI/ML) algorithms in the Internet of Thing (IoT) s ervice environments.

IoT applications will be deployed in heterogeneous and different area s such as the energy, transportation, automation and manufacturing in dustries as well as the information and communication technology (IC T) industry. Many IoT sensors and devices can connect to an IoT servi ce environment where IoT objects cannot interoperate with each other and can interact with different applications. The IoT service may not run in a single administrative domain. If market demand exists, the cross-domain service scenarios for IoT applications could be widely d eployed. Future IoT applications occur at multiple domains of heterog eneity with various time scales.

The IoT service requirements for common architectures and public APIs poses some challenges to the underlying service environment and netw orking technologies. Some IoT applications require significant securi ty and privacy as well as significant resource and time constraints. These mission-critical applications can be separated from many common IoT applications that current technology may not provide. It means t hat IoT service requirements are difficult to classify common require ments and functional requirements depending on IoT service scenario.

Recently, artificial intelligence technologies can help the context-a ware IoT service scenarios apply rule-based knowledge accumulation. T he IoT service assumes that many sensing devices are connected to sin gle or multiple IoT network domains. Each sensor sends small packets to the IoT servers periodically or non-periodically. Detection data c ontains periodic status information that monitors whether the system is in a normal state or not. In some cases, alert information is incl uded for quick processing. Most IoT applications can operate in two m odes. One is a simple monitoring mode and the other is an abnormal mo de for rapid processing. In a simple monitoring phase, the IoT device periodically sends sensing data to the server. If the measured data is outside the normal range, the IoT service can change the operating mode to an abnormal phase and activate future probes. Alarm conditio ns should be promptly notified to responsible persons. For mission-cr itical applications, reliable communication with robust QoS requireme nts in terms of error and latency is required.

Periodic data accumulation from IoT devices is cumbersome. Under norm al conditions, the IoT data is simply accumulated without further act ion. In an unusual situation, incoming IoT data can cause an urgent a ction to notify the administrator of the problem. Streaming data traffic from thousands of IoT devices is annoying to store in the database because it is not easy to extract unidentified or future incidents. Only a significant portion of the incoming data stream can be stored in a real-time database that is time-sensitive and capable of rapid query processing. A combination of different IoT detection data, including location, time, and status, allows you to sort and categorize a portion of streaming data when an additional inspection is required, and perform real-time processing. One of the missions of the IoT database is to be able to extract preliminary symptoms of unexpected accidents from a large amount of streaming data.

If some transmitted data is important to invoke the corresponding action, there are some questions about whether the incoming data is correct. If the incoming data contains accurate and time-critical events, appropriate real-time control and management can be performed. However, if the incoming data is inaccurate or intentionally corrupted, additional accidents may occur. In these cases, incoming data can trigger to initiate additional inspections to protect against future unacceptable situations. But, if time-critical data is missed due to errors in the sensing devices and the delivery protocol, there is no reason to configure IoT networks and devices at a high cost.

It is not easy to analyze data collected through IoT devices installed to monitor complex IoT service environments. If the sensor malfunctions, the data of the sensor cannot be trusted. Additional investigation should be done if abnormal status from specific sensors is collected. The data of the redundant sensor installed in the same area should be received or combined with other sensor information adjacent to the sensor to determine the abnormal state.

For sensors installed in a specific area, sensing records will remain for a certain period of time. IoT service operators can look at the operational history of the sensor for a period of time to determine what problems were encountered when data was collected. When an abnormal situation occurs, IoT sensor should investigate whether it noticed normal operations and notified the IoT service operator. If the abnormal situation is not properly detected, the operator should analyze whether it was caused by malfunction of the IoT sensor/other reasons.

In the IoT service environment, it is possible to analyze the situation accurately by applying recent artificial intelligence and machine learning technologies. If there is an operational record of the past, it is possible to determine when an abnormal situation arises. Most problems are likely to be repeated, so if the past learning experience is accumulated, the anomaly of IoT services can be easily and immediately identified. In addition, when information gathered from various sensors is synthesized, it is possible to accurately determine whether abnormal situations have occurred.

Various types of IoT sensors are installed with certain purposes. It expects that all the IoT sensors intend to monitor the occurrence of special abnormal situations in advance. Therefore, it should be set in advance what actions are required when a specific anomaly occurs. The appropriate work is performed on the abnormal situation according to the procedure, predefined by the human. By using artificial intelligence and machine learning algorithms, the appropriate actions are taken when an abnormal situation is detected from various IoT sensors.

2. Challenging Issues of IoT network

This section describes the challenging issues of data sensing, collection, transfer, and intelligent decision from untrusted data quality and unexpected situations of IoT service environments.

2.1. Untrusted and incorrect IoT devices

IoT traffic is similar to traditional Internet traffic with small packet sizes. Mobile IoT traffic can cause some errors and delays because wireless links are unstable and signal strength may be degraded with device mobility. If the signal strength of the IoT device with a power limit is not so strong, the reception quality of the IoT server may not be sufficient to obtain the measurement data.

For mission-critical applications, such as smart-grid and factory-automation, expensive IoT sensors with self-rechargeable batteries and redundant hardware logic may be required. However, unexpected abnormal situations may occur due to sensor malfunctions. There are trade-offs between implementation cost and efficiency for cost-effective IoT services. When smart-grid and factory-automation applications are equipped with IoT devices, the acceptable quality from IoT solutions can be required. Sometimes, expensive and duplicated IoT solutions may be needed.

2.2. Traffic burstiness of IoT network

IoT traffic includes two types of traffic characteristic: periodic with small packet sizes and bursty with high bandwidth. Under normal conditions, the IoT traffic periodically transmits status information with a small bandwidth, several kilobits/sec. However, in an abnormal state, IoT devices need a high bandwidth, up to several tens of megabits/sec, in order to identify actual events and investigate accurate status information. In addition, traffic volume can explosively increase in response to emergencies. For example, in the case of smart-grid application, the bandwidth of several kilobits/sec is usually used, and when an urgent situation occurs, a broadband channel is required up to several tens of megabits/sec.

The other traffic can be integrated at an IoT network to increase bandwidth efficiency. If an emergency situation occurs in the IoT service, IoT traffic volumes suddenly increase, in which case network processing capacity may be not sufficient. If the IoT service is integrated with voice and video applications, the problem can become more complex. As time goes by, traffic congestion and bottlenecks are frequent in some areas. In addition, if an existing service policy changes (for example, prioritizing certain traffic or suddenly changing the route), other unexpected problems may be encountered. Various congestion control and load balancing algorithms with the help of artificial intelligence can be applied to handle time-varying traffic on a network.

Until now, much research has been done on traffic variability in an integrated network service environment. All networks have their own traffic characteristics, depending on geographical area, number of subscribers, subscribers' preferences, and types of applications used. In the case of IoT traffic, the normal bandwidth is very small. If the IoT traffic volume increases abruptly in an abnormal situation, the network may suffer unacceptable delay and loss. If emergency situations detected by IoT networks occur in a smart grid or intelligent transportation system, the processing power of the IoT network alone cannot solve the problem and the help of existing network resources is inevitable.

2.3. Management overheads of heterogeneous IoT sensors

Traffic management in an integrated network environment is not easy. In order to operate the network steadily, a network operator has its own know-hows and experiences. If there are plenty of network resources, it is easy to set up a bypass route even if network failure or congestion occurs in a specific area. For operating network steadily, network resources may be designed to be over-provisioned in order to cope with various possible outages. A network operator predicts the amount of traffic generated by the corresponding equipment and grasps to what extent a transmission bandwidth is required. If traffic fluctuation is very severe, the network operator can allocate network resources in advance. In case of frequent failures or severe traffic fluctuation, some network resources are separated in order not to affect normal traffic.

More than a billion IoT devices are expected to connect to smartphones, tablets, wearables, and vehicles. Therefore, IoT services are targeted at mobile applications. In particular, intelligent transportation systems need the help of IoT technology to provide traffic monitoring and prevent public or private traffic accidents. IoT technology can play an important role in reducing traffic congestion, saving people's travel time and costs, and providing a pleasant journey.

The IoT service has troublesome administrative problems to configure an IoT network which consists of IoT servers, gateways, and many sensing devices. The small-sized but large-numbered IoT devices may incur administrative overhead since all the IoT devices should be initialized and the bootstrapping information of IoT resources should be loaded into the IoT service environments. Whenever some IoT devices are newly added and some devices have to be removed, the dynamic reconfiguration of IoT resources is essential. In addition, the IoT device's preinstalled software should be regularly inspected and upgraded according to its version. Frequent upgrades and changes to some IoT devices may require autonomic management and bootstrapping techniques.

Network management generally assumes that all network resources operate reliably with acceptable quality. In most failure situations, the network operator decides to switch to a redundant backup device or bypass the failed communication path. If some IoT devices are not stable, duplicate IoT devices can be installed for the same purpose. If IoT resources are not duplicated, various mechanisms are needed to reduce the damage. Therefore, it is necessary to prioritize the management tasks to be performed first when an abnormality occurs in the IoT service environment. However, managing duplicate networks can cause another problem. If two IoT devices are running at the same time, the recipient can get redundant information. If two or more unusual situations occur at the same time, it is difficult to solve the problem since tasks for urgent processing should be distinguished from tasks that can be performed over time.

In addition, the operations manager's mistakes or misunderstanding of problem situations can lead to other unexpected complications. Therefore, artificial intelligence technologies can help what kind of network management work is required when an unexpected complicated situation occurs even though a procedure for an abnormal situation is already prepared.

2.4. Power management of IoT devices

RF-enabled wireless energy transfer (WET) technology has the advantage of enhancing poor user experience quality, which is a critical issue that can occur in the existing energy harvesting equipment of IoT equipment or conventional battery-powered communication, and can have sustainable throughput performance. WET technology is expected to become the core of next-generation communication because it can transmit stable and continuous power to equipment through electromagnetic waves that propagate far-field. This technology is expected to have advantages in communication stability and battery replacement cycle, which are limitations of existing IoT battery equipment.

The WET service simultaneously charges energy and transmits data at the hybrid access point, and in this process, network operation management technology that helps each user manage battery life is essential. To this end, the network must be able to manage the power and traffic budget of multiple APs and enhance the battery life of each IoT device. By efficiently managing energy and power transmission between multiple APs operating at the same time, it can be helpful for battery-life and reliable communication of a large number of IoT devices used at the same time. For example, for users who are constantly far away from the AP and take a long time to charge, the nearest AP can continuously transmit power. In addition, transmissions can proceed to more distant APs to avoid congestion on specific APs.

Until now, only traffic in the network service environment is considered. However, in an environment where WET becomes more common, it seems difficult to manage network resources only with the existing traffic-oriented network environment. For efficient network operation in an environment that considers energy and traffic at the same time, period control or network slicing of a new concept will be required.

2.5. Computational resource management for IoT services

IoT networks for various industries require intelligent IoT services designed to make decisions and take actions in various situations by applying data analysis techniques based on data collected from IoT devices. For example, an IoT system for an autonomous vehicle should analyze vehicle information (speed, acceleration, steering angle, etc.) and camera images collected from the vehicle to control vehicle speed, direction, and vehicle convenience functions, etc. An IoT system for augmented/virtual reality needs to analyze lots of information such as the user's visual information, motion information, and object location information to perform display control suitable for the user's movement.

Since these IoT applications need to collect data and make context-aware decisions in real-time, they generally require low latency. However, as the complexity and computation requirements of IoT applications increase, it is hard to satisfy the required latency time by utilizing only IoT devices due to the limited computing capability.

Therefore, in order to overcome the limited computational capability of IoT devices and satisfy the service latency requirements of users, many studies have focused on offloading computational workload to computing nodes in the IoT network such as gateways and cloudlets, or to cloud by extending the IoT network to the cloud. While offloading computational workload, the optimal computing nodes are selected in consideration of various factors such as status of the computing nodes and communication bandwidth. After the optimal node selection, the computational workloads are processed in the selected nodes and data transmissions are performed.

However, various problems exist in such computational workload offloading solutions. Communication overhead occurs because data must be exchanged between computing nodes in order to distribute and process computation. In addition, in the process of data movement, users' privacy issues are also emerging. In conclusion, as IoT applications become computational-intensive and users'perceptions of data are changing, it is necessary to overcome the limited computational capability of IoT devices by applying various computation distribution technologies, and in this process, various problems mentioned above need to be considered.

2.6. Network management of heterogeneous IoT services

The number of applications targeting not only mobile devices but also IoT devices is increasing as many more IoT devices are connected through a shared network. Correspondingly, the application services targeting different types of IoT devices have become more diversified. For example, mobile broadband, massive IoT, and mission-critical IoT have different requirements concerning mobility, charging, security, police control, latency, and reliability. In terms of IoT devices for massive IoT, their locations tend to be fixed which results in less mobility, but high throughput while maintaining their massive connection. On the other hand, autonomous driving and unmanned aerial vehicle containing mission-critical IoT require low latency within milliseconds and highly reliable transmission while considering their mobility. As a result, it is difficult to guarantee such different services with the existing mobile communication network.

While constructing new networks for each service has many limitations in terms of spatial and financial aspects, network slicing allows operators to manage different services with more flexibility and agility within the existing network at low cost. The concept of network slicing is to run multiple logical customized networks, referred to as slices, on a shared infrastructure. Devices using the same application services are associated with the same slice to separately manage their requirements from that of devices in other slices. However, to efficiently manage the network slicing, coordination of physical resources between slices is necessary. These physical resources include frequency bandwidth, cache memory, and backhaul link capacity. For efficient management, such resources should be dynamically coordinated between slices considering traffic characteristics, amount of data requests, and transmission protocol.

3. Overview of AI/ML-based IoT services

In this section, successful applications of artificial intelligence i n IoT domains are provided. The common property of IoT applications a nd services is that they require fast analytics rather than later ana lytics with piled data. Recently, neural-network-based artificial int elligence technologies are widely used across many IoT applications.

Simple IoT applications include dynamic contexts that share common fe atures among social relations at the same administration domain. IoT devices in the same domain can provide their service contexts to the IoT server. When a dynamic change occurs in an IoT service context, t he IoT device needs real-time processing to activate urgent events, a lert notifications, update, and reconnect contexts. The IoT service m ust support real-time interactions between the IoT device and the sys tem in the same domain. The IoT service contexts must be shared betwe en physical objects and social members in the same domain as well.

Artificial intelligence technologies have been shown promising in man y areas, including IoT. For example, contextual information for a car -sharing business must interact with customers, car owners, and car s haring providers. All entities in the value chain of a car sharing bu siness must share the corresponding situation to pick up, board, and return shared cars. Communication networks and interactive informatio n, including registration and payment, can be shared tightly among th e entities. Home IoT service environment can be equipped with sensors for theft detection, door lock, temperature, fire detection, gas det ection, short circuit, air condition to name a few. Office IoT servic e environments, including buildings such as shopping centers and bus/ airport terminals, have their own sensors, including alarm sensors. W hen an alarm signal is detected by the sensor, the physical position and occurrence time of the sensor is determined in advance. All signa ls from various sensors are analyzed comprehensively to make the righ t decision. If some sensors frequently malfunction, the situation can be grasped more accurately by analyzing the information of the adjac ent sensor. In particular, when installing multiple sensors in a part icular building (e.g., surveillance camera, location monitoring, temp erature, etc.), a much wider range of sensors can be used when utiliz ing artificial intelligence and machine learning technologies.

(Smart home) Smart home concept span over multiple IoT applications, health, energy, entertainment, education, etc. It involves voice reco gnition, natural language processing, image-based object recognition, appliance management, and many more artificial intelligence technolo gies integrated with IoT. Smart connected-devices monitor the house t o provide better control over home supplies and expenses. The energy consumption and efficiency of home appliances are monitored and analy zed with deep learning based technologies, such as artificial neural network, long-short-term-memory, etc.

(Smart city) Smart city, as well, contains multiple IoT domains, tran sportation, infrastructure, energy, agriculture, etc. Since heterogen eous data from different domains are gathered in smart cities, variou s artificial intelligence approaches are studied in smart-city applic ation. Public transportation behaviors and crowd movements patterns a re important issues, and they are often dealt with neural network bas ed methods, long-short-term-memory and convolutional neural network.

(Smart energy) As two-way communication energy infrastructure is depl oyed, smart grid has become a big IoT application, which requires int elligent data processing. The traditional energy providers are highly interested in recognizing local energy consumption patterns and fore casting the needs in order to make appropriate decisions on real-time. Moreover, the energy consumers, as well, want analyzed information o n their own energy consumption behaviors. Recently, many works on ene rgy consumption prediction, energy flexibility analysis, etc. are act ively ongoing. Most works are based on the latest deep learning techn ologies, such as multi-layered-perceptron, recurrent neural network, long-short-term-memory, autoencoder, etc.

(Smart transportation) The intelligent transportation system is anoth er source of big data in IoT domains. Many use cases, such as traffic flow and congestion prediction, traffic sign recognition, vehicle in trusion detection, etc., have been studied. Moreover, a lot of advanc ed artificial intelligence technologies are required in autonomous an d smart vehicles, which require many intelligent sub-tasks, such as p edestrian's detection, obstacle avoidance, etc.

(Smart healthcare) IoT and artificial intelligence are integrated int o the healthcare and wellbeing domain as well. By analyzing food imag es with convolutional neural network on mobile devices, dietary intak es can be measured. With voice signal captured from sensor devices, v oice pathologies can be detected. Moreover, recurrent neural network and long-short-term-memory technologies are actively being studied fo r early diagnosis and prediction of diseases with time series medical data.

(Smart agriculture) To manage a vast area of land, IoT and artificial intelligence technologies are recently used in agriculture domains. Deep neural network and convolutional neural network are utilized for crop detection or classification and disease recognition in the plan ts. Moreover, for automatic farming with autonomous machine operation, obstacle avoidance, fruit location, and many more sub-tasks are hand led with advanced artificial intelligence technologies.

4. Requirements for AI/ML-based IoT services

In this section, requirements for AI/ML-based IoT data collection and delivery, intelligent and context-aware IoT services, and applying A

I/ML to IoT data will be described.

4.1. Requirements for AI/ML-based IoT data collection and delivery

IoT services store a vast amount of data that IoT devices periodically generate, and the refining and analyzing are costly. Effective analysis of IoT data has been considered to be the most important factor in data processing, but the analysis of efficient data collection and delivery methods are becoming other significant factors as the amount of the data collected is explosively increasing.

In particular, as a number of IoT devices have been deployed within the IoT network, controlling data collection and delivery for each of them has become impossible. The introduction of AI/ML techniques for simultaneous and efficient management of the IoT devices should be considered as a countermeasure. For IoT data collection and delivery, the following two factors will need to be considered, IoT devices energy and data quality.

(IoT Device Energy) As many IoT devices have begun to be deployed within the IoT network, it is impossible to deliver energy to many IoT devices simultaneously. Consequently, the efficient battery use has become an important issue.

If IoT data collection and delivery periods are too short, a lifetime of the IoT device will be shortened through the reckless use of IoT device energy. Thereby, it increases the cost required to provide IoT service. On the other hand, if IoT data collection and delivery period are too long, the quality of the IoT services provided will be reduced due to the lack of details in the data for situation recognition and real-time processing. Therefore, taking into account the energy consumption of the IoT devices, research on proper IoT data collection and delivery period is necessary.

(IoT device using WET) When many IoT devices use WET to charge energy, transmit data, and perform calculations, it is difficult to deliver desired amount of power to all users. Therefore, efficient energy transfer for communication is essential.

When the UE acquires and transmits energy using WET, it is necessary to consider the amount of energy to transmit to each AP or user to satisfy QoS or user experience. We also need information about the optimal amount of energy transfer for battery life. Therefore, research should be conducted considering the power consumption and traffic of each AP and UE.

(Data Quality) Since the data collected from the majority of IoT devices usually contain redundant information, it causes additional costs for the data collection and refinement processes. Therefore, it will be necessary to select and deliver meaningful information from redundant IoT data to reduce unnecessary cost on the IoT network. To do so, it will need the research to identify the relationships among the data collected various devices and interpret the information that the data contains.

(Optimal IoT Device Operation) Each IoT service may have its own defined information requirements and it is imperative that the data with certain quality level should be collected to extract the core information for each situation. As scheme of data collection considerably affects the data quality, the operation of IoT devices should be effectively adjusted to ensure the data quality. In general, high resolution of data collected from IoT devices as well as data preprocessing are required for high quality data. However, these methods leading tremendous operation cost of IoT devices causes a lifetime of IoT networks, degrading the QoS of IoT services. Eventually, an optimal IoT device operation technique for each IoT service must be considered.

4.2. Requirements for intelligent and context-aware IoT services

In a context-aware IoT service environment, it is important to establish a context to be aware of in advance since IoT devices will be deployed according to a pre-designed architecture and to check how characteristics of IoT data and data-to-data characteristics are expressed under these circumstances. For the data produced by IoT devices, since it contains the device's relative location information, sensing value over time, event information, it should be reviewed to provide the target context-aware service using this information. Some of the necessary technologies will be described in the following.

(Physical Clustering) To increase the accuracy of context-awareness, the provision of context-aware services should be considered in a situation where the relationship between IoT devices with respect to physical layout or physical environment is taken into account. Setting a rule using the service provider's domain knowledge may be possible, but introducing the physical clustering into a diverse IoT environment (e.g., in bedroom, kitchen, balcony, or a space connected through an open door) will require identifying the physical relationship between the devices using data generated from IoT devices.

(Extra Data Processing) In order to prevent degradation of service quality from errors in data values or device malfunctions, extra sensors should be placed in the majority of IoT environments. In a context-aware service, they contain the same information, so the technologies filtering the data that contains only essential part among the same information while preventing data errors would be required.

(Unreported data handling) If an event is detected on a particular IoT device, it will transmit data regardless of the device's sensing and delivery interval. At this time, the data of IoT devices which are physically clustered are needed to accurately detect events that occurred, and it is difficult to expect that these devices will provide data at the same time. A gateway can request data from clustered devices, but it has a problem for real-time processing for emergency situations. Therefore, handling unreported data will be required based on previously collected data.

(Abnormal data in AI/ML) In the case of context-aware services that operates based on the predetermined rule, the flexibility to cope with emergency situations that have not been considered is low, and thus AI/ML algorithms are required to intelligently cope with a myriad of situations. However, many abnormal data are generated depending on environmental conditions such as device status, so AI/ML algorithms that can operate in that imperfect environment should be considered.

(Edge computing in IoT) There are two necessary prerequisites required in context-aware IoT services: IoT devices real-time management and IoT network architecture supporting the high-volume data transmission. When an abnormal situation is discovered, high-volume data should be utilized to adequately to monitor the situation through the IoT device's real-time management. As contrasted with conventional cloud computing structures, an edge computing structure, where IoT data processing servers are located in closer proximity to IoT devices, provides higher energy efficiency and lesser communication delay.

4.3. Requirements for applying AI/ML to IoT data

In this subsection, the requirements for applying AI/ML to IoT data are described.

4.3.1. Training AI/ML algorithm

To use AI/ML algorithm, two elements are required, AI/ML model and training data. The presence of training dataset in good quality is an important factor of the AI/ML model performance since the model is iteratively trained with the training data. However, for anomaly detection, there is not enough training data since not only the probability of anomaly occurrence is very low but also it is almost impossible to retrieve the ground truth value even when the situation has occurred. Therefore, using domain knowledge, AI/ML learning based on abnormal situation data generation or simulation should be considered. For example, for an external intrusion detection application within a smart home, when a camera and a motion sensor detect an intruder, a light sensor checks the measuring value. If the light does not turn on, then the IoT application recognizes it as an abnormal situation. In this way, by using the domain knowledge, the rule regarding the operational scenario of the IoT application is generated as the training data, and the generated training data can be used for model learning. This will not only enable learning the anomaly detection algorithm in IoT application but also improving the accuracy. Therefore, IoT application, in which it is difficult to acquire dataset in good quality, will require data generation based on domain knowledge for AI/ML.

4.3.2. AI/ML inference in IoT application

In order for AI ML technology to be applied to IoT applications, the training data and the input data for model testing and inferencing must have the same characteristics such as dimension, time interval, types of features, etc. However, due to the volatile IoT data characteristics that vary from situations in many IoT applications, it is difficult to directly apply the AI/ML algorithms. For example, in a simple monitoring phase, the IoT devices periodically send sensing data, and AI/ML have no difficulty in operating. However, in an abnormal mode, the IoT applications require a fast response, and IoT devices transmit data at shorter intervals than normal, which changes the characteristics of the data being input to the AI/ML algorithm. Therefore, data preprocessing technology handling the abnormal data will be required in advance, such as data imputation, correcting data anomalies, and Interpolation of unreported data.

4.3.3. AI/ML models update in IoT application

While IoT devices deployed in an IoT environment continuously collect and transmit the necessary data to an IoT server, the IoT server delivers intelligent IoT services using AI/ML models, based on these collected data. As IoT devices monitor a wide variety of IoT environmental information such as time-relevant and place-relevant information, the data gathered from IoT devices changes significantly depending on the events occurred in an IoT environment. For this reason, complete information is no longer reflected during AI/ML training and this, sooner or later, affects the quality of IoT services provided. Therefore, AI/ML models must be updated periodically upon the data subsequently collected. However, if the models are updated in a too short period, not only the high management (or labor) cost for updating is required, but also it may detrimentally affect the overall performance of the models. Therefore, models should be updated by fairly considering between the cost required for updating AI/ML models and the merits gained from the usage of updated AI/ML models.

4.4. Requirements for distributed computing for AI/ML-based IoT services

The AI/ML-based IoT services consist of large-scale computations. IoT devices themselves may not have enough computing power to handle these large-scale computations. Therefore, it is necessary to overcome the limitations of computing power of IoT devices by distributing computational workload across gateways, cloudlets, and clouds. Since there are many challenges in distributing computational workloads, the following factors must be considered:

(Energy Consumption) In most AI/ML-based IoT services, data collected from IoT devices is transmitted to the edge or cloud. However, since IoT devices have limited battery life and it is impractical to charge them frequently, it is necessary to consider the energy consumption of IoT devices in distributed computing. The energy consumption of IoT devices is caused by many factors such as uplink data transmission and the amount of computation. In order to minimize the energy consumption of IoT devices, those factors should be considered in distributed computing.

(Latency) In distributed computing, overall latency consists of many factors such as communication delay, queuing delay, and processing time. If all factors are not considered properly, overall latency may increase. Therefore, it is necessary to reduce the overall latency by considering various delay factors.

(Communication Overhead) In order to distribute the computational workload, it is necessary to exchange input/output data and intermediate results between nodes. In other words, additional data should be moved through the network, which leads to communication overhead. How to reduce this overhead should also be considered.

(Data Privacy) Data from IoT devices must be transmitted to other nodes for distributed processing. At this time, if the data to be sent is related to the user's personal information or sensitive data, the user will be reluctant to send the data. In this regard, the consideration of data privacy is required for distributed computing.

Also, from a technical point of view, distributing computational workloads requires the following technologies:

(Computing Node Selection) In order to distribute computational workloads, the appropriate target nodes must be selected. In general, it is necessary to select a node suitable for the purpose of distribution in consideration of node state information and connection information between nodes. In this case, queue status and communication bandwidth will be important factors for purpose of minimizing delay, and power consumption for transmission between nodes and battery status of nodes will be important factors for the purpose of minimizing energy consumption. If the purpose is to improve service availability, nodes?malfunction and failure rates will be critical considerations. In this way, it is necessary to determine the importance of factors to be considered according to the purpose of distribution and to select appropriate nodes based on this.

(Cooperation between Computing Nodes) In order to minimize the latency of IoT applications, cooperation between computing nodes is essential. After the selection of possible computing nodes based on the IoT network status, sub-task partitioning and distributed task allocation among nodes are performed. In general, most AI/ML algorithms can be divided into several sub-tasks and the sub-tasks can be allocated to the appropriate computing nodes based on several factors such as computational capacity, and network bandwidth. However, while allocating several sub-tasks among nodes, not only latency but also data privacy and energy consumption should be considered. For example, in order to protect user data privacy, it is necessary to allocate an initial part of AI/ML algorithms to the end device. Also, for reducing energy consumption, it is necessary to minimize the amount of computation in the end device as much as possible and reduce the amount of uplink data transmission. While considering the above several issues, the optimally distributed task allocation techniques which aim to minimize latency are required.

(Network Slicing) Slice coordination is necessary for efficient management of network slicing. Because different slices have different service requirements, a balanced allocation of physical resources such as frequency bandwidth, cache memory, and backhaul link capacity between slices is important to guarantee requirements associated with IoT devices. For a balanced allocation of physical resources, not only the amount of feasible physical resources but also the requirements and traffic characteristics of IoT devices should be considered. For example, massive IoT devices, in which the size of each packet tends to be short, require a low outage probability for massive connection and this can be achieved with HARQ-based grant-free transmission with a relatively small amount of frequency bandwidth. On the other hand, mission-critical IoT devices require both low latency and high reliability. Although grant-free transmission can reduce the latency caused by grant scheduling overhead, such transmission is vulnerable to preemption. To lower the probability of preemption, securing a sufficient amount of frequency bandwidth is essential compared to the previous case. The task of allocating the physical resources considering the requirements and traffic characteristics of numerous IoT devices becomes more critical in an environment where traffic characteristics and requirements vary over time. Because of these considerations, the numerical solution cannot be computed. Therefore, learning from historical data becomes important and AI/ML algorithms can be applied to cope with the overall problems to intelligently manage network slicing.

5. State of arts of the artificial intelligence/machine learning technologies for IoT services

In this section, well-known machine learning and artificial intelligence technologies applicable to IoT applications are reviewed.

5.1. Machine learning and artificial intelligence technologies review

The classical machine learning models can be divided into three types, supervised, unsupervised, and reinforcement learnings. Therefore, in this subsection, machine learning and artificial intelligence technology reviews are done in four different categories: supervised, unsupervised, reinforcement, and neural-network-based.

5.1.1. Supervised learning for IoT

Supervised learning is a task-based type of machine learning, which approximates function describing the relationship and causality between input and output data. Therefore, the input data needs to be clearly defined with proper output data since supervised learning models learn explicitly from direct feedback.

(K-Nearest Neighbor) Given a new data point in K-Nearest Neighbor (KNN) classifier, it is classified according to its K number of the closest data points in the training set. To find the K nearest neighbors of the new data point, it needs to use a distance metric which can affect classifier performance, such as Euclidean, Mahalanobis or Hamming. One limitation of KNN in applying for IoT network is that it is unscalable to large datasets because it requires the entire training dataset to classify a newly incoming data. However, KNN required less processing power capability compared to other complex learning methods.

(Naive Bayes) Given a new data point in Naive Bayes classifiers, it is classified based on Bayes' theorem with the "naive" assumption of independence between the features. Since Naive Bayes classifiers don't need a large number of data points to be trained, they can deal with high-dimensional data points. Therefore, they are fast and highly scalable. However, since its "naive" assumptions are somewhat strong, a certain level of prior knowledge on the dataset is required.

(Support Vector Machine) Support Vector Machine (SVM) is a binary and non-probabilistic classifier which finds the hyperplane maximizing the margin between the classes of the training dataset. SVM has been the most pervasive machine learning technology until the study on neural network technologies are advanced recently. However, SVM still has advantages over neural network based and probabilistic approaches in terms of memory usage and capability to deal with high-dimensional data. In this manner, SVM can be used for IoT applications with severe data storage constraint.

(Regression) Regression is a method for approximating the relationships of the dependent variable, which is being estimated, with the independent variables, which are used for the estimation. Therefore, this method is widely used for forecasting and inferring causal relationships between input data and output data in time-sensitive IoT application.

(Random Forests) In random forests, instead of training a single decision tree, a group of trees is trained. Each tree is trained on a subset of the training set using a randomly chosen subset of M input variables. Random forests considering various tree structures have very high accuracy, so it can be utilized in the accuracy-critical IoT applications.

5.1.2. Unsupervised learning for IoT

Unsupervised learning is a data-driven type of machine learning which finds hidden structure in unlabeled dataset without feedback during the learning process. Unlike supervised learning, unsupervised learning focuses on discovering patterns in the data distributions and gaining insights from them.

(K-means clustering) K-means clustering aims to assign observations into K number of clusters in which each observation belongs to the cluster having the most similarities. The measure of similarity is the distance between K cluster centers and each observation. K-means is a very fast and highly scalable clustering algorithm, so it can be used for IoT applications with real-time processing requirements such as smart transportation.

(Density-based spatial clustering of applications with noise) Density-Based approach to Spatial Clustering of Applications with Noise (DBSCAN) is a method that clusters dataset based on the density of its data samples. In this model, dense regions which include data samples with many close neighbors are considered as clusters, and data samples in low-density regions are classified as outliers [Kriegal]. Since this method is robust to outliers, DBSCAN is efficient data clustering method for IoT network environments with untrusted big datasets in practice.

(Dimensionality Reduction) Dimensionality reduction aims to transform high-dimensional data into low-dimensional data while minimizing information loss. In other words, it serves to extract some features that can best represent data among the features. By using dimensionality reduction, we can reduce the complexity of the model and shorten the computation time. Examples of dimensionality reduction include principal component analysis (PCA), T-stochastic neighbor embedding (T-SNE).

5.1.3. Reinforcement learning for IoT

Reinforcement learning is a reactive type of machine learning that learn a series of actions in a given set of possible states, actions, and rewards or penalties. It can be seen as the exploring decision-making process and choosing the action series with the most reward or the least penalty which can be cost, priority, time to name a few. Reinforcement learning can be helpful for selecting action of IoT device by providing a guideline.

(Q-learning) Q-Learning is a model-free, off-policy reinforcement learning algorithm based on the well-known Bellman Equation. The goal is to learn an action-selection policy maximizing the Q-value, which tells an agent what action to take. It can be used for IoT device to determine which action it should take according to conditions.

(State-Action-Reward-State-Action) Though State-Action-Reward-State-Action (SARSA) is a much similar algorithm to Q-learning, the main difference is that it is an on-policy algorithm in which agent interacts with the environment and updates the policy based on actions taken. It means that the Q-value is updated by an action performed by the current policy instead of the greed policy that maximizes Q-value. In this perspective, it is relevant when an action of one IoT device will greatly influence the condition of the environment.

(Deep Q Network) Deep Q network (DQN) is developed to solve the exploration problem for unseen states. In the case of Q-learning, the agent is not capable of estimating value for unseen states. To handle this generality problem, DQN leverages neural network technology. As a variation of the classic Q-Learning algorithm, DQN utilizes a deep convolutional neural net architecture for Q-function approximation. In real environments not all possible states and conditions are not able to be observed. Therefore, DQN is more relevant than Q-learning or SARSA in real applications such as IoT. Since DQN could be used within only discrete action space, it can be utilized for traffic routing in the IoT network.

(Deep Deterministic Policy Gradient) DQN has solved generality and exploration problem of the unseen or rare states. Deep Deterministic Policy Gradient (DDPG) takes DQN into the continuous action domain. DDPG is a deterministic policy gradient based actor-critic, model-free algorithm. The actor decides the best action for each state and critic is used to evaluate the policy, the chosen action set. In IoT applications, DDPG can be utilized for the tasks that require controlled in continuous action spaces, such as energy-efficient temperature control, computation offloading, network traffic scheduling, etc.

(Trust Region Policy Optimization) The concept of Trust Region Policy Optimization (TRPO) is to update parameters within Trust Region. In order to update the parameters of the new policy within the defined region, the parameters need to be updated with samples obtained from local points by solving conservative policy iteration. Moreover, the value of step size is constrained to update parameters within the defined Trust Region In IoT communication networks, considering that actions of TRPO are confidently decided within its boundary, TRPO can be implemented for load balancing, power allocation, service provisioning, etc.

(Proximal Policy Optimization) Unlike TRPO, Proximal Policy Optimization (PPO) uses the first derivative to update the parameters of its policy. The computation complexity of PPO is much alleviated compared to that of TRPO. While the step size of TRPO is constantly adjusted to satisfy its constraint, the step size of PPO simply uses clipping to fix the maximum value of its update. For exploration and exploitation, PPO computes action entropy to determine whether an agent requires more exploration to maximize its objective function. PPO is suitable for training agents in an environment where action space is too large for DQN algorithm. Therefore, tasks such as user association, power control, and resource allocation in wireless communications can benefit from PPO.

5.1.4. Neural Network based algorithms for IoT

(Recurrent Neural Network) Recurrent Neural Network (RNN) is a discriminative type of supervised learning model that takes serial or time-series input data. RNN is specifically developed to address issue of time dependency of sequential time-series input data. It processes sequences of data through internal memory, and it is useful in IoT applications with time-dependent data, such as identifying time-dependent patterns of sensor data, estimating consumption behavior over time, etc.

(Long Short Term Memory) As an extension of RNN, Long Short Term Memory (LSTM) is a discriminative type of supervised learning model that is specialized for serial or time-series input data as well [Hochreiter]. The main difference of LSTM from RNN is that it utilizes the concept of gates. It actively controls forget gates to prevent the long term time dependency from waning. Therefore, compared to RNN, it is more suitable for data with long time relationship and IoT applications requiring analysis on the long lag of dependency, such as activity recognition, disaster prediction, to name a few [Chung].

(Convolutional Neural Network) Convolutional neural network (CNN) is a discriminative type of supervised learning model. It is developed specifically for processing 2-dimensional image data by considering local connectivity, but now generally used for multidimensional data such as multi-channel sound signals, IoT sensor values, etc. As in CNN neurons are connected only to a small subset of the input and share weight parameters, CNN is much more sparse compared to fully connected network. However, it needs a large training dataset, especially for visual tasks. In CNN, a new activation function for neural network, Rectified Linear Unit (ReLU), was proposed, which accelerates training time without affecting the generalization of the network [Krizhevsky]. In IoT domains, it is often used for detection tasks that require some visual analysis.

(Variational Autoencoder) Autoencoder (AE) is a generative type of unsupervised learning model. AE is trained to generate output to reconstruct input data, thus it has the same number of input and output units. It is suitable for feature extraction and dimensionality reduction. Because of its behavior to reconstructing the input data at the output layer, it is often used for machinery fault diagnosis in IoT applications. The most popular type of AE, Variational Autoencoder (VAE) is a generative type of semi-supervised learning model. Its assumptions on the structure of the data are weak enough for real applications and its training process through backpropagation is fast [Doersch]. Therefore, VAE is suitable in IoT applications where data tends to be diverse and scarce.

(Generative Adversarial Network) Generative Adversarial Network (GAN) is a hybrid type of semi-supervised learning model which contain two neural networks, namely the generative and discriminative networks [Goodfellow]. The generator is trained to learn the data distribution from a training dataset in order to generate new data which can deceive the latter network, so-called the discriminator. Then, the discriminator learns to discriminate the generated data from the real data. In IoT applications, GAN can be used in situations when something needs to be generated from the available data, such as localization, way-finding, and data type conversion.

5.2. Technologies for lightweight and real-time intelligence

As the era of IoT has come, some sort of light-weight intelligence is needed to support smart objects. Prior to the era of IoT, most of the works on learning did not consider resource-constrained environments. Especially, deep learning models require many resources such as processing power, memory, stable power source, etc. However, it has been recently shown that the parameters of the deep learning models contain redundant information, so that some parts of them can be delicately removed to reduce complexity without much degradation of performance [Ba], [Denil]. In this section, the technologies to achieve real-time and serverless learning in IoT environments are introduced.

(network compression) Network compression is a method to convert a dense network into a sparse one. With this technology the network can be reduced in its size and complexity. By pruning irrelevant parts or sharing redundant parameters, the storage and computational requirements can be decreased [Han]. After pruning, the performance of the network is examined and the pruning process is repeated until the performance reaches the minimum requirements for the specific applications and use cases. As many parameters are removed or shared, the memory required is reduced, as well as computational burden and energy. Especially as most energy in neural network is used to access memory, the consumed energy dramatically drops. Although its main limitation is that there is not a general solution to compress all kinds of network, but it rather depends on the characteristics of each network. However, network compression is still the most widespread method to make deep learning technologies to be lightweight and IoT-friendly.

(approximate computing) Approximate computing is an approach to support deep learning in smart devices [Venkataramani], [Moons]. It is based on the facts that the results of deep learning do not need to be exact in many IoT applications but still valid if the results are in an acceptable range. By integrating approximate computing into deep learning, not only the execution time but also the energy consumption is reduced [Mohammadi]. Based on the optimal trade-off between accuracy and run-time or energy consumption, the network can be adjustably approximated. The network approximate technology can be well-used in such situations when the response time is more important than sophisticatedly analyzed results. Although it is a technology to facilitate real-time and lightweight intelligence, the process of training models and converting it to approximate network require some amount of resource. Therefore, the approximated model can be deployed on smart devices but the learning and approximation processes still need to take places on resource rich platforms.

(early exit inference) Deep learning models take advantages of nonlinear properties, from the forwarding process through a large number of layers, but often allow inferences with a reasonable level of accuracy without going through all the layers. Starting with this idea, early exit points in the middle of a layer has been studied to determine the reliability of the calculation results at that point, and if they have a reasonable level of reliability, they are no longer calculated. These technologies, which can achieve quick inference at the expense of some accuracy, can be useful in making decisions which are not fatal but require quick responses in resource-constrained IoT environments.

5.3. Federated learning

Federated learning is a method by which multiple IoT devices work together to train a single predictive model. Based on this, lower latency, less power, and smarter models can be obtained, and there are also advantages in terms of privacy. Federated learning allows mobile phones to jointly train predictive models to keep all data on the device, allowing machine learning to be performed without the need to store data in the cloud, making it well-suited for applying artificial intelligence in IoT devices.

(Horizontal Federated Learning) Horizontal federated learning is mainly used when all training models are the same and each user has the same data type. The clients update individually, and the server collects all local updates, calculates global updates based on the aggregated information and sends them to all users. It is mainly used in technologies such as voice assistants in smart homes.

(Federated Transfer Learning) Federated transfer learning, unlike the horizontal federated learning mentioned above, aims to train in situations with different data sets and different sample spaces. Send multiple data from different spaces, transfer them in the same space, and train the model. This is likely to be used in situations such as collaborations in hospitals with other patients.

(Centralized Federated Learning) Centralized federated learning trains on a central server and trains all clients in parallel in each round of communication. The central server employs a weighted average algorithm such as federated averaging (FedAvg) and then distributes the weights to each client. Servers are critical to managing clients by aggregating and distributing updates and protecting privacy.

(Decentralized federated learning) Decentralized federated learning is a network topology without a central server, and all learning subjects are connected through P2P. In every round of communication, clients are trained based on their own data and consensus on global updates is reached based on models received from neighboring clients. Based on this, it can be used even when there is no communication with the server and can be applied to technologies such as blockchain.

6. Use cases of AI/ML into IoT service

Many IoT service environments are equipped with camera, door lock, temperature sensor, fire detector, gas detector, alarm, and so on. Each sensor is deployed with particular purposes of each own to provide a specific service. However, there is a problem that the sensor utilization is not high enough due to the provision of the service using only a single sensor rather than multiple sensors and their mutual relations. Therefore, the quality of the service provided is not high as well. To enhance the sensor utilization and the service quality, all signals from various sensors should be analyzed comprehensively to make the right decision. This section describes the use cases for introducing AI / ML techniques in actual IoT service, utilizing multiple sensors. In advance of each use case description of various IoT service domains, characteristics of each domain to adopt AI/ML techniques are investigated.

6.1. Surveillance and Security in Smart Home

To minimize users' inconvenience and ensure their safety, surveillance and safety IoT applications provided within smart homes require fast notification with good level of precision IoT service quality for abnormal conditions detection. To do this, both data preprocessing techniques and AI/ML technologies for analysis of anomalies with high accuracy will be required.

6.1.1. Characteristics of Smart Home for AI/ML processing

(Training Data Generation) For Surveillance and Security, the processed data is necessary because there is little data for anomalies and the ground truth values are hardly available. Therefore, first, the steps to detect and calibrate the abnormal data are essential before the anomaly data should be generated using domain knowledge. First, constructing simulators about targeted smart home and generating events against external intrusions and then collecting the anomaly data can be considered. Furthermore, based on the data collected in the actual environment, anomaly data generation can proceed by breaking the relationship between sensors considering possible links between them within any intrusive environment.

(AI/ML Algorithm) One of the characteristics of the IoT environment for surveillance and safety is that a massive amount of data is collected and real-time responses are required. For the kNN algorithm, since the more data sets, the stronger against the noise and the higher the accuracy. If the appropriate dataset is used, the fast response can be expected. It makes suitable for the service environment to be considered. In addition, considering real-time data forecasting and analysis via LSTM, it is believed that improved accuracy for real-time anomalies detection can be expected.

6.1.2. Use case

(To be continued)

6.2. Smart home for elderly welfare

In the currently existing smart home IoT system for elders, it is difficult to detect exact physical and mental health status from IoT data because it is a very complicated problem mixed with various factors such as depression levels, heart rates, surroundings, etc. If health status can be detected correctly, it could be used for various single living-elderly care services. To do this, various techniques will be required such as protecting user's privacy and recognizing complex situation.

6.2.1. Characteristics of Smart Home for elderly welfare

IoT devices for in-home healthcare operations need to be small and unobtrusive. This form factor limitation means that these devices are also energy constrained. At the same time, the devices need to be ultra-reliable and provide timely and accurate data, coupled with processing capabilities to extract vital information. However, it is not an easy task for single-living elders to manage the status of all IoT devices in the house. To support reliable smart healthcare services for elders, the trade-offs between the energy-efficiency of the devices and the reliability of the collected data should be properly considered. Additionally, to manage and handle various emergency situations, AI/ML-based real-time data monitoring and automatic behavior recognition, sophisticated situation aware technology should be required.

6.3. Smart Building Management

Smart buildings often consist of heterogeneous IoT devices. These devices cooperate and their data is integrated for efficient autonomous building management. Many of the events in a large building may not require deep, complicated learning or processing. Some of them may require a fast response than an accurate analysis. Above all, a lot of events simultaneously occur and can arise heavy loads on the main server. The edge-computing techniques can be used to offload the main server's tasks.

6.3.1. Characteristics of Smart Building for AI/ML processing

(Training Data Generation) In smart buildings, heterogeneous IoT devices are deployed. They are diverse in their types, functions, performances, etc. To utilize the data from diverse devices, data needs to be able to well-integrated. Therefore, it is better for data to be in a common data format, or it needs to be able to transform into one another. The other characteristic is that the IoT devices may interact in local and global environments of the building. Therefore, the scope of the dataset used in training can play a critical role in developing AL/ML model for building management.

(AI/ML Algorithm) To offload and reduce the burden of the main server and to provide fast, efficient decision makings, the IoT and the other network-related devices can use their computing resources. Various edge-computing techniques can be applied to do so, such as developing light-weighted AI/ML models that can be easily deployed in the edge devices or balancing the learning and processing computation load from the server to the edge devices.

6.3.2. Use case

(To be continued)

6.4. Manufacturing optimization in Smart factory

When IoT meets smart factory Industry, Industrial Internet of Things (IIoT) technology integrating various technologies such as AI/ML, big data analytics, massive sensor connection, Machine to Machine communication (M2M) and automation are generally utilized. Its IIoT applications has initiated a profound change within companies making them increasingly connected and allowing them to exploit data to optimize their production processes, since IIoT is able to support real-time monitoring of each process line, thanks to different types of sensors positioned in critical points.

However, to anticipate and prevent possible failures, the health state of assets is always monitored, which generating tremendous volume of data, and instantons and intelligent responses to unexpected behavior are required.

6.4.1. Characteristics of Smart Factory for AI/ML processing

(To be continued)

6.4.2. Use case

(Real-time predictive maintenance) It is in needs to monitor the autonomous machines deployed in the manufacturing plants and forecast any maintenance requirements in prior to the machine failure.

(Demand Forecasting) With forecasting, future demand can be predicted to meet the targeted delivery deadlines in advance. This is one of the widespread strategies adopted across the industry where a number of manufacturing plants are presently transforming from reactive to predictive inventory management.

6.5. Smart parking

Parking operators are nowadays in the process of transforming their business model to satisfy a more demanding end-user, by looking for new technologies that allow the enhancement of drivers?user-experience. These drivers are currently facing mobility issues in urban areas, where the correlation of traffic congestion in cities is highlighted with drivers looking for parking spaces. As a consequence, current drivers are spending more time searching for a free place, contributing to higher fuel consumption and increased environmental pollution. Therefore, intelligent smart parking system is required to collect and process real-time parking occupancy information.

6.5.1. Characteristics of Smart Parking for AI/ML processing

In smart parking system, any end-user has access to real-time parking occupancy information both through urban furniture (information displays) and mobile/automobile apps, reducing the time each driver spends looking for a parking space. Moreover, parking operators can enhance their ongoing operations by knowing in every instant what is

happening and by analyzing reports on the behavioural patterns detected by the platform, improving their operational and enforcement efficiency. To achieve this, the usage of a video camera will be required to monitor in real time the available slots in a parking park. Additionally, the media acquired from the deployed cameras will be processed using image recognition AI/ML model allowing to perceive in real time the status of each parking spot. A lightweight AI/ML model will be initially used to support real-time processing, however, if any potential change is detected (a parking slot was potentially filled up for instance) more powerful AI/ML models can be utilized to confirm/dismiss the potential change. It can be also confirmed which entity is occupying the slot (car, bike, truck person, etc.)

6.6. Augmented Reality (AR) in IoT

Augmented reality technologies enable visualizing digital information and combining it with the physical world. As massive data are generated from IoT, visualization of digital information helps monitoring the status of circumstance and understanding the IoT system. AR techniques can be applied to several IoT applications such as autonomous driving, smart manufacturing industries. In such applications, massive IoT data are collected from IoT sensors including cameras, temperature & humidity sensors, and should be processed and analyzed in real-time.

6.6.1. Characteristics of Augmented Reality for AI/ML processing

As augmented reality techniques usually detect surrounding objects and interact with the physical world in real-time, it is computational-intensive and delay-sensitive. Also, the augmented reality devices which visualize digital information are usually mobile devices which have limited battery life. Therefore, augmented reality techniques should consider both latency minimization and energy consumption minimization. To achieve these goals, a lightweight AI/ML model would be used and edge computing techniques can be used to offload tasks to the edge server. However, as lots of data collected by IoT sensors, the data transmission time can be a critical issue in satisfying the latency restriction. Therefore, to support augmented reality in IoT systems, distributed and cooperative edge computing techniques which can distribute tasks at the edge server and mobile devices would be needed.

6.7. Smart Farm

Smart farm is a convenient and efficient agricultural form that manages the growing environment in an optimal state without time and

space restrictions by using automated facilities and information and communication technology. The most common example of smart farm implementation is a method in which sensors installed throughout the farm collect various information to help managers make decisions. An intelligent smart farm is required to process incoming information in real time and make decisions.

6.7.1. Characteristics of Smart Farm for AI/ML processing

Smart Farm collects meteorological information such as temperature, humidity, CO2, and solar radiation from IoT devices. In Smart Farm, various types of time series data are received sporadically, and compared to other IoT services, real time processing is not necessary. Data obtained from multiple sensors generally come in different time slots, and these data are combined into a single input. This combined input can be designed to control air conditioners and heaters, open and close windows, and feed nutrients using the AI/ML model.

7. IANA Considerations

This document requests no action by IANA.

8. Acknowledgements

9. Contributors

10. Informative References

[Hochreiter]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[Chung] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555v1 [cs.NE], 2014.

[Krizhevsky]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097-1105.. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[Doersch] C. Doersch, "Tutorial on variational autoencoders," arXiv preprint arXiv:1606.05908v2 [stat.ML], 2016.

[Goodfellow]I. Goodfellow et al., "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672-2680.

[Ba] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, 2014, pp. 2654-2662.

[Denil] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, and M. Ranzato, "Predicting parameters in deep learning," in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 2148-2156.

[Han] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, 2015, pp. 1135-1143.

[Venkataramani]S. Venkataramani, A. Ranjan, K. Roy, and A. Raghunathan, "AxNN: Energy-efficient neuromorphic systems using approximate computing," in Proc. Int. Symp. Low Power Electron. Design, ACM, 2014, pp. 27-32. [Moons]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[Moons] B. Moons, B. De Brabandere, L. Van Gool, and M. Verhelst, "Energy- efficient ConvNets through approximate computing," in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Lake Placid, NY, USA: IEEE, 2016, pp. 1-8.

[Mohammadi] Mohammadi, Mehdi, et al. "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys & Tutorials, 2018, pp. 2923-2960.

[Kriegel] Kriegel, HansPeter, et al. "Densitybased clustering," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2011, pp. 231-240.

Authors' Addresses

Jun Kyun Choi (editor)
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon, Korea

Email: jkchoi59@kaist.ac.kr

Jae Seob Han
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon, Korea

Email: j89449@kaist.ac.kr

Gyeong Ho Lee
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon, Korea

Email: gyeongho@kaist.ac.kr

Hyun Seo Park
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon, Korea

Email: tkf92001@kaist.ac.kr